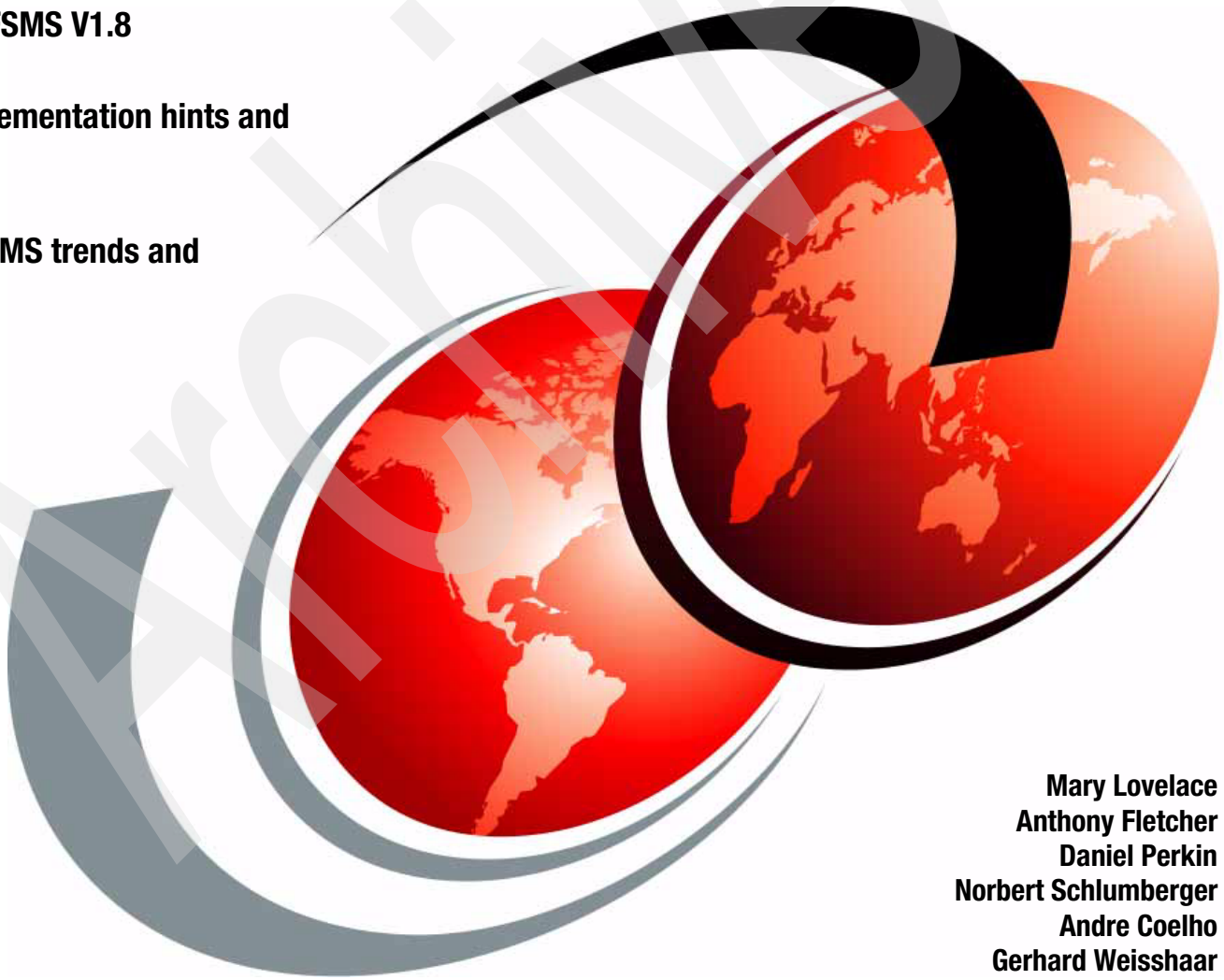
IBM

z/OS V1R8 DFSMS Technical Update

- Understand the features and functions in z/OS DFSMS V1.8
- Read implementation hints and tips
- Learn DFSMS trends and directions



Mary Lovelace
Anthony Fletcher
Daniel Perkin
Norbert Schlumberger
Andre Coelho
Gerhard Weisshaar

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS V1R8 DFSMS Technical Update

April 2008

SG24-7435-00

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

First Edition (April 2008)

This edition applies to Version 1, Release 8 of z/OS DFSMS (product number 5694-A01).

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo)  ®
z/OS®
CICS®
DB2®
DFSMS™
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSMSrmm™
DFSORT™
DS8000™
Enterprise Storage Server®
FlashCopy®
IBM®
MVS™
OS/390®
Redbooks®
RACF®
RAMAC®
REXX™
RMF™
Sysplex Timer®
System Storage™
TotalStorage®

The following terms are trademarks of other companies:

Shared Virtual Array, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

FRx, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Each release of DFSMS™ builds upon the previous version to provide enhanced storage management, data access, device support, program management, and distributed data access for the z/OS® platform in a system-managed storage environment.

This IBM® Redbooks® publication provides a technical overview of the functions and enhancements in z/OS V1R8 DFSMS and follow-on releases. It provides you with the information that you need to understand and evaluate the content of these DFSMS releases, along with practical implementation hints and tips. Also included are enhancements that were made available through an enabling PTF that have been integrated into z/OS V1R7 DFSMS.

This publication is written for storage professionals and system programmers who have experience with the components of DFSMS. It provides sufficient information so that you can start prioritizing the implementation of new functions and evaluate their applicability in your DFSMS environment.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Mary Lovelace is a Consulting IT Specialist at the International Technical Support Organization. She has more than 20 years of experience with IBM in large systems, storage and storage networking product education, system engineering and consultancy, and systems support. She has written many IBM Redbooks on TotalStorage® Productivity Center and z/OS storage products.

Anthony Fletcher is an IT Specialist working with the Global Services Delivery (GSD) z/OS software platform of IBM Global Services On Demand Infrastructure Services, based in New Zealand but working for New Zealand and Australia. He has 36 years of experience in z/OS and OS/390® and their predecessors and related components both as a customer of IBM and with IBM Global Services. He is a Team Leader for the mainframe operations of four diverse clients in the banking, airline, investment, and telecommunications industries. He holds a degree in Electrical Engineering from SALFORD University, Lancashire, UK. His main areas of expertise include DFSMS, DFSMSrmm™, and DFSMShsm™, and he has a working knowledge of RACF®. He also has experience in installing non-IBM products for the GSD platform.

Daniel Perkin is a Software Engineer with IBM DASD test organization in Tucson, Arizona. He has nine years of experience in large systems and storage. His main areas of expertise include z/OS Systems Programming, DASD, and copy services.

Norbert Schlumberger is an IT Architect with IBM Germany. He has 30 years of experience in storage software and storage management for IBM and customer systems, including 18 years of experience in DFSMSrmm. He also has experience in DFSMShsm and a good knowledge of RACF. Norbert's areas of expertise include performing conversions from vendor tape management products to DFSMSrmm, new DFSMSrmm implementations, and marketing support for DFSMSrmm including IBM 3494/IBM 3495 ATLs, VTSs, and vendor robotics. He also has experience in DFSMShsm and a good knowledge of RACF. He has worked at IBM for 33 years.

Andre Coelho is an IT Storage Specialist in Rio de Janeiro, Brazil, but working for IBM West Internal Accounts, Boulder, Colorado. He has 22 years of experience in z/OS and predecessors. His areas of expertise include DFSMSdfp™, DFSMShsm, and DFSMSrmm.

Gerhard Weisshaar is a Senior Consultant working with Empalis group in Germany. He has more than 20 years of experience in z/OS and its predecessors. His areas of expertise include implementation of storage and storage networking software and hardware, as well as z/OS Security Server. He has written extensively on the chapters in this book related to DFSMShsm and DFSMSrmm.

Thanks to the following people for their contributions to this project:

Sangam Racherla
Emma Jacobs
International Technical Support Organization, San Jose Center

Bob Haimowitz
International Technical Support Organization, Raleigh Center

Stephen Branch
IBM San Jose

Philip Chauvet
Victor Liang
Brian Corkill
Terri Menendez
Miguel Perez
Helen Witter
Art Bariska
Sherrie Neiderbrach
Jeff Suarez

Mike Wood
IBM UK

Bernd-Rainer Bresser
Reinhard Siegel
IBM Germany

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review book form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Something old, something new

This chapter provides:

- A summary of the function provided by z/OS V1R7 DFSMS in “What was new in z/OS V1R7 DFSMS” on page 2
- An overview of what is new in z/OS V1R8 DFSMS in “What is new in z/OS V1R8 DFSMS” on page 3

1.1 What was new in z/OS V1R7 DFSMS

The followings section summarize the enhancements made in various components in z/OS V1R7 DFSMS. For details on these enhancements refer to *z/OS V1R7 DFSMS Technical Update*, SG24-7225.

1.1.1 DFSMSdfp

DFSMSdfp V1R7 provided enhancements in the following areas:

- Implementation of large format data sets
- Implementation of the device support address space (DEVMAN)
- Support for processor multiple subchannels
- Provision of REPRO MERGECAT Fromkey/Tokey option
- Catalog enhancements
  - VVDS implicit size specification
  - Automatic Catalog access tuning
- VSAM data set extent constraint removal
- VSAM RLS 64-bit data buffers
- SMS Volume and ACS allocation test enhancements
  - SMS volume status change from NOTCON by VARY command
  - SMS ACS message processing enhancements
  - SMS ACS environment enhancements
- Extended Remote Copy Plus
- DEVSERV QLIB command
- PDSE restartable address space (This was introduced with V1R6 but has significant changes by way of APAR fixes since the initial release.)
- Function removals - although not enhancements in the usual sense
  - ISAM removal
  - JOBCAT and STEPCAT removal reminder
  - VSAM Attributes removal reminder

1.1.2 DFSMSdss

DFSMSdss™ provided enhancements to support the new large format data sets.

Large format sequential data sets are now supported for the following DFSMSdss functions:

- Logical COPY
- Logical and physical DUMP and RESTORE
- Stand-Alone RESTORE
- Logical and physical RELEASE
- PRINT
- DEFRAG

DFSMSdss also supports the use of large format data sets for the following:

- Output from logical and physical DUMP
- Input to logical, physical, and Stand-Alone RESTORE
- Input or output for COPYDUMP.\

Large format data sets are processed in exactly the same way as basic format data sets by most of the commands above. The remainder of this chapter highlights how large format data sets are processed where there is a difference between the processing of large format data sets and basic format data sets.

1.1.3 DFSMShsm

DFSMShsm V1R7 provided enhancements to support the following:

- Support for large format data sets
- Fast subsequent migration improvements
- Extended tape table of contents (TTOC)
- Removal of ABARS requirement for INCLUDE statement
- Cancellation of individual HSM tasks
- Using wild cards with HMIGRATE
- Saving LRECL of migrated data sets in the MCD
- New recycle processing options for connected sets
- Audit Media Controls Resume
- Dump encryption support

1.1.4 DFSMSrmm

DFSMSrmm V1R7 provided the following enhancements:

- Facility to issue DFSMSrmm TSO commands from the operator console
- Improved security control over DFSMSrmm functions
- Enterprise Enablement
- Support for the use of large format data sets

1.1.5 OAM

DFSMSdfp V1R7 provided the following enhancements to OAM:

- Tape dispatcher display
- Immediate recall to DB2®
- Clear old location option
- Return to MVS™ scratch exit
- Enhanced MOVEVOL utility

1.2 What is new in z/OS V1R8 DFSMS

Details of the new functions and enhancements in z/OS V1R8 DFSMS are provided in later chapters, as listed below. The following sections are summaries of what is covered in the remainder of this book.

1.2.1 DFSMSdfp

DFSMSdfp V1R8 has been enhanced in the following areas:

- Improvements in tape data set security by new options in DEVSUPxx member
- Catalog
- SMS Fast Path Volume Select
- SMS SCDS create ACDS
- SMS Serviceability
- VSAM code modernization
- DFSMS RLS Enhanced recovery

  This is an internal improvement that occurs automatically.
- DFSMS RLS
  - Problem diagnosis - See 3.6.2, “VSAM RLS DIAG command on z/OS V1R8 system” on page 26.
  - Performance enhancement - See 3.6.4, “VSAM RLS performance enhancement” on page 27.
  - SMF records for analysis - See 3.6.7, “Monitoring the VSAM RLS use of 64-bit data buffers” on page 30.
- PDSE Program Object format 5 (PO5) module format compatibility on pre-z/OS V1R8 systems

  Refer to 3.8.1, “PDSE PO5 module format compatibility on pre-z/OS V1R8 systems” on page 55, for details.
- PDSE larger buffer specification

  See 3.8.2, “PDSE 64-bit virtual storage use enablement” on page 55.
  - PDSE_HSP-SIZE
  - PDSE1_HSP_SIZE
  - PDSE_DIRECTORY_SIZE
  - PDSE1_DIRECTORY_SIZE
- PDSE buffer hold beyond close

  See 3.8.4, “Retain buffers beyond PDSE close” on page 62.
  - PDSE_BUFFER_BEYOND_CLOSE
  - PDSE1_BUFFER_BEYOND_CLOSE
- DEVMAN Rapid Index rebuild
- ICFRU

  This was added to DFSMS V1R7 but not documented in *z/OS V1R7 DFSMS Technical Update*, SG24-7225. For an ICFRU Readiness review example see 3.10, “Integrated Catalog Forward Recovery Utility” on page 73.

1.2.2 DFSMSdfp OAM enhancements

DFSMS OAM V1R8 has been enhanced in the following areas:

- Binary large object support
- Immediate backup copy
- Automated selection of RECYCLE volumes
- Global display keyword

1.2.3 DFSMSdss enhancements

DFSMSdss V1R8 has been enhanced in the following areas (see 5.1, “Defining logical and physical processing” on page 134):

- DSS LOGICAL COPY
- DSS extended keywords to differentiate between logical and physical operations

1.2.4 DFSMShsm enhancements

DFSMShsm V1R8 has been enhanced in the following areas:

- Fast replication tape support
- Data set recover from fast replication backup and dump versions
- ARECOVER individual data set restore
- Better handling of errors on alternate duplex tapes
- Recycle SYNCHDEV at intervals
- Migration scratch queue for non-VSAM data sets
- New command ALTERPRI

1.2.5 DFSMSrmm enhancements

DFSMSrmm V1R8 has been enhanced in the following areas:

- Improvements in tape data set security
- Enterprise RMM CIMOM enhancements

2

Useful information to get you started

This chapter contains useful information that we found by trial and error, including common errors and undocumented tidbits of information.

2.1 Cheating with OAM

After implementing the immediate backup and automated selection of RECYCLE volumes functions, you may want to verify that it works, but you might not have any tape volumes that are actually full. Here is one method that could serve to verify both new functions:

1. Set SMS up so that an immediate backup occurs to tape.
2. Run OSREQ STORE to store an object.
3. Observe the tape mount and the immediate back up of data to the first backup.
4. For the volume where the backup data was written, issue the command:

   ```
   MODIFY OAM,UPDATE,VOLUME,xxxxxx,FULL,Y
   ```

5. Run OSREQ STORE to store a second object.
6. Observe the tape mount and the immediate back up of data.
7. Issue the command MODIFY OAM,START,RECYCLE,(ALLBK1),DISPLAY,PV=100.

   You will see the volume you marked as full.
8. Issue the command MODIFY OAM,UPDATE,VOLUME,xxxxxx,FULL,N for the volume you had previously marked full.

2.2 HSM fast migration reconnect users

There is a problem affecting users of the HSM fast reconnect to migrated copies of data sets that in certain circumstances can lead to loss of data. Refer to “OA19493” on page 464, for further information.

2.3 Users of Fast Subsequent Migration

There is a possibility of data loss if DFSMSdss and DFSMShsm with Fast Subsequent Migration (FSM) is used on any level of OS390 V2.10 or z/OS V1.1 to V1.6.

From z/OS V1.7 or later there is no problem if all use of DFSMSdss is through the API. This means that DFSMShsm use and any other use through the API is not a problem, and a problem only arises if DFSMSdss is used through JCL/BATCH.

If you fall into the affected group, or if in doubt, you should consider the following:

- Users of DFSMSdss together with DFSMShsm FSM should refer to the current version of Information APAR OA20117.

  This APAR describes a situation where a data set's change bit could be turned off if DFSMSdss is used to restore the data set outside the control of DFSMShsm, and DFSMShsm is configured to use FSM.

  This is because DFSMSdss, when run using JCL to invoke it, always turns the data set changed bit off on any data set restored.

This could result in what should be a new migration being skipped when DFSMShsm considers the data set for migration (see Figure 2-1). If the bit is turned off when the data set is evaluated, and a migrated copy of the data set is found in the DFSMShsm MCDS, it may consider that to be a suitable migrated copy, and will revalidate the entry and delete the data set from DASD. A subsequent recall would then find an old copy of the data set.

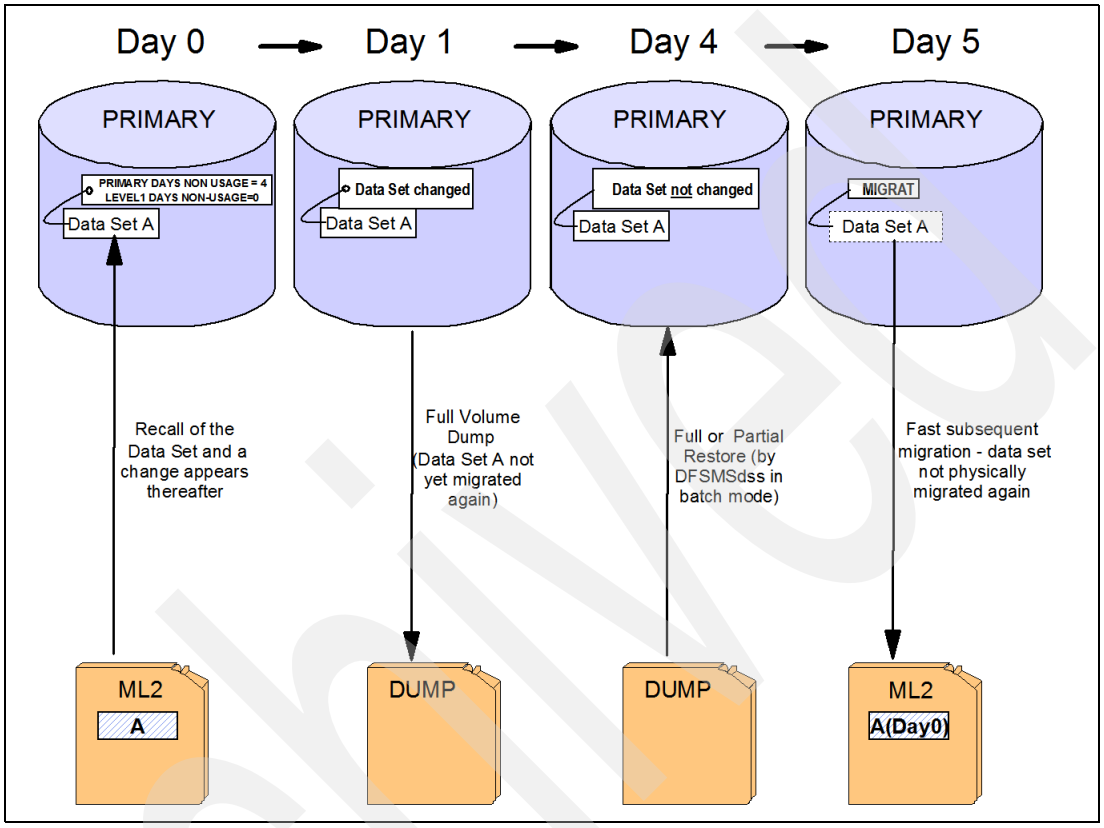


Figure 2-1 FSM potential data loss scenario

Using DFSMShsm to manage data set restores manages the situation properly even if, as is the case, the migrated copy was moved by DFSMSdss.

If in doubt as to whether all data set restores are under the control of DFSMShsm, you should consider suspending the use of DFSMShsm FSM until resolution is available.

- Apar OA20907 has been opened to provide a patch option for DFSMSdss that changes the behavior of DFSMSdss to *not* reset the changed bit if it was found to be on in the restored copy. This makes its behavior consistent with the invocation of DFSMSdss via DFSMShsm. You should monitor the status of APAR OA20907.

2.4 Tape security

DFSMS support introduces new options for securing tape data sets using the System Authorization Facility (SAF). These are designed to allow you to define profiles to protect data sets on tape using the DATASET class without the need to activate the TAPEDSN option or the TAPEVOL class. DFSMS also provides options that you can use to specify that all data sets on a tape volume should have common authorization and that users are authorized to overwrite existing files on a tape volume.

The new options were introduced as a result of customer requirements. So, it is time to rethink the way that you are currently protecting data on tape.

There are new options available for use in the DEVSUPxx member of PARMLIB to change the behavior of open data set security checking in case of tape data sets. Refer to Chapter 9, “Tape security” on page 331, for details.

2.5 RECYCLE error - potential data loss

There is a red alert about potential data loss after a RECYCLE error on DFSMShsm V1R8. Figure 2-2 contains the description of the red alert. The APAR for this problem is APAR OA18465. Refer to “OA18465” on page 462 for additional information.

```
RED ALERT: V1R8 DFSMShsm potential data loss after RECYCLE error.

ABSTRACT:
POTENTIAL DATA LOSS AFTER RECYCLE ERROR.

DESCRIPTION:
A problem was detected in the V1R8 DFSMShsm RECYCLE function which under
certain circumstances, can potentially result in data loss. The problem occurs
only when the recycle function encounters certain errors with its input tape.
The most common type of error would be the failure to mount the input tape,
resulting in the failure of the recycle function with message ARC0833I rc31.
The problem detected causes DFSMShsm to lose knowledge of the logical EOF on
the output tape. As a result, the next time that DFSMShsm attempts to write to
the recycle output tape, it will write from load point, overwriting data
previously written to the tape. Attempts to recall/recover the overwritten data
sets will fail with ARC1001I rc68 reas16.

THE PTF for APAR OA18465/UA29904 will cause the recycle processing to maintain
the logical EOF information for its output tape. Error scenarios that would
have caused DFSMShsm to lose knowledge of the logical EOF on the recycle output
tape will now result in an abend of the recycle task. Although the attempt to
recycle the input tape will fail, the potential for the output tape being
subsequently overwritten is eliminated.

V1R8 customers are strongly advised to avoid running the RECYCLE function on
V1R8 DFSMShsm until the PTF for APAR OA18465/UA29904 has been installed. Please
see APAR OA18465 for more information.

The DFSMShsm development team has created a tape assessment tool for those
customers that have run RECYCLE already. Information on the tool can be found
in APAR OA18465.

RECOMMENDED ACTIONS:
Do not run RECYCLE on a V1R8 DFSMShsm prior to installing PTF UA29904 for
OA18465. All installations who have run RECYCLE on a V1R8 DFSMShsm are
requested to mark any partial ML2 and backup tapes full using the DELVOL volser
MIGRATION(MARKFULL) or DELVOL volser BACKUP(MARKFULL) command, as appropriate.

NOTE: DFSMShsm RECYCLE can be run safely on systems with z/OS V1R7 and below.
```

Figure 2-2 Output about DFSMShsm red alert

2.6 Management class retention limit and tape data sets

After OA17011 is applied, tape volumes are prematurely released during RMM housekeeping because the special expiration date explicitly specified in the JCL by the user is overridden

when the management class assigned to the tape data set specifies Retention Limit = 0. Management class attributes should not apply to tape data sets.

This error is addressed by APAR OA20293. Refer to “OA17011” on page 474 and “OA20293” on page 471 for details.

3

DFSMSdfp V1R8 enhancements

In this chapter we discuss new and changed functions in DFSMSdfp. The following topics are covered:

- Catalog enhancements
- SMS volume selection performance enhancement
- Copy SCDS to ACDS
- SMS serviceability
- VSAM code modernization
- RLS updates
- Device manager enhancements
- PDSE enhancements
- PDSE buffer management statistics
- ICFRU

3.1 Catalog enhancements

z/OS V1R8 introduces improvements in LISTCAT processing and output as well as the ability to specify the number of catalog address space requests available for user requests.

3.1.1 LISTCAT

IDCAMS LISTCAT processing in z/OS V1R8 has been enhanced to provide better performance, especially for large catalogs. The performance improvement is automatic in V1R8, and no action needs to be taken in order to exploit the new function.

Important: The output from LISTCAT commands has also changed, which might affect products that process the output.

Examples

A new header line appears on each page of LISTCAT output. Additionally, there have been changes in the way LISTCAT LEVEL processing works for GDGs and ALIASes. Figure 3-1 shows an example of pre-z/OS V1R8 LISTCAT LEVEL output, specifying the ALIAS name in the LISTCAT LEVEL command.

```
 LISTCAT LEVEL(SYSDOC)
GDG BASE ------ SYSDOC.CIMN.SYSLOGD
     IN-CAT --- COMCIC.ICFCAT
NONVSAM ------- SYSDOC.CIMN.SYSLOGD.G0832V00
     IN-CAT --- COMCIC.ICFCAT
.
.
.
IDCAMS  SYSTEM SERVICES                                          TIME:
17:06:24
         THE NUMBER OF ENTRIES PROCESSED WAS:
                   AIX ------------------0
                   ALIAS ----------------0
                   CLUSTER --------------0
```

Figure 3-1 Pre-z/OS V1R8 LISTCAT output

Figure 3-2 shows an example of z/OS V1R8 LISTCAT LEVEL output illustrating the header and ALIAS changes.

```
 LISTCAT LEVEL(SYSDOC)
                              LISTING FROM CATALOG -- COMCIC.ICFCAT
GDG BASE ------ SYSDOC.CIMN.SYSLOGD
     IN-CAT --- COMCIC.ICFCAT
NONVSAM ------- SYSDOC.CIMN.SYSLOGD.G0832V00
     IN-CAT --- COMCIC.ICFCAT
.
.
.
IDCAMS  SYSTEM SERVICES                                          TIME:
17:04:40
                              LISTING FROM CATALOG -- SCI18A.DASDPLEX.CATALOG
ALIAS --------- SYSDOC
     IN-CAT --- SCI18A.DASDPLEX.CATALOG
IDCAMS  SYSTEM SERVICES                                          TIME:
17:04:40
                              LISTING FROM SELECTED CATALOGS
         THE NUMBER OF ENTRIES PROCESSED WAS:
                   AIX ------------------0
                   ALIAS ----------------1
                   CLUSTER --------------0
```

Figure 3-2 z/OS V1R8 LISTCAT output

Migration and coexistence

Information APAR II14250 contains details about the differences in IDCAMS LISTCAT processing in z/OS V1R8. This is generally not a consideration unless, for example, you have a vendor product that processes LISTCAT output.

Maintenance

We recommend that you have the PTFs for APARs OA16912, OA18720, and OA20169 applied prior to using the new support. In addition, the PTF for APAR OA16372 should be applied.

OA16912 and OA18720 represent the roll-up APARs for this new support. APAR OA20169 is the APAR resulting from OA18184's PTF being marked PE. It is not closed at the time of this writing, but is important because it fixes the PE introduced with OA18184 as well as the issue OA18184 was intended to fix, where LISTCAT was not issuing error messages for offline volumes.

OA16372 represents VSAM GET/PUT requests failing when accessing a VSAM data set and the data set has more than three volumes that have been recataloged out of the original extend order. Refer to "OA16372" on page 468 for a full description.

3.1.2 Dynamic service count

z/OS V1R8 introduces the ability to alter the number of catalog address space service tasks available for user requests. An insufficient number of service tasks is usually indicated by running a MODIFY CATALOG,REPORT command and observing a HIGHEST # OF SERVICE TASKS value approaching the SERVICE TASK UPPER LIMIT value. Exceeding the limit results in performance problems.

The default number of service tasks is 200 and the maximum potential number is 999. Of these, 90% are used as the maximum number of user catalog requests that can be processed concurrently. Figure 3-3 shows an example of the MODIFY CATALOG,REPORT command using the default value. It shows a value of 180 for the SERVICE TASK UPPER LIMIT, which is 90% of 200. In this case the HIGHEST # OF SERVICE TASKS value is only 16, so there is no need to change the default upper limit value.

```
MODIFY CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 831
*CAS*************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ1180                            *
*  CATALOG ADDRESS SPACE ASN = 0034                               *
*  SERVICE TASK UPPER LIMIT  =  180                               *
*  SERVICE TASK LOWER LIMIT  =   60                               *
*  HIGHEST # SERVICE TASKS   =   16                               *
*  CURRENT # SERVICE TASKS   =   16                               *
*  MAXIMUM # OPEN CATALOGS   = 1,024                              *
```

Figure 3-3 CATALOG REPORT showing default value

If the HIGHEST # OF SERVICE TASKS does approach or exceed the 180 boundary, we use this new function to change the upper limit value.

Implementing

The SYSCAT statements in the LOADxx member of PARMLIB cannot be used to change the default. A SYSCATxx member in SYS1.NUCLEUS must be used. Many installations presently use the LOADxx member for SYSCAT statements, so a new SYSCATxx member might have to be allocated. We have provided an example of a job to allocate the SYSCATxx member in Figure 3-4. The number of service tasks defined is specified in columns 65–67 of the SYSCATxx member. This example changes the service task count to 999.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
******************************** Top of Data ***************************
//SYSCAT JOB CLASS=A,MSGCLASS=T,NOTIFY=&SYSUID,REGION=6M
//SYSCAT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=SYS1.NUCLEUS(SYSCATLG),DISP=SHR,DCB=(RECFM=U)
//SYSUT1 DD *
SBOX0011  MCAT.SANDBOX.Z18.SBOX00                              Y 999
//*
******************************** Bottom of Data ************************
```

Figure 3-4 Example SYSCATxx member

More information about the format and use of the SYSCATxx member can be found in *DFSMS Managing Catalogs,* SC26-7409.

After we have removed the SYSCAT statements from LOADxx and IPLed using the new SYSCATxx member we added to PARMLIB, we issue another MODIFY CATALOG,REPORT command. Figure 3-5 is a catalog report that now shows 900, or 90% of the 999 figure we specified in SYSCATxx.

```
MODIFY CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 998
*CAS*******************************************************
*  CATALOG COMPONENT LEVEL   = HDZ1180                     *
*  CATALOG ADDRESS SPACE ASN = 0034                        *
*  SERVICE TASK UPPER LIMIT  =  900                        *
*  SERVICE TASK LOWER LIMIT  =   60                        *
*  HIGHEST # SERVICE TASKS   =   21                        *
*  CURRENT # SERVICE TASKS   =   21                        *
*  MAXIMUM # OPEN CATALOGS   = 1,024                       *
```

Figure 3-5 CATALOG REPORT showing maximum value

Migration and coexistence

If you decide to IPL a pre-V1R8 level after making these changes, you first need to undo the changes you made to SYSCATxx and LOADxx when implementing this V1R8 support.

3.2 SMS volume selection performance enhancement

SMS has been enhanced with a new function to speed up volume selection. In storage groups with hundreds or thousands of candidate volumes, it can take an unacceptably long time for SMS to determine a target for placement of a new data set.

Turning fast path volume selection on causes SMS to select volumes normally until DADSM rejects 100 volumes for insufficient free space. It then excludes all volumes that do not have sufficient free space in the current SMS volume statistics, which has the effect of reducing the overall number of candidate volumes and therefore the number of attempts SMS must make in order to find a suitable target. This is not a function that is normally exercised often. It is a performance improvement for the worst-case scenario when you have a lot of volumes and the majority of them are nearly full. This worst-case scenario is more likely to occur when you have a large number of volumes in a single storage group. That type of environment benefits the most from this new function.

There are two limits to using fast volume selection:

- It does not apply to striping allocation. Striping allocation has already excluded the volumes above the high threshold from volume selection.
- It may inadvertently exclude volumes that have sufficient free space but for which the SMS volume statistics indicate that they do not, as described in the *MVS Initialization and Tuning Reference,* SA22-7592, and shown in Figure 3-6.

```
Invalid volume statistics can occur as a result of the following events:

     The VTOC index is broken.

     OEM products bypass CVAF processing.

     In an SMSplex when the SMS synchronization time interval has not
     yet been driven to update the SMS configuration with the most
     current space statistics.  These statistics are based on updates
     that can occur on another system in the SMSplex.
```

Figure 3-6 Occasions for bypass due to SMS volume statistics

Implementing

Fast volume selection is requested by using the new FAST_VOLSEL parameter. This can be set on or off by:

- Using the IGDSMSxx member of PARMLIB
- Issuing the SETSMS FAST_VOLSEL command

An example of setting fast volume selection to on using the IGDSMSxx member of PARMLIB is provided in Figure 3-7:

```
 SYS1.PARMLIB(IGDSMS03) - 01.04
 ===>
***************************** Top of Data ********
SMS ACDS(SYS1.SMS.ACDS)
    COMMDS(SYS1.SMS.COMMDS)
    INTERVAL(15)
    DINTERVAL(150)
    DEADLOCK_DETECTION(15,4)
    FAST_VOLSEL(ON)
    SMF_TIME(YES)
    CF_TIME(1800)
    RLSINIT(YES)
    RLS_MAX_POOL_SIZE(100)
    REVERIFY(NO)
```

Figure 3-7 Placing the FAST_VOLSEL parameter in PARMLIB

An example of setting fast volume selection on using the SETSMS FAST_VOLSEL command is provided in Figure 3-8. The command does not come back with a message indicating that it is complete. You can verify its present state by issuing the D SMS,OPTIONS command, which is also shown in Figure 3-8.

```
SETSMS FAST_VOLSEL(ON)
.
.
.
D SMS,OPTIONS
IGD002I 17:01:47 DISPLAY SMS 253
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
.
.
.
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
GDS_RECLAIM = YES       DSSTIMEOUT = 0
BLOCKTOKENSIZE = NOREQUIRE       FAST_VOLSEL = ON
IGD002I 17:01:47 DISPLAY SMS
TRACE    = ON     SIZE = 128K     TYPE = ALL
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = ON   SMSSJF = ON   SMSSSI = ON   ACSINT = ON
    OPCMD  = ON   CONFC  = ON   CDSC   = ON   CONFS  = ON
    MSG    = ON   ERR    = ON   CONFR  = ON   CONFA  = ON
    ACSPRO = ON   IDAX   = ON   DISP   = ON   CATG   = ON
    VOLREF = ON   SCHEDP = ON   SCHEDS = ON   VTOCL  = ON
    VTOCD  = ON   VTOCR  = ON   VTOCC  = ON   VTOCA  = ON
    RCD    = ON   DCF    = ON   DPN    = ON   TVR    = ON
    DSTACK = ON   UAFF   = ON   DEBUG  = ON
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
```

Figure 3-8 Turning FAST_VOLSEL(ON) with the SETSMS command and verifying state

Once FAST_VOLSEL is turned on, you see a new IGD17294I message when fast volume selection is used. An example of this appears in Figure 3-9, which is the output from an IEFBR14 job allocating a data set in a space-constrained storage group.

```
IEF236I ALLOC. FOR IEFBR14 D301
IGD17294I FAST VOLUME SELECTION IS USED TO ALLOCATE DATA SET
MHLRES3.MHLSMS.XLD301
IGD101I SMS ALLOCATED TO DDNAME (SEQOUTG )
       DSN (MHLRES3.MHLSMS.XLD301                          )
        STORCLAS (MHLSMS) MGMTCLAS (        ) DATACLAS (       )
        VOL SER NOS= SLD14D
IEF142I IEFBR14 D301 - STEP WAS EXECUTED - COND CODE 0000
IGD104I MHLRES3.MHLSMS.XLD301                                RETAINED,
DDNAME=SEQOUTG
IEF373I STEP/D301    /START 2007052.1126
IEF374I STEP/D301    /STOP  2007052.1126 CPU    0MIN 00.00SEC SRB    0MIN
00.00S
```

Figure 3-9 Fast volume selection in action

Migration and coexistence

If you IPL a pre-V1R8 driver you need to remove FAST_VOLSEL from the IGDSMSxx member of PARMLIB if you included it while implementing z/OS V1R8.

3.3 COPY SCDS to ACDS

This enhancement allows users to create an ACDS from any valid SCDS without first having to activate the ACDS. An example of its use is be to create an ACDS for use at a separate disaster recovery location.

Preallocating the ACDS

One of the requirements for this command is that the target ACDS for the COPYSCDS command must be pre-allocated. In Figure 3-10 we provide an example of an IDCAMS job to allocate a 60-track ACDS. Check the size of your existing SCDS to determine how many tracks your target ACDS should be.

```
//ALLCACDS JOB CLASS=A,MSGCLASS=T,NOTIFY=&SYSUID,REGION=6M
//STEP EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER(NAME(MHLRES3.TEST.ACDS) LINEAR VOL(SBOX11) -
  TRK(60 60) SHAREOPTIONS(2,3)) -
  DATA(NAME(MHLRES3.TEST.ACDS.DATA)REUSE)
```

Figure 3-10 Pre-allocating your ACDS

Using the new COPYSCDS command

The requirements are:

- The ACDS must be pre-allocated.
- The source SCDS should be validated before being copied.

SMS automatically verifies that:

- The acds_dsn is not the currently active ACDS.
- The scds_dsn is a valid SCDS.

If you do break any of these rules, SMS issues an IGD088I message, as in the case where we attempt to copy our SCDS into the active ACDS, as seen in Figure 3-11.

```
SETSMS COPYSCDS(SYS1.SMS.SCDS,SYS1.SMS.ACDS)
IGD088I COPYSCDS COMMAND FAILED SCDS SYS1.SMS.ACDS 313
- REASON CODE 6135
```

Figure 3-11 Example of the IGD088I message with COPYSCDS failure

The command format for the new COPYSCDS command is:

```
SETSMS COPYSCDS(scds_dsn,acds_dsn)
```

Where *scds_dsn* is the name of the SCDS that is copied to the target ACDS acds_dsn. Figure 3-12 is an example of the SYSLOG output that you should expect to see after successfully issuing this command.

```
SETSMS COPYSCDS(SYS1.SMS.SCDS,MHLRES3.TEST.ACDS)
IEF196I IEF237I 831A ALLOCATED TO SYS00016
IEF196I IEF237I D30C ALLOCATED TO SYS00017
IGD019I SCDS SYS1.SMS.SCDS SUCCESSFULLY COPIED AS ACDS TO 243
MHLRES3.TEST.ACDS
IEF196I IGD104I MHLRES3.TEST.ACDS                          RETAINED,
IEF196I DDNAME=SYS00017
IEF196I IGD104I SYS1.SMS.SCDS                              RETAINED,
IEF196I DDNAME=SYS00016
```

Figure 3-12 SYSLOG output from a successful COPYSCDS command

Migration and coexistence

There are no migration or coexistence considerations for the COPYSCDS support.

3.4 SMS serviceability

A new DEBUG parameter has been added to the SETSMS TRACE command to add trace points at strategic locations and reduce the size of SMS trace data and therefore the amount of time needed to analyze the data. This serves to reduce the total time necessary for problem resolution.

This parameter should only be used at the direction of the system programmer when engaged in problem determination.

Using the DEBUG parameter

The new DEBUG parameter is set with the SETSMS TRACE command and is displayed with the D SMS,TRACE command. IBM provides the specific commands at problem determination time. An example is provided in Figure 3-13.

```
SETSMS DESELECT(ALL)
IEE712I SETSMS   PROCESSING COMPLETE
SETSMS TRACE(ON) TYPE(ALL) SIZE(1M) SELECT(DEBUG)
IEE712I SETSMS   PROCESSING COMPLETE
D SMS,TRACE
IGD002I 12:23:29 DISPLAY SMS 020
TRACE    = ON     SIZE = 1024K     TYPE = ALL
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = OFF  SMSSJF = OFF  SMSSSI = OFF  ACSINT = OFF
    OPCMD  = OFF  CONFC  = OFF  CDSC   = OFF  CONFS  = OFF
    MSG    = OFF  ERR    = OFF  CONFR  = OFF  CONFA  = OFF
    ACSPRO = OFF  IDAX   = OFF  DISP   = OFF  CATG   = OFF
    VOLREF = OFF  SCHEDP = OFF  SCHEDS = OFF  VTOCL  = OFF
    VTOCD  = OFF  VTOCR  = OFF  VTOCC  = OFF  VTOCA  = OFF
    RCD    = OFF  DCF    = OFF  DPN    = OFF  TVR    = OFF
    DSTACK = OFF  UAFF   = OFF  DEBUG  = ON
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
```

Figure 3-13 Turning the DEBUG parameter ON and displaying its status

Migration and coexistence

There are no migration or coexistence considerations when using this new function.

3.5 VSAM code modernization

VSAM code has been modernized so that it automatically dumps for errors that normally indicate a system problem or invalid data set. No action is required to implement the automatic dumps.

Additionally, the current procedure for collecting data for a specific VSAM return code is complex and prone to errors. New parameters for the MODIFY CATALOG command have been added to assist with the first-time data capture of user-specified VSAM problems.

Usage

The new VDUMPON and VDUMPOFF parameters for the MODIFY CATALOG command, as described in *DFSMS Managing Catalogs,* SC26-7409, can be seen in Figure 3-14.

```
MODIFY CATALOG,VDUMPON(pdf,rc,compid,error)

pdf       Specifies the VSAM Problem Determination Function code (one
          to three characters from 0 to 255), or * (asterisk).
rc        Specifies the VSAM return code in decimal format (one to
          three characters from 0 to 255), or * (asterisk).
compid    Specifies the component code (0 - 5), or * (asterisk).
error     Specifies the VSAM error code in decimal format (one to
          three characters from 0 to 255), or * (asterisk).
Notes:

1.  Specifying an asterisk (*) for any parameter indicates that a wild
    card search is to be done for that value. Up to three asterisks
    (without intervening spaces) may be specified where a single
    asterisk is allowed, but the extra asterisks have no effect on the
    command's output.
2.  If a parameter is to be omitted, it must be specified as an
    asterisk (*).  For example, VDUMPON(*,*,*,2) is syntactically
    valid, while VDUMPON(2), VDUMPON( , , ,2), and VDUMPON(2, ,2) are
    all invalid.
3.  At least one of the parameters must be specified with a value
    other than asterisks. For example, VDUMPON(*,*,*,*) is not
    allowed.
4.  User-initiated VSAM dynamic dumping does not occur unless you
    specify VDUMPON.
5.  You can set only one VDUMPON at a time. Each entry overwrites the
    previous information. After a match occurs, the information is
    cleared and no further user-initiated dumps will be taken.
.
.
.
Issuing MODIFY CATALOG,VDUMPOFF will clear the VDUMP options.
```

Figure 3-14 The new VDUMPON and VDUMPOFF parameters

Example

In order to demonstrate use of this new function, we wrote an IDCAMS job that attempts to delete a non-existant VSAM data set, which results in a RC 8 ERROR 16 VSAM RECORD NOT FOUND failure. The example job is shown in Figure 3-15 and the SYSLOG output for this is shown in Figure 3-16 on page 24.

```
//DELVSM JOB MSGCLASS=H,REGION=0M
//DELSTG EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
     DELETE SLT.DATASET.DOESNT.EXIST CLUSTER PURGE
```

Figure 3-15 Deleting a non-existant VSAM data set

```
MODIFY CATALOG,VDUMPON(*,8,*,16)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG VSAM DUMP OPTIONS 957
*CAS*******************************************************
*    STATUS  FUNC CODE  RETURN CODE  COMPONENT  ERROR CODE      *
*       ON      ***         008         ***        016          *
*CAS*******************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
$HASP100 DELVSM   ON INTRDR                            FROM TSU13819
MHLRES3
IRR010I  USERID MHLRES3 IS ASSIGNED TO THIS JOB.
ICH70001I MHLRES3 LAST ACCESS AT 11:22:32 ON MONDAY, MARCH 12, 2007
$HASP373 DELVSM   STARTED - INIT 1    - CLASS B - SYS CIMN
IEA794I SVC DUMP HAS CAPTURED: 963
DUMPID=001 REQUESTED BY JOB (CATALOG )
DUMP TITLE=VSAM DYNAMIC RPL DUMP - IDA019RA+1D8A FEEDBACK CODE:
            0A080010
.
.
.
IEA611I COMPLETE DUMP ON SYSDOC.DUMP.DA.D070312.T112257.CIMN.S00001 975
DUMPID=001 REQUESTED BY JOB (CATALOG )
```

Figure 3-16 Example VSAM dump for VSAM record not found

Migration and coexistence

There are no migration or coexistence considerations when using this new function.

3.6 RLS updates

In z/OS V1R8 the following VSAM RLS features were introduced:

- The ability for users to detect VSAM RLS latch contention has been introduced with the VSAM RLS DIAG command (see “VSAM RLS DIAG command on z/OS V1R8 system” on page 26).
- A performance enhancement is provided to speed up the process of terminating the SMSVSAM address spaces across the members of a SYSPLEX (see “VSAM RLS performance enhancement” on page 27).

z/OS V1R7 64-bit (2 GB bar) implementation continuation: The SMF recording of use of 64-bit buffers that was intended for delivery in z/OS V1R7 was held over and is now available in V1R8. See 3.6.5, “VSAM RLS use of 64-bit data buffers” on page 28.

Important: Before using VSAM RLS in a DFSMS V1.8 environment you should review OA17415. All users of VSAM RLS reading or writing to a multi-volume VSAM data set where the volumes have been recataloged in a different order than originally extended to are affected. Refer to “OA17415” on page 469 for more details.

3.6.1 RLS enhanced recovery

In order to talk about RLS enhanced recovery, we should discuss VSAM record level sharing (RLS). It is a data set access mode that allows multiple address spaces, CICS® application owning regions on multiple z/OS systems, DFSMShsm control data sets access, DFSMStvs, and jobs to access data at the same time. VSAM RLS requires that the data sets be System Managed Storage (SMS) data sets.

The purpose of the RLS function is to enhance access and recovery capabilities to existing VSAM files. One particular area of enhancement is in the area of reliability, availability, and serviceability (RAS). You can divide that into two separate but related areas:

- First time data capture - to document internal logic errors at the earliest point in time possible. It consists of any system failure introduced by coding errors or *bugs*.
- System error recovery - to recover from an identified system error. It can be identified by a health check or by error detecting features on the base operating system. It will try to recover as much as possible, minimizing the impact to the customer. The highest priority is to prevent damage to customer data. After that, it can minimize a failing function and automate steps to restore the system to full functionality.

RLS works using a VSAM address space named SMSVSAM. Using it, the requesting program gains control through cross memory and data access.

The original RLS design included extensive first time data capture. Each of the RLS subcomponents added many health checks, as well as local recovery routines, which detect and document internal logic errors. If a health check identified an error, an SVC D command is issued with a 0F4 completion code along with a unique return/reason code identifying the error. The local recovery routine would then receive control and issue an SDUMP macro to obtain an SVC dump. This design has proven very successful and has enabled many internal logic errors to be solved with the first occurrence of the problem.

The original RLS design also included system error recovery. Following an internal logic error, most of the RLS subcomponents opted to terminate the SMSVSAM address space in order to prevent possible data integrity problems and correct environmental errors. By terminating the SMSVSAM address space, the *damaged* system environment is cleared and a new *clean* SMSVSAM address space is created. Additionally, the process of terminating and restarting the SMSVSAM address space was automated so that the customer does not have to manually perform these steps. The first exploiter of this resource, CICS, had added some automation to this specific case, receiving system signs of individual regions, so it can act accordingly, closing and reopening files on the failing system.

This idea of terminating the SMSVSAM address space because of environmental problems has not proven to be the best approach. It provided protection, but at the same time, produced availability problems. A logic error that occurs during a single read/write request to a single data set can result in a system outage that impacts all files opened on the system. And as most of the subcomponents adopted the same strategy, it started to terminate the SMSVSAM address space for reasons other than data integrity. Even a display command has the potential for terminating the SMSVSAM address space.

With z/OS V1.R8 the development team redesigned the system error recovery area. Instead of terminating the SMSVSAM address space, due to internal logic errors, it remains up, and the individual function with error is fails. The first time data capture design remains as before. See “OA17734” on page 467 for more detailed information.

Tip: We recommend that you read Chapters 5 and 6 of the IBM Redbooks publication *VSAM Demystified*, SG24-6105, for an introduction to this subject.

Coexistence and migration

See Table 3-1 for information about coexistence and migration for the redesigns of system error recovery area.

Table 3-1 Coexistence table about the redesigned system error recovery area

Element or feature	DFSMSdfp
When change was introduced	z/OS V1R8.0
Applies to migration from	None
Timing	After the first IPL of z/OS V1R8
Target system hardware requirements	None
Target system software requirements	None
Other system (coexistence or fallback) requirements	None
Restrictions	None
System impacts	None

3.6.2 VSAM RLS DIAG command on z/OS V1R8 system

A situation that can arise when using VSAM RLS is that one or more tasks using RLS appear to hang. One of the causes for this is that latches are in contention. Use the following command to display latch contention:

```
D SMS,SMSVSAM,DIAG(CONTENTION)
```

If there are latches in contention the user is informed with detailed information that is intended to allow the problem to be resolved.

In Figure 3-17 we show the result of issuing the command on a system that has no latches in contention, which means that message IGW342I is issued.

```
D SMS,SMSVSAM,DIAG(CONTENTION)
IGW342I VSAM RLS DIAG STATUS (V.01) 335
NO CONTENTION BY REGISTERED RESOURCES EXISTS
```

Figure 3-17 D SMS, SMSVSAM,DIAG(CONTENTION) command and output no latches

If there are latches in contention, message IGW343I is issued.

In Figure 3-18 we show sample data resulting from the command when a latch was found to be in contention. This information may be used to identify the holder from the ASID shown, which may lead to a decision as to whether the holder could be cancelled or whether it would be necessary to wait.

```
IGW343I VSAM RLS DIAG STATUS (V.01) 035
|----RESOURCE---| |------ WAITER ------| |--HOLDER---| ELAPSED
TYPE        ID    JOB NAME ASID   TASK   ASID   TASK     TIME
-------- -------- -------- ---- -------- ---- -------- --------
LATCH    7F38B928 SMSVSAM  000A 007CBB58 000A 007CBE88 00:01:07
    DESCRIPTION: IDAVTCP  - VTCPage Storage Pool
```

Figure 3-18 Example of information available if RLS latch contention is detected

3.6.3 VSAM RLS DIAG command on pre-z/OS V1R8 systems

There is no support for the VSAM RLS DIAG command on lower level systems, but toleration maintenance is available to handle the command if it should be issued. In Figure 3-17 on page 26 we show the toleration maintenance available for systems prior to z/OS V1R8.

Table 3-2 RLS DIAG command toleration maintenance

APAR	Description	Effect
OA14568	Eliminates confusing message relating to another internal function	Generates message IGW495I D SMS,SMSVSAM,DIAG COMMAND FUNCTION HAS FOUND THE DIAG TABLE HAS NO ENTRIES

In Figure 3-19 we show the result of issuing the command on a z/OS V1R7 system that has VSAM RLS enabled and has the PTF for APAR OA14568 installed.

```
D SMS,SMSVSAM,DIAG(CONTENTION)
IGW495I D SMS,SMSVSAM,DIAG COMMAND FUNCTION HAS FOUND 074
THE DIAG TABLE HAS NO ENTRIES.
```

Figure 3-19 D SMS, SMSVSAM,DIAG(CONTENTION) command and output on z/OS V1R7

3.6.4 VSAM RLS performance enhancement

z/OS V1R8 introduced a performance enhancement that skips cleaning up the XES lock table when the SMSVSAM address space is terminated. This function is automatically available, but can be triggered by using the following command if necessary:

```
V SMS,SMSVSAM,TERMINATESERVER
```

If the command is issued on a z/OS V1R8 system, since RLS is a sysplex-wide function, it affects other systems, and toleration maintenance is required for this.

Toleration on pre-z/OS V1R8 systems

There is no provision for the performance improvement on lower level systems, but toleration maintenance is available. In Figure 3-3 on page 16 we show the toleration maintenance available for systems prior to z/OS V1R8.

Note: The current status of these and related APARs should be used.

Table 3-3 RLS XES cleanup performance improvement toleration maintenance

APAR	Description	Effect
OA13332 - See “OA13332” on page 458.	Prerequisite for OA11708	Adds SYSPLEX support required for OA11708
OA11708 - See “OA11708” on page 457.	Reduces termination delays when V SMS,SMSVSAM,TERMINATESERVER command is issued	Manages locks more rapidly
OA14666 - See “OA14666” on page 459.	Post requisite for OA11708	Completes OA11708

3.6.5 VSAM RLS use of 64-bit data buffers

VSAM RLS has specific requirements for use of 64-bit buffers (over the 2 GB bar). These are discussed in detail in *z/OS V1R7 DFSMS Technical Update*, SG24-7225, but an update to what is published there is needed.

The basic requirements are:

- Any data set to be used for RLS above the bar buffers must be associated with a SMS data class with the RLS ABOVE THE BAR attribute set to Y.
- SYS1.PARMLIB member IGDSMSxx must include a non-zero value for RlsAboveThebarMaxPoolSize, as shown in Figure 3-20 on page 29:

  ```
  RlsAboveThebarMaxPoolSize = 500
  ```

- SYS1.PARMLIB member IGDSMSxx must include Rls_MaxCfFeatureLevel = A. This is what has changed from the comment in *z/OS V1R7 DFSMS Technical Update*, SG24-7225, where a value of Z was shown as allowable.
- SYS1.PARMLIB member IGDSMSxx may include a non-zero value for RlsFixedPoolSize. Figure 3-20 on page 29 shows RlsFixedPoolSize = 50
- In Figure 3-20 on page 29 we show the output from the D SMS,OPTIONS command showing the RLS requirements set.

```
D SMS,OPTIONS
IGD002I 19:54:14 DISPLAY SMS 857
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
ACSDEFAULTS = NO        PDSESHARING = EXTENDED
OVRD_EXPDT = NO         SYSTEMS = 8
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 256MB
USE_RESOWNER = YES      RLS_MAX_POOL_SIZE = 100MB
RLSINIT = YES           RLSTMOUT = 0
COMPRESS = GENERIC      LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
QTIMEOUT = 300          TVSNAME = 064
AKP = 1000              TV_START_TYPE = WARM
MAXLOCKS = (0,0)
CICSVR_INIT = YES       CICSVR_DSNAME_PREFIX = DWWUSER.V3R1M0
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = A
RlsAboveThebarMaxPoolSize = 500
RlsFixedPoolSize = 50
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2000M
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
GDS_RECLAIM = YES       DSSTIMEOUT = 0
BLOCKTOKENSIZE = NOREQUIRE       FAST_VOLSEL = ON
IGD002I 19:54:14 DISPLAY SMS
TRACE    = OFF    SIZE = 1024K     TYPE = ALL
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = OFF  SMSSJF = OFF  SMSSSI = OFF  ACSINT = OFF
    OPCMD  = OFF  CONFC  = OFF  CDSC   = OFF  CONFS  = OFF
    MSG    = OFF  ERR    = OFF  CONFR  = OFF  CONFA  = OFF
    ACSPRO = OFF  IDAX   = OFF  DISP   = OFF  CATG   = OFF
    VOLREF = OFF  SCHEDP = OFF  SCHEDS = OFF  VTOCL  = OFF
    VTOCD  = OFF  VTOCR  = OFF  VTOCC  = OFF  VTOCA  = OFF
    RCD    = OFF  DCF    = OFF  DPN    = OFF  TVR    = OFF
    DSTACK = OFF  UAFF   = OFF  DEBUG  = ON
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
PDSE1_BUFFER_BEYOND_CLOSE = NO
GDS_RECLAIM = YES       DSSTIMEOUT = 0
BLOCKTOKENSIZE = NOREQUIRE       FAST_VOLSEL = ON
IGD002I 19:54:14 DISPLAY SMS
TRACE    = OFF    SIZE = 1024K     TYPE = ALL
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = OFF  SMSSJF = OFF  SMSSSI = OFF  ACSINT = OFF
    OPCMD  = OFF  CONFC  = OFF  CDSC   = OFF  CONFS  = OFF
    MSG    = OFF  ERR    = OFF  CONFR  = OFF  CONFA  = OFF
    ACSPRO = OFF  IDAX   = OFF  DISP   = OFF  CATG   = OFF
    VOLREF = OFF  SCHEDP = OFF  SCHEDS = OFF  VTOCL  = OFF
    VTOCD  = OFF  VTOCR  = OFF  VTOCC  = OFF  VTOCA  = OFF
    RCD    = OFF  DCF    = OFF  DPN    = OFF  TVR    = OFF
    DSTACK = OFF  UAFF   = OFF  DEBUG  = ON
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
```

Figure 3-20 D SMS,OPTIONS showing RLS above 2 GB bar requirements set

3.6.6 DSS LOGICAL COPY WARNING

A VSAM KSDS with data and index components on different volumes may lose its SMS RLSDATA information during DSS COPY. This may result in the correct attributes for VSAM

RLS use in general, and the 64-bit requirements in particular being dropped.

Refer to “OA18319” on page 462 for further information.

3.6.7 Monitoring the VSAM RLS use of 64-bit data buffers

You can monitor the use of VSAM RLS 64-bit data buffers, and check on the status of data classes eligible and which data sets are eligible.

You can set up recording of SMF type 42 records then check the data returned in the appropriate subtype records as listed below. For more information refer to Chapter 7 under “Monitoring the use of 64-bit data buffers,” in the manual *DFSMS Using the New Functions,* SC26-7473.

If you have access to IBM RMF™ then it is preferable to use it to look at the various statistics provided. If you do not have RMF then the sample programs provided in the appendix and illustrated below can be used to gain insight into what is happening on the system.

Note: Before using any of the SMF records for analysis, review the current status of APAR OA17704, as shown in “OA11708” on page 457.

SMF records for analysis

The SMF records for analysis are:

- SMF Record type 42 Subtype 16 (Data Set Summary) indicates whether a data set is enabled for 64-bit data buffers and whether it uses storage above 2-gigabytes. Any data set that you want SMF to monitor usage for must be nominated by means of the command:

  `V SMS,MONDS(`*`datasetname`*`),ON`

  Where *datasetname* is a fully qualified name or one with one trailing asterisk (*).

  For details on these SMF records for analysis refer to 3.6.8, “SMF Record type 42 Subtype 16 information example” on page 30.

- SMF Record type 42 Subtype 18 (CF Cache Partition Usage) provides statistics on the use of coupling facility cache storage classes. For details on these SMF records for analysis see “SMF Record type 42 Subtype 18 information example” on page 36.

- SMF Record type 42 Subtype 19 (LRU record) provides statistics for data buffers above and below 2 gigabytes. For details on these SMF records for analysis see “SMF Record type 42 Subtype 19 information example” on page 44.

3.6.8 SMF Record type 42 Subtype 16 information example

SMS writes SMF Record type 42 subtype 16 to document VSAM RLS data set activity.

Note: Before attempting to use SMF type 42 subtype 16 records in relation to 64-bit buffer usage, APAR OA17704 (see “OA17704” on page 456) should be installed.

We wrote a simple program called SMF42TG to scan the SMF records and summarize activity. The program itself and how to construct it are documented in “SMF record type 42 subtype 16 data display program” on page 522.

In Figure 3-21 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT42TG program. The JCL could be changed to make a permanent extract of the SMF data, or to read of an already created SMF data extract.

```
//MHLRES1L JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN2
//*DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS.G3196V00
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=32760,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(42(16)))
/*
//  EXEC PGM=SMF42TG
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF42TG.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 3-21 JCL to extract SMF records and run program SMF42TG

Note: As a result of the problems described by APAR OA17704 (see “OA17704” on page 456) there may be abnormal displays in the output. The fact that a record is not correctly built in the SMF records is usually indicated by the cycle time records being zero or abnormally large.

There are four sets of records contained in the SMF records type 42 subtype 16, two sections relating to below the 2 GB bar and two relating to above the 2 GB bar. Hence, a large amount of output could be produced. Samples of each type of output are presented from a run of the program. The ability to use buffers above the 2 GB bar was made available with z/OS V1R7, but the SMF data about activity was not available until z/OS V1R8.

Program SMF42TG output uses headings for the output that are taken from the SMF records that it was obtained from. To assist with interpretation of the output, we provide extracts from parts of the SMF mapping macro IGWSMF. In order to distinguish between the SMF record type 42 subtype 16 fields from below and above the 16 GB bar, those relating to below the 2 GB bar are prefixed SMF42, and those relating to above the 2 GB bar are prefixed SMF2A (2A is hexidecimal for 42).

The records (in each group) consist of SYSPLEX-wide data and Data set/MVS system summary data.

For below the 2 GB bar, the SYSPLEX-wide records come in group SMF42GA1 and the MVS system records come in group SMF42G1A. Similar groups for above the 2 GB bar are SMF2AGA1 and SMF2AG1A.

Example from group SMF42GA1 - SYSPLEX-wide summary

In Figure 3-22 we show part of the output from running SMF42GA1.

```
SMF42GA1 SYSPLEX D/S RESPONSE SUMMARY SET #:      0000001
YYYYDDD HH:MM:SS SMF42GAA SMF42GAB                                SMF42GAC
2007080 20:30:00 0000600  HSM.BCDS.RLS.DATA                       HSM.BCDS.RLS
        SMF42GAE                SMF42GAF                    SMF42GAH                      SMF42GAI SMF42GAJ
        SC54GRT                 CSERLS                      RLS_CACHE                     UNKNOWN  GT4KNOTACT
        SMF42GAP         SMF42GCA SMF42GCB SMF42GCC SMF42GCD SMF42GCE SMF42GCF SMF42GCI SMF42GCL SMF42GCM SMF42GCN
                         0000009  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
.
.
.
```

Figure 3-22 SMF42TG output for group SMF41GA1

A number of sets of output may be produced depending on the configuration of the SYSPLEX.

The specific meanings of the fields can be found in the manual *MVS System Management Facilities (SMF)*, SA22-7630, or can be interpreted from the IGWSMF macro in SYS1.MACLIB on the system, which may be more up to date than the version in the manual.

In Figure 3-23 on page 33 we show part of the IGWSMF macro that relates to group SMF42GA1 records.

In Figure 3-22:

- Fields SMF42GAA, SMF42GAB, and SMF42GAC show, respectively, the interval time, data set name, and VSAM sphere name.

  The SMF42GAA value can be used as an indicator that the record is valid. A value of 0 or a very high number is likely to indicate that the SMF record may be suspect.
- Fields SMF42GAE, SMF42GAF, SMF42GAH, SMF42GAI, and SMF42GAJ show, respectively, the Storage Class Name, the CACHESET name, the DFP CACHE STRUCTURE name, the indicator of the component being processed, and CF processing status indictors.

  The SMF42GAI field may no longer be supported. All test runs show it as a field of x'00000000').
- The remaining fields can be interpreted from the IGWSMF macro.
- Information is provided in Appendix B, "Code samples" on page 477, where the program is described on how to alter it to process other SMF fields from those initially provided.

```
*   ---------------------------------------------------------------
*    CF DATASET RESPONSE TIME SUMMARY (SMF42 SUBTYPE 16)
*   ---------------------------------------------------------------
SMF42OGA DSECT ,
SMF42GA1 DS   0CL1472              SYSPLEX WIDE SUMMARY        @21A
SMF42GAA DS   1FL4                 INTERVAL LENGTH. THIS IS THE
*                                  TOTAL TIME OF THE MEASUREMENT
*                                  PERIOD. (IN SECONDS)        @10A
SMF42AOO DS   CL12                 RESERVED                    @MDK
*
SMF42GAB DS   CL44                 DATA SET NAME               @10A
SMF42AO1 DS   1FL4                 RESERVED                    @10A
*
SMF42GAC DS   CL44                 VSAM SPHERE NAME            @10A
SMF42AO2 DS   1FL4                 RESERVED                    @10A
*
SMF42GAD DS   1FL2                 LENGTH OF THE STORAGE CLASS
*                                  NAME                        @10A
SMF42GAE DS   CL30                 STORAGE CLASS NAME          @10A
*
SMF42AO3 DS   1FL2                 CACHESETNAME LENGTH         @10A
SMF42GAF DS   CL30                 CACHESET NAME               @10A
*
SMF42GAG DS   CL2                  RESERVED                    @10A
SMF42GAH DS   CL30                 DFP CACHE STRUCTURE NAME    @10A
*
SMF42GAI DS   1FL4                 INDICATOR OF COMPONENT BEING
*                                  PROCESSED                   @10A
SMF42GAIO EQU   X'80000000'        DATA COMPONENT              @10A
SMF42GAI1 EQU   X'40000000'        INDEX COMPONENT             @10A
*                                                              @10A
SMF42GAJ DS   CL12                 INDICATES DFSMS GREATER     @P5A
*                                  THAN 4K CF CACHING STATUS   @P5A
*                                  VALUES ARE ALL, NONE,       @P5A
*                                  UPDATESONLY, GT4KNOTACT     @P5A
SMF42GAK DS   1FL4                 NUMBER OF LOCK REQUESTS
*                                  PROCESSED                   @10A
SMF42GAL DS   1FL4                 NUMBER OF TRUE CONTENTION
*                                  LOCK REQUESTS               @10A
SMF42GAM DS   1FL4                 NUMBER OF FALSE CONTENTION
*                                  LOCK REQUESTS               @10A
SMF42GZ1 DS   1FL2                 SMSDIRECTWEIGHT             @10A
SMF42GZ2 DS   1FL2                 SMSSEQUENTIALWEIGHT         @10A
.
.
.
.
```

Figure 3-23 IGWSMF record extract for group SMF42GA1

Example from group SMF42G1A - data set/MVS summary

In Figure 3-24 we show part of the output from running SMF42GA1.

```
SMF42G1A DATA SET/MVS SYSTEM SUMMARY SET #: 0000001
YYYYDDD HH:MM:SS SMF42GBA SMF42GBB                                SMF42GBC
2007080 20:30:00 0000600  HSM.BCDS.RLS.DATA                      HSM.BCDS.RLS
        SMF42GBE                    SMF42GBF                   SMF42GBH                      SMF42GBI SMF42AO9
        SC54GRT                     CSERLS                     SC64                          UNKNOWN  GT4KNOTACT
        SMF42GBP         SMF42GIA SMF42GIB SMF42GIC SMF42GID SMF42GIE SMF42GIF SMF42GIL SMF42GIR SMF42GIS
                         0000009  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000 .
.
.
```

Figure 3-24 SMF42TG output for group SMF41G1A

A number of sets of output may be produced depending on the configuration of the SYSPLEX.

The specific meaning of the fields can be found in the manual *MVS System Management Facilities (SMF)*, SA22-7630, or can be interpreted from the IGWSMF macro in SYS1.MACLIB on the system, which may be more up to date than the version in the manual.

In Figure 3-25 we show part of the IGWSMF macro that relates to group SMF42G1A records.

```
SMF42G1A DS   0CL1488              DATA SET/MVS SYSTEM SUMMARY @21A
*                                 SECTION                     @21C
SMF42GBA DS   1FL4                INTERVAL LENGTH. THIS IS THE
*                                 TOTAL TIME OF THE MEASUREMENT
*                                 PERIOD. (IN SECONDS)        @10A
SMF42A05 DS   CL12                RESERVED                    @10A
*
SMF42GBB DS   CL44                DATA SET NAME               @10A
SMF42A06 DS   1FL4                RESERVED                    @10A
*
SMF42GBC DS   CL44                VSAM SPHERE NAME            @10A
SMF42A07 DS   1FL4                RESERVED                    @10A
*
SMF42GBD DS   1FL2                LENGTH OF THE STORAGE CLASS
*                                 NAME                        @10A
SMF42GBE DS   CL30                STORAGE CLASS NAME          @10A
*
SMF42A08 DS   1FL2                CACHESETNAME LENGTH         @10A
SMF42GBF DS   CL30                CACHESET NAME               @10A
*
SMF42A12 DS   1FL2                RESERVED                    @10A
SMF42GBG DS   CL30                DFP CACHE STRUCTURE NAME    @10A
*
SMF42GBH DS   CL8                 MVS SYSTEM NAME             @10A
SMF42A11 DS   CL8                 RESERVED                    @10A
*
SMF42GBI DS   1FL4                INDICATOR OF COMPONENT BEING
*                                 PROCESSED                   @10A
SMF42GBI0 EQU   X'80000000'       DATA COMPONENT              @10A
SMF42GBI1 EQU   X'40000000'       INDEX COMPONENT             @10A
SMF42A09 DS   CL12                INDICATES DFSMS GREATER     @P5A
*                                 THAN 4K CF CACHING STATUS   @P5A
*                                 VALUES ARE ALL, NONE,       @P5A
*                                 UPDATESONLY, GT4KNOTACT     @P5A
SMF42GBK DS   1FL4                NUMBER OF LOCK REQUESTS
*                                 PROCESSED                   @10A
SMF42GBL DS   1FL4                NUMBER OF TRUE CONTENTION
*                                 LOCK REQUESTS               @10A
SMF42GBM DS   1FL4                NUMBER OF FALSE CONTENTION
*                                 LOCK REQUESTS               @10A
SMF42GZ8 DS   1FL2                SMSDIRECTWEIGHT             @10A
SMF42GZ9 DS   1FL2                SMSSEQUENTIALWEIGHT         @10A
*
.
```

Figure 3-25 IGWSMF record extract for group SMF42G1A

In Figure 3-25:

- Fields SMF42GBA, SMF42GBB, and SMF42GBC show, respectively, the interval time, data set name, and VSAM sphere name.

  The SMF42GBA value can be used as an indicator that the record is valid. A value of 0 or a very high number is likely to indicate that the SMF record may be suspect.

- Fields SMF42GBE, SMF42GBF, SMF42GBH, SMF42GBI, and SMF42A09 show, respectively, the Storage Class Name, the CACHESET name, the DFP CACHE

STRUCTURE name, the indicator of the component being processed, and CF processing status indictors.

The SMF42GBI field may no longer be supported. All test runs show it as a field of x'00000000').

- The remaining fields can be interpreted from the IGWSMF macro.

Information is provided in Appendix B, "Code samples" on page 477, where the program is described on how to alter it to process other SMF fields from those initially provided.

Example from group SMF2AGA1 - SYSPLEX-wide summary

Because of the problems with the content of the SMF records, as described in 3.6.8, "SMF Record type 42 Subtype 16 information example" on page 30, actual output from the program for above the 2 GB bar is not shown. If the program is run after the fixes have been implemented, the output appears the same as from the below the 2 GB bar examples above, but with 2A substituted for 42.

Example from group SMF2AG1A - data set/MVS summary

Because of the problems with the content of the SMF records, as described in 3.6.8, "SMF Record type 42 Subtype 16 information example" on page 30, actual output from the program for above the 2 GB bar is not shown. If the program is run after the fixes have been implemented, the output appears the same as from the below the 2 GB bar examples above, but with 2A substituted for 42.

3.6.9 SMF Record type 42 Subtype 18 information example

SMS writes SMF Record type 42 subtype 18 to document cache use by VSAM RLS.

We have written a sample program called SMF42TI to scan the SMF records and summarize activity. The program itself and how to construct it is documented in "SMF Record type 42 subtype 18 data display program" on page 534.

In Figure 3-26 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT42TI program. The JCL could be changed to make a permanent extract of the SMF data, or to read an already created SMF data extract.

```
//MHLRES1L JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//*DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS.G3196V00
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=32760,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(42(18)))
/*
//  EXEC PGM=SMF42TI
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF42TI.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 3-26 JCL to extract SMF records and run program SMF42TI

There are three different sets of output processed by SMF42TI if data is found in the records. In Table 3-4 we show the relationship between the three types of data reported on by SMF42TI.

In Figure 3-27 on page 39 we show example output from running program SMF42TI.

Table 3-4 SMF42TI report summary information

Data description	SMF structure name	Report identifier
Coupling Facility Cache Partition Summary (SMF42 subtype 18) - data for this partition across all cache structures	SMF420IA	SMF41IA1
Coupling Facility Cache Partition Summary (SMF42 subtype 18) - data for this partition for a singe cache structure	SMF420IC	SMF42IC1
Coupling Facility Cache Partition Summary (SMF42 subtype 18) - directory/element ratio data for this cache structure	SMF420IE	SMF42IE1

```
SMF TYPE 42 S/TYPE 18 RECS. COLS USE SMF NAMES
SMF42IA1 TOTALS SET #:   0000001
HH:MM:SS YYYYDDD SMF42IAA SMF42IBG          SMF42IBH SMF42IO1          SMF42IAD SMF42IAF SMF42IAG SMF42IAH SMF42IAI SMF42IAJ
11:00:00 2007074 0000599  DIRECT            0000001  PARTITION  #      0000001  0000000  0000000  0000000  0000000  0000000
                 SMF42IAK SMF42IAL SMF42IAM SMF42IAN SMF42IAO SMF42IAP SMF42IAQ SMF42IAR SMF42IAS SMF42IAT SMF42IAU SMF42IAV
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
                 SMF42IAW SMF42IAX SMF42IAY SMF42IAZ SMF42IBA SMF42IBB SMF42IBC SMF42IBD SMF42IBE SMF42IBF
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF42IA1 TOTALS SET #:   0000002
HH:MM:SS YYYYDDD SMF42IAA SMF42IBG          SMF42IBH SMF42IO1          SMF42IAD SMF42IAF SMF42IAG SMF42IAH SMF42IAI SMF42IAJ
11:00:00 2007074 0000599  SEQUENTIAL        0000001  PARTITION  #      0000002  0000000  0000000  0000000  0000000  0000000
                 SMF42IAK SMF42IAL SMF42IAM SMF42IAN SMF42IAO SMF42IAP SMF42IAQ SMF42IAR SMF42IAS SMF42IAT SMF42IAU SMF42IAV
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
                 SMF42IAW SMF42IAX SMF42IAY SMF42IAZ SMF42IBA SMF42IBB SMF42IBC SMF42IBD SMF42IBE SMF42IBF
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
.
.
SMF42IC1 TOTALS SET #:   0000001
HH:MM:SS YYYYDDD SMF42ICA SMF42ICB                   SMF42IDG          SMF42IDH SMF42ICD SMF42ICF SMF42ICG SMF42ICH
11:00:00 2007074 0000599  RLS_CACHE                 DIRECT            0000001  0000001  0000000  0000000  0000000
                 SMF42ICI SMF42ICJ SMF42ICK SMF42ICL SMF42ICM SMF42ICN SMF42ICO SMF42ICP SMF42ICQ SMF42ICR SMF42ICS
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
                 SMF42ICT SMF42ICU SMF42ICV SMF42ICW SMF42ICX SMF42ICY SMF42ICZ SMF42IDA SMF42IDB SMF42IDC SMF42IDD SMF42IDE
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF42IC1 TOTALS SET #:   0000002
HH:MM:SS YYYYDDD SMF42ICA SMF42ICB                   SMF42IDG          SMF42IDH SMF42ICD SMF42ICF SMF42ICG SMF42ICH
11:00:00 2007074 0000599  RLS_CACHE                 SEQUENTIAL        0000001  0000002  0000000  0000000  0000000
                 SMF42ICI SMF42ICJ SMF42ICK SMF42ICL SMF42ICM SMF42ICN SMF42ICO SMF42ICP SMF42ICQ SMF42ICR SMF42ICS
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
                 SMF42ICT SMF42ICU SMF42ICV SMF42ICW SMF42ICX SMF42ICY SMF42ICZ SMF42IDA SMF42IDB SMF42IDC SMF42IDD SMF42IDE
                 0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
.
.
SMF42IE1 TOTALS SET #:   0000001
HH:MM:SS YYYYDDD SMF42IEA SMF42IEB                   SMF42IEC SMF42IED SMF42IEE SMF42IEF SMF42IEG SMF42IEH SMF42IEI SMF42IEJ
14:00:00 2007074 0000600  RLS_CACHE                 0000000  0000000  0000252  0000255  0000252  0000255  0000252  0000255

SMF42IE1 TOTALS SET #:   0000002
HH:MM:SS YYYYDDD SMF42IEA SMF42IEB                   SMF42IEC SMF42IED SMF42IEE SMF42IEF SMF42IEG SMF42IEH SMF42IEI SMF42IEJ
14:00:00 2007074 0000600  RLS_CACHE                 0000000  0000000  0000252  0000255  0000252  0000255  0000252  0000255
.
.
```

Figure 3-27 SMF42TI program sample output

Program SMF42TI output uses headings for the output that is taken from the SMF records that it was obtained from.

The specific meaning of the fields can be found in *MVS System Management Facilities (SMF)*, SA22-7630, or can be interpreted from the IGWSMF macro in SYS1.MACLIB on the system, which may be more up to date than the version in the manual.

In Figure 3-27 the output is shown in three groups. The meanings of the fields are similar, and we recommend understanding the output review of the records in SMFMAN or in the IGWSMF macro, as the values presented depend on the system configuration.

- In groups SMF41IA1, SMF42IC1, and SMF42IE1, respectively, fields SMF42IAA, SMF42IAC, and SMF42IAE show the interval time. The value shown can be used as an indicator that the record is valid. A value of 0 or a very high number is likely to indicate that the SMF record may be suspect.

- For group SMF42IA1 in Table 3-5 we show the interpretation of the field names as taken from the IGWSMF macro.

Table 3-5 SMF42IA1 field interpretation

Field name	Meaning
SMF42IBG	Partition type
SMF42IBH	SMS specified cache weight
SMF42IO1	Partition name word
SMF42IAD	CF cache partition number

- For group SMF42IC1 in Table 3-6 we show the interpretation of the field names as taken from the IGWSMF macro.

Table 3-6 SMF42IC1 field interpretation

Field name	Meaning
SMF42ICB	cache structure name
SMF42IDG	partition type
SMF42IDH	SMS specified cache weight
SMF42ICD	CF cache partition number

- For group SMF42IE1 in Table 3-7 we show the interpretation of the field names as taken from the IGWSMF macro.

Table 3-7 SMF42IE1 field interpretation

Field name	Meaning
SMF42IEB	SMS cache structure name
SMF42IEC	Number of times the directory portion of the dir/ele ratio changes
SMF42IED	Number of times the element portion of the dir/ele ratio changes
SMF42IEE	Low ratio value in this interval - directory part
SMF42IEF	Low ratio value in this interval - element part
SMF42IEG	High ratio value in this interval - directory part
SMF42IEH	High ratio value in this interval - element part
SMF42IEI	Current ratio value in this interval - directory part

Field name	Meaning
SMF42IEJ	Current ratio value in this interval - element part

- The remaining fields can be interpreted from the IGWSMF macro.
- Information is provided in Appendix B, “Code samples” on page 477, where the program is described for how to alter it to process other SMF fields from those initially provided.

To assist with interpretation of the output in this document, we provide extracts from parts of the SMF mapping macro IGWSMF (Figure 3-28 on page 42 through to Figure 3-29 on page 43).

Note: Changes may be made to the SMF mapping macro to correspond with changes to the data recorded, so for accurate analysis, the actual IGWSMF macro should be used.

```
*   ---------------------------------------------------------------
*    COUPLING FACILITY CACHE PARTITION SUMMARY (SMF42 SUBTYPE 18)
*    DATA FOR THIS PARTITION ACROSS ALL CACHE STRUCTURES.
*   ---------------------------------------------------------------
SMF420IA DSECT ,
SMF42IA1 DS   0D             CF CACHE PARTITION TOTALS
*                                   SECTION                    @10A
SMF42IAC DS   0CL196                CACHE BUFFER POOLS
*                                    THERE ARE 23 POOLS:
*                                    11 SEQUENTIAL, 11 DIRECT,
*                                    AND 1 DEFAULT
*                                    THERE IS A SEPERATE SECTION FOR
*                                    EACH PARTITION.  THIS MEANS THAT
*                                    THERE WILL BE A MINUMUM OF 23
*                                    TOTALS SECTIONS AND 23 SUMMARY
*                                    SECTIONS.
SMF42IAA DS    1FL4                 INTERVAL LENGTH. THIS IS THE
*                                    TOTAL TIME OF THE MEASUREMENT
*                                    PERIOD. (IN SECONDS)       @10A
SMF42IOO DS    CL12                 RESERVED                   @10A
*
SMF42IAB DS    CL32                 RESERVED                   @10A
*
SMF42IBG DS    CL16                 PARTITION TYPE             @10A
*                                   'SEQUENTIAL' DATA ONLY     @10A
*                                   'DIRECT' DATA ONLY         @10A
*                                   'COMBINED' DATA            @10A
*
SMF42IBH DS    1FL4                 SMS SPECIFIED CACHE WEIGHT @10A
SMF42IO1 DS    CL12                 PARTITIONNAMEWORD          @10A
SMF42IAD DS    1FL4                 COUPLE FACILITY CACHE
*                                   PARTITION NUMBER           @10A
SMF42IAE DS    1FL4                 STATUS CONDITIONS          @10A
SMF42IAF DS    1FL4                 READ HIT COUNTER           @10A
SMF42IAG DS    1FL4                 READ MISS DIRECTORY HIT
*                                   COUNTER                    @10A
SMF42IAH DS    1FL4                 READ MISS ASSIGN SUPP
*                                   COUNTER                    @10A
SMF42IAI DS    1FL4                 READ MISS, NAME ASSIGNED
*                                   COUNTER                    @10A
SMF42IAJ DS    1FL4                 READ MISS TARGET STORAGE
*                                   CLASS FULL COUNTER         @10A
SMF42IAK DS    1FL4                 WRITE HIT CHANGE BIT 0
*                                   COUNTER                    @10A
SMF42IAL DS    1FL4                 WRITE HIT CHANGE BIT 1
*                                   COUNTER                    @10A
SMF42IAM DS    1FL4                 WRITE MISS NOT REGISTERED
*                                   COUNTER                    @10A
SMF42IAN DS    1FL4                 WRITE MISS INVALID STATE
*                                   COUNTER                    @10A
SMF42IAO DS    1FL4                 WRITE MISS TARGET STORAGE CLASS
*                                   FULL COUNTER               @10A
```

Figure 3-28 IGWSMF macro part of DSECT SMF420IA section relating to subtype 18

```
*   ---------------------------------------------------------------
*    COUPLING FACILITY CACHE PARTITION SUMMARY (SMF42 SUBTYPE 18)
*    DATA FOR THIS PARTITION FOR A SINGLE CACHE STRUCTURE
*   ---------------------------------------------------------------
SMF42OIC DSECT ,
SMF42IC1 DS    0D                  CF CACHE PARTITION SUMMARY
*                                  SECTION                    @10A
SMF42ICC DS    0CL196              CACHE BUFFER POOLS
*                                  THERE ARE 23 POOLS:
*                                  11 SEQUENTIAL, 11 DIRECT,
*                                  AND 1 DEFAULT
*                                  THERE IS A SEPERATE SECTION FOR
*                                  EACH PARTITION.  THIS MEANS THAT
*                                  THERE WILL BE A MINUMUM OF 23
*                                  TOTALS SECTIONS AND 23 SUMMARY
*                                  SECTIONS.
SMF42ICA DS    1FL4                INTERVAL LENGTH. THIS IS THE
*                                  TOTAL TIME OF THE MEASUREMENT
*                                  PERIOD. (IN SECONDS)       @10A
SMF42IO2 DS    CL12                RESERVED                   @10A
*
SMF42IO4 DS    1FL2                RESERVED                   @10A
SMF42ICB DS    CL30                CACHE STRUCTURE NAME       @10A
*
SMF42IDG DS    CL16                PARTITION TYPE             @10A
*                                  'SEQUENTIAL' DATA ONLY     @10A
*                                  'DIRECT' DATA ONLY         @10A
*                                  'COMBINED' DATA            @10A
*
SMF42IDH DS    1F                  SMS SPECIFIED CACHE WEIGHT @10A
SMF42IO3 DS    CL12                RESERVED                   @10A
*
SMF42ICD DS    1FL4                COUPLE FACILITY CACHE
*                                  PARTITION NUMBER           @10A
SMF42ICE DS    1FL4                STATUS CONDITIONS          @10A
SMF42ICF DS    1FL4                READ HIT COUNTER           @10A
SMF42ICG DS    1FL4                READ MISS DIRECTORY HIT
*                                  COUNTER                    @10A
SMF42ICH DS    1FL4                READ MISS ASSIGN SUPP
*                                  COUNTER                    @10A
SMF42ICI DS    1FL4                READ MISS, NAME ASSIGNED
*                                  COUNTER                    @10A
SMF42ICJ DS    1FL4                READ MISS TARGET STORAGE
*                                  CLASS FULL COUNTER         @10A
SMF42ICK DS    1FL4                WRITE HIT CHANGE BIT 0
*                                  COUNTER                    @10A
SMF42ICL DS    1FL4                WRITE HIT CHANGE BIT 1
*                                  COUNTER                    @10A
SMF42ICM DS    1FL4                WRITE MISS NOT REGISTERED
*                                  COUNTER                    @10A
SMF42ICN DS    1FL4                WRITE MISS INVALID STATE
*                                  COUNTER                    @10A
SMF42ICO DS    1FL4                WRITE MISS TARGET STORAGE CLASS
*                                  FULL COUNTER               @10A
SMF42ICP DS    1FL4                DIRECTORY ENTRY RECLAIM
*                                  COUNTER                    @10A
```

Figure 3-29 IGWSMF macro part of DSECT SMF420IC section relating to subtype 18

```
* ------------------------------------------------------------------
*    Coupling Facility Cache Partition Summary (SMF42 subtype 18)
*    Directory/Element ratio data for this cache structure
* ------------------------------------------------------------------
SMF420IE DSECT ,                                                @11A
SMF42IE1 DS    0CL80              CF Cache Directory/element    @11A
*                                 ratio section                @11A
SMF42IEA DS    1FL4               Interval length              @11A
SMF42IO6 DS    CL12               reserved                     @11A
SMF42IO7 DS    CL2                reserved                     @11A
SMF42IEB DS    CL30               DFSMS cache structure name   @11A
SMF42IEC DS    1FL4               Number of times the          @11A
*                                 directory portion of the     @11A
*                                 dir/ele ratio changes        @11A
SMF42IED DS    1FL4               Number of times the          @11A
*                                 Element   portion of the     @11A
*                                 dir/ele ratio changes        @11A
SMF42IEE DS    1FL4               Low  ratio value in this     @11A
*                                 interval - directory part    @11A
SMF42IEF DS    1FL4               Low  ratio value in this     @11A
*                                 interval - Element   part    @11A
SMF42IEG DS    1FL4               High ratio value in this     @11A
*                                 interval - directory part    @11A
SMF42IEH DS    1FL4               High ratio value in this     @11A
*                                 interval - Element   part    @11A
SMF42IEI DS    1FL4               Current ratio value in this  @11A
*                                 interval - directory part    @11A
SMF42IEJ DS    1FL4               Current ratio value in this  @11A
*                                 interval - element part      @11A
SMF420IE_LEN EQU *-SMF420IE       Length of section            @11A
*
```

Figure 3-30 IGWSMF macro DSECT SMF420IE section relating to subtype 18

3.6.10 SMF Record type 42 Subtype 19 information example

SMS can be configured to write SMF Record type 42 subtype 19 to document cache use.

We have written a simple program called SMF42TJ to scan the SMF records and summarize activity. The program itself and how to construct it is documented in “SMF record type 42 subtype 19 data display program” on page 547.

In Figure 3-31 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT42TJ program. The JCL could be changed to make a permanent extract of the SMF data, or to read from an already created SMF data extract.

```
//MHLRES1L JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//*DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS.G3196V00
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=32760,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(42(19)))
/*
//  EXEC PGM=SMF42TJ
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF42TJJ.PDS
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 3-31 JCL to extract SMF records and run program SMF42TJ

In Figure 3-32 we show example output from running program SMF42TJ.

```
SMF42JNA SYSPLEX SET #:  0000001
HH:MM:SS YYYYDDD SMF42JNA SMF42JN7 SMF42JNG SMF42JNH SMF42JNI SMF42JNJ SMF42JNK SMF42JNL SMF42JNM SMF42JNN
17:30:00 2007073 0000599  0000000  0000040  0000160  0000000  0000000  0000000  0000000  0000000  0000000

SMF42JPA LPAR SET #:     0000001
HH:MM:SS YYYYDDD SMF42JPA SMF42JPB SMF42JP6 SMF42JPG SMF42JPH SMF42JPI SMF42JP2 SMF42JPJ SMF42JPK SMF42JPL SMF42JPM SMF42JPN
17:30:00 2007073 0000600  SC64     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF42JPA LPAR SET #:     0000002
HH:MM:SS YYYYDDD SMF42JPA SMF42JPB SMF42JP6 SMF42JPG SMF42JPH SMF42JPI SMF42JP2 SMF42JPJ SMF42JPK SMF42JPL SMF42JPM SMF42JPN
17:30:00 2007073 0000600  SC70     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF42JPA LPAR SET #:     0000003
HH:MM:SS YYYYDDD SMF42JPA SMF42JPB SMF42JP6 SMF42JPG SMF42JPH SMF42JPI SMF42JP2 SMF42JPJ SMF42JPK SMF42JPL SMF42JPM SMF42JPN
17:30:00 2007073 0000600  SC65     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF42JPA LPAR SET #:     0000004
HH:MM:SS YYYYDDD SMF42JPA SMF42JPB SMF42JP6 SMF42JPG SMF42JPH SMF42JPI SMF42JP2 SMF42JPJ SMF42JPK SMF42JPL SMF42JPM SMF42JPN
17:30:00 2007073 0000599  SC63     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000


SMF2AJNA SYSPLEX SET #:  0000001
HH:MM:SS YYYYDDD SMF2AJNA SMF2AJN7 SMF2AJNG SMF2AJNH SMF2AJNI SMF2AJNJ SMF2AJNK SMF2AJNL SMF2AJNM SMF2AJNN
17:30:00 2007073 0000599  0000000  0000020  0000080  0000000  0000000  0000000  0000000  0000000  0000000

SMF2AJPA LPAR SET #:     0000001
HH:MM:SS YYYYDDD SMF2AJPA SMF2AJPB SMF2AJP6 SMF2AJPG SMF2AJPH SMF2AJPI SMF2AJP2 SMF2AJPJ SMF2AJPK SMF2AJPL SMF2AJPM SMF2AJPN
17:30:00 2007073 0000600  SC64     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF2AJPA LPAR SET #:     0000002
HH:MM:SS YYYYDDD SMF2AJPA SMF2AJPB SMF2AJP6 SMF2AJPG SMF2AJPH SMF2AJPI SMF2AJP2 SMF2AJPJ SMF2AJPK SMF2AJPL SMF2AJPM SMF2AJPN
17:30:00 2007073 0000600  SC70     0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF2AJPA LPAR SET #:     0000003
HH:MM:SS YYYYDDD SMF2AJPA SMF2AJPB SMF2AJP6 SMF2AJPG SMF2AJPH SMF2AJPI SMF2AJP2 SMF2AJPJ SMF2AJPK SMF2AJPL SMF2AJPM SMF2AJPN
17:30:00 2007073 0000600  SC65     0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000

SMF2AJPA LPAR SET #:     0000004
HH:MM:SS YYYYDDD SMF2AJPA SMF2AJPB SMF2AJP6 SMF2AJPG SMF2AJPH SMF2AJPI SMF2AJP2 SMF2AJPJ SMF2AJPK SMF2AJPL SMF2AJPM SMF2AJPN
17:30:00 2007073 0000599  SC63     0000000  0000040  0000000  0000000  0000000  0000000  0000000  0000000  0000000  0000000
```

Figure 3-32 SMF42TJ program sample output

The fields prefixed SMF42 are below the 2 GB bar, and those prefixed SMF2A are above the 2 GB bar. The sample program and resulting output do not cover all the SMF records that are available. The program as documented in the appendix can be adapted to extract and display alternative records, or could be extended to cover more of the records.

The significance of the contents of these records depends on the particular environment being used. These examples demonstrate that SMF was producing records for both below and above the 2 GB bar activity, but as it was done on a low usage system (as far as RLS was concerned), many values show as zero.

Interpretation of the field meaning can be done by following the examples for subtypes 16 and 18 above.

Program SMF42TJ output uses headings for the output that are taken from the SMF records that it was obtained from. To assist with interpretation of the output, we provide extracts from parts of the SMF mapping macro IGWSMF (Figure 3-33 on page 47 and Figure 3-34 on page 49).

Note: Changes may be made to the SMF mapping macro to correspond with changes to the data recorded, so for accurate analysis, the actual IGWSMF macro should be used for reference.

```
* ------------------------------------------------------------
*    Coupling Facility Local Buffer Manager (SMF42 subtype 19)
*    Local Buffer Manager LRU Statistics Summary. sysplex totals
* ------------------------------------------------------------
SMF42OJA DSECT ,
SMF42JA1 DS    0D                      sysplex totals section
*
SMF42JAC DS    0CL1384
*
SMF42JNA DS    1FL4              Interval length. This is the
*                                total time of the measurement
*                                period. (in seconds) total
SMF42JOO DS    CL12              reserved
*
SMF42JNB DS    CL16              reserved
SMF42JND DS    CL8               Reserved
*
SMF42JNE DS    2F
*                                Average Cpu time for all
*                                systems in the sysplex.  This
*                                is the amount of CPU time
*                                (on average) that each system
*                                consumed while processing the
*                                Local Buffer Manager LRU
*                                alogrithm.
*
SMF42JNF DS    2F
*                                Total CPU time for this
*                                record (in milli-seconds)
*                                (sysplex totals)
*
SMF42JNO DS    CL4               Reserved
SMF42JN7 DS    1FL4              Total number of write
*                                requests
*                                (sysplex totals)
SMF42JNG DS    1FL4              Average number of buffer
*                                manager LRU intervals
*                                processed during this period.
*                                (sysplex totals)
*
SMF42JNH DS    1FL4              Total number of buffer
*                                manager LRU intervals
*                                processed during this period.
*                                (across the sysplex)
*
SMF42JNI DS    1FL4              Average
*                                Number of buffer manager
*                                LRU intervals processed
*                                where BMF was over the goal
*                                and normal algorithms were
*                                bypassed to reclaim buffers.
*
SMF42JNJ DS    1FL4              Total
*                                Number of buffer manager
*                                LRU intervals processed
*                                where BMF was over the goal
*                                and normal algorithms were
```

Figure 3-33 IGWSMF macro relating to SMF type 42 subtype 19 (parts of the SMF420JA DSECT)

The above example shows the records for below the 2 GB bar records where they are prefixed SMF42xxx. The IGWSMF macro contains corresponding definitions for the above the bar records where they are prefixed SMF2Axxx.

```
* ------------------------------------------------------------
*  Coupling Facility Local Buffer Manager (SMF42 subtype 19)
*  Local Buffer Manager LRU Statistics Summary. single system
* ------------------------------------------------------------
SMF42OPA DSECT ,
SMF42PA1 DS    0D
*
SMF42PAC DS    0CL800
*
SMF42JPA DS    1FL4              Interval length. This is the
*                                total time of the measurement
*                                period. (in seconds) total
*
SMF42JO1 DS    CL12              Reserved
*
SMF42JPB DS    CL8               MVS system name.
SMF42JPC DS    CL8               Reserved
SMF42JPD DS    CL8               Reserved
*
SMF42JPE DS    CL8               Average CPU time spent by
*                                BMF LRU processing during
*                                each cycle (in milli-seconds)
*
SMF42JPF DS    CL8               Total CPU time for this
*                                record (in milli-seconds)
*
SMF42JP1 DS    CL12              Reserved
SMF42JP6 DS    1FL4              Total number of write
*                                requests
SMF42JPG DS    1FL4              Number of buffer manager
*                                LRU intervals processed
*                                during this period.
*
SMF42JPH DS    1FL4              Number of buffer manager
*                                LRU intervals processed
*                                where BMF was over the goal
*                                and normal algorithms were
*                                bypassed to reclaim buffers.
*
SMF42JPI DS    1FL4              Total number of times that BMF
*                                was called in this interval
*
SMF42JP2 DS    1FL4              Number of buffer manager
*                                LRU intervals processed
*                                where BMF was over the goal,
*                                accelerated the aging, but
*                                did not go into panic mode.
SMF42JPJ DS    1FL4              Buffer manager number of
*                                hits during this interval.
SMF42JPK DS    1FL4              Buffer manager number of
*                                'hits' current percentage
*                                during this interval. Value
*                                for the last LRU cycle before
*                                the SMF record was processed
```

Figure 3-34 GWSMF macro section SMF type 42 subtype 19 (parts of the SMF420PA DSECT)

The above example shows the records for below the 2 GB bar records where they are prefixed SMF42xxx. The IGWSMF macro contains corresponding definitions for the above the bar records where they are prefixed SMF2Axxx.

3.7 Device manager enhancements

z/OS V1R8 introduces several new enhancements to DEVMAN, the device manager address space. These expand upon the DEVMAN CTRACE support and parameters introduced in z/OS V1R7.

3.7.1 Rapid index rebuild

This enhancement significantly improves the performance of the convert routine used when rebuilding a VTOC index. It has been implemented by moving the responsibility for this function into the DEVMAN address space, which now reads the entire contents of the volume's VTOC into a data space with as little as one channel program. Prior to this enhancement, DADSM read and processed each track in the VTOC one at a time, causing significant I/O overhead due to multiple start subchannel operations. The original DADSM code is retained so that if the DEVMAN address space is unavailable the rebuild can still take place.

Maintenance

In order to use this support, APAR PK19625 must be installed for ICKDSF R17. Without this APAR support, the old DADSM method is used. You must also have the PTF for APAR OA17478 installed. It fixes a situation where a BUILDIX failure can occur as well as an ABEND0C4 problem.

Implementing

With DFSMS V1R8 the VTOC convert routine automatically uses DEVMAN if it is available.

ICKDSF R17 is used as before with no change to the command format, but it must have maintenance installed to test for and call the DEVMAN routines. ICKDSF R17 can be common to lower levels of DFSMS where, if the enabling APAR PK19625 is installed, it does not find DEVMAN or may find DEVMAN but not the convert routine, in which case the original DADSM routines are used.

Examples

In Figure 3-35 we show sample JCL to rebuild the indexed VTOC on volume SLD000. This job first converts our existing indexed VTOC to OSFORMAT and then converts back to an indexed VTOC.

```
//INITJO3  JOB SLT,MSGCLASS=H,REGION=1024K
/*JOBPARM SYSAFF=*
//*
//DSF     EXEC PGM=ICKDSF,PARM=NOREPLYU
//SYSPRINT DD SYSOUT=*
//DISK DD UNIT=3390,VOL=SER=SLD000,DISP=OLD
//SYSIN    DD *
  BUILDIX DDNAME(DISK) OS
  BUILDIX DDNAME(DISK) IX
```

Figure 3-35 ICKDSF BUILDIX job

Figure 3-36 is an example of the IEC604I and ICKDSF messages that are seen in SYSLOG as a result of executing the JCL provided in Figure 3-35.

```
$HASP373 INITJO3 STARTED - INIT 1    - CLASS A - SYS CIMN
ICK502I   BUILDIX FUNCTION STARTED
ICK503I   D000 REQUEST RECEIVED TO CONVERT VTOC TO OSFORMAT
ICK504I   D000 VTOC FORMAT IS CURRENTLY IXFORMAT, REQUEST ACCEPTED
IEC604I VTOC CONVERT ROUTINE ENTERED ON D000,SLD000,DOS,DEVMAN
ICK513I   D000 BUILDIX PROCESSING COMPLETED:  VTOC IS NOW IN OSFORMAT
ICK502I   BUILDIX FUNCTION STARTED
ICK503I   D000 REQUEST RECEIVED TO CONVERT VTOC TO IXFORMAT
ICK504I   D000 VTOC FORMAT IS CURRENTLY OSFORMAT, REQUEST ACCEPTED
ICK513I   D000 BUILDIX PROCESSING COMPLETED:  VTOC IS NOW IN IXFORMAT
$HASP395 INITJO3 ENDED
```

Figure 3-36 SYSLOG output from the indexed VTOC build JCL

Message IEC604I has been updated to show when DEVMAN is being used by the addition of DEVMAN to the end of the message.

A typical example of message IEC604I in SYSLOG without using DEVMAN would be as shown in Figure 3-37.

```
IEC604I VTOC CONVERT ROUTINE ENTERED ON D000,SLD000,DOS
```

Figure 3-37 IEC604I VTOC convert message not using DEVMAN

A typical example of message IEC604I in SYSLOG using DEVMAN is as shown in Figure 3-38.

```
IEC604I VTOC CONVERT ROUTINE ENTERED ON D000,SLD000,DOS,DEVMAN
```

Figure 3-38 IEC604I VTOC convert message using DEVMAN

3.7.2 New MODIFY DEVMAN parameters

Three new MODIFY DEVMAN parameters have been added. Their functions as described in *MVS System Commands,* SA22-7627, and are shown in Figure 3-39.

```
F DEVMAN,{DUMP}
         {REPORT}
         {RESTART}

The following are brief descriptions of the parameters.

DUMP
   Captures a diagnostic dump of the device manager address space,
   including the dataspace that contains device manager CTRACE records.

   Note:  The device manager CTRACE component name is SYSDMO. To connect
   device manager to an output writer, use the command TRACE
   CT,ON,COMP=SYSDMO.

REPORT
   Provides basic information about the current activity and module
   levels for the device manager address space.

RESTART
   Terminates the device manager address space and restarts the device
   manager in a new address space. The system allows any subtasks that
   are active in the device manager address space at the time of the
   restart to finish processing. The time allowed for subtask completion
   is determined by using the average time taken by previous subtasks.
   The system abnormally ends any subtasks that do not complete in time
   before it restarts the address space.
```

Figure 3-39 New DEVMAN parameters

There are two primary considerations when using the MODIFY DEVMAN,RESTART command:

- You can use RESTART to avoid IPL when you install software. You can install most device manager APARs by refreshing LLA (F LLA,REFRESH) and then restarting the device manager (F DEVMAN,RESTART).
- You can end and not restart the device manager address space by using the CANCEL DEVMAN command. When you end the address space in this way, you must restart the device manager with the DEVMAN cataloged procedure.

Examples

These commands should only be issued at the direction of the system programmer.

In Figure 3-40 we show an example of the MODIFY DEVMAN,DUMP command.

```
MODIFY DEVMAN,DUMP
IEA794I SVC DUMP HAS CAPTURED: 857
DUMPID=019 REQUESTED BY JOB (DEVMAN  )
DUMP TITLE=COMPONENT=DEVICE MANAGER,COMPID=DF133,ISSUER=DMOVS00
          1,JOBNAME=DEVMAN
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00020)
IEF196I         DSN (SYSDOC.DUMP.DA.D070218.T195615.CIMN.S00019  )
IEF196I         STORCLAS (DUMPSC) MGMTCLAS (        ) DATACLAS (
IEF196I )
IEF196I         VOL SER NOS= DMPDP3
IEF196I IGD104I SYSDOC.DUMP.DA.D070218.T195615.CIMN.S00019   RETAINED,
IEF196I DDNAME=SYS00020
IEA611I COMPLETE DUMP ON SYSDOC.DUMP.DA.D070218.T195615.CIMN.S00019 865
DUMPID=019 REQUESTED BY JOB (DEVMAN  )
FOR ASID (0419)
```

Figure 3-40 Dumping the device manager address space

The REPORT parameter shows the current status of the device manager address space. Figure 3-41 is a report from a system where we would expect no device manager specific functions to be executing.

```
MODIFY DEVMAN,REPORT
DMO0030I DEVICE MANAGER REPORT 341
**** DEVMAN ************************************************
* FMID:  HDZ1180                                           *
* APARS: UA28529  DMOAT002                                 *
* NO SUBTASKS ARE ACTIVE                                   *
**** DEVMAN ************************************************
```

Figure 3-41 DEVMAN report with no subtasks active

If issued at the precise moment a VTOC index is being rebuilt, it shows DEVMANs current status, as seen in Figure 3-42.

```
MODIFY DEVMAN,REPORT
DMO0030I DEVICE MANAGER REPORT 818
**** DEVMAN ************************************************
* FMID:  HDZ1180                                           *
* APARS: UA28529  DMOAT002                                 *
* SUBTASKS:                                                *
* JOBNAME  STARTED   SERVICE   UNIT  STATUS                *
* -------------------------------------------------------- *
* INITJ03  17.52.22  READVTOC  D000  SUBTASK RUNNING       *
**** DEVMAN ************************************************
```

Figure 3-42 DEVMAN report with ICKDSF BUILDIX job running

The device manager address space can be restarted with the MODIFY DEVMAN,RESTART command. Figure 3-43 shows an example of this.

```
MODIFY DEVMAN,RESTART
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 DEVMAN   ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0419.
$HASP250 DEVMAN PURGED -- (JOB KEY WAS C04EEE54)
IEF196I         1 //IEESYSAS JOB TIME=1440,
IEF196I           // MSGLEVEL=1
IEF196I         2 //DEVMAN   EXEC IEESYSAS,PROG=DMOVS001
IEF196I  STMT NO. MESSAGE
IEF196I         2 IEFC001I PROCEDURE IEESYSAS WAS EXPANDED USING SYSTEM
IEF196I LIBRARY SYS1.IBM.PROCLIB
IEF196I         3 XXIEESYSAS PROC PROG=IEFBR14
IEF196I                   00050000
IEF196I         4 XXIEFPROC  EXEC PGM=&PROG
IEF196I                   00100000
IEF196I           XX* THE IEESYSAS PROCEDURE IS SPECIFIED IN THE
IEF196I                   00150000
IEF196I           XX* PARAMETER LIST TO IEEMB881 BY MVS COMPONENTS
IEF196I                   00200000
IEF196I           XX* STARTING FULL FUNCTION SYSTEM ADDRESS SPACES.
IEF196I                   00250000
IEF196I           IEFC653I SUBSTITUTION JCL - PGM=DMOVS001
DMO0010I DEVICE MANAGER INITIALIZATION STARTED
DMO0000I DEVICE MANAGER INITIALIZATION COMPLETE
```

Figure 3-43 Restarting the DEVMAN address space

Migration and coexistence

There are no migration or coexistence considerations when using the new DEVMAN support.

3.8 PDSE enhancements

In z/OS V1R8 PDSE there are the following enhancements:

- Program objects built may use Program Object format 5 (PO5).
- In prior releases of DFSMS, PDSE cached directory buffers in the SYSBMFDS data space. With z/OS V1R8, PDSE has completely eliminated the 31-bit SYSBMF data space and now uses 64-bit buffers. Directory pages are now cached in 64-bit storage in the SMSPDSE and SMSPDSE1 address space. Several of the PDSE control block cell pools have been moved from 31-bit SMSPDSE and SMSPDSE1 address space storage to 64-bit address space storage.
- In this section we cover the enhancements to Partitioned Data Set Extended (PDSE) HSP and directory storage size that are available now and the option to retain buffers beyond close.

Significant changes to PDSE were introduced with z/OS V1R6, and we recommend that you refer to the IBM Redbooks publication *Partitioned Data Set Extended Usage Guide*, SG24-6106, for background information.

3.8.1 PDSE PO5 module format compatibility on pre-z/OS V1R8 systems

There is support for use of the PDSE Program Object format 5 (PO5) module format on prior releases. In Table 3-8 we show the toleration maintenance available for systems prior to z/OS V1R8.

Table 3-8 PDSE PO5 format toleration maintenance

APAR	Description
OA13525	Allows use of load modules built on z/OS V1R8 to be used on prior systems

3.8.2 PDSE 64-bit virtual storage use enablement

Prior to z/OS V1R8, the SMSPDSE and SMSPDSE1 address spaces were 31-bit implementations. This limited the number of concurrently opened PDSE members that could exist in a system to approximately one million.

With z/OS V1R8, both address spaces are implemented in 64-bit mode. Now you can specify up to 16 GB, which allows a much larger number of PDSE members to be open.

The options that enabled by use of 64-bit storage are:

- PDSE_DIRECTORY_STORAGE (up to 16 GB) - related to SMSPDSE address space
- PDSE1_DIRECTORY_STORAGE (up to 16 GB) - related to SMSPDSE1 address space
- PDSE_HSP-SIZE (up to 2047 MB) - related to SMSPDSE address space
- PDSE1_HSP-SIZE (up to 2047 MB) - related to SMSPDSE1 address space

We recommend that you refer to the options available for member IGDSMSxx of PARMLIB in *MVS Initialization and Tuning Guide,* SA22-7591, and *MVS Initialization and Tuning Reference,* SA22-7592.

3.8.3 PDSE 64-bit virtual storage option setting

For the PDSE1 options, you can implement changes by using the SETSMS command (in which case they may revert to prior values at an IPL unless PARMLIB is also updated).

For PDSE and PDSE1 options you can update the appropriate values in the SYS1.PARMLIB member IGDSMSxx, in which case they take effect at an IPL.

You can issue the SET SMS=xx command to have the system implement the options contained in SYS1.PARMLIB member IGDSMS*xx* (where *xx* is a user-specified value). For some options the address space may have to be restarted (applies to SMSPDSE1 only).

To verify the effect of making changes, we did the following:

- Displayed the current options
- Changed the specific options of interest
- Displayed the options to verify that the change had taken effect

In Figure 3-44 on page 57 we show the SMS options before making changes. The options that we will modify are highlighted bold.

Note: The system default for both PDSE_DIRECTORY_STORAGE(n) and PDSE1_DIRECTORY_STORAGE(n) is 2047, and in the display these both show as 2000M.

The system default for PDSE_HSP_SIZE(n) and for PDSE1_HSP_SIZE(n) are also 0, which indicates that the respective hiperspace should not be started. However, in the display PDSE1_HSP_SIZE shows as 256 MB, indicating that the value has been set rather than defaulted at system IPL. In this case, the value was set in SYS1.PARMLIB as PDSE1_HSP_SIZE(256). If we change it by command it reverts to 256 MB at the next IPL.

```
D SMS,OPTIONS
IGD002I 13:26:24 DISPLAY SMS 566
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
ACSDEFAULTS = NO        PDSESHARING = EXTENDED
OVRD_EXPDT = NO         SYSTEMS = 8
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 256MB
USE_RESOWNER = YES      RLS_MAX_POOL_SIZE = 100MB
RLSINIT = YES           RLSTMOUT = 0
COMPRESS = GENERIC      LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
QTIMEOUT = 300          TVSNAME = 064
AKP = 1000              TV_START_TYPE = WARM
MAXLOCKS = (0,0)
CICSVR_INIT = YES       CICSVR_DSNAME_PREFIX = DWWUSER.V3R1M0
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = Z
RlsAboveThebarMaxPoolSize = 0
RlsFixedPoolSize = 0
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2000M
PDSE1_DIRECTORY_STORAGE = 2000M
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
GDS_RECLAIM = YES       DSSTIMEOUT = 0
GDS_RECLAIM = YES       DSSTIMEOUT = 0
BLOCKTOKENSIZE = NOREQUIRE       FAST_VOLSEL = OFF
IGD002I 13:26:24 DISPLAY SMS
```

Figure 3-44 Current IGDSMSxx option display

SETSMS commands issued and results

In Figure 3-45 we show the command SETSMS PDSE_DIRECTORY_STORAGE(2000M) being issued, and the response that indicates that it is invalid (because it is for the PDSE environment).

```
SETSMS PDSE_DIRECTORY_STORAGE(2000M)
IGD029I ERROR FOR SETSMS COMMAND 531
ERROR IS INVALID KEYWORD: PDSE_DIRECTORY_STORAGE
```

Figure 3-45 SETSMS PDSE_DIRECTORY_STORAGE(2000M) showing error

In Figure 3-46 we show the command SETSMS PDSE1_DIRECTORY_STORAGE(2000M) being issued. In this case the command is accepted, but without any confirmation of acceptance, so we follow the SETSMS command with part of the output from the D SMS,OPTIONS command to show that it has been accepted.

```
SETSMS PDSE1_DIRECTORY_STORAGE(2500M)
D SMS,OPTIONS
IGD002I 16:11:03 DISPLAY SMS 578
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
ACSDEFAULTS = NO        PDSESHARING = EXTENDED
OVRD_EXPDT = NO         SYSTEMS = 8
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 256MB
USE_RESOWNER = YES      RLS_MAX_POOL_SIZE = 100MB
RLSINIT = YES           RLSTMOUT = 0
COMPRESS = GENERIC      LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
QTIMEOUT = 300          TVSNAME = 064
AKP = 1000              TV_START_TYPE = WARM
MAXLOCKS = (0,0)

CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = Z
RlsAboveThebarMaxPoolSize = 0
RlsFixedPoolSize = 0
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2000M
PDSE1_DIRECTORY_STORAGE = 2500M
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
```

Figure 3-46 SETSMS PDSE1_DIRECTORY_STORAGE(2500M) and D SMS,OPTIONS commands

In Figure 3-47 we show the command
SETSMS PDSE1_HSP_SIZE(512) being issued. In this case the command is accepted, but without any confirmation of acceptance, so we follow the SETSMS command with part of the output from the D SMS,OPTIONS command to show that it has been accepted.

```
SETSMS PDSE1_HSP_SIZE(512)
D SMS,OPTIONS
IGD002I 17:39:30 DISPLAY SMS 646
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
ACSDEFAULTS = NO        PDSESHARING = EXTENDED
OVRD_EXPDT = NO         SYSTEMS = 8
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 512MB
```

Figure 3-47 SETSMS PDSE1_HSP_SIZE(512) and D SMS,OPTIONS commands

SET SMS=xx command method of updating SMS options

In order to use the SET SMS=xx command, a member of SYS1.PARMLIB must be available. For this demonstration we created IGDSMS88 as a copy of the member used for the last system IPL.

In Figure 3-48 we show part of the contents of IGDSMS88 with the lines being changed highlighted in bold.

```
SMS ACDS(SYS1.SMS.ACDS)
    COMMDS(SYS1.SMS.COMMDS)
    INTERVAL(15)
    DINTERVAL(150)
    DEADLOCK_DETECTION(15,4)
    SMF_TIME(YES)
    CF_TIME(1800)
    RLSINIT(YES)
    RLS_MAX_POOL_SIZE(100)
    REVERIFY(NO)
    ACSDEFAULTS(NO)
    PDSESHARING(EXTENDED)
    PDSE_RESTARTABLE_AS(YES)
    PDSE1_MONITOR(YES)
    PDSE1_LRUCYCLES(200)
    PDSE1_LRUTIME(50)
    PDSE_HSP_SIZE(256)
    PDSE1_HSP_SIZE(512)
    PDSE1_BMFTIME(3600)
    PDSE_DIRECTORY_STORAGE(2400M)
    PDSE1_DIRECTORY_STORAGE(2500M)
```

Figure 3-48 IGDSMS88 to be used with SET SMS=88 command

In Figure 3-49 we show the command SET SMS=88 being issued and the results as automatically listed by the SET SMS command.

The new values for the following can be seen highlighted bold:

- PDSE_HSP_SIZE
- PDSE1_HSP_SIZE
- PDSE_DIRECTORY_STORAGE(2400M)
- PDSE1_DIRECTORY_STORAGE(2400M)

Note: In order for the new value for PDSE1_HSP_SIZE to take effect it is necessary to restart SMSPDSE1. The new value for PDSE_HSP_SIZE does not take effect as a result of the update. That only occurs with an IPL. However, the contents of IGDSMS88 have been validated by issuing the SET SMS=88 command.

```
SET SMS=88
IEE252I MEMBER IGDSMS88 FOUND IN SYS1.PARMLIB
IGD031I SMS PARAMETERS 775
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
CACHETIME = 3600        SMF_TIME = YES
CF-TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           ACSDEFAULTS = NO
USE_RESOWNER = YES      DSNTYPE = PDS
GDS_RECLAIM = YES       PDSESHARING = EXTENDED
OVRD_EXPDT = NO         RLS_MAX_POOL_SIZE = 100MB
SYSTEMS = 8             COMPRESS = GENERIC
PDSE_HSP_SIZE = 256MB   PDSE1_HSP_SIZE = 512MB
RLSINIT = YES           RLSTMOUT = 0
CICSVR_INIT = NO        CICSVR_DSNAME_PREFIX = DWW.
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = Z
RlsAboveTheBarMaxPoolSize = 0
RlsFixedPoolSize = 0
DSSTIMEOUT =  0         FAST_VOLSEL = OFF
PDSE_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2400M
PDSE1_DIRECTORY_STORAGE = 2500M
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
BLOCKTOKENSIZE =  NOREQUIRE
```

Figure 3-49 SET SMS=88 command and listing of results

3.8.4 Retain buffers beyond PDSE close

This enhancement can provide improved performance when using PDSE members that are opened and closed very often. Before z/OS V1R8, when a PDSE was closed, the buffers for directory and member data were purged. If the member was opened again, the buffers had to be acquired again. Performance can be improved by specifying that the buffers be retained after the PDSE member is closed.

This option applies to all PDSE members, so if used it results in more storage in use for the PDSE memory cache.

Restriction: PDSE extended sharing must be active for the buffer beyond PDSE close enhancement.

PDSE extended sharing

You can define the extent of PDSE sharing. PDSEs can be shared between users of a single processor only (normal) or shared between users on any processor in a multisystem complex (extended). You select the type of sharing by using the PDSESHARING keyword in the IGDSMSxx member of SYS1.PARMLIB.

Extended sharing is available only when PDSESHARING(EXTENDED) has been specified in the IGDSMSxx member of SYS1.PARMLIB. Extended sharing applies when a PDSE is allocated with DISP=SHR. It cannot be used with DISP=OLD.

With extended sharing, any number of users or systems can concurrently share a PDSE or members within it for input (read) or output (write). While multiple members can be updated by different users on the same system, only one user can update a member at a given time. In addition, only one system can access the PDSE when the update is being done.

3.8.5 PDSE retain buffers beyond PDSE close enablement

For the PDSE1 options, you can implement changes by using the SETSMS command (in which case they may revert to prior values at an IPL unless PARMLIB is also updated). SMSPDSE1 must be restarted after the SETSMS command has been issued.

For PDSE and PDSE1 options you can update the appropriate values in the SYS1.PARMLIB member IGDSMSxx, in which case they take effect at an IPL.

You can issue the SET SMS=xx command to have the system implement the options contained in SYS1.PARMLIB member IGDSMS*xx* (where *xx* is a user-specified value). SMSPDSE1 must be restarted after the SETSMS command has been issued.

To verify the effect of making changes, we did the following:

- Displayed the current options.
- Changed the specific options of interest.
- Displayed the options to verify that the change had taken effect.

In Figure 3-50 we show the SMS options before making changes. The options that we modify are highlighted bold. The options that relate to caching of PDSE buffers after close are:

- PDSE_BUFFER_BEYOND_CLOSE(YES|NO)
- PDSE1_BUFFER_BEYOND_CLOSE(YES|NO)

The default in both cases is NO.

```
D SMS,OPTIONS
IGD002I 19:08:05 DISPLAY SMS 930
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
SMF_TIME = YES          CACHETIME = 3600
CF_TIME = 1800          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           DSNTYPE = PDS
ACSDEFAULTS = NO        PDSESHARING = EXTENDED
OVRD_EXPDT = NO         SYSTEMS = 8
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 256MB
USE_RESOWNER = YES      RLS_MAX_POOL_SIZE = 100MB
RLSINIT = YES           RLSTMOUT = 0
COMPRESS = GENERIC      LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
QTIMEOUT = 300          TVSNAME = 064
AKP = 1000              TV_START_TYPE = WARM
MAXLOCKS = (0,0)
CICSVR_INIT = YES       CICSVR_DSNAME_PREFIX = DWWUSER.V3R1M0
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = Z
RlsAboveThebarMaxPoolSize = 0
RlsFixedPoolSize = 0
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2000M
PDSE1_DIRECTORY_STORAGE = 2000M
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
GDS_RECLAIM = YES       DSSTIMEOUT = 0
BLOCKTOKENSIZE = NOREQUIRE       FAST_VOLSEL = OFF
```

Figure 3-50 Display SMS OPTIONS before BUFFER BEYOND CLOSE changes

In Figure 3-51 we show the effect of issuing the following command and the resulting error since it is not supported:

```
SETSMS PDSE_BUFFER_BEYOND_CLOSE(YES)
```

```
SETSMS PDSE_BUFFER_BEYOND_CLOSE(YES)
IGD029I ERROR FOR SETSMS COMMAND 934
ERROR IS INVALID KEYWORD: PDSE_BUFFER_BEYOND_CLOSE
```

Figure 3-51 SETSMS PDSE_BUFFERBEYOND_CLOSE command and error message

In Figure 3-52 we show the effect of issuing the following command and confirmation that it was accepted. In this case the PDSE environment is still set OFF. The SMSPDSE1 address space needs to be restarted to take advantage of the changed setting.

```
SETSMS PDSE1_BUFFER_BEYOND_CLOSE(YES)
```

```
SETSMS PDSE1_BUFFER_BEYOND_CLOSE(YES)
IGW040I Buffer past close not active for SMSPDSE
IGW040I Buffer past close active for SMSPDSE1
```

Figure 3-52 SETSMS PDSE1_BUFFER_BEYOND_CLOSE command and confirmation

SET SMS=xx command method of updating SMS options

In order to use the SET SMS=xx command, a member of SYS1.PARMLIB must be available. For this demonstration we created IGDSMS89 as a copy of the member used for the last system IPL.

In Figure 3-53 we show part of the contents of IGDSMS89 with the lines being changed highlighted bold.

```
SMS ACDS(SYS1.SMS.ACDS)
    COMMDS(SYS1.SMS.COMMDS)
    PDSESHARING(EXTENDED)
    PDSE_RESTARTABLE_AS(YES)
    PDSE1_MONITOR(YES)
    PDSE1_LRUCYCLES(200)
    PDSE1_LRUTIME(50)
    PDSE1_HSP_SIZE(256)
    PDSE1_BMFTIME(3600)
    PDSE_BUFFER_BEYOND_CLOSE(YES)
    PDSE1_BUFFER_BEYOND_CLOSE(YES)
```

Figure 3-53 GDSMS89 to be used with SET SMS=89 command

In Figure 3-54 we show the following command and the results as listed automatically that show that the changes have been made, in this case for both PDSE and PDSE1.

```
SET SMS=89
```

```
SET SMS=89
IEE252I MEMBER IGDSMS89 FOUND IN SYS1.PARMLIB
IGD031I SMS PARAMETERS 985
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
INTERVAL = 15           DINTERVAL = 150
CACHETIME = 3600        SMF_TIME = YES
CF-TIME = 3600          PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600     PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60       PDSE1_LRUTIME = 50
PDSE_LRUCYCLES = 15    PDSE1_LRUCYCLES = 200
LOCAL_DEADLOCK = 15     GLOBAL_DEADLOCK = 4
REVERIFY = NO           ACSDEFAULTS = NO
USE_RESOWNER = YES      DSNTYPE = PDS
GDS_RECLAIM = YES       PDSESHARING = EXTENDED
OVRD_EXPDT = NO         RLS_MAX_POOL_SIZE = 100MB
SYSTEMS = 8             COMPRESS = GENERIC
PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 256MB
RLSINIT = NO            RLSTMOUT = 0
CICSVR_INIT = NO        CICSVR_DSNAME_PREFIX = DWW.
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
Rls_MaxCfFeatureLevel = Z
RlsAboveTheBarMaxPoolSize = 0
RlsFixedPoolSize = 0
DSSTIMEOUT =  0         FAST_VOLSEL = OFF
PDSE_MONITOR = (YES,0,0)
PDSE1_MONITOR = (YES,0,0)
PDSE_DIRECTORY_STORAGE = 2000M
PDSE1_DIRECTORY_STORAGE = 2000M
PDSE_BUFFER_BEYOND_CLOSE = YES
PDSE1_BUFFER_BEYOND_CLOSE = YES
BLOCKTOKENSIZE =  NOREQUIRE
```

Figure 3-54 SET SMS=89 command and listing of results

For PDSE this has verified that the syntax of member IGDSMS89 is good, but the change does not take effect until an IPL is done using this member.

For PDSE1, the change takes place with an IPL using this member, but the SMSPDSE1 address space uses the new setting if it was restarted.

3.8.6 DFSMS V1R8 PDSE enhancements in storage requirements

Implementation of PDSE_BUFFER_BEYOND_CLOSE or PDSE1_BUFFER_BEYOND_CLOSE results in a requirement for more storage than before, so the ability to specify higher values for PDSE_HSP_SIZE, PDSE1_HSP_SIZE, PDSE_DIRECTORY_STORAGE, or PDSE1_DIRECTORY_STORAGE may be important.

There is no new instrumentation provided to specifically address effects on the Buffer Management Facility (BMF) of these higher values because the PDSE address spaces are always using 64-bit storage for buffers.

However, the techniques presented in Section 9.8 of *Partitioned Data Set Extended Usage Guide*, SG24-6106, remain valid. The SMF42T1 program referenced in that section must be re-assembled using the V1R8 version of the macros.

A simplified version of the information in *Partitioned Data Set Extended Usage Guide*, SG24-6106, is presented in 3.9, “PDSE buffer management statistics” on page 66.

3.9 PDSE buffer management statistics

SMF is used to record data that may be useful in determining how the PDSE buffer management function is performing, and may allow values to be tuned to improve performance or reduce over commitment of real storage. BMF is responsible for managing access to and from the PDSE hiperspace, and other than through SMF there is no way to determine whether hiperspace is being used.

SMF record type 42, subtype 1 provides overall buffer use totals for SMSPDSE and, if implemented, separately for SMSPDSE1, and totals for each storage class being used by the SMSPDSE[1] address spaces and their hiperspaces.

3.9.1 BMF data capture preparation

In order to have SMF capture BMF data, the following items must be addressed:

- PDSE_BMFTIME and, if SMSPDSE1 is in use, PDSE1_BMFTIME must be set to appropriate values. The default setting of 3600 seconds (1 hour) may be too long for meaningful analysis. Note that if no PDSE eligible activity occurs, no BMF data is captured because hiperspace cache is not used. After eligible PDSE data sets are accessed, the BMF data recording function starts, and it continues even after the triggering PDSE is no longer in use.
- Note that there is no eye catcher in the data to distinguish BMF data associated with SMSPDSE from that associated with SMSPDSE1. However, as each address space has its own LRUCYCLES definition, making these different does provide a way of inferring which is which. The default setting for SMSPDSE is 15, and for SMSPDSE1 is 50. Note that LRUCYCLES is one of the variables that may need to be adjusted to limit CPU or real storage use, so if they are changed, be sure to maintain some difference between the two.

3.9.2 BMF analysis preparation

We provide a sample program that formats the information from what was recorded by SMF. The program needs to be assembled and linked into an appropriate data set before use. Refer to “SMF record type 42 sub type 1 data” on page 478 for information about building the program.

So that the user can relate the program output to the SMF manuals and other sources of information, the field names as defined in the SMF macros are used.

The SMF records that are collected from the BMF totals are shown in Figure 3-55.

```
SMF42O1A DSECT ,
SMF42BMF DS    0CL0036   Description of BMF totals section
SMF42TNA DS    1FL4      Total number of storage classes
SMF42TMT DS    1FL4      Interval length (total time of measurement
                         period)
SMF42TRT DS    1FL4      Total number of member data page reads
SMF42TRH DS    1FL4      Total number of member data page read hits
                         (found in BMF)
SMF42TDT DS    1FL4      Total number of directory data page reads
SMF42TDH DS    1FL4      Total number of directory data page read hits
                         (found in BMF)
SMF42BUF DS    1FL4      Total number of active BMF 4K buffers   @18A
SMF42BMX DS    1FL4      High water mark of BMF buffers          @18A
SMF42LRU DS    1FL2      BMF LRU interval time                   @18A
SMF42UIC DS    1FL2      BMF LRU cycles before buffer cast out   @18A
```

Figure 3-55 Extract of the BMF SMF data fields used

For the same SMF record as above, the data collected for the storage classes are shown in Figure 3-56.

```
SMF42O1B DSECT ,
SMF42SC  DS    0CL0048   Description of Stor Class Sum section @OW00573
SMF42PNA DS    0CL0032   Storage class name
SMF42PNL DS    1FL2      Storage class name length
SMF42PNN DS    1CL0030   Storage class name
SMF42SRT DS    1FL4      Total number of member data page reads
SMF42SRH DS    1FL4      Total number of member data page read hits    X
                         (found in BMF)
SMF42SDT DS    1FL4      Total number of directory data page reads
SMF42SDH DS    1FL4      Total number of directory data page read hits X
                         (found in BMF)
```

Figure 3-56 Extract of the BMF storage class fields used

Depending on the structure of the system, there may be one or more BMF totals sections, which are shown in the program output as BMF TOTALS SET #: *nnnnnnn*, where *nnnnnnn* is a numeric value, and there may be one or more storage class sections that are identified by their names as defined to SMS. Normally, there is one set of SMF records for SMSPDSE and one for SMSPDSE1.

The extract program assumes that the data it is processing is an extract from the SMF data prepared by the SMF-supplied extraction program IFASMFDP. The SMFPRMxx member of SYS1.PARMLIB should be checked to ensure that SMF record type 42 is being selected or that it is not being suppressed.

The extract may be taken from the running copy of the VSAM 'SYS1.MAN*' data sets or from a previously extracted sequential data set. IFASMFDP manages the correct handing of the input data set, and regardless of the form of the input, produces a sequential data set. In order to limit the amount of data extracted, the recommended control statement input to IFASMFDP limits the output to select only records of type 42 subtype 1.

If the IFASMFDP is set to extract all type 42 records, all those that are not type 42 subtype 1 are discarded by the SMF42T1.

Listings of the program source and examples of JCL to assemble and link the program are shown in “SMF record type 42 sub type 1 data” on page 478. Assembly is required only once for initial implementation of a given level of the operating system, then if maintenance is applied to the IGWSMF macro.

An example of the job required to extract the SMF records then run the formatting program is shown in Figure 3-57. This assumes that the formatting program is called SMF42T1 and that it has previously been assembled and stored in the data set PDSERES.SMF42T1J.PDS (an example name only). The IFASMFDP selection specification is part of the OUTDD statement OUTDD(OUTDD,TYPE(42(1))).

```
//MHLRES1S JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//EXTRACT EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=A
//ADUPRINT DD SYSOUT=A
//*DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//DUMPOUT DD DUMMY
//OUTDD   DD DSN=&T1,SPACE=(CYL,(10,5)),RECFM=VB,LRECL=5096,
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA
//SYSIN DD  *
          INDD(DUMPIN,OPTIONS(DUMP))
          OUTDD(OUTDD,TYPE(42(1)))
/*
//FORMAT  EXEC PGM=SMF42T1
//STEPLIB DD DISP=SHR,DSN=PDSERES.SMF42T1J.PDS
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DSN=&T1
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 3-57 Example of JCL to format SMF record type 42 subtype 1 records

An example of output from the IFASMFDP program is shown in Figure 3-58. The full output shows all the records found, and the right-most column shows the number of records selected. In this partial output example, no records were selected for types 41 and 43, and 41 records were found for the type 42 record, which is consistent with the selection criteria specified on the OUTDD control statement.

```
IFA010I SMF DUMP PARAMETERS
IFA010I END(2400) -- DEFAULT
IFA010I START(0000) -- DEFAULT
IFA010I DATE(1900000,2099366) -- DEFAULT
IFA010I OUTDD(OUTDD,TYPE(42(1))) -- SYSIN
IFA010I INDD(DUMPIN,OPTIONS(DUMP)) -- SYSIN
IFA020I OUTDD    -- SYS07053.T123632.RA000.MHLRES1S.T1.H01
IFA020I DUMPIN   -- SMFDATA.ALLRECS.G3142V00
                                    SUMMARY ACTIVITY REPORT
      START DATE-TIME  02/21/2007-18:40:00                        END DATE-TIME  02/21/2007-20:10:00
      RECORD       RECORDS         PERCENT      AVG. RECORD   MIN. RECORD   MAX. RECORD       RECORDS
        TYPE          READ        OF TOTAL           LENGTH        LENGTH        LENGTH       WRITTEN
           2             1           .01 %            18.00            18            18             1
.
.

        40          216         2.98 %           74.00            74            74             0
        41            6          .08 %          332.00           332           332             0
        42          651         8.98 %          605.10           176        32,620             2
```

Figure 3-58 Example of part of the output from IFASMFDP showing selection of type 42 records

3.9.3 SMF statistics interpretation

There are no absolute values that are correct or wrong for the tuning parameters. The SMF data is useful to determine what is happening, then after a change verify that the change had some effect.

An example of output from the SMF type 42 record formatting program is shown in Figure 3-59. Note that the value for SMF42TNA of 1corresponds to the one SCLASS entries that follows it. By checking the output from the **d sms,options** command we infer from the value of 0006000 for SMF42LRU that this is data from SMSPDSE.

```
SMF TYPE 42 S/TYPE 1 RECORDS. COLS USE SMF NAMES
BMF TOTALS SET #:  0000001
HH:MM:SS YYYYDDD SMF42TNA SMF42TMT SMF42TRT SMF42TRH SMF42TDT SMF42TDH SMF42BUF SMF42BMX SMF42LRU SMF42UIC
19:32:51 2007052 0000001  0003599  0000000  0000000  0000000  0000000  0000010  0000121  0006000  0000015

SMF42PNN (SCLASS): **NONE**
HH:MM:SS YYYYDDD SMF42SRT SMF42SRH SMF42SDT SMF42SDH
19:32:51 2007052 0000000  0000000  0000000  0000000
```

Figure 3-59 Example 1 of output from program formatting SMF type 42 subtype 1 records

A further example of output from the SMF type 42 record formatting program is shown in Figure 3-60. Note that the value for SMF42TNA of 1 corresponds to the one SCLASS entry that follows it. By checking the output from the **d sms,options** command we infer from the value of 0005000 for SMF42LRU that this is data from SMSPDSE1.

```
BMF TOTALS SET #:  0000001
HH:MM:SS YYYYDDD SMF42TNA SMF42TMT SMF42TRT SMF42TRH SMF42TDT SMF42TDH SMF42BUF SMF42BMX SMF42LRU SMF42UIC
19:32:51 2007052 0000003  0003599  0000001  0000000  0000056  0000047  0000017  0001399  0005000  0000200

SMF42PNN (SCLASS): STANDARD
HH:MM:SS YYYYDDD SMF42SRT SMF42SRH SMF42SDT SMF42SDH
19:32:51 2007052 0000000  0000000  0000000  0000000
```

Figure 3-60 Example 2 of output from program formatting SMF type 42 subtype 1 records

For reference, some of the data reported shows what the current settings are, and there would not be any change in these unless some specific action was taken, but the various counts can be expected to change.

For each PDSE data space in use, that is always one and optionally two, there will be data for the BMF and also for the processing as it relates to the storage classes involved.

The data that is recorded by SMF and ultimately returned in the SMF type 42 subtype 1 records are as shown in Figure 3-55 on page 67 and Figure 3-56 on page 67.

The following comments are presented in the order shown in the collection data fields in the examples mentioned above.

BMF totals data

There is no specific data in the BMF totals data section to indicate which of the SMSPDSE or SMSPDSE1 data spaces it applies to. In the SMF42T1 program output, the sections are identified in the output in the program header as:

```
BMF TOTALS SET #:  000000n
```

Where *n* is replaced by the relative number of the BMF totals section found.

The default system settings for LRUCYCLES and BMFTIME are different for the SMSPDSE and SMSPDSE1 data sets, so as long as a difference is retained in any site settings between one or another of these, you can determine which one is for SMSPDSE and which for SMSPDSE1. Example 3-1 shows two lines from a sample report. You can see the values 0005000 and 000200 under the SMF42LRU and SMF42UIC columns, respectively. From the output from the command **d sms,options** you can see both the LRUCYCLES and BMFTIME values, and from that determine which set of BMF totals is which. Refer to the SMF42LRU explanation below for comments on the apparent multiplication factor applied over what was specified in the IGDSMSxx parameters.

Example 3-1 Sample program showing SMF42LRU and SMF42UIC columns

```
YYYYDDD SMF42TNA SMF42TMT SMF42TRT SMF42TRH SMF42TDT SMF42TDH SMF42BUF SMF42BMX SMF42LRU SMF42UIC

2004327 0000002  0003599  0000000  0000000  0000010  0000008  0000011  0001080  0005000  0000200
```

Note: The number assigned to the BMF totals heading represents the relative order that the data appears in the SMF type 42 subtype 1 records. For the life of a given IPL, it is likely that the order will remain the same if there is more than one, but it is possible that after a subsequent IPL, the order will have changed.

The columns are:

- SMF42TNA - total number of storage classes

  Number of storage classes defined in the system. This is useful to correlate the number of storage classes reported on in the storage class data report.

- SMF42TMT - interval length for data gathering

  This is the value defined in the BMF_TIME fields and may not be what is being used, because this value is documented as overridden by the SMF INTVAL if the SMS OPTION SMS_TIME is set to YES.

- SMF42TRT - total number of member data page reads

  This is the overall number of member data page reads, including those found in BMF.

- SMF42TRH - total number of member data page reads found in BMF

  This is the number of member data page reads satisfied from BMF. When BMF is functioning in support of a PDSE, we expect to see that SMF42TRH is not zero, and that it is less than SMF42TRT. Whether or not member data is cached by BMF depends on the MSR value specified in the storage class associated with a particular data set being low.

- SMF42TDT - total number of directory data page reads

  This is the overall number of directory data page reads, including those found in BMF.

- SMF42TDH - total number of directory data page reads found in BMF

  This is the number of directory data page reads satisfied from BMF. When BMF is functioning in support of a PDSE, we expect to see that SMF42TDH is not zero, and that it is less than SMF42TDT. Directory data page reads should be managed by BMF whether or not the data set is associated with a storage class.

- SMF42BUF - total number of active BMF buffers

  This number should correspond to the number of 4-K pages that will fit in the hiperspace as defined by the HSP_SIZE.

- SMF42BMX - high water mark of BMF buffers

  This number represents the largest number of BMF buffers used from the total number of BMF buffers available. If the number presented here is consistently significantly lower that the number of buffers available, this may be an indication that the HSP_SIZE can safely be reduced. Likewise, if the number presented here is consistently close to the total number of buffers available, it may be necessary to consider increasing the HSP_SIZE. However, the number of buffers in use is influenced by the values used for LRUCYCLES and LRUTIME, and it may be that adjusting these should be done before considering changing HSP_SIZE, as that requires a system IPL to implement.

- SMF42LRU - BMF LRU interval time

  This value corresponds to the PDSE[1]_LRUTIME value as specified or as defaulted in the IGDSMSxx PARMLIB options member, or as subsequently altered by SETSMS command. Note that the value as present is shown as multiplied by 100.

- SMF42UIC - BMF LRU cycles before buffer cast out

  This value corresponds to the PDSE[1]_LRUCYCLES value as specified in the IGDSMSxx options member, or as subsequently altered by SETSMS command. Note that

LRUTIME is not an actual time value, but an indication of how many times a buffer may remain in the buffer but not allocated when buffer management cycle runs, which is controlled by LRUCYCLES.

Attention: The higher that LRUCYCLES is set the more CPU time is used running the BMF process, and the lower that LRUTIME is set the more real storage remains allocated. Tuning LRUCYCLES and LRUTIME for each of the SMSPDSE and SMSPDSE1 address spaces can be carried out without a system IPL. Any changes should be carefully monitored in conjunction with measurements of the committed CPU and real storage frames. Any sudden increase in CPU or REAL storage commitment may reflect the introduction of a PDSE into the system.

Storage class summary data

There is one of these sections for each storage class in the system related to each of the SMSPDSE or SMSPDSE1 address spaces. For ease of interpretation, storage classes intended to manage the PDSE buffer management processes should be named according to a convention that indicates that they are intended for PDSE data set management, and indicate whether the management intended was to have the PDSE cached in the PDSE hiperspaces.

- SMF42PNL - storage class data section name length

  Reference information

- SMF42PNN - Storage class name

  Reference information

- SMF42SRT - total number of member data page reads

  This is the overall number of members read.

- SMF42SRH - total number of member data pages read from BMF

  This is the number of member data page reads supplied from the BMF buffers.

  When a PDSE associated with this storage class is being serviced by BMF, the SMF42SRH value should not be zero, but it will be less than the total number of member data reads. This shows zero if the storage class does not have a low MSR value.

- SMF42SDT - total number of directory data page reads

  This is the total number of data pages read using this storage class.

- SMF42SDH - total number of directory data page reads serviced by BMF.

  This is the total number of directory data pages read that were found in the BMF buffers. Directory pages are eligible for BMF processing whether or not the PDSE is associated with an eligible storage class.

  The value in SMF42SDH should not be zero, and will likely be less than the value in SMF42SDT if the BMF function is managing the data to and from hiperspace.

Recommendation for further analysis

If the SMF42SRH results are significantly lower than SMF42SRT, it is likely to be an indication that for some reason few PDSE data sets are being cached to hiperspace. If the PDSE data sets in use are being used by single or very few users at a time, then there may be no concern. However, if a PDSE is intended to be shared, then the benefits of using hiperspace should be investigated.

One comparatively simple way to measure whether a particular data set comes into that category is to assign it its own storage class for test purposes. That requires the creation of

the storage class if not already available, and probably requires an adjustment to the ACS routines to ensure that the data set is assigned, by itself, to the test storage class. Once implemented, the SMF type 42 subtype 1 records automatically pick up the additional storage class, and the effect of changes to the MSR can be monitored.

Once the data set is being managed by BMF, any specific changes made to improve the BMF member data hit ratio can then be fitted back into the original storage class.

3.10 Integrated Catalog Forward Recovery Utility

The Integrated Catalog Forward Recovery Utility (ICFRU) was originally available as a separate product. With z/OS V1R7, it was integrated into the base product. With z/OS V1R8 the documentation was expanded to include details of a technique to verify readiness of the product for use in a real disaster.

In order to use ICFRU a mechanism must exist to restore a valid version of a given catalog, so a backup process must exist, and the correct SMF records need to be recorded and saved in appropriate data sets.

The full description of ICFRU is contained in Appendix A of *DFSMS Managing Catalogs,* SC26-7409.

3.10.1 ICFRU system flow

In summary, ICFRU is a pair of programs that are used to update a valid catalog from activity records that have been recorded in the system SMF data sets.

Program CRURRSV is Integrated Catalog Forward Recovery Record Selection and Validation, and CRURRAP is Integrated Catalog Forward Recovery Record Analysis and Processing.

You can control the execution of ICFRU programs through the specification of execution parameters (the same parameters for CRURRSV and CRURRAP) and by providing the appropriate SMF record data and an IDCAMS EXPORT of the original catalog.

To use the ICFRU you need:

- The name of the catalog to be recovered
- Recovery start date and time - the date and time at which the backup to be used for recovery was made
- Recovery stop date and time - the date and time that correspond to the closing of the catalog (or the time after which no updates were possible)
- An SMF gap time - the (approximate) interval just smaller than the minimum time used to fill (or switch) an SMF recording data set
- A multi-system clock difference (approximate) maximum difference between the TOD clocks of any two systems sharing the catalog
- All SMF dump data sets spanning the recovery period
- A sort utility and appropriate sort control
- The IDCAMS EXPORT data set to be used as the basis for recovery

In Figure 3-61 we show the overall flow of the ICFRU process.

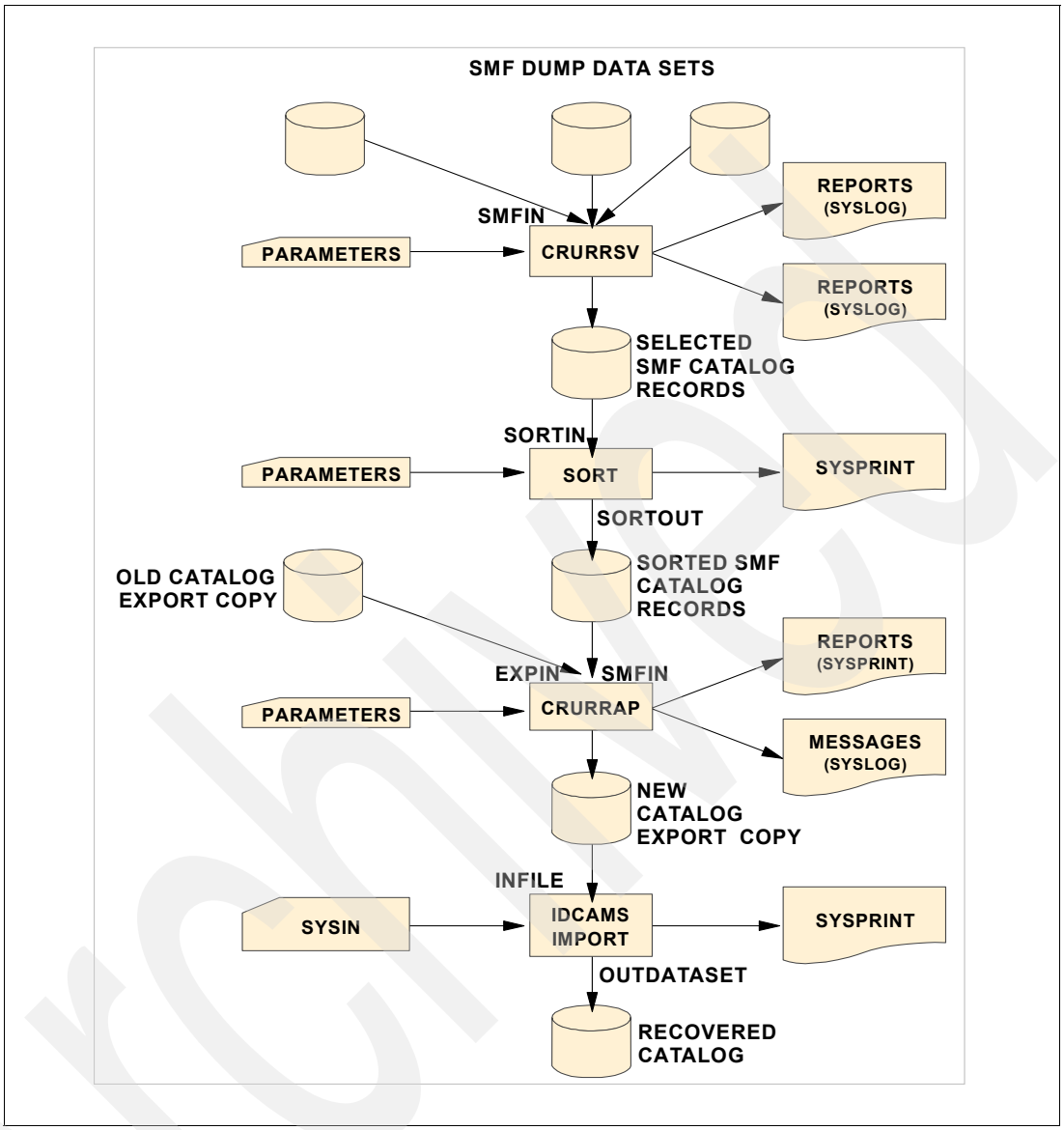


Figure 3-61 ICFRU Process flow

3.10.2 ICFRU installation readiness overview

This section provides a practical example based on Appendix A, section “Confirming Installation Readiness,” of the manual *DFSMS Managing Catalogs,* SC26-7409.

Note: This is a simulation of the process that allows you to verify that all items required are available. You must set up the complete process for recovery that runs automatically to ensure that you have all the data necessary to restore catalogs in the event of an unexpected failure.

The simulation is done by carrying out the following phases:

- Phase 1

  One-time preparation steps that need only be run once. See 3.10.3, “Installation readiness preparation” on page 75.

- Phase 2

  Execution steps that may be repeated as necessary. See 3.10.4, “Installation readiness verification” on page 77.

3.10.3 Installation readiness preparation

In this section we discuss installation readiness preparation.

Step 1 - verify that SMF is recording record types 61, 65, and 66

In Figure 3-62 on page 76 we show command D SMS,OPTIONS being issued and the results. The options of interest are bolded. The TYPE entries for both SUBSYS(STC..) and SYS.. must both cover the required records. In this case all records other than record type 99 are being collected. If the display had shown that any of record types 61, 65, or 66 were not included, the SMFPRMxx PARMLIB member must be updated and implemented.

Note: It is necessary to carry out this check on all systems that share the same catalogs.

```
D SMF,O
IEE967I 18.47.14 SMF PARAMETERS 451
        MEMBER = SMFPRM00
        MULCFUNC -- DEFAULT
        BUFUSEWARN(25) -- DEFAULT
        BUFSIZMAX(0128M) -- DEFAULT
        SYNCVAL(00) -- DEFAULT
        DUMPABND(RETRY) -- DEFAULT
        SUBSYS(STC,NOINTERVAL) -- SYS
        SUBSYS(STC,NODETAIL) -- SYS
        SUBSYS(STC,EXITS(IEFUSO)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFUJP)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFUJI)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFACTRT)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFU85)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFU84)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFU83)) -- PARMLIB
        SUBSYS(STC,EXITS(IEFU29)) -- PARMLIB
        SUBSYS(STC,TYPE(0:98,100:255)) -- PARMLIB
        SYS(NODETAIL) -- PARMLIB
        SYS(NOINTERVAL) -- PARMLIB
        SYS(EXITS(IEFU29)) -- PARMLIB
        SYS(EXITS(IEFUTL)) -- PARMLIB
        SYS(EXITS(IEFUJI)) -- PARMLIB
        SYS(EXITS(IEFUSO)) -- PARMLIB
        SYS(EXITS(IEFUJP)) -- PARMLIB
        SYS(EXITS(IEFUSI)) -- PARMLIB
        SYS(EXITS(IEFUJV)) -- PARMLIB
        SYS(EXITS(IEFACTRT)) -- PARMLIB
        SYS(EXITS(IEFU85)) -- PARMLIB
        SYS(EXITS(IEFU84)) -- PARMLIB
        SYS(EXITS(IEFU83)) -- PARMLIB
        SYS(TYPE(0:98,100:255)) -- PARMLIB
        NOBUFFS(MSG) -- PARMLIB
        LASTDS(MSG) -- PARMLIB
        LISTDSN -- PARMLIB
        SID(SC64) -- PARMLIB
        DDCONS(NO) -- PARMLIB
        JWT(2400) -- PARMLIB
        MEMLIMIT(NOLIMIT) -- PARMLIB
        STATUS(010000) -- PARMLIB
        MAXDORM(3000) -- PARMLIB
        INTVAL(10) -- PARMLIB
        REC(PERM) -- PARMLIB
        NOPROMPT -- PARMLIB
        DSNAME(SYS1.SC64.MAN3) -- PARMLIB
        DSNAME(SYS1.SC64.MAN2) -- PARMLIB
        DSNAME(SYS1.SC64.MAN1) -- PARMLIB
```

Figure 3-62 D SMF,OPTIONS command and results

Step 2 - create initial IDCAMS EXPORT data set and GDG

For the purposes of this simulation it is advisable to ensure that the catalog is free of errors since any errors may affect the final compare. In a real disaster situation, whatever state the catalog is in when it is backed up (for example, EXPORTED) is all you have to deal with.

You need to retain information about the status of the catalog and the EXPORT process, so data sets are defined to capture this information. In this example Generation Data Group (GDG) data sets are used, but any method that provides unique data set names can be used.

You need to establish data set and GDG information one time, then the same structure for subsequent runs.

In Figure 3-63 we show the job to create the data sets and GDGs. The DELETE command for the GDGs can be omitted the first time the job is run.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//*************************************************************
//* THE GDGS AND MODEL DSCBS FOR THE DATA SETS ASSOCIATED WITH
//* EXPORTING CATALOG UCAT.VSBOX01 ARE DEFINED. BY CONVENTION
//* ALL BACKUP DATA SETS WILL START WITH MHLRES1.BCAT .
//*************************************************************
//SETUPDS   EXEC PGM=IDCAMS
//LIST      DD DSN=MHLRES1.BCAT.LIST.DCB,DISP=(NEW,CATLG),
//          VOL=SER=SBOX01,UNIT=SYSDA,SPACE=(0,0),
//          DCB=(RECFM=VBA,LRECL=125,BLKSIZE=4250)
//EXPORT    DD DSN=MHLRES1.BCAT.EXPORT.DCB,DISP=(NEW,CATLG),
//          VOL=SER=SBOX01,UNIT=SYSDA,SPACE=(0,0),
//          DCB=(RECFM=VBS,LRECL=32404)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  DELETE          MHLRES1.BCAT.CATALOG.LISTING  GDG
  DELETE          MHLRES1.BCAT.CATALOG.BACKUP   GDG
  DELETE          MHLRES1.BCAT.PROBLEM.LISTING  GDG
  DEFINE GDG (NAME(MHLRES1.BCAT.CATALOG.LISTING) LIM(7) NEMP SCR -
             FOR(9999))
  DEFINE GDG (NAME(MHLRES1.BCAT.CATALOG.BACKUP) LIM(7) NEMP SCR -
             FOR(9999))
  DEFINE GDG (NAME(MHLRES1.BCAT.PROBLEM.LISTING) LIM(7) NEMP SCR -
             FOR(9999))
  LISTC LEVEL(MHLRES1.BCAT) ALL
```

Figure 3-63 ICFRU readiness check - set up data sets and GDGs

3.10.4 Installation readiness verification

To verify:

1. Create an IDCAMS EXPORT. This is similar to the job that is run regularly.
2. Wait a while.
3. Create an IDCAMS EXPORT. This is similar to the job that is run regularly after the one from step 1.

4. Create an EXPORT data set from the EXPORT created in step 1 together with SMF records that cover the period between steps 1 and 3.
5. Compare the EXPORT created in step 4 with that taken at step 3.

Note: In a real situation, at step 5 the EXPORT from step 4 is what will be used as an IMPORT to create the replacement catalog rather then be used for comparison processes, as shown here.

In Figure 3-64 we show the job to run the catalog diagnosis and make the first export of catalog UCAT.FLETCHER.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*************************************************************
//* DIAGNOSE THE CATALOG
//* LIST ITS ALIASES
//* LIST ITS SELF-DESCRIBING ENTRY COMPLETELY
//* LIST ITS ENTRIES WITH JUST THE VOLUME INFORMATION
//DIAGLIST  EXEC PGM=IDCAMS
//SETBKDS   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(NEW,PASS),
//          DCB=MHLRES1.BCAT.EXPORT.DCB,
//          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
//SYSPRINT  DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(MOD,PASS),
//          DCB=MHLRES1.BCAT.LIST.DCB,
//          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
//SYSIN     DD  *
  DIAG  ICFCAT INDATASET(UCAT.FLETCHER)
  LISTCAT ENTRY(UCAT.FLETCHER) ALL
  LISTCAT ENTRY(UCAT.FLETCHER) ALL CAT(UCAT.FLETCHER)
  LISTCAT VOLUME CAT(UCAT.FLETCHER)
/*
//*************************************************************
//* EXPORT THE CATALOG IF THE DIAGNOSTICS WERE OKAY
//EXPCAT    EXEC PGM=IDCAMS,COND=(8,LE)
//CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,PASS),
//          DCB=MHLRES1.BCAT.EXPORT.DCB,
//          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
//SYSPRINT  DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(MOD,PASS),
//          DCB=MHLRES1.BCAT.LIST.DCB,
//          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
//SYSIN DD  *
 EXPORT UCAT.FLETCHER OFILE(CATBACK) TEMP
/*
//*************************************************************
//* CATALOG THE BACKUP AND LISTING IF EXPORT WAS OK
//CATAL     EXEC PGM=IEFBR14,COND=(8,LE)
//CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,CATLG)
//SYSLIST   DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(OLD,CATLG)
//*************************************************************
//* COPY THE LISTING AND DISCARD THE BACKUP IF EXPORT FAILED
//REPOUT    EXEC PGM=IDCAMS,COND=(0,EQ,CATAL)
//SYSPRINT  DD SYSOUT=*
//CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,DELETE)
//CATLIST   DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(OLD,DELETE)
//CATPROB   DD DSN=MHLRES1.BCAT.PROBLEM.LISTING(+1),DISP=(NEW,CATLG),
//          DCB=MHLRES1.BCAT.LIST.DCB,
//          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
//SYSIN     DD *
 REPRO INFILE(CATLIST) OUTFILE(CATPROB)
/*
```

Figure 3-64 ICFRU readiness check IDCAMS diagnose and first EXPORT job

In Figure 3-65 and Figure 3-66 on page 81 we show the JES2 output from running the job listed in Figure 3-64 on page 79.

```
 J E S 2  J O B  L O G  --  S Y S T E M  S C 6 4  --  N O D E  W T S C P L X 2

12.39.14 JOB21908 ---- FRIDAY,    16 MAR 2007 ----
12.39.14 JOB21908  IRR010I  USERID MHLRES1  IS ASSIGNED TO THIS JOB.
12.39.14 JOB21908  ICH70001I MHLRES1  LAST ACCESS AT 12:33:00 ON FRIDAY, MARCH 16, 2007
12.39.14 JOB21908  $HASP373 MHLRES1I STARTED - INIT 1    - CLASS A - SYS SC64
12.39.14 JOB21908  IEF403I MHLRES1I - STARTED - TIME=12.39.14 - ASID=002A - SC64
12.39.15 JOB21908  -                                      --TIMINGS (MINS.)--            ----PAGING COUNTS---
12.39.15 JOB21908  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG   PAGE   SWAP    VIO SWAPS STEPNO
12.39.15 JOB21908  -MHLRES1I          DIAGLIST    00     99    .00    .00    .00    554   0      0      0      0     0     1
12.39.15 JOB21908  -MHLRES1I          EXPCAT      00     75    .00    .00    .00    920   0      0      0      0     0     2
12.39.15 JOB21908  -MHLRES1I          CATAL       00     10    .00    .00    .00     25   0      0      0      0     0     3
12.39.15 JOB21908  -MHLRES1I          REPOUT   FLUSH      0    .00    .00    .00      0   0      0      0      0     0     4
12.39.15 JOB21908  IEF404I MHLRES1I - ENDED - TIME=12.39.15 - ASID=002A - SC64
12.39.15 JOB21908  -MHLRES1I ENDED.  NAME-MHLRES2              TOTAL CPU TIME=   .00  TOTAL ELAPSED TIME=   .00
12.39.15 JOB21908  $HASP395 MHLRES1I ENDED
------ JES2 JOB STATISTICS ------
  16 MAR 2007 JOB EXECUTION DATE
           50 CARDS READ
          124 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            8 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,                JOB21908
          // NOTIFY=&SYSUID,TIME=1440,REGION=6M                                 00020000
          /*JOBPARM L=999,SYSAFF=*                                             00030000
          //*************************************************************
          //* DIAGNOSE THE CATALOG
          //* LIST ITS ALIASES
          //* LIST ITS SELF-DESCRIBING ENTRY COMPLETELY
          //* LIST ITS ENTRIES WITH JUST THE VOLUME INFORMATION
          IEFC653I SUBSTITUTION JCL - (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,NOTIFY=MHLRES1,TIME=1440,REGION=6M
          2 //DIAGLIST  EXEC PGM=IDCAMS
        3 //SETBKDS   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(NEW,PASS),
          //          DCB=MHLRES1.BCAT.EXPORT.DCB,
          //          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
        4 //SYSPRINT  DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(MOD,PASS),
          //          DCB=MHLRES1.BCAT.LIST.DCB,
          //          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
        5 //SYSIN     DD  *
          //*************************************************************
          //* EXPORT THE CATALOG IF THE DIAGNOSTICS WERE OKAY
        6 //EXPCAT    EXEC PGM=IDCAMS,COND=(8,LE)
        7 //CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,PASS),
          //          DCB=MHLRES1.BCAT.EXPORT.DCB,
          //          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
        8 //SYSPRINT  DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(MOD,PASS),
          //          DCB=MHLRES1.BCAT.LIST.DCB,
          //          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
        9 //SYSIN DD  *
            //*************************************************************
          //* CATALOG THE BACKUP AND LISTING IF EXPORT WAS OK
       10 //CATAL     EXEC PGM=IEFBR14,COND=(8,LE)
       11 //CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,CATLG)
       12 //SYSLIST   DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(OLD,CATLG)
          //*************************************************************
          //* COPY THE LISTING AND DISCARD THE BACKUP IF EXPORT FAILED
       13 //REPOUT    EXEC PGM=IDCAMS,COND=(0,EQ,CATAL)
       14 //SYSPRINT  DD SYSOUT=*
       15 //CATBACK   DD DSN=MHLRES1.BCAT.CATALOG.BACKUP(+1),DISP=(OLD,DELETE)
       16 //CATLIST   DD DSN=MHLRES1.BCAT.CATALOG.LISTING(+1),DISP=(OLD,DELETE)
       17 //CATPROB   DD DSN=MHLRES1.BCAT.PROBLEM.LISTING(+1),DISP=(NEW,CATLG),
          //          DCB=MHLRES1.BCAT.LIST.DCB,
          //          UNIT=SYSALLDA,SPACE=(TRK,(15,15),RLSE)
       18 //SYSIN     DD *
```

Figure 3-65 ICFRU Readiness check IDCAMS Diagnose and first EXPORT job JES2 output (1 of 2)

```
 STMT NO. MESSAGE
-
ICH70001I MHLRES1  LAST ACCESS AT 12:33:00 ON FRIDAY, MARCH 16, 2007
IEF236I ALLOC. FOR MHLRES1I DIAGLIST
IGD101I SMS ALLOCATED TO DDNAME (SETBKDS )
        DSN (MHLRES1.BCAT.CATALOG.BACKUP.G0005V00        )
        STORCLAS (STANDARD) MGMTCLAS (MCDB22) DATACLAS (        )
        VOL SER NOS= MLD00C
IGD101I SMS ALLOCATED TO DDNAME (SYSPRINT)
        DSN (MHLRES1.BCAT.CATALOG.LISTING.G0005V00       )
        STORCLAS (STANDARD) MGMTCLAS (MCDB22) DATACLAS (        )
        VOL SER NOS= MLD10C
IEF237I JES2 ALLOCATED TO SYSIN
IEF237I 8019 ALLOCATED TO SYS00001
IGD104I UCAT.FLETCHER                                RETAINED,  DDNAME=SYS00001
IEF142I MHLRES1I DIAGLIST - STEP WAS EXECUTED - COND CODE 0000
IGD106I MHLRES1.BCAT.CATALOG.BACKUP.G0005V00         PASSED,    DDNAME=SETBKDS
IGD106I MHLRES1.BCAT.CATALOG.LISTING.G0005V00        PASSED,    DDNAME=SYSPRINT
IEF285I   MHLRES1.MHLRES1I.JOB21908.D0000101.?         SYSIN
IEF373I STEP/DIAGLIST/START 2007075.1239
IEF374I STEP/DIAGLIST/STOP  2007075.1239 CPU    0MIN 00.02SEC SRB    0MIN 00.00SEC VIRT  1236K SYS   360K EXT     144K SYS
IEF236I ALLOC. FOR MHLRES1I EXPCAT
IGD103I SMS ALLOCATED TO DDNAME CATBACK
IGD103I SMS ALLOCATED TO DDNAME SYSPRINT
IEF237I JES2 ALLOCATED TO SYSIN
IEF237I 8019 ALLOCATED TO SYS00002
IGD104I UCAT.FLETCHER                                RETAINED,  DDNAME=SYS00002
IEF142I MHLRES1I EXPCAT - STEP WAS EXECUTED - COND CODE 0000
IGD106I MHLRES1.BCAT.CATALOG.BACKUP.G0005V00         PASSED,    DDNAME=CATBACK
IGD106I MHLRES1.BCAT.CATALOG.LISTING.G0005V00        PASSED,    DDNAME=SYSPRINT
IEF285I   MHLRES1.MHLRES1I.JOB21908.D0000102.?         SYSIN
IEF373I STEP/EXPCAT  /START 2007075.1239
IEF374I STEP/EXPCAT  /STOP  2007075.1239 CPU    0MIN 00.03SEC SRB    0MIN 00.00SEC VIRT   444K SYS   360K EXT    1752K SYS
IEF236I ALLOC. FOR MHLRES1I CATAL
IGD103I SMS ALLOCATED TO DDNAME CATBACK
IGD103I SMS ALLOCATED TO DDNAME SYSLIST
IEF142I MHLRES1I CATAL - STEP WAS EXECUTED - COND CODE 0000
IGD107I MHLRES1.BCAT.CATALOG.BACKUP.G0005V00         ROLLED IN, DDNAME=CATBACK
IGD107I MHLRES1.BCAT.CATALOG.LISTING.G0005V00        ROLLED IN, DDNAME=SYSLIST
IEF373I STEP/CATAL   /START 2007075.1239
IEF374I STEP/CATAL   /STOP  2007075.1239 CPU    0MIN 00.00SEC SRB    0MIN 00.00SEC VIRT     4K SYS   316K EXT       0K SYS
IEF202I MHLRES1I REPOUT - STEP WAS NOT RUN BECAUSE OF CONDITION CODES
IEF272I MHLRES1I REPOUT - STEP WAS NOT EXECUTED.
IEF373I STEP/REPOUT  /START 2007075.1239
IEF374I STEP/REPOUT  /STOP  2007075.1239 CPU    0MIN 00.00SEC SRB    0MIN 00.00SEC VIRT     0K SYS     0K EXT       0K SYS
IEF375I  JOB/MHLRES1I/START 2007075.1239
IEF376I  JOB/MHLRES1I/STOP  2007075.1239 CPU    0MIN 00.05SEC SRB    0MIN 00.00SEC
```

Figure 3-66 ICFRU Readiness check IDCAMS Diagnose and first EXPORT job JES2 output (2 of 2)

This was not the first run of this job since the initial data sets and GDGs were set up, so the relative GDG number is G0005V00 rather than G0001V00 as might be expected.

The significant data sets from this run are MHLRES1.BCAT.CATALOG.BACKUP.G0005V00 and MHLRES1.BCAT.CATALOG.LISTING.G0005V00.

The output from this job, as captured in the MHLRES1.BCAT.CATALOG.LISTING.* output, should be checked for errors, and if necessary any errors found should be corrected and the job run again.

Once the run is clean, as in this case, the significant part of the output is the last part of Figure 3-67.

```
 EXPORT UCAT.FLETCHER OFILE(CATBACK) TEMP
IDC0005I NUMBER OF RECORDS PROCESSED WAS 6
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 03/16/07 AT 12:39:15
IDC1147I IT IS RECOMMENDED THAT DIAGNOSE AND EXAMINE BE RUN BEFORE
IDC1147I IMPORT OF CATALOG
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 3-67 ICFRU Readiness check IDCAMS Diagnose and first EXPORT job captured output - last part

For the purposes of simulating a catalog recovery, we noted the time and date that the EXPORT was created, as listed in the message:

```
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 03/16/07 AT 12:39:15
```

In order to guarantee that there has been some catalog activity since the initial EXPORT was created, we created a job to define sixteen data sets.

In Figure 3-68 we show the job to allocate these data sets.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*********************************************************************
//* CATALOG A NUMBER OF DATASETS TO ENSURE THERE HAS BEEN SOME
//* CATALOG ACTIVITY
//* N.B. RUN THE CORRESPONDING DELETE JOB AFTER THE SECOND
//* EXPORT HAS BEEN CREATED
//DEFDS     EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//DD01      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST01,UNIT=SYSALLDA
//DD02      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST02,UNIT=SYSALLDA
//DD03      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST03,UNIT=SYSALLDA
//DD04      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST04,UNIT=SYSALLDA
//DD05      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST05,UNIT=SYSALLDA
//DD06      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST06,UNIT=SYSALLDA
//DD07      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST07,UNIT=SYSALLDA
//DD08      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST08,UNIT=SYSALLDA
//DD09      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST09,UNIT=SYSALLDA
//DD10      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST10,UNIT=SYSALLDA
//DD11      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST11,UNIT=SYSALLDA
//DD12      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST12,UNIT=SYSALLDA
//DD13      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST13,UNIT=SYSALLDA
//DD14      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST14,UNIT=SYSALLDA
//DD15      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST15,UNIT=SYSALLDA
//DD16      DD DISP=(,CATLG),SPACE=(TRK,(1,1)),
//          DSN=MHLCAT9.BCAT.TEST16,UNIT=SYSALLDA
//SYSIN     DD  *
  LISTCAT LVL(MHLCAT9.BCAT)
/*
```

Figure 3-68 ICFRU Readiness check - allocate test data sets

We then ran the same job as used to create the first EXPORT — the job shown in Figure 3-64 on page 79. The output was basically the same as when run the first time except that the data sets generated were suffixed G0006V00 instead of G0005V00.

In Figure 3-69 we show the last part of the MHLRES1.BCAT.CATALOG.LISTING.G0006V00 data set, which shows the result of the EXPORT command.

```
 EXPORT UCAT.FLETCHER OFILE(CATBACK) TEMP
IDC0005I NUMBER OF RECORDS PROCESSED WAS 22
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 03/16/07 AT 12:51:25
IDC1147I IT IS RECOMMENDED THAT DIAGNOSE AND EXAMINE BE RUN BEFORE
IDC1147I IMPORT OF CATALOG
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 3-69 ICFRU Readiness job last part of MHLRES1.BCAT.CATALOG.LISTING.G0002V00

For the purposes of simulating a catalog recovery, we noted the time and date that the second EXPORT was created, as listed in the message:

```
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 03/16/07 AT 12:51:25
```

We then identified the SMF data sets that contained the SMF data from just before the first export was created until just after the second export was created from 03/16/07 AT 12:39:15 to 03/16/07 AT 12:51:25.

The data sets were identified as SMFDATA.ALLRECS.G3350V00 through to SMFDATA.ALLRECS.G3351V00.

We then used this information in the job to run program ICFRRSV.

In Figure 3-71 on page 85 we show the JCL to run the CRURRSV program. It has been set up to use the SMF records that may contain updates to the catalog, and set up with the start and stop date and times as taken from the IDCAMS EXPORT job and the SMF records.

The significant PARM values of catalog name, start_time, stop_time, and gap time are highlighted in Figure 3-70.

```
// PARM=('UCAT.FLETCHER',
//       '03/16/07','12:39:15',
//       '03/16/07','12:51:25',
//       '0030',
```

Figure 3-70 CRURRSV program PARM values

The appropriate SMF data set names and the output data set have been specified.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*****************************************************************
//* THIS JCL EXECUTES CRURRSV TO SELECT THE APPROPRIATE SMF
//* RECORDS FROM THE DATASETS COVERING THE PERIOD WHERE RECORDS
//* MAY BE LOST FROM THE CATALOG.
//* THE INPUT IS THE SMF DATA SET(S), OUTPUT IS A CONSOLIDATED
//* SET OF SMF RECORDS. SELECTION IS CONTROLED THROUGH THE PARM
//*
//RRSV  EXEC PGM=CRURRSV,
// PARM=('UCAT.FLETCHER',
//       '03/16/07','12:39:15',
//       '03/16/07','12:51:25',
//       '0030',
//       '0000')
//*PARM=('CATALOG.NAME',
//*      'STARTDATE','STARTTIME',   MM/DD/YY HH:MM:SS
//*      'STOPDATE,'STOPTIME',
//*      'GAPTIME',          (MINUTES)
//*      'CLOCKDIFFERENCE')  (SECONDS)
//SYSDUMP  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLOG   DD SYSOUT=*
//SMFIN    DD DISP=SHR,DCB=BUFNO=60,DSN=SMFDATA.ALLRECS.G3350V00
//         DD DISP=SHR,DCB=BUFNO=60,DSN=SMFDATA.ALLRECS.G3351V00
//SMFOUT   DD DISP=(,CATLG),DCB=BUFNO=60,
//         UNIT=SYSDA,SPACE=(CYL,(10,2)),
//         DSN=MHLRES1.BCAT.SMF.CAT.RECS.FLETCHER
```

Figure 3-71 ICFRU Readiness check - CRURRSV job

CRURRSV produces several reports based on the analysis of the SMF data. For explanation of the information you should refer to Appendix A of *DFSMS Managing Catalogs,* SC26-7409.

In Figure 3-72 we show the output from running the job as shown in Figure 3-71 on page 85.

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRSV SYSPRINT                 03/16/07 (07.075)  17:42:17    PAGE 01
                 RECORD SELECTION AND VALIDATION REPORT
                          EXECUTION PARAMETERS
CATALOG NAME                UCAT.FLETCHER
RECORD SELECTION START      03/16/07 (07.075)  12:39:15
RECORD SELECTION STOP       03/16/07 (07.075)  12:51:25
SIGNIFICANT GAP TIME        0030 MINUTES
MAXIMUM CLOCK DIFFERENCE    0000 SECONDS
REPORT FOR ALL SYSTEMS
RECORD SELECTION AND VALIDATION CONDITION CODE IS 00
          0 ANOMALIES (LOST DATA, GAPS) DETECTED
         16 RECORDS SELECTED FOR UCAT.FLETCHER
         16 DEFINE (TYPE 61) RECORDS SELECTED
          0 DELETE (TYPE 65) RECORDS SELECTED
          0 ALTER  (TYPE 66) RECORDS SELECTED
      01 SYSTEM(S) RECORDED CHANGES TO THIS CATALOG
         SC64
FOR CATALOG UCAT.FLETCHER
03/16/07 (07.075)  11:29:33.57 OLDEST SMF CATALOG RECORD FOUND
03/16/07 (07.075)  12:44:24.44 OLDEST SMF CATALOG RECORD SELECTED
03/16/07 (07.075)  12:44:24.59 NEWEST SMF CATALOG RECORD SELECTED
03/16/07 (07.075)  12:44:24.59 NEWEST SMF CATALOG RECORD FOUND
FOR ALL SMF RECORD TYPES
03/16/07 (07.075)  06:40:01.08 OLDEST SMF RECORD FOUND (ANY TYPE)
03/16/07 (07.075)  16:30:08.64 NEWEST SMF RECORD FOUND (ANY TYPE)
          1 SYSTEM IDENTIFIERS WERE FOUND
     26,478 TOTAL SMF RECORDS WERE READ
          2 SMF SWITCH (TYPE 90, SUBTYPE 6) RECORDS WERE FOUND
          0 SMF EOD (TYPE 90, SUBTYPE 7) RECORDS WERE FOUND
          0 SMF IPL (TYPE 0) RECORDS WERE FOUND
          0 SMF LOST DATA (TYPE 7) RECORDS WERE FOUND
          0 FORWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
            0030 MINUTES WERE FOUND
          0 BACKWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
            0030 MINUTES WERE FOUND
RECORD SELECTION AND VALIDATION CONDITION CODE IS 00
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRSV SYSPRINT                 03/16/07 (07.075)  17:42:17    PAGE 02
                 RECORD SELECTION AND VALIDATION REPORT
                          EXECUTION PARAMETERS
CATALOG NAME                UCAT.FLETCHER
RECORD SELECTION START      03/16/07 (07.075)  12:39:15
RECORD SELECTION STOP       03/16/07 (07.075)  12:51:25
SIGNIFICANT GAP TIME        0030 MINUTES
MAXIMUM CLOCK DIFFERENCE    0000 SECONDS
REPORT FOR SYSTEM SC64
RECORD SELECTION AND VALIDATION CONDITION CODE IS 00
          0 ANOMALIES (LOST DATA, GAPS) DETECTED
         16 RECORDS SELECTED FOR UCAT.FLETCHER
         16 DEFINE (TYPE 61) RECORDS SELECTED
          0 DELETE (TYPE 65) RECORDS SELECTED
          0 ALTER  (TYPE 66) RECORDS SELECTED
FOR CATALOG UCAT.FLETCHER
03/16/07 (07.075)  11:29:33.57 OLDEST SMF CATALOG RECORD FOUND
03/16/07 (07.075)  12:44:24.44 OLDEST SMF CATALOG RECORD SELECTED
03/16/07 (07.075)  12:44:24.59 NEWEST SMF CATALOG RECORD SELECTED
03/16/07 (07.075)  12:44:24.59 NEWEST SMF CATALOG RECORD FOUND
FOR ALL SMF RECORD TYPES
03/16/07 (07.075)  06:40:01.08 OLDEST SMF RECORD FOUND (ANY TYPE)
03/16/07 (07.075)  16:30:01.08 NEWEST SMF RECORD FOUND (ANY TYPE)
     26,478 TOTAL SMF RECORDS WERE READ
          2 SMF SWITCH (TYPE 90, SUBTYPE 6) RECORDS WERE FOUND
          0 SMF EOD (TYPE 90, SUBTYPE 7) RECORDS WERE FOUND
          0 SMF IPL (TYPE 0) RECORDS WERE FOUND
          0 SMF LOST DATA (TYPE 7) RECORDS WERE FOUND
          0 FORWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
            0030 MINUTES WERE FOUND
          0 BACKWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
            0030 MINUTES WERE FOUND
RECORD SELECTION AND VALIDATION CONDITION CODE IS 00
```

Figure 3-72 ICFRU Readiness check - CRURRSV job output

In addition to the report output, a separate SYSLOG file is produced. This is useful for determining the number of SMF switches that occurred and whether there might be gaps in the SMF data.

In Figure 3-73 we show the output from the CRURRSV SYSLOG.

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRSV SYSLOG                  03/16/07 (07.075)  17:42:17
CRU023I SWITCH SMF RECORD FOUND FOR SYSID SC64
        03/16/07 (07.075)  06:40:01.17 RECORD BEING PROCESSED - DUMP FOLLOWS

+0000 00AEC8  00A40000 1E5A0024 9F750107 075FE2C3 F6F40000 00000024 000C0001 00000030   *.u...!.......¬SC64.............*
+0020 00AEE8  00740001 0006F0F1 E2D4C640 40404040 40404040 40404040 40404040 40404040   *.....01SMF                     *
+0040 00AF08  40404040 0064FD95 0107056F E2E8E2F1 4BE2C3F6 F44BD4C1 D5F24040 40404040   *    ...n...?SYS1.SC64.MAN2     *
+0060 00AF28  40404040 40404040 40404040 40404040 40404040 40404040 E2E8E2F1 4BE2C3F6   *                         SYS1.SC6*
+0080 00AF48  F44BD4C1 D5F14040 40404040 40404040 40404040 40404040 40404040 40404040   *4.MAN1                         *
+00A0 00AF68  40404040                                                                 *                               *
CRU023I SWITCH SMF RECORD FOUND FOR SYSID SC64
        03/16/07 (07.075)  11:40:00.57 RECORD BEING PROCESSED - DUMP FOLLOWS

+0000 00AEC8  00A40000 1E5A0040 16790107 075FE2C3 F6F40000 00000024 000C0001 00000030   *.u...!. .....¬SC64.............*
+0020 00AEE8  00740001 0006F0F1 E2D4C640 40404040 40404040 40404040 40404040 40404040   *.....01SMF                     *
+0040 00AF08  40404040 0064FD95 0107056F E2E8E2F1 4BE2C3F6 F44BD4C1 D5F14040 40404040   *    ...n...?SYS1.SC64.MAN1     *
+0060 00AF28  40404040 40404040 40404040 40404040 40404040 40404040 E2E8E2F1 4BE2C3F6   *                         SYS1.SC6*
+0080 00AF48  F44BD4C1 D5F24040 40404040 40404040 40404040 40404040 40404040 40404040   *4.MAN2                         *
+00A0 00AF68  40404040                                                                 *                               *
```

Figure 3-73 CFRU Readiness check - CRRRSV SYSLOG output

Once the SMF records have been selected, they must be sorted into the form required by CRURRAP.

In Figure 3-74 we show the JCL to sort the SMF records. The input and output data sets are bolded.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*****************************************************************
//* THIS JCL EXECUTES SORT TO PREPARE FOR INPUT TO CRURRAP
//*
//RRSV  EXEC PGM=ICEMAN
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DISP=SHR,
//         DSN=MHLRES1.BCAT.SMF.CAT.RECS.FLETCHER
//SORTOUT  DD DISP=(,CATLG,DELETE),UNIT=SYSDA,
//         SPACE=(CYL,(10,2),RLSE),
//         DSN=MHLRES1.BCAT.SORTED.SMF.CAT.RECS.FLETCHER
//SYSIN DD *
 OPTION DYNALLOC=SYSDA,FILSZ=E10000
 SORT FIELDS=(218,44,CH,A,262,1,BI,A,11,4,PD,D,7,4,BI,D)
/*
```

Figure 3-74 ICFRU Readiness check - SMF sort job

The output from the SORT should be checked to verify that the correct number of records as reported by CRURRSV have been sorted.

In Figure 3-75 we show the message output resulting from running the JCL as shown in Figure 3-74 on page 87.

```
 BLOCKSET     SORT  TECHNIQUE SELECTED
 VISIT http://www.ibm.com/storage/dfsort FOR DFSORT PAPERS, EXAMPLES AND MORE
 - CONTROL STATEMENTS FOR 5694-A01, Z/OS DFSORT V1R5 - 17:42 ON FRI MAR 16, 2007 -
  OPTION DYNALLOC=SYSDA,FILSZ=E10000
  SORT FIELDS=(218,44,CH,A,262,1,BI,A,11,4,PD,D,7,4,BI,D)
 RECORD TYPE IS V - DATA STARTS IN POSITION 5
 C5-K21008 C6-K90007 C7-K90000 C8-K90007 E4-K90007 C9-BASE   E5-K18181 E6-K18181 B0-Q96745 E7-K11698
 ICEAM1 ENVIRONMENT IN EFFECT - ICEAM1 INSTALLATION MODULE SELECTED
 MHLRES1I.RRSV    .        , INPUT LRECL = 32760, BLKSIZE = 4096, TYPE = VBS
 MAIN STORAGE = (MAX,6291456,6291456)
 MAIN STORAGE ABOVE 16MB = (6214096,6214096)
 OPTIONS: OVFLO=RC0 ,PAD=RC0 ,TRUNC=RC0 ,SPANINC=RC16,VLSCMP=N,SZERO=Y,RESET=Y,VSAMEMT=Y,DYNSPC=256
 OPTIONS: SIZE=6291456,MAXLIM=1048576,MINLIM=450560,EQUALS=Y,LIST=Y,ERET=RC16 ,MSGDDN=SYSOUT
 OPTIONS: VIO=N,RESDNT=ALL ,SMF=NO    ,WRKSEC=Y,OUTSEC=Y,VERIFY=N,CHALT=N,DYNALOC=(SYSDA
,004),ABCODE=MSG
 OPTIONS: RESALL=4096,RESINV=0,SVC=109 ,CHECK=Y,WRKREL=Y,OUTREL=Y,CKPT=N,STIMER=Y,COBEXIT=COB2
 OPTIONS: TMAXLIM=6291456,ARESALL=0,ARESINV=0,OVERRGN=65536,CINV=Y,CFW=Y,DSA=0
 OPTIONS: VLSHRT=N,ZDPRINT=Y,IEXIT=N,TEXIT=N,LISTX=N,EFS=NONE    ,EXITCK=S,PARMDDN=DFSPARM ,FSZEST=N
 OPTIONS: HIPRMAX=OPTIMAL,DSPSIZE=MAX ,ODMAXBF=0,SOLRF=Y,VLLONG=N,VSAMIO=N,MOSIZE=MAX
 OPTIONS: NULLOUT=RC0
 EXCP ACCESS METHOD USED FOR SORTOUT
 EXCP ACCESS METHOD USED FOR SORTIN
 DC 8184 TC 0 CS DSV   KSZ 58 VSZ 58
 FSZ=8184 BC  IGN=10000 E  AVG=16380 0  WSP=10 C  DYN=0 0
 B1-K19866 B2-K17569 EC-K10929 B4-K17569 E8-K21008
 OUTPUT LRECL = 32760, BLKSIZE = 27998, TYPE = VBS  (SDB)
 IN MAIN STORAGE SORT
 INSERT 0, DELETE 0
 RECORDS - IN: 16, OUT: 16
 NUMBER OF BYTES SORTED: 4864
 AVERAGE RECORD LENGTH = 304 BYTES
 TOTAL WORK DATA SET TRACKS ALLOCATED: 0 , TRACKS USED: 0
 MEMORY OBJECT STORAGE USED = 1M BYTES
 HIPERSPACE STORAGE USED = 0K BYTES
 DATA SPACE STORAGE USED = 0K BYTES
 END OF DFSORT
```

Figure 3-75 ICFRU Readiness check - SORT job messages

The sorted SMF records must be processed by CRURRAP to produce a data set that is to be used as a file for IDCAMS IMPORT. For the purposes of this ICFRU readiness check, it is not used as import by IDCAMS, but it will be compared with the second IDCAMS EXPORT made during this check.

In Figure 3-76 we show the job set up to construct a pseudo IDCAMS EXPORT data set that could be used to re-create a lost catalog.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*************************************************************
//* THIS JCL EXECUTES CRURRAP TO GENERATE THE EQUIVALENT OF
//* AND IDCAMS EXPORT.
//* THE INPUT IS THE SMF DATA SET(S) AS PRODUCED BY CRURRSV AND
//* SORTED BY SMF.
//* EXACTLY THE SAME SELECTION PARAMETERS MUST BE SET AS SPECIFIED
//* TO CRURRSV.
//*
//RRSV  EXEC PGM=CRURRAP,
// PARM=('UCAT.FLETCHER',
//       '03/16/07','12:39:15',
//       '03/16/07','12:51:25',
//       '0030',
//       '0000')
//*PARM=('CATALOG.NAME',
//*      'STARTDATE','STARTTIME',   MM/DD/YY HH:MM:SS
//*      'STOPDATE','STOPTIME',
//*      'GAPTIME',          (MINUTES)
//*      'CLOCKDIFFERENCE')  (SECONDS)
//SYSDUMP  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLOG   DD SYSOUT=*
//SMFIN    DD DISP=SHR,DCB=BUFNO=60,
//         DSN=MHLRES1.BCAT.SORTED.SMF.CAT.RECS.FLETCHER
//EXPIN    DD DISP=SHR,DCB=BUFNO=60,
//         DSN=MHLRES1.BCAT.CATALOG.BACKUP.G0006V00
//EXPOUT   DD DISP=(,CATLG),DCB=BUFNO=60,
//         UNIT=SYSDA,SPACE=(CYL,(10,2)),
//         DSN=MHLRES1.BCAT.NEW.CATALOG.EXPORT.FLETCHER
```

Figure 3-76 ICFRU Readiness check - CRURRAP job

The PARM values used for CRURRAP must be the same as those used for CRURRSV.

For the purposes of this readiness check we compare it with the second IDCAMS EXPORT data set. The two should be equal.

In Figure 3-77 through to Figure 3-79 on page 91 we show the output from the CRURRAP job as defined in Figure 3-76 on page 89.

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRAP SYSPRINT                 03/16/07 (07.075) 20:10:32     PAGE 1
                  RECORD ANALYSIS AND PROCESSING REPORT
                            EXECUTION PARAMETERS
CATALOG NAME                UCAT.FLETCHER
RECORD SELECTION START      03/16/07 (07.075)  12:39:15
RECORD SELECTION STOP       03/16/07 (07.075)  12:51:25
SIGNIFICANT GAP TIME        0030 MINUTES
MAXIMUM CLOCK DIFFERENCE    0000 SECONDS
 RECORD ANALYSIS AND PROCESSING CONDITION CODE IS 08
                             ERROR REPORT
        16 TOTAL ERRORS (CONDITION CODES 12 AND 8)
         0 RECORDS REJECTED FROM EXPIN                (LOGGED, DUMPED, CC=12)
               0 RECORDS WITH INVALID LENGTHS              (CRU302I)
               0 RECORDS WITH INVALID CATALOG RECORD TYPES (CRU303I)
        16 ERRORS IN EVENT SEQUENCE INVOLVING THE MOST CURRENT RECORD
                                                    (LOGGED, DUMPED, CC=8)
              16 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
                     0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU203I)
                     0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU204I)
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
                    16 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU209I)
               0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRAP SYSPRINT                 03/16/07 (07.075) 20:10:32     PAGE 2
                           ANOMALY REPORT
         0 ANOMALIES (CONDITION CODES 4 AND 0)
         0 SYNCHRONIZATION CHECKS INVOLVING THE MOST CURRENT RECORD
                           BUT WITH NO EVENT SEQUENCE ERROR  (CRU113I)
                                                    (LOGGED, DUMPED, CC=4)
               0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
               0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
               0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
               0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
               0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)
         0 ERRORS IN EVENT SEQUENCE INVOLVING A SUPERSEDED RECORD
                                                          (LOGGED, CC=4)
               0 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
                     0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU106I)
                     0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU107I)
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)
                     0 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU112I)
               0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)
         0 SYNCHRONIZATION CHECKS INVOLVING A SUPERSEDED RECORD
                           BUT WITH NO EVENT SEQUENCE ERROR  (CRU020I)
                                                    (NOT LOGGED, CC=0)
               0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
               0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
               0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
               0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
               0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
```

Figure 3-77 ICFRU Readiness check - CRURRAP job output (part 1 of 3)

```
CRURRAP SYSPRINT                03/16/07 (07.075) 20:10:32     PAGE 3
         REPORT OF RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
        22 TOTAL RECORDS PROCESSED (NO ERROR/NO ANOMALY, CONDITION CODE 0)
               6 MOST CURRENT RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
                     0 SMF INSERT FOR A NEW RECORD              (CRU002I)
                     0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
                     0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
                     0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
                     0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
                     0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)
                     0 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU008I)
                     0 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU009I)
                     6 EXPORT RECORD CARRIED FORWARD            (CRU001I)
              16 SUPERSEDED RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
                     0 SMF INSERT FOR A NEW RECORD              (CRU012I)
                     0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
                     0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
                     0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
                     0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
                     0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
                     0 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU018I)
                     0 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU019I)
                    16 EXPORT RECORD SUPERSEDED                 (CRU011I)
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
```

Figure 3-78 ICFRU Readiness check - CRURRAP job output (part 2 of 3)

```
CRURRAP SYSPRINT                03/16/07 (07.075) 20:10:32     PAGE 4
                  REPORT OF RECORDS BY DATA SET
        32 TOTAL RECORDS IN THE NEW EXPORT DATA SET (EXPOUT)
                    10 CONTROL RECORDS
                    22 CATALOG RECORDS
              16 RECORDS FORWARDED FROM THE OLD EXPORT DATA SET (EXPIN)
                    10 CONTROL RECORDS
                     6 CATALOG RECORDS
              16 CATALOG RECORDS SELECTED FROM THE SMF DATA SET (SMFIN)
        32 TOTAL RECORDS FROM THE OLD EXPORT DATA SET (EXPIN)
              16 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET
                    10 CONTROL RECORDS
                     6 CATALOG RECORDS
              16 RECORDS SUPERSEDED OR DELETED (BASED ON SMF DATA)
               0 RECORDS REJECTED BECAUSE OF ERRORS
                     0 INVALID LENGTH
                     0 UNRECOGNIZED CATALOG RECORD TYPE
        16 TOTAL RECORDS FROM THE SMF DATA SET (SMFIN)
              16 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET
               0 RECORDS SUPERSEDED OR DELETED BY NEWER SMF RECORDS
               0 RECORDS REJECTED
                     0 NOT AN MVS SMF RECORD
                     0 NOT AN SMF CATALOG RECORD
                     0 NOT AN SMF CATALOG RECORD FOR THIS CATALOG
                     0 DATE/TIME EARLIER THAN EFFECTIVE START TIME
                     0 DATE/TIME LATER THAN EFFECTIVE STOP TIME
        32 TOTAL OF ALL OUTPUT RECORDS
        16 TOTAL OF ALL RECORDS DISCARDED
        48 TOTAL OF ALL INPUT RECORDS
```

Figure 3-79 ICFRU Readiness check - CRURRAP job output (part 3 of 3)

In addition to the standard job output, as with CRURRSV, CRURRAP also produces a SYSLOG.

In Figure 3-80 we show the SYLOG output. In this case the data reflected documents the checking that occurred to determine whether a record found by the ICFRU analysis is more or less up to date than that shown in the second IDCAMS EXPORT run.

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRAP SYSLOG                  03/16/07 (07.075)  20:10:32
CRU104I SPECIFIED START PRECEDES EXPORT, ANOMALIES POSSIBLE

CRU209I SMF INSERT IS MOST CURRENT BUT IS PRECEDED BY EXPORT RECORD FOR
        (A) MHLCAT9.BCAT.TEST01                      /00
        NEWER: DEFINE FROM SYS SC64 AT 12:44:24.44 ON 03/16/07 (07.075)
        OLDER: EXPORT RECORD
        SMF INSERT RECORD IS WRITTEN TO THE NEW EXPORT DATA SET

+0000 4F8000  01300000 1E3D0045 FBCC0107 075FE2C3 F6F44040 4040C9D5 00000028 000A0001   *.............¬SC64    IN........*
+0020 4F8020  00000032 009E0001 40F1C3C1 E3D4C7D4 E340D4C8 D3D9C5E2 F1C90045 FB950107   *........ 1CATMGMT MHLRES1I...n..*
+0040 4F8040  075FD4C8 D3D9C5E2 F14040E4 C3C1E34B C6D3C5E3 C3C8C5D9 40404040 40404040   *.¬MHLRES1  UCAT.FLETCHER       *
+0060 4F8060  40404040 40404040 40404040 40404040 40404040 404040C1 D4C8D3C3 C1E3F94B   *                        AMHLCAT9.*
+0080 4F8080  C2C3C1E3 4BE3C5E2 E3F0F140 40404040 40404040 40404040 40404040 40404040   *BCAT.TEST01                     *
+00A0 4F80A0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040   *                                *
+00C0 4F80C0  40404040 40404040 40404040 00644040 00600034 C1000001 2DD4C8D3 C3C1E3F9   *             ..  .-..A....MHLCAT9*
+00E0 4F80E0  4BC2C3C1 E34BE3C5 E2E3F0F1 40404040 40404040 40404040 40404040 40404040   *.BCAT.TEST01                    *
+0100 4F8100  40404040 40000014 01FFFFFF FFFFFFFF FF080707 5F00000F 01000016 0400E2C2   *     ..............¬.........SB*
+0120 4F8120  D6E7C5F1 3010200F 08000000 00000000                                       *OXE1............               *

CRU011I EXPORT RECORD WAS SUPERSEDED AND WAS THE OLDEST RECORD FOR
        (A) MHLCAT9.BCAT.TEST01                      /00
        RECORD IS BYPASSED, ACTION WAS TAKEN FOR A MORE CURRENT RECORD

CRU209I SMF INSERT IS MOST CURRENT BUT IS PRECEDED BY EXPORT RECORD FOR
        (A) MHLCAT9.BCAT.TEST02                      /00
        NEWER: DEFINE FROM SYS SC64 AT 12:44:24.45 ON 03/16/07 (07.075)
        OLDER: EXPORT RECORD
        SMF INSERT RECORD IS WRITTEN TO THE NEW EXPORT DATA SET

+0000 4F8000  01300000 1E3D0045 FBCD0107 075FE2C3 F6F44040 4040C9D5 00000028 000A0001   *.............¬SC64    IN........*
+0020 4F8020  00000032 009E0001 40F1C3C1 E3D4C7D4 E340D4C8 D3D9C5E2 F1C90045 FB950107   *........ 1CATMGMT MHLRES1I...n..*
+0040 4F8040  075FD4C8 D3D9C5E2 F14040E4 C3C1E34B C6D3C5E3 C3C8C5D9 40404040 40404040   *.¬MHLRES1  UCAT.FLETCHER       *
+0060 4F8060  40404040 40404040 40404040 40404040 40404040 404040C1 D4C8D3C3 C1E3F94B   *                        AMHLCAT9.*
+0080 4F8080  C2C3C1E3 4BE3C5E2 E3F0F240 40404040 40404040 40404040 40404040 40404040   *BCAT.TEST02                     *
+00A0 4F80A0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040   *                                *
+00C0 4F80C0  40404040 40404040 40404040 00644040 00600034 C1000001 2DD4C8D3 C3C1E3F9   *             ..  .-..A....MHLCAT9*
+00E0 4F80E0  4BC2C3C1 E34BE3C5 E2E3F0F2 40404040 40404040 40404040 40404040 40404040   *.BCAT.TEST02                    *
+0100 4F8100  40404040 40000014 01FFFFFF FFFFFFFF FF080707 5F00000F 01000016 0400E2C2   *     ..............¬.........SB*
+0120 4F8120  D6E7C5F1 F7010200F 08000000 00000000                                      *OXA7............               *

CRU011I EXPORT RECORD WAS SUPERSEDED AND WAS THE OLDEST RECORD FOR
        (A) MHLCAT9.BCAT.TEST02                      /00
        RECORD IS BYPASSED, ACTION WAS TAKEN FOR A MORE CURRENT RECORD


.
.
.
```

Figure 3-80 ICFRU Readiness check - CRURRAP SYSLOG output

There are several instances of these checks. They are not all shown in the example because they all reflect the same situation.

For the purpose of the ICFRU readiness check, the last step is to compare the second IDCAMS EXPORT data set with the one constructed by running CRURRSV, DFSORT™, and CRURRAP.

In Figure 3-81 we show the job used to do the compare using the standard IEBCOMPR utility.

```
//MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*****************************************************************
//* THIS JCL EXECUTES IEBCOMPR TO COMPARE THE NEW EXPORT DATA SET
//* PRODUCED BY CRURRAP WITH THE SECOND EXPORT DATA SET CREATED
//* AS PART OF THE ICFRU READINESS CHECK
//*
//* THIS COMPARISON IS ONLY FOR TESTING THE CATALOG UPDATE PROCESS
//* SINCE THE SECOND EXPORT DATA SET WILL NOT EXIST IN A REAL
//* RECOVERY SITUATION
//*
//COMP  EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=MHLRES1.BCAT.CATALOG.BACKUP.G0006V00
//SYSUT2   DD DISP=SHR,DSN=MHLRES1.BCAT.NEW.CATALOG.EXPORT.FLETCHER
//SYSIN    DD DUMMY
```

Figure 3-81 ICFRU Readiness check - IEBCOMPR compare job

In Figure 3-82 we show the full output from the IEBCOMPR job because the utility does not document the data sets in its output, so it is necessary to show the JES2 output showing the SYSUT1 and SYSUT2 DDNAMEs with data sets.

```
 J E S 2  J O B  L O G  --  S Y S T E M  S C 6 4  --  N O D E  W T S C P L X 2

20.11.18 JOB22060 ---- FRIDAY,    16 MAR 2007 ----
20.11.18 JOB22060  IRR010I  USERID MHLRES1  IS ASSIGNED TO THIS JOB.
20.11.18 JOB22060  ICH70001I MHLRES1  LAST ACCESS AT 20:10:32 ON FRIDAY, MARCH 16, 2007
20.11.18 JOB22060  $HASP373 MHLRES1I STARTED - INIT 1    - CLASS A - SYS SC64
20.11.18 JOB22060  IEF403I MHLRES1I - STARTED - TIME=20.11.18 - ASID=002A - SC64
20.11.18 JOB22060  -                                        --TIMINGS (MINS.)--            ----PAGING COUNTS---
20.11.18 JOB22060  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG   PAGE   SWAP    VIO SWAPS STEPNO
20.11.18 JOB22060  -MHLRES1I          COMP        00     53    .00    .00    .00    209   0      0      0      0     0     1
20.11.18 JOB22060  IEF404I MHLRES1I - ENDED - TIME=20.11.18 - ASID=002A - SC64
20.11.18 JOB22060  -MHLRES1I ENDED.  NAME-MHLRES2              TOTAL CPU TIME=   .00  TOTAL ELAPSED TIME=   .00
20.11.18 JOB22060  $HASP395 MHLRES1I ENDED
------ JES2 JOB STATISTICS ------
  16 MAR 2007 JOB EXECUTION DATE
           17 CARDS READ
           54 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            3 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //MHLRES1I JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,                JOB22060
          // NOTIFY=&SYSUID,TIME=1440,REGION=6M                                 00020000
          /*JOBPARM L=999,SYSAFF=*                                             00030000
          //*************************************************************
          //* THIS JCL EXECUTES IEBCOMPR TO COMPARE THE NEW EXPORT DATA SET
          //* PRODUCED BY CRURRAP WITH THE SECOND EXPORT DATA SET CREATED
          //* AS PART OF THE ICFRU READINESS CHECK
          //*
          //* THIS COMPARISON IS ONLY FOR TESTING THE CATALOG UPDATE PROCESS
          //* SINCE THE SECOND EXPORT DATA SET WILL NOT EXIST IN A REAL
          //* RECOVERY SITUATION
          //*
          IEFC653I SUBSTITUTION JCL - (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,NOTIFY=MHLRES1,TIME=1440,REGION=6M
        2 //COMP  EXEC PGM=IEBCOMPR
        3 //SYSPRINT DD SYSOUT=*
        4 //SYSUT1   DD DISP=SHR,DSN=MHLRES1.BCAT.CATALOG.BACKUP.G0006V00
        5 //SYSUT2   DD DISP=SHR,DSN=MHLRES1.BCAT.NEW.CATALOG.EXPORT.FLETCHER
        6 //SYSIN    DD DUMMY
ICH70001I MHLRES1  LAST ACCESS AT 20:10:32 ON FRIDAY, MARCH 16, 2007
IEF236I ALLOC. FOR MHLRES1I COMP
IEF237I JES2 ALLOCATED TO SYSPRINT
IGD103I SMS ALLOCATED TO DDNAME SYSUT1
IGD103I SMS ALLOCATED TO DDNAME SYSUT2
IEF237I DMY  ALLOCATED TO SYSIN
IEF142I MHLRES1I COMP - STEP WAS EXECUTED - COND CODE 0000
IEF285I   MHLRES1.MHLRES1I.JOB22060.D0000101.?         SYSOUT
IGD104I MHLRES1.BCAT.CATALOG.BACKUP.G0006V00         RETAINED,  DDNAME=SYSUT1
IGD104I MHLRES1.BCAT.NEW.CATALOG.EXPORT.FLETCHER     RETAINED,  DDNAME=SYSUT2
IEF373I STEP/COMP    /START 2007075.2011
IEF374I STEP/COMP    /STOP  2007075.2011 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT   376K SYS   320K EXT       4K SYS
IEF375I  JOB/MHLRES1I/START 2007075.2011
IEF376I  JOB/MHLRES1I/STOP  2007075.2011 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC
                                        COMPARE UTILITY                                  PAGE 0001
    END OF JOB-TOTAL NUMBER OF RECORDS COMPARED = 00000032
```

Figure 3-82 ICFRU Readiness check - IEBCOMPR compare job output

3.10.5 ICFRU implementation final steps

One ICFRU has been validated as a means to reconstruct IDCAMS EXPORT data sets, the processes necessary to regularly capture IDCAMS EXPORT copies of catalogs and to capture SMF records need to be set up.

Note: It is important to take into account the necessity to capture the SMF records from all systems that have access to the catalogs in question, and they must all participate in the recovery process.

4

OAM enhancements

In this chapter we discuss new and changed functions in OAM. The following topics are covered:

- Binary large object support
- Immediate backup copy
- Automated selection of RECYCLE volumes
- Global display keyword
- Update from z/OS V1.7

4.1 Binary large object support

DB2 UDB for OS/390 Version 6 introduced the support for large objects (LOBs). They can contain text documents, images, or movies, and can be stored directly in the DBMS with sizes up to 2 gigabytes per object and 65,536 TB for a single LOB column in a 4,096 partition table. The introduction of these new data types has implied some changes in the administration processes and programming techniques. The IBM Redbook *LOBs with DB2 for z/OS: Stronger and Faster,* SG24-7270, describes the usage of LOBs with DB2 for z/OS. In z/OS V1R8, OAM has been enhanced for use with DB2's LOB support.

If LOB support is not enabled, OAM stores objects larger than 32 K in multiple rows in the 32 K DB2 table. A 256 MB object for example would take approximately 8,000 rows to store. When LOB support is enabled, objects up to 256 MB in size can be stored in a single row. This has the effect of improving performance and reducing the frequency of lock escalations and time outs.

4.1.1 Implementing

These are the steps we used in order to implement the LOB function:

1. Ensure that the values for LOBVALA and LOBVALS in DB2's DSNZPARM are sufficient for supporting large objects. They must allow for objects greater than 256 M in size in order to use OAM's new function. See the IBM Redbooks publication *LOBs with DB2 for z/OS: Stronger and Faster,* SG24-7270, for more information about preparing DB2 for LOBs.
2. Verify that the MOS=xxx value in your IEFSSNxx PARMLIB member is sufficient. The default is 50 M, and future attempts to use OSREQ to store objects larger than 50 M fail if this value is not specified. Figure 4-1 shows an example of an IEFSSNxx member with MOS=xxx customized to support objects up to 256 M in size.

```
EDIT       SYS1.PARMLIB(IEFSSNR7) - 01.76
Command ===>
000213 SUBSYS SUBNAME(OAM1)
000214   INITRTN(CBRINIT)
000215   INITPARM('MSG=EM,MOS=256')
```

Figure 4-1 Setting the MOS=xxx value in IEFSSNxx

3. Add the LOB keyword in your IEFSSNxx PARMLIB member. The allowed values are A, P, and N. LOB=N is the default. Specifying A puts objects for all storage groups into a LOB storage structure, specifying P places objects for a partial list of storage groups into a LOB storage structure, and specifying N does not put any objects into a LOB storage structure. Figure 4-2 is an excerpt from the *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, publication explaining each value in detail.

```
LOB=x
   Specifies whether or not OAM exploits DB2 LOB support for large
   objects that exceed 32 KB (32640 bytes). LOB has the following
   options:
      LOB=A specifies that, for all storage groups, objects that
      exceed 32 KB are to be stored in a LOB storage structure
      when stored to DB2. LOB=A indicates to OAM that the
      installation has created LOB storage structures and
      associated V_OSM_LOB_BASE_TBL views for ALL object storage
      groups defined in the ACDS. This results in optimal
      performance when you want to store large objects(greater
      than 32 KB) to DB2, because OAM does not query DB2 to see if
      the LOB base table view exists. If the LOB base table view
      does not exist, the large object store fails.
      LOB=P indicates to OAM that the installation has created LOB
      storage structures and associated V_OSM_LOB_BASE_TBL views
      for a PARTIAL list of object storage groups defined in the
      ACDS. This requires OAM to query DB2 to see if the LOB base
      table view exists for a given object storage group for each
      large object stored. If the LOB base table view does exist
      for a given object storage group, large objects are stored
      in the associated LOB storage structure. If the LOB base
      table view does not exist, large objects are stored in the
      32 KB data table.
      LOB=N specifies that objects that exceed 32 KB are to be
      stored in a 32 KB data table when stored to DB2. This is the
      default option.
```

Figure 4-2 Definition of the LOB parameter in PARMLIB

Figure 4-3 shows an example of an IEFSSNxx member customized to use LOB support for a partial list of the object storage groups defined to SMS.

```
EDIT       SYS1.PARMLIB(IEFSSNR7) - 01.76
Command ===>
000213 SUBSYS SUBNAME(OAM1)
000214   INITRTN(CBRINIT)
000215   INITPARM('MSG=EM,MOS=256,LOB=P')
```

Figure 4-3 Setting the LOB=x value in IEFSSNxx

4. Modify and run the CBRSMR18 migration job. This job performs the migration from the z/OS V1R7 version of the Object Storage Database to the z/OS V1R8 version that supports DB2 large objects. Running this job is required even if you do not intend on using the LOB function. The latest version of CBRSMR18 is found in SAMPLIB, and it has been updated by recent APARs.

5. Modify and run the CBRILOB sample job found in SAMPLIB. This job defines the VSAM ESDS that is used by DB2 to create the LOB storage structure and creates the LOB base table, base table view, auxiliary table, and index that comprise the LOB storage structure within the object storage table hierarchy. It should be customized to create the LOB structure for every storage group that you intend to have exploit large objects. If you specify LOB=A in IEFSSNxx, all groups need to have the LOB structure defined using this job. Figure 4-4 shows some tips for calculating primary and secondary space needs within this job if you do it manually.

```
Calculating the primary and secondary space needs for OSMLBTS, OTLOBX1, OSMLATS
and OTLOBAX1 in a customized CBRILOB job.
Each formula results in number of cylinders.
The result could be calculated as number of tracks if you divide by 49152
instead of 720K.
OSMLBTS:
 (Number of objects stored in this table * 78 ) / 720K
OTLOBX1:
 (Number of objects stored in this table * 50) / 720K
OSMLATS:
 (Number of objects stored in this table * average object size) / 720K
OTLOBAX1:
  (Number of objects stored in this table * 21 ) / 720K
```

Figure 4-4 Calculating primary and secondary space needs in CBRILOB

6. Modify and run the CBRPBIND sample job found in SAMPLIB. This job performs a DB2 bind for the packages needed to access the OAM storage object group, OAM administration, and OAM configuration tables. It has been updated by recent APARs and it now uses VALIDATE(RUN) instead of VALIDATE(BIND). The *DFSMS Using the New Functions,* SC26-7473, publication contains more information about when you might want to change this default to VALIDATE(BIND).
7. Verify the state of your OAM/DB2 environment and the functionality of the LOB support, as discussed in “Validating” on page 98.

4.1.2 Validating

After implementing the LOB function, you should verify the state of your OAM/DB2 environment. The OSREQ command and SMF analysis are two good methods to accomplish this task.

OSREQ

OSREQ is a TSO/E command processing tool that can be used to manipulate objects in your OAM/DB2 environment. Refer to *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, for more information about this tool. OSREQ requires that the DSNALI module, or DSNCLI module when OAM is running under CICS, be APF authorized and in the LINKLST concatenation or in the STEPLIBs of any jobs using OSREQ to store and access objects. If you attempt to use OSREQ and these prerequisites are not met, message CBR0401I is issued with a return code 16 and reason code D8010000. CBRSAMIV is a sample member in SAMPLIB that runs OSREQ in a batch job.

Figure 4-5 is an example of a customized CBRSAMIV job that stores a new 200 million byte object, lists it, queries it, retrieves it, and deletes it.

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//STEPLIB  DD   DSN=DB2M8.SDSNLOAD,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OSREQ STORE    OAMTEST.LOB200A OAMTEST.LOB200A LENGTH(200000000)
 LISTCAT ENTRIES('OAMTEST.LOB200A') ALL
 OSREQ QUERY    OAMTEST.LOB200A OAMTEST.LOB200A
 OSREQ RETRIEVE OAMTEST.LOB200A OAMTEST.LOB200A  COMPARE VIEW(PRIMARY)
 OSREQ DELETE   OAMTEST.LOB200A OAMTEST.LOB200A
/*
```

Figure 4-5 Customized JCL for the CBRSAMIV sample job

Running the customized CVRSAMIV sample job results in the output seen in Figure 4-6.

```
IEF375I  JOB/CBROAMIV/START 2007057.1654
IEF376I  JOB/CBROAMIV/STOP  2007057.1654 CPU    0MIN 01.89SEC SRB    0MIN
00.00S
READY
 OSREQ STORE    OAMTEST.LOB200A OAMTEST.LOB200A LENGTH(200000000)
OSREQ STORE successful. Return code = 00000004, reason code = 04020480.
OSREQ STORE response time is 6845 milliseconds.
OSREQ STORE data rate is 28533 kilobytes/second.
READY
 LISTCAT ENTRIES('OAMTEST.LOB200A') ALL
NONVSAM ------- OAMTEST.LOB200A
     IN-CAT --- UCAT.VSBOX01
     HISTORY
DATASET-OWNER-----(NULL)     CREATION--------2007.057
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----GROUP00
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS--OBJDASD
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
 OSREQ RETRIEVE OAMTEST.LOB200A OAMTEST.LOB200A  COMPARE VIEW(PRIMARY)
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 12 milliseconds.
OSREQ RETRIEVE successful. Return code = 00000000, reason code = 00000000.
OSREQ RETRIEVE response time is 1611 milliseconds.
OSREQ RETRIEVE data rate is 121236 kilobytes/second.
Data comparison for object OAMTEST.LOB200A OAMTEST.LOB200A successful.
READY
 OSREQ DELETE   OAMTEST.LOB200A OAMTEST.LOB200A
OSREQ DELETE successful. Return code = 00000000, reason code = 00000000.
OSREQ DELETE response time is 13 milliseconds.
READY
END
```

Figure 4-6 Output from running the CBRSAMIV sample job

The return and reason code from the STORE operation above indicates that in this case the catalog entry was created for this collection and that the storage class specified for the collection was overridden. The *DFSMSdfp Diagnosis*, GY27-7618, publication is very useful in determining what the OSREQ return codes mean.

SMF

SMF data should be analyzed in order to validate that the new LOB support and associated performance improvements are being used when running the CBRSAMIV sample job.

OAM writes SMF Record type 85 subtype 2/3/6 to document the OSREQ macros use of the LOB support.

We have written a simple program called SMF85TA to scan the SMF records and summarize activity. The program itself and how to construct it is documented in “SMF record type 85 subtype 1-7 data display program” on page 513

In Figure 4-7 we show the JCL to extract the SMF records and run the program.

If you do not want output from all the types that the program can process, change the SMFselection statement to only include those subtypes that you do want. For example, change OUTDD(OUTDD,TYPE(85(1,2,3,4,5,6,7))) to OUTDD(OUTDD,TYPE(85(2,3,4,5,6))) to only exclude subtypes 1 and 7.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT85I program. The JCL could be changed to make a permanent extract of the SMF data, or to read an already created SMF data extract.

```
//MHLRES1O JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//SMFEXTR EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=5096,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(85(1,2,3,4,5,6,7)))
/*
//  EXEC PGM=SMF85TA
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TA.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 4-7 SMF85TA program execution JCL

In Figure 4-8 on page 102 and Figure 4-9 on page 103 we show an example of output from the execution of program SMF85TA with all records selected in the SMFEXTR step. At the bottom in bold, you can see an OSREQ DELETE with a value of 81000000. Looking at the CBRSMF member of MACLIB, you can see that the 8 indicates “PRIMARY COPY OF OBJECT DELETED FROM DASD” and the 1 indicates “PRIMARY COPY OF THE OBJECT WAS DELETED FROM A LOB STORAGE STRUCTURE”.

Looking further up in the output you can see an OSREQ RETRIEVE with a value of 80040000. If we look at CBRSMF again, we can see that the 8 indicates “PRIMARY COPY OF OBJECT RETRIEVED FROM DASD” and the 4 indicates “PRIMARY COPY OF THE OBJECT WAS RETRIEVED FROM A LOB STORAGE STRUCTURE”. This same method is also used to translate values for output from OSREQ STORE. Another important thing about STORE is that ST2FLGS8 equals X'80' when an immediate backup copy is scheduled, as seen in the CBRSMF data pasted in Figure 4-12 on page 106. The SMF Type 85 Subtype 39 record is preferred, however, for analyzing Immediate Backup Copy results, as seen in “SMF” on page 112.

```
SMF TYPE 85 SUBTYPE 1-7 RECORDS
SMFDTE/TME:               2007061 14:14:01.013
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:01.020
STYPE:                    2 OSREQ STORE
COLN/OBJN:                OAMTEST.MARO2OA                              OAMTEST.OBJOO2OA
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000004 07241088 80C00000

SMFDTE/TME:               2007061 14:14:01.021
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:02.022
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:02.023
STYPE:                    4 OSREQ QUERY
COLN/OBJN:                OAMTEST.MARO2OA                              OAMTEST.OBJOO2OA
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:02.026
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:04.031
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:04.032
STYPE:                    5 OSREQ CHANGE
COLN/OBJN:                OAMTEST.MARO2OA                              OAMTEST.OBJOO2OA
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00                              00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 20000000


SMFDTE/TME:               2007061 14:14:04.033
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:05.034
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:05.035
STYPE:                    4 OSREQ QUERY
COLN/OBJN:                OAMTEST.MARO2OA                              OAMTEST.OBJOO2OA
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000


SMFDTE/TME:               2007061 14:14:05.038
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000
```

Figure 4-8 SMF85TA output - all subtypes part (1 of 2)

```
SMFDTE/TME:               2007061 14:14:06.039
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                          00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:06.040
STYPE:                    4 OSREQ QUERY
COLN/OBJN:                OAMTEST.MARO20A                            OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:06.041
STYPE:                    3 OSREQ RETRIEVE
COLN/OBJN:                OAMTEST.MARO20A                            OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00                    00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 80040000

SMFDTE/TME:               2007061 14:14:06.042
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                          00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:07.043
STYPE:                    1 OSREQ ACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                          00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 00000000

SMFDTE/TME:               2007061 14:14:07.044
STYPE:                    6 OSREQ DELETE
COLN/OBJN:                OAMTEST.MARO20A                            OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00                    00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:                 00000000 00000000 81000000

SMFDTE/TME:               2007061 14:14:07.045
STYPE:                    7 OSREQ UNACCESS
COLN/OBJN:
SGN/SCN/MCN/LEN/TTOK/TOK:                          00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
```

Figure 4-9 SMF85TA output - all subtypes part (2 of 2)

In Figure 4-10 we show example of output from the execution of program SMF85TA with records 1 and 7 not selected in the SMFEXTR step. Using this method to leave out the OSREQ ACCESS and UNACCESS data serves to reduce the amount of data that appears in your output, thereby simplifying things.

```
SMF TYPE 85 SUBTYPE 1-7 RECORDS
SMFDTE/TME:          2007061 14:14:01.020
STYPE:           2 OSREQ STORE
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000004 07241088 80C00000

SMFDTE/TME:          2007061 14:14:02.023
STYPE:           4 OSREQ QUERY
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 00000000

SMFDTE/TME:          2007061 14:14:04.032
STYPE:           5 OSREQ CHANGE
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00            00000000000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 20000000

SMFDTE/TME:          2007061 14:14:05.035
STYPE:           4 OSREQ QUERY
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 00000000

SMFDTE/TME:          2007061 14:14:06.040
STYPE:           4 OSREQ QUERY
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00  OBJDASD  OBJDASD  00000000001  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 00000000

SMFDTE/TME:          2007061 14:14:06.041
STYPE:           3 OSREQ RETRIEVE
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00            00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 80040000

SMFDTE/TME:          2007061 14:14:07.044
STYPE:           6 OSREQ DELETE
COLN/OBJN:           OAMTEST.MAR020A                  OAMTEST.OBJ0020A
SGN/SCN/MCN/LEN/TTOK/TOK: GROUP00            00000500000  00000000000000000000000000000000 D6E2D4C97F667A08
VSN/MT/RC/RS/FLGS:           00000000 00000000 81000000
```

Figure 4-10 SMF85TA output - excluding subtypes 1 and 7

In Figure 4-11 to Figure 4-15 on page 109 we show the SMF records that relate to this output. The Flags field, as shown in the output, reflects the flag bits as mapped by the flag fields in the SMF records corresponding to the particular subtype.

SMF record type 85 subtype 1 is used to map subtype records 1, 2, 3, 4, 5, 6, and 7.

Note: You should not use the contents of the CBRSMF macro as presented here. When you assemble the program, or wish to refer to the macro, it will be found in SYS1.MACLIB.

```
ST1      DSECT                    SUBTYPES 1 - 7
ST1COLN  DS    CL44' '            COLLECTION NAME
ST1OBJN  DS    CL44' '            OBJECT NAME
ST1SGN   DS    CL8' '             STORAGE GROUP NAME
ST1SCN   DS    CL8' '             STORAGE CLASS NAME
ST1MCN   DS    CL8' '             MANAGEMENT CLASS NAME
ST1OFF   DS    BL4'0'             OFFSET FOR PARTIAL OBJECT
*                                 RETRIEVE (SUBTYPE 3), ZERO FOR
*                                 ALL OTHERS.
ST1LEN   DS    BL4'0'             LENGTH,
*                                 SUBTYPE 1 - UNUSED
*                                 SUBTYPE 2 - LENGTH OF OBJECT STORED
*                                 SUBTYPE 3 - NUMBER OF BYTES RETRIEVED
*                                 SUBTYPE 4 - NUMBER OF QEL ELEMENTS
*                                             RETURNED.
*                                 SUBTYPE 5 - UNUSED
*                                 SUBTYPE 6 - LENGTH OF OBJECT DELETED
*                                 SUBTYPE 7 - UNUSED
ST1TTOK  DS    CL16' '            OSREQ TRACKING TOKEN, SUPPLIED
*                                 WITH TTOKEN KEYWORD ON OSREQ
*                                 MACRO
ST1TOK   DS    CL8' '             OSREQ ACCESS TOKEN
ST1VSN   DS    CL6' '             VOLUME SERIAL NUMBER
ST1VMT   DS    CL2' '             VOLUME MEDIA TYPE
ST1RC    DS    BL4'0'             OSREQ RETURN CODE, IN REGISTER 15
*                                 FOLLOWING OSREQ MACRO
ST1RS    DS    BL4'0'             OSREQ REASON CODE, IN REGISTER 15
*                                 FOLLOWING OSREQ MACRO
ST1FLGS  DS    BL4'0'             PROCESSING FLAGS. MEANING
*                                 DEPENDENT ON RECORD SUBTYPE.
```

Figure 4-11 SMF record type 85 subtype 1 significant fields (extract from CBRSMF macro) (1 of 5)

```
*********************************************************************
*                                                                   *
* SUBTYPE 2 - OSREQ STORE FLAGS                                     *
*                                                                   *
*********************************************************************
ST2FLGS0 EQU   X'80'             OBJECT STORE TO DASD
ST2FLGS1 EQU   X'40'             OBJECT STORE TO OPTICAL
ST2FLGS2 EQU   X'20'             OBJECT STORE TO TAPE
ST2FLGS3 EQU   X'10'             UNUSED
ST2FLGS4 EQU   X'08'             UNUSED
ST2FLGS5 EQU   X'04'             WHEN ON, THE OSREQ STORE
*                                REQUEST RESULTED IN THE MOUNTING
*                                OF A SHELF-RESIDENT REMOVABLE
*                                MEDIA VOLUME (TAPE OR OPTICAL)
*                                BY A HUMAN OPERATOR. ONLY VALID
*                                IF BIT 1 OR 2 IS ON.
ST2FLGS6 EQU   X'02'             WHEN ON, THE OSREQ STORE
*                                REQUEST RESULTED IN THE MOUNTING
*                                OF A LIBRARY-RESIDENT REMOVABLE
*                                MEDIA VOLUME (TAPE OR OPTICAL)
*                                BY A HUMAN OPERATOR. ONLY VALID
*                                IF BIT 1 OR 2 IS ON.
ST2FLGS7 EQU   X'01'             WHEN ON, THE OSREQ STORE
*                                REQUEST WAS SATISIFIED USING
*                                AN ALREADY MOUNTED REMOVEABLE
*                                MEDIA VOLUME (TAPE OR OPTICAL).
*                                ONLY VALID
*                                IF BIT 1 OR 2 IS ON.
ST2FLGS8 EQU    X'80'            WHEN ON, AN IMMEDIATE BACKUP COPY
*                                WAS SCHEDULED FOR THIS OBJECT.  @L5A
ST2FLGS9 EQU    X'40'            WHEN ON, THE OBJECT IS STORED
*                                TO LOB STORAGE STRUCTURE        @L5A
```

Figure 4-12 SMF record type 85 subtype 1 significant fields (extract from CBRSMF macro) (2 of 5)

```
*********************************************************************
*                                                                   *
* SUBTYPE 3 - OSREQ RETRIEVE FLAGS                                  *
*                                                                   *
*********************************************************************
ST3FLGS0 EQU    X'80'           WHEN ON, PRIMARY COPY OF OBJECT
*                               RETRIEVED FROM DASD.
ST3FLGS1 EQU    X'40'           WHEN ON, PRIMARY COPY OF OBJECT
*                               RETRIEVED FROM OPTICAL.
ST3FLGS2 EQU    X'20'           WHEN ON, PRIMARY COPY OF OBJECT
*                               RETRIEVED FROM TAPE.
ST3FLGS3 EQU    X'10'           WHEN ON, EITHER THE FIRST OR THE
*                               SECOND BACKUP COPY OF THE OBJECT WAS
*                               RETRIEVED FROM OPTICAL AS RESULT OF
*                               VIEW=BACKUP OR VIEW=BACKUP2 BEING
*                               SPECIFIED ON THE OSREQ MACRO.
*                               REFER TO BIT 10 TO INDICATE WHICH
*                               BACKUP COPY WAS RETRIEVED.     @L2C
ST3FLGS4 EQU    X'08'           WHEN ON, EITHER THE FIRST OR THE
*                               SECOND BACKUP COPY OF THE OBJECT WAS
*                               RETRIEVED FROM TAPE AS RESULT OF
*                               VIEW=BACKUP OR VIEW=BACKUP2 BEING
*                               SPECIFIED ON THE OSREQ MACRO.
*                               REFER TO BIT 10 TO INDICATE WHICH
*                               BACKUP COPY WAS RETRIEVED.     @P1C
ST3FLGS5 EQU    X'04'           WHEN ON, EITHER THE FIRST OR THE
*                               SECOND BACKUP COPY OF THE OBJECT WAS
*                               RETRIEVED FROM OPTICAL AS A RESULT
*                               OF THE PRIMARY COPY OF THE OBJECT
*                               RESIDING ON AN UNREADABLE OPTICAL
*                               DISK VOLUME AND THE AUTOMATIC ACCESS
*                               TO BACKUP WAS ACTIVE.
*                               REFER TO BIT 10 TO INDICATE WHICH
*                               BACKUP COPY WAS RETRIEVED.     @P1C
ST3FLGS6 EQU    X'02'           WHEN ON, EITHER THE FIRST OR THE
*                               SECOND BACKUP COPY OF THE OBJECT WAS
*                               RETRIEVED FROM TAPE AS A RESULT
*                               OF THE PRIMARY COPY OF THE OBJECT
*                               RESIDING ON AN UNREADABLE OPTICAL
*                               DISK VOLUME AND THE AUTOMATIC ACCESS
*                               TO BACKUP WAS ACTIVE.
*                               REFER TO BIT 10 TO INDICATE WHICH
*                               BACKUP COPY WAS RETRIEVED.     @P1C
ST3FLGS7 EQU    X'01'           WHEN ON, THE OSREQ RETRIEVE
*                               REQUEST RESULTED IN THE MOUNTING
*                               OF A SHELF-RESIDENT REMOVABLE
*                               MEDIA VOLUME (TAPE OR OPTICAL)
*                               BY A HUMAN OPERATOR. ONLY VALID
*                               IF BIT 1, 2, 3, 5 OR 6 IS ON.
```

Figure 4-13 SMF record type 85 subtype 1 significant fields (extract from CBRSMF macro) (3 of 5)

```
ST3FLGS8 EQU    X'80'           WHEN ON, THE OSREQ RETRIEVE
*                               REQUEST RESULTED IN THE MOUNTING
*                               OF A LIBRARY-RESIDENT REMOVABLE
*                               MEDIA VOLUME (TAPE OR OPTICAL)
*                               BY A HUMAN OPERATOR. ONLY VALID
*                               IF BIT 1, 2, 3, 5 OR 6 IS ON.
ST3FLGS9 EQU    X'40'           WHEN ON, THE OSREQ RETRIEVE
*                               REQUEST WAS SATISIFIED USING
*                               AN ALREADY MOUNTED REMOVEABLE
*                               MEDIA VOLUME (TAPE OR OPTICAL).
*                               ONLY VALID
*                               IF BIT 1, 2, 3, 5 OR 6 IS ON.
ST3FLGS10 EQU    X'20'          WHEN ON, THE SECOND BACKUP COPY OF
*                               THE OBJECT WAS RETRIEVED         @L4C
ST3FLGS11 EQU    X'10'          WHEN ON, A RECALL WAS SCHEDULED
*                               FOR THIS OBJECT                  @L4A
ST3FLGS12 EQU    X'08'          WHEN ON, A RECALL WAS EXPLICITYLY
*                               SPECIFIED ON THE OSREQ RETRIEVE
*                               REQUEST                          @L4A
ST3FLGS13 EQU    X'04'          WHEN ON, THE PRIMARY COPY OF
*                               THE OBJECT WAS RETRIEVED FROM
*                               A LOB STORAGE STRUCTURE          @L5A
*
*********************************************************************
*                                                                   *
* SUBTYPE 4 - OSREQ QUERY FLAGS                                     *
*                                                                   *
*********************************************************************
*********************************************************************
*                                                                   *
* SUBTYPE 5 - OSREQ CHANGE FLAGS                                    *
*                                                                   *
*********************************************************************
ST5FLG0  EQU    X'80'           WHEN ON, MANAGEMENT CLASS
*                               SPECIFIED ON OSREQ CHANGE.
ST5FLG1  EQU    X'40'           WHEN ON, STORAGE CLASS
*                               SPECIFIED ON OSREQ CHANGE.
ST5FLG2  EQU    X'20'           WHEN ON, RETENTION PERIOD
*                               SPECIFIED ON OSREQ CHANGE.
```

Figure 4-14 SMF record type 85 subtype 1 significant fields (extract from CBRSMF macro) (4 of 5)

```
*********************************************************************
*                                                                   *
* SUBTYPE 6 - OSREQ DELETE FLAGS                                    *
*                                                                   *
*********************************************************************
ST6FLG0  EQU    X'80'           WHEN ON, PRIMARY COPY OF OBJECT
*                               DELETED FROM DASD.
ST6FLG1  EQU    X'40'           WHEN ON, PRIMARY COPY OF OBJECT
*                               DELETED FROM OPTICAL.
ST6FLG2  EQU    X'20'           WHEN ON, PRIMARY COPY OF OBJECT
*                               DELETED FROM TAPE.
ST6FLG3  EQU    X'10'           WHEN ON, BACKUP COPY OF OBJECT
*                               DELETED FROM OPTICAL.
ST6FLG4  EQU    X'08'           WHEN ON, BACKUP COPY OF OBJECT
*                               DELETED FROM TAPE.
ST6FLG5  EQU    X'04'           WHEN ON, 2ND BACKUP COPY OF OBJECT
*                               DELETED FROM OPTICAL.          @L2A
ST6FLG6  EQU    X'02'           WHEN ON, 2ND BACKUP COPY OF OBJECT
*                               DELETED FROM TAPE.             @L2A
ST6FLG7  EQU    X'01'           WHEN ON, THE PRIMARY OCPY OF
*                               THE OBJECT WAS DELETED FROM
*                               A LOB STORAGE STRUCTURE        @L5A
*********************************************************************
*                                                                   *
* SUBTYPE 7 - OSREQ UNACCESS FLAGS                                  *
*                                                                   *
*********************************************************************
```

Figure 4-15 SMF record type 85 subtype 1 significant fields (extract from CBRSMF macro) (5 of 5)

Migration and coexistence

Migration considerations include running the CBRILOB, CBRSMR18, and CBRPBIND jobs as described in 4.1.1, “Implementing” on page 96. Running CBRSMR18 is required even if you do not intend to exploit large object support. Pre-V1R8 systems in the same OAMplex as a V1R8 system must have the fix for APAR OA12683 applied, regardless of whether the V1R8 system is exploiting large object support. DB2 Version 6.1 or later is required before OAM can use DB2’s LOB support.

Maintenance

The fix for APAR OA16562 must be applied prior to running the CBRILOB and CBRSMR18 jobs associated with LOB implementation. This APAR updates those sample jobs.

4.2 Immediate backup copy

You now have the ability to immediately back up your objects when they first get stored. Prior to this support, objects would not get backed up until the object storage management cycle (OSMC) was run, which would often times be several hours later.

4.2.1 Implementing

Two changes must be made in the management class definition for the objects that you want to exploit this function. The BACKUP FREQUENCY field must be set to 0 and the AUTO BACKUP field must be set to Y, as seen in Figure 4-16. When these two fields are updated, an immediate backup copy will automatically be used.

```
MANAGEMENT CLASS ALTER             Page 3 of 5
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Management Class Name : OBJDASD

To ALTER Management Class, Specify:
  Backup Attributes
    Backup Frequency  . . . . . . . . 0       (0 to 9999 or blank)
    Number of Backup Vers . . . . . . 1       (1 to 100 or blank)
      (Data Set Exists)
    Number of Backup Vers . . . . . . 1       (0 to 100 or blank)
      (Data Set Deleted)
    Retain days only Backup Ver . . . 60      (1 to 9999, NOLIMIT or blank)
      (Data Set Deleted)
    Retain days extra Backup Vers . . 30      (1 to 9999, NOLIMIT or blank)
    Admin or User command Backup  . . BOTH    (BOTH, ADMIN or NONE)
    Auto Backup . . . . . . . . . . . Y       (Y or N)
    Backup Copy Technique . . . . . . P       (P=Conc Preferred, R=Conc
                                              Required or S=Standard)
Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
```

Figure 4-16 Altering an existing management class in ISMF for immediate backup

4.2.2 Validating

After you have implemented immediate backup copy, there are several methods that you can use to validate its function.

RETCODE2

As described in the binary large object support section, OSREQ can be used to store objects. A new RETCODE2 parameter can be coded when using the OSREQ store function, and indicates whether the store request successfully scheduled an immediate backup copy for the object. The RETCODE2 output is only valid when the STORE is successful, in which case one of the following return codes is provided:

- 0: Immediate backup copy request successfully scheduled.
- 4: Immediate backup copy request not required.
- 8: An attempt to schedule an immediate backup for this object was not successful because OSMC is not up and running.
- 14: An attempt to schedule an immediate backup for this object was not successful due to an unexpected scheduling error.

Figure 4-17 provides an example JCL showing an OSREQ STORE with the RETCODE2 parameter.

```
//CBROAMIV   JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),REGION=0M
/*JOBPARM SYSAFF=SC64
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//STEPLIB  DD   DSN=DB2M8.SDSNLOAD,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OSREQ STORE    OAMTEST.LOB200C OAMTEST.LOB200C LENGTH(2000) RETCODE2
```

Figure 4-17 Example JCL for OSREQ STORE with RETCODE2 parameter

JOBLOG output from running this sample JCL can be seen in Figure 4-18. This output indicates that the immediate backup was scheduled successfully.

```
 IEF375I  JOB/CBROAMIV/START 2007057.1950
 IEF376I  JOB/CBROAMIV/STOP  2007057.1950 CPU    0MIN 00.05SEC SRB    0MIN
00.00
1READY
  OSREQ STORE    OAMTEST.LOB200C OAMTEST.LOB200C LENGTH(2000) RETCODE2
 OSREQ STORE successful. Return code = 00000004, reason code = 04020480,
retcode2 = 00000000.
 OSREQ STORE response time is 181 milliseconds.
 OSREQ STORE data rate is 10 kilobytes/second.
 READY
```

Figure 4-18 Example of an OSREQ STORE with successful immediate backup

SYSLOG

You can look at SYSLOG immediately after an OSREQ STORE is run in order to see if the immediate backup copy is performed. Figure 4-19 shows an example of the mount seen when backing up to tape.

```
$HASP373 CBROAMIV STARTED - INIT 1    - CLASS A - SYS SC64
IEF403I CBROAMIV - STARTED - TIME=17.13.54 - ASID=002A - SC64
+IGD01009I MC ACS GETS CONTROL &ACSENVIR=STORE
+IGD01010I SG ACS GETS CONTROL &ACSENVIR=STORE
IEC501A M 0B91,TST006,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
-                                          --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS STEPNO
-CBROAMIV          STEP1       00    379    .03    .00    .19   537K
 0      0      0      0     0     1
IEF404I CBROAMIV - ENDED - TIME=17.14.06 - ASID=002A - SC64
-CBROAMIV ENDED.  NAME-                     TOTAL CPU TIME=   .03
TOTAL ELAPSED TIME=   .19
$HASP395 CBROAMIV ENDED
$HASP309 INIT 1    INACTIVE ******** C=ABCDE
IEC205I CBRRT001,OAM,OAM,FILESEQ=1, COMPLETE VOLUME LIST, 241
DSN=OAM.BACKUP.DATA,VOLS=TST006,TOTALBLOCKS=1228
IEF234E K 0B91,TST006,PVT,OAM,OAM
```

Figure 4-19 SYSLOG indicating OAM is backing up to tape after OSREQ STORE

SMF

OAM writes SMF record type 85 subtype 39 to document the use of the immediate backup facility. It also uses subtype 2 to document when an immediate backup copy has been scheduled as a result of running the OSREQ macro. See the SMF section for large object support for more information about immediate backup and subtype 2.

We have written a simple program called SMF85TI to scan the SMF records and summarize activity. The program itself and how to construct it is documented in “SMF record type 85 subtype 39 data display program” on page 490.

In Figure 4-40 on page 130 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT85I program. The JCL could be changed to make a permanent extract of the SMF data, or to read of an already created SMF data extract.

```
//MHLRES1O JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(1O,5)),
// RECFM=VB,LRECL=5O96,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(85(39)))
/*
//  EXEC PGM=SMF85TO
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TI.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 4-20 SMF85TI program execution JCL

In Figure 4-21 we show an example of output from the execution of program SMF85TI.

```
SMF TYPE 85 SUBTYPE 39 RECORDS
COLN/CNID:        OAMTEST.LOB2000                    00000000035
OBJN/SGN/OLEN:     OAMTEST.LOB2000                  GROUP00 00000002000
VSN/MT/TKN/FLGS:    TST008   00000000004      82800000

COLN/CNID:        OAMTEST.LOB200P                    00000000036
OBJN/SGN/OLEN:     OAMTEST.LOB200P                  GROUP00 00020000000
VSN/MT/TKN/FLGS:    TST003   00000000008      82800000

COLN/CNID:        OAMTEST.LOB200Q                    00000000037
OBJN/SGN/OLEN:     OAMTEST.LOB200Q                  GROUP00 00020000000
VSN/MT/TKN/FLGS:    TST003   00000000622      82800000

COLN/CNID:        OAMTEST.LOB200R                    00000000038
OBJN/SGN/OLEN:     OAMTEST.LOB200R                  GROUP00 00020000000
VSN/MT/TKN/FLGS:    TST004   00000000004      82800000

COLN/CNID:        OAMTEST.LOB200S                    00000000039
OBJN/SGN/OLEN:     OAMTEST.LOB200S                  GROUP00 00020000000
VSN/MT/TKN/FLGS:          01077952576      80000000
```

Figure 4-21 SMF85TI output

The left-hand side of the report contains abbreviations from the SMF records. Record type 85 subtype 39 contains the fields of interest to verify that the function of OAM immediate backup is occurring. In the SMF records, as shown in Figure 4-22 on page 115, the fields are prefixed with ST39. In the report the fields are shown without the ST39, and have been compressed to have more than one entry per line.

The fourth example shows a successful operation involving object OAMTEST.LOB200R and tape volume TST004. The Flgs field shows 8280000, which (as per the SMF record extract below) indicates that the primary copy is stored to DASD and the backup is on tape, and that the write to tape was successful. The VSN field is also filled in with TST004.

The fifth example shows an unsuccessful operation involving object OAMTEST.LOB200S. The flgs field shows 8000000, which indicates that the primary copy is on DASD, but there is no indication of a successful backup.

In Figure 4-22 on page 115 we show the SMF records that relate to this output.

Note: You should not use the contents of the CBRSMF macro as presented here. When you assemble the program, or wish to refer to the macro, it will be found in SYS1.MACLIB.

```
ST39     DSECT                  SUBTYPE 39                          @L5A
ST39COLN DS     CL44' '         COLLECTION NAME                     @L5A
ST39CNID DS     BL4'0'          COLLECTION ID                       @L5A
ST39OBJN DS     CL44' '         OBJECT NAME                         @L5A
ST39SGN  DS     CL8' '          STORAGE GROUP NAME                  @L5A
ST39MCN  DS     CL8' '          MANAGEMENT CLASS NAME               @L5A
ST39OLEN DS     BL4'0'          OBJECT LENGTH                       @L5A
ST39SVSN DS     CL6' '          SOURCE VOLUME SERIAL NUMBER OF
*                                OPTICAL VOLUME OR TAPE VOLUME
*                                FROM WHICH THE PRIMARY OBJECT
*                                WAS READ.
*                                ONLY VALID IF THE BIT 1 OR 2
*                                IS ON IN FIELD ST39FLGS             @L5A
ST39SMT  DS     CL2' '          SOURCE MEDIA TYPE OF THE VOLUME
*                                FROM WHICH THE PRIMARY OBJECT
*                                WAS READ.
*                               ONLY VALID IF THE BIT 1 OR 2
*                                IS ON IN FIELD ST39FLGS             @L5A
ST39TVSN DS     CL6' '          TARGET VOLUME SERIAL NUMBER OF
*                                OPTICAL VOLUME OR TAPE VOLUME
*                                ON WHICH THE BACKUP COPY OF
*                                THE OBJECT WAS WRITTEN.
*                                ONLY VALID IF THE BIT 1 OR 2
*                                IS ON IN FIELD ST39FLGS             @L5A
ST39TMT  DS     CL2' '          TARGET MEDIA TYPE OF THE VOLUME
*                                ON WHICH THE BACKUP COPY OF
*                                THE OBJECT WAS WRITTEN.
*                                ONLY VALID IF THE BIT 1 OR 2
*                                IS ON IN FIELD ST39FLGS             @L5A
ST39BTKN DS     BL4'0'         VOLUME LOCATION TOKEN ASSOCIATED
*                                WITH THE COPY OF THE OBJECT
*                                ON THE VOLUME SPECIFIED IN THE
*                                ST39TVSN FIELD.                     @L5A
ST39FLGS DS     BL4'0'         PROCESSING FLAGS                    @L5A
ST39FLG0 EQU    X'80'          WHEN ON, THE PRIMARY COPY
*                                IS STORED TO DASD.                  @L5A
ST39FLG1 EQU    X'40'          WHEN ON, THE PRIMARY COPY
*                                IS STORED TO OPTICAL.               @L5A
ST39FLG2 EQU    X'20'          WHEN ON, THE PRIMARY COPY
*                                IS STORED TO TAPE.                  @L5A
ST39FLG3 EQU    X'10'          RESERVED                            @L5A
ST39FLG4 EQU    X'08'          RESERVED                            @L5A
ST39FLG5 EQU    X'04'          WHEN ON, THE BACKUP COPY
*                                IS STORED TO OPTICAL.               @L5A
ST39FLG6 EQU    X'02'          WHEN ON, THE BACKUP COPY
*                                IS STORED TO TAPE.                  @L5A
ST39FLG7 EQU    X'01'          RESERVED                            @L5A
ST39FLG8 EQU    X'80'          WHEN ON, WRITE TO BACKUP COPY
*                                WAS SUCCESSFUL.                     @L5A
```

Figure 4-22 SMF record type 85 subtype 39 significant fields (extract from CBRSMF macro)

Migration and coexistence

Pre-V1R8 systems in the same OAMplex as a V1R8 system must have the fix for APAR OA12683 applied.

Maintenance

The fix for APAR OA18519 must be applied or immediate backup copy could fail if OSREQ STORE is consuming all of the available DB2 threads.

4.3 Automated selection of RECYCLE volumes

OAM now provides a method to automatically recycle volumes based on user-defined criteria. It is intended for use in recycling full tape volumes associated with object or object backup storage groups. Optical volumes do not use this function and continue to be recycled individually with the MODIFY,OAM,START,MOVEVOL command.

4.3.1 Implementing

The MODIFY OAM,START command has been enhanced with a RECYCLE parameter. This command is used to start the RECYCLE based on criteria specified in the command and in PARMLIB. As seen in the *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, publication, the syntax of the command is as shown in Example 4-1.

Example 4-1 Syntax of the MODIFY OAM,START,RECYCLE command

```
MODIFY OAM,START,RECYCLE,scope{,PV=xxx}[,LIM=yy|,DISPLAY]
    Displays a list of candidate volumes that meet user-defined criteria
    to the hardcopy log, and in turn, automatically selects recycle
    candidate volumes and initiates the MOVEVOL with RECYCLE process on
    those candidates until either the user-specified limit is reached or
    no more volumes meeting criteria are available.

    scope
        Indicates one of the following:

      Specified name of an object or object backup storage group,
      indicates that only tape volumes marked full, that belong to
      the specified object or object backup storage group, are
      considered candidates for this RECYCLE command.

      ALLGRP all full tape volumes that belong to all primary object
      storage groups defined in the ACTIVE SCDS are considered candidates
      for this RECYCLE command.

      ALLBK1 all full tape volumes that belong to all first backup
      storage groups defined in the ACTIVE SCDS are considered
      candidates for this RECYCLE command.

      ALLBK2 all full tape volumes that belong to all second backup
      storage groups defined in the ACTIVE SCDS are considered
      candidates for this RECYCLE command.

    PV
```

```
    =nnn An optional keyword indicating the valid data threshold to be
    used in determining whether a volume is a candidate for RECYCLE.
    Full tape volumes that have a percentage of valid data less than
    or equal to nnn are candidates for RECYCLE. If PV=nnn is not
    specified, the percent valid to be used to determine RECYCLE
    candidates is derived from the PERCENTVALID default value as
    defined through the SETOAM command in the CBROAMxx PARMLIB member.
    Valid values for nnn are 0 to 100.

DISPLAY
    An optional parameter that produces a list of volumes that meet
    criteria to be recycle candidates. This list is sorted by the
    percentage of valid data on each volume and is written to hardcopy
    system log through the CBR9875I message. This option does not
    initiate Recycle processing, and can be issued at anytime, whether
    a RECYCLE command is actively processing or not. The list of
    candidate volumes might be large as it shows all volumes that meet
    the user-specified criteria for RECYCLE.
    If DISPLAY is not specified then LIM=yy must be specified.

LIM
    =yy If the DISPLAY parameter is not specified, this keyword is
    required to indicate the maximum number of volumes to be selected
    for RECYCLE processing. Valid values for yy are 1 to 40.
    If LIM=yy is not specified, then DISPLAY must be specified.
```

New keywords have been added for the SETOAM statement in the CBROAMxx member of PARMLIB. They are:

- MAXRECYCLETASKS
- SGMAXRECYCLETASKS
- PERCENTVALID

Figure 4-23 shows the definitions for these new keywords from the *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, publication.

```
MAXRECYCLETASKS(nn)
      Can be specified at the global level. The nn is the maximum number of
      MOVEVOL tasks that can be run concurrently by the RECYCLE function.
      Valid values for nn are 0 - 15. The default is 1 if no value is
      specified. A value of 0 indicates that no RECYCLE operations can be
      run at the storage group or global level.
SGMAXRECYCLETASKS(nn)
      An optional parameter that you can specify at the storage group
      level. The nn is the maximum number of MOVEVOL tasks that can be
      run concurrently by the RECYCLE function for a storage group. The
      value for SGMAXRECYCLETASKS cannot exceed the value for
      MAXRECYCLETASKS. Valid values for nn are 0 - 15. The default is 1
      if no value is specified. A value of 0 indicates that no RECYCLE
      operations can be run at the storage group level specified.

      If you only want to recycle volumes from one group, the setting
      for all other groups would be 0 to ensure that the group with a
      non-zero value receives all the recycling processing.

      If you specify a value for a group that is higher than the value
      for another group, the system selects more of the volumes to
      recycle from the group with the higher value. However, RECYCLE
      processing might be working with the original order of volumes
      that are sorted by the amount of valid data for each volume, and
      might select volumes from other groups to satisfy the limit before
      it processes the higher-value group.
PERCENTVALID(nnn)
      You can only specify this keyword at the global (all storage groups)
      level. nnn represents the global default percentage of valid data
      threshold that is used to determine whether a full tape volume is a
      candidate for RECYCLE processing. This SETOAM value is used only if
      the optional PV= keyword is not specified on the RECYCLE command. The
      PERCENTVALID value that is specified on the RECYCLE command takes
      precedence over the PERCENTVALID value in the SETOAM statement. Valid
      values for nnn are 0 - 100.  The default is 0 if no value is
      specified.
```

Figure 4-23 The new SETOAM statements for CBROAMxx in PARMLIB

An example of an updated CBROAMxx PARMLIB member can be seen in Figure 4-24.

```
 SYS1.PARMLIB(CBROAM00) - 01.44
 ===>
***************************** Top of Data ********
SETOAM TAPEDISPATCHERDELAY(45)
       TAPERECYCLEMODE(MVSSCRATCH)
       STORAGEGROUP(GROUP00 TAPEUNITNAME(3590-1))
       STORAGEGROUP(OBJBKP TAPEUNITNAME(3590-1))
       MAXRECYCLETASKS(2)
       STORAGEGROUP(GROUP00 SGMAXRECYCLETASKS(2))
       PERCENTVALID(10)
SETOSMC MAXRECALLTASKS(2)
        RECALLALL(15)
        CLEAROLDLOC(TAPE)
        RECALLTAPE(0)
        FIRSTBACKUPGROUP(OBJBKP)
        STORAGEGROUP(GROUP00 RECALLOFF(ON)
                     FIRSTBACKUPGROUP(OBJBKP))
```

Figure 4-24 Example CBROAMxx PARMLIB member updated for RECYCLE

4.3.2 Validating

In this section we discuss methods to validate and use the new function.

START RECYCLE

The DISPLAY option can be used to display RECYCLE candidates without actually performing the MOVEVOL with RECYCLE function. Figure 4-25 is an example of a RECYCLE command with the DISPLAY parameter for all first backup storage groups. You can also see that we are overriding the PERCENTVALID value that we have in PARMLIB.

```
MODIFY OAM,START,RECYCLE,(ALLBK1),DISPLAY,PV=50
CBR9880I OAM START RECYCLE command starting.
CBR9875I Recycle Candidates: 720
The following volumes are candidates for OAM RECYCLE command
processing using pv=50, lim=N/A, scope=(ALLBK1), maxrecycletasks=2.
  VOLSER  %VAL  SGNAME   STAT     VOLSER  %VAL  SGNAME   STAT
  TST016    16  OBJBKP
CBR9879I OAM Recycle: End of OAM Recycle candidate volumes.
CBR9881I OAM START RECYCLE command ending successfully. Reason is
display specified.
```

Figure 4-25 Showing RECYCLE candidates with the DISPLAY option

Figure 4-26 is an example of an actual RECYCLE taking place. In this case we again override the PERCENTVALID that we previously specified in PARMLIB because the PERCENTVALID value of the candidate volume is greater. We also setting LIM=2, although we could have set LIM=1 because we have only one candidate volume. The LIM= keyword is required when the DISPLAY parameter is not specified.

```
MODIFY OAM,START,RECYCLE,(ALLBK1),LIM=2,PV=50
CBR9880I OAM START RECYCLE command starting.
CBR9875I Recycle Candidates: 725
The following volumes are candidates for OAM RECYCLE command
processing using pv=50, lim=2, scope=(ALLBK1), maxrecycletasks=2.
  VOLSER  %VAL  SGNAME   STAT     VOLSER  %VAL  SGNAME   STAT
  TST016    16  OBJBKP
CBR9879I OAM Recycle: End of OAM Recycle candidate volumes.
CBR9800I OAM Move Volume Recycle starting for volumes TST016 and N/A.
CBR9852I Move Volume Utility processing objects in storage group
GROUP00 for volume TST016.
IEC501A M 0B91,TST016,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
IEC205I CBRRT001,OAM,OAM,FILESEQ=1, COMPLETE VOLUME LIST, 730
DSN=OAM.BACKUP.DATA,VOLS=TST017,TOTALBLOCKS=1
IEF234E K 0B90,TST017,PVT,OAM,OAM
IEC501A M 0B90,TST017,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
CBR9858I Move Volume Utility status for volume TST016. Total: 6,
Attempted: 6, Successful: 6, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volumes TST016 and N/A.
CBR9881I OAM START RECYCLE command ending successfully. Reason is no
more volumes available.
IEC205I CBRRT001,OAM,OAM,FILESEQ=1, COMPLETE VOLUME LIST, 760
DSN=OAM.BACKUP.DATA,VOLS=TST017,TOTALBLOCKS=1233
IEF234E K 0B91,TST016,PVT,OAM,OAM
CBR2164I Tape volume TST016 has had all objects expired or deleted
and has been returned to OAM scratch status.
IEF234E K 0B90,TST017,PVT,OAM,OAM
```

Figure 4-26 Issuing the RECYCLE for first backup storage group tape volumes

SMF

OAM writes SMF record type 85 subtype 40 to document the use of the RECYCLE facility.

We have written a simple program called SMF85TJ to scan the SMF records and summarize activity. The program itself and how to construct it is documented in “SMF record type 85 subtype 40 data display program” on page 496.

In Figure 4-27 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT85I program. The JCL could be changed to make a permanent extract of the SMF data, or to read an already created SMF data extract.

```
//MHLRES1O JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN2
//*DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS.G3196VOO
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=5096,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(85(40)))
/*
//  EXEC PGM=SMF85TJ
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TJ.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 4-27 SMF85TJ program execution JCL

In Figure 4-28 we show an example of output from the execution of program SMF85TI. This example shows that volumes TST003 and TST004 were successfully processed by the OSMC command migration facility.

```
SMF TYPE 85 SUBTYPE 40 RECORDS
STRD/ENDD/VOLN/PCTV/LIM: 2007-02-28 2007-02-28 00000000002  00000000010  00000000003
VSN:TST003
VSN:TST004
```

Figure 4-28 SMF85TJ output

In Figure 4-29 we show the SMF records that relate to this output.

Note: You should not use the contents of the CBRSMF macro as presented here. When you assemble the program, or wish to refer to the macro, it will be found in SYS1.MACLIB.

```
ST40     DSECT                   SUBTYPE 40
ST40STRD DS     CL10' '          DATE RECYCLE COMMAND STARTED     @L5A
ST40ENDD DS     CL10' '          DATE RECYCLE COMMAND ENDED       @L5A
ST40VOLN DS     BL2'0'           NUMBER OF VOLSERS COMPLETED      @L5A
ST40PCTV DS     BL2'0'           PERCENT VALID USED FOR COMMAND   @L5A
ST40LIM  DS     BL2'0'           LIMIT USED FOR COMMAND           @L5A
         DS     BL2'0'           RESERVED                         @L5A
ST40END  DS     0C               END OF BASE SECTION              @L5A
*********************************************************************
* SUBTYPE 40 VOLUME ARRAY SECTION                                   *
*********************************************************************
ST40VOLD DSECT                   ARRAY OF VOLS COMPLETED RECYCLE @L5A
ST40VSN  DS     4OCL6' '         VOLSER                          @L5A
```

Figure 4-29 SMF record type 85 subtype 40 significant fields (extract from CBRSMF macro)

More SMF

OAM also writes SMF record type 85 subtype 32/33/34/35 to document the use of this facility. For MOVEVOL (subtype 35), bit 0 in ST32FLGS is set to ON to indicate that the MOVEVOL was invoked automatically under software control as the result of a RECYCLE.

We have written a simple program called SMF85TH to scan the SMF records and summarize activity. The program itself and how to construct it are documented in “SMF record type 85 subtype 32-35 data display program” on page 502.

In Figure 4-30 we show the JCL to extract the SMF records and run the program.

```
//MHLRES1O JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//SMFEXTR EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(1O,5)),
// RECFM=VB,LRECL=5096,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
          INDD(DUMPIN,OPTIONS(DUMP))
          OUTDD(OUTDD,TYPE(85(32,33,34,35)))
/*
//  EXEC PGM=SMF85TH
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TH.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 4-30 SMF85TH program execution JCL

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the sSMFT85I program. The JCL could be changed to make a permanent extract of the SMF data, or to read of an already created SMF data extract.

If you do not want output from all the types that the program can process, change the SMF selection statement to only include those subtypes that you do want. For example, change OUTDD(OUTDD,TYPE(85(32,33,34,35))) to OUTDD(OUTDD,TYPE(85(35))) to only select subtype 35.

In Figure 4-31 we show an example of output from the execution of program SMF85TH. You can see that a RECYCLE command was processed for volume TST003 at 11:44 on day 2007059.

```
SMF TYPE 85 SUBTYPE 32-35 RECORDS
SMFDTE/TME:                          2007059 09:01:52.000
STYPE/SGN/VSNO/VSN1/MT:              32 GROUPOO                  (STORAGE GROUP PROCESSING)
PDWO/PDWK/PDRO/PDRK/PDDO/PDDK:       00000000000  00000000000  00000000004  00000078128  00000000004  00000078128
POWO/POWK/PORO/PORD/PODO/PODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
PTWO/PTWK/PTRO/PTRD/PTDO/PTDK:       00000000004  00000078128  00000000000  00000000000  00000000000  00000000000
BOWO/BOWK/BORO/BORK/BODO/BODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
BTWO/BTWK/BTRO/BTRK/BTDO/BTDK:       00000000001  00000019532  00000000000  00000000000  00000000000  00000000000
B2OWO/B2OWK/B2ORO/B2ORK/B2ODO/B2ODK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
B2TWO/B2TWK/B2TRO/B2TRK/B2TDO/B2TDK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
DTUP/DTDE/4KIN/4KDE/32KI/32KD/NCE:   00000000017  00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
FLGS/NTE/RCLD/RCLK/LOBI/LOBD:        81000000     00000000000  00000000000  00000000000  00000000000  00000000004

SMFDTE/TME:                          2007059 10:00:00.047
STYPE/SGN/VSNO/VSN1/MT:              32 OBJBKP                   (STORAGE GROUP PROCESSING)
PDWO/PDWK/PDRO/PDRK/PDDO/PDDK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
POWO/POWK/PORO/PORD/PODO/PODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
PTWO/PTWK/PTRO/PTRD/PTDO/PTDK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
BOWO/BOWK/BORO/BORK/BODO/BODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
BTWO/BTWK/BTRO/BTRK/BTDO/BTDK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
B2OWO/B2OWK/B2ORO/B2ORK/B2ODO/B2ODK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
B2TWO/B2TWK/B2TRO/B2TRK/B2TDO/B2TDK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
DTUP/DTDE/4KIN/4KDE/32KI/32KD/NCE:   00000000000  00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
FLGS/NTE/RCLD/RCLK/LOBI/LOBD:        80800000     00000000000  00000000000  00000000000  00000000000  00000000000

SMFDTE/TME:                          2007059 11:44:00.048
STYPE/SGN/VSNO/VSN1/MT:              35 OBJBKP   TST003 N/A    06 (MOVE VOLUME (MOVEVOL) UTILITY)
PDWO/PDWK/PDRO/PDRK/PDDO/PDDK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
POWO/POWK/PORO/PORD/PODO/PODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
PTWO/PTWK/PTRO/PTRD/PTDO/PTDK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
BOWO/BOWK/BORO/BORK/BODO/BODK:       00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
BTWO/BTWK/BTRO/BTRK/BTDO/BTDK:       00000000006  00000039072  00000000006  00000039072  00000000000  00000000000
B2OWO/B2OWK/B2ORO/B2ORK/B2ODO/B2ODK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
B2TWO/B2TWK/B2TRO/B2TRK/B2TDO/B2TDK: 00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
DTUP/DTDE/4KIN/4KDE/32KI/32KD/NCE:   00000000006  00000000000  00000000000  00000000000  00000000000  00000000000  00000000000
FLGS/NTE/RCLD/RCLK/LOBI/LOBD:        82000000     00000000000  00000000000  00000000000  00000000000  00000000000
```

Figure 4-31 SMF85TH output

In Figure 4-32 on page 124 to Figure 4-36 on page 128 we show the SMF records that relate to this output.

SMF record type 85 subtype 32 is used to map subtype records 32, 33, 34, and 35.

Note: You should not use the contents of the CBRSMF macro as presented here. When you assemble the program, or wish to refer to the macro, it will be found in SYS1.MACLIB.

```
ST32     DSECT                  SUBTYPES 32 - 35
ST32SGN  DS     CL8' '          STORAGE GROUP NAME
ST32VSNO DS     CL6' '          VOLUME SERIAL NUMBER OF OPTICAL
*                               VOLUME. ONLY VALID FOR SUBTYPES
*                               34 AND 35, CONTAINS BLANKS FOR
*                               OTHER SUBTYPES.
ST32VSN1 DS     CL6' '          VOLUME SERIAL NUMBER OF OPPOSITE
*                               SIDE OF OPTICAL DISK
*                               VOLUME. ONLY VALID FOR SUBTYPES
*                               34 AND 35, CONTAINS BLANKS FOR
*                               OTHER SUBTYPES.
ST32OMT  DS     CL2' '          OPTICAL MEDIA TYPE. ONLY VALID
*                               FOR SUBTYPES 34 AND 35, CONTAINS
*                               BLANKS FOR OTHER SUBTYPES.
         DS     CL2' '          RESERVED
*********************************************************************
* COUNTS OF PRIMARY OBJECTS (AND KILOBYTES) WRITTEN, READ AND
* DELETED FROM DASD.
*********************************************************************
ST32PDWO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               WRITTEN TO DASD.
ST32PDWK DS     BL4'0'          NUMBER OF KILOBYTES PRIMARY
*                               OBJECTS WRITTEN TO DASD.
ST32PDRO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               READ FROM DASD.
ST32PDRK DS     BL4'0'          NUMBER OF KILOBYTES PRIMARY
*                               OBJECTS READ FROM DASD.
ST32PDDO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               DELETED FROM DASD.
ST32PDDK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS DELETED FROM DASD.
*********************************************************************
* COUNTS OF PRIMARY OBJECTS (AND KILOBYTES) WRITTEN, READ AND
* DELETED FROM OPTICAL.
*********************************************************************
ST32POWO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               WRITTEN TO OPTICAL.
ST32POWK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS WRITTEN TO OPTICAL.
ST32PORO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               READ FROM OPTICAL.
ST32PORK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS READ FROM OPTICAL.
ST32PODO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               DELETED FROM OPTICAL.
ST32PODK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS DELETED FROM OPTICAL.
```

Figure 4-32 SMF record type 85 subtype 32 significant fields (extract from CBRSMF macro) (1 of 5)

```
*********************************************************************
* COUNTS OF PRIMARY OBJECTS (AND KILOBYTES) WRITTEN, READ AND
* DELETED FROM TAPE.
*********************************************************************
ST32PTWO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               WRITTEN TO TAPE.
ST32PTWK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS WRITTEN TO TAPE.
ST32PTRO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               READ FROM TAPE.
ST32PTRK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS READ FROM TAPE.
ST32PTDO DS     BL4'0'          NUMBER OF PRIMARY OBJECTS
*                               LOGICALLY DELETED FROM TAPE.
ST32PTDK DS     BL4'0'          NUMBER OF KILOBYTES OF PRIMARY
*                               OBJECTS LOGICALLY DELETED FROM
*                               TAPE.
*********************************************************************
* COUNTS OF BACKUP OBJECTS (AND KILOBYTES) WRITTEN, READ AND
* DELETED FROM OPTICAL.
*********************************************************************
ST32BOWO DS     BL4'0'          NUMBER OF BACKUP  OBJECTS
*                               WRITTEN TO OPTICAL.
ST32BOWK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS WRITTEN TO OPTICAL.
ST32BORO DS     BL4'0'          NUMBER OF BACKUP OBJECTS
*                               READ FROM OPTICAL.
ST32BORK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS READ FROM OPTICAL.
ST32BODO DS     BL4'0'          NUMBER OF BACKUP  OBJECTS
*                               DELETED FROM OPTICAL.
ST32BODK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS DELETED FROM OPTICAL.
*********************************************************************
* COUNTS OF BACKUP OBJECTS (AND KILOBYTES) WRITTEN, READ AND
* DELETED FROM TAPE.
*********************************************************************
ST32BTWO DS     BL4'0'          NUMBER OF BACKUP  OBJECTS
*                               WRITTEN TO TAPE.
ST32BTWK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS WRITTEN TO TAPE.
ST32BTRO DS     BL4'0'          NUMBER OF BACKUP  OBJECTS
*                               READ FROM TAPE.
ST32BTRK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS READ FROM TAPE.
ST32BTDO DS     BL4'0'          NUMBER OF BACKUP  OBJECTS
*                               LOGICALLY DELETED FROM TAPE.
ST32BTDK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP
*                               OBJECTS LOGICALLY DELETED FROM
*                               TAPE.
```

Figure 4-33 SMF record type 85 subtype 32 significant fields (extract from CBRSMF macro) (2 of 5)

```
*********************************************************************
* COUNTS OF BACKUP2 OBJECTS (AND KILOBYTES) WRITTEN, READ AND    @L2A
* DELETED FROM OPTICAL.                                         @L2A
*********************************************************************
ST32B2OWO DS     BL4'0'          NUMBER OF BACKUP2 OBJECTS       @L2A
*                                WRITTEN TO OPTICAL.             @L2A
ST32B2OWK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS WRITTEN TO OPTICAL.     @L2A
ST32B2ORO DS     BL4'0'          NUMBER OF BACKUP2 OBJECTS       @L2A
*                                READ FROM OPTICAL.              @L2A
ST32B2ORK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS READ FROM OPTICAL.      @L2A
ST32B2ODO DS     BL4'0'          NUMBER OF BACKUP2  OBJECTS      @L2A
*                                DELETED FROM OPTICAL.           @L2A
ST32B2ODK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS DELETED FROM OPTICAL.   @L2A
*********************************************************************
* COUNTS OF BACKUP2 OBJECTS (AND KILOBYTES) WRITTEN, READ AND    @L2A
* DELETED FROM TAPE.                                            @L2A
*********************************************************************
ST32B2TWO DS     BL4'0'          NUMBER OF BACKUP2 OBJECTS       @L2A
*                                WRITTEN TO TAPE.                @L2A
ST32B2TWK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS WRITTEN TO TAPE.        @L2A
ST32B2TRO DS     BL4'0'          NUMBER OF BACKUP2 OBJECTS       @L2A
*                                READ FROM TAPE.                 @L2A
ST32B2TRK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS READ FROM TAPE.         @L2A
ST32B2TDO DS     BL4'0'          NUMBER OF BACKUP2 OBJECTS       @L2A
*                                LOGICALLY DELETED FROM TAPE.    @L2A
ST32B2TDK DS     BL4'0'          NUMBER OF KILOBYTES OF BACKUP2  @L2A
*                                OBJECTS LOGICALLY DELETED FROM  @L2A
*                                TAPE.                           @L2A
```

Figure 4-34 SMF record type 85 subtype 32 significant fields (extract from CBRSMF macro) (3 of 5)

```
*********************************************************************
* COUNTS OF ACTIVITY AGAINST THE OBJECT STORAGE DATABASE
* (OBJECT DIRECTORY TABLE, 4K OBJECT STORAGE TABLE AND 32K
* OBJECT STORAGE TABLE).
*********************************************************************
ST32DTUP DS     BL4'0'          NUMBER OF ROWS UPDATED IN THE
*                               OBJECT DIRECTORY TABLE.
ST32DTDE DS     BL4'0'          NUMBER OF ROWS DELETED FROM THE
*                               OBJECT DIRECTORY TABLE.
ST324KIN DS     BL4'0'          NUMBER OF ROWS INSERTED INTO THE
*                               4K OBJECT STORAGE TABLE.
ST324KDE DS     BL4'0'          NUMBER OF ROWS DELETED FROM THE
*                               4K OBJECT STORAGE TABLE.
ST3232KI DS     BL4'0'          NUMBER OF ROWS INSERTED INTO THE
*                               32K OBJECT STORAGE TABLE.
ST3232KD DS     BL4'0'          NUMBER OF ROWS DELETED FROM THE
*                               32K OBJECT STORAGE TABLE.
ST32NCE  DS     BL4'0'          NUMBER OF OPTICAL CARTRIDGES
*                               EXPIRED.  VALID ONLY FOR
*                               SUBTYPE 32.
ST32FLGS DS     BL4'0'          PROCESSING FLAGS
ST32FLG0 EQU    X'80'           WHEN ON, THIS PROCESS WAS
*                               INVOKED AUTOMATICALLY UNDER
*                               SOFTWARE CONTROL.
ST32FLG1 EQU    X'40'           WHEN ON, THIS PROCESS WAS
*                               INVOKED BY A MODIFY OAM,START
*                               COMMAND.
ST32FLG2 EQU    X'20'           WHEN ON, THIS PROCESS WAS
*                               INVOKED USING AN ISMF LINE
*                               OPERATOR.
ST32FLG3 EQU    X'10'           WHEN ON, VOL RECOVERY WAS
*                               INVOKED W/ BACKUP1 KEYWORD OR
*                               DEFAULTED TO BACKUP1            @L2A
ST32FLG4 EQU    X'08'           WHEN ON, VOL RECOVERY WAS
*                               INVOKED W/ BACKUP2 KEYWORD      @L2A
ST32FLG5 EQU    X'04'           WHEN ON, VOL RECOVERY OR MOVEVOL WAS
*                               SPECIFIED WITH DELETE OPTION    @L3A
ST32FLG6 EQU    X'02'           WHEN ON, VOL RECOVERY OR MOVEVOL WAS
*                               SPECIFIED WITH RECYCLE OPTION   @L3A
ST32FLG7 EQU    X'01'           WHEN ON, INDICATED PROCESSING OBJECT
*                               STORAGE GROUP                   @L3A
ST32FLG8 EQU    X'80'           WHEN ON, INDICATED PROCESSING BACKUP
*                               OBJECT STORAGE GROUP            @L3A
ST32FLG9 EQU    X'40'           WHEN ON, CYCLE ENDTIME EXCEEDED @L3A
*
```

Figure 4-35 SMF record type 85 subtype 32 significant fields (extract from CBRSMF macro) (4 of 5)

```
ST32NTE  DS     BL4'0'          NUMBER OF TAPE VOLUMES EXPIRED
*                               VALID ONLY FOR SUBTYPE 32       @L3A
ST32RCLD DS     BL4'0'          NUMBER OF RECALLED OBJECTS
*                               VALID ONLY FOR SUBTYPE 32       @00A
ST32RCLK DS     BL4'0'          NUMBER OF KILOBYTES OF
*                               RECALLED OBJECTS
*                               VALID ONLY FOR SUBTYPE 32       @00A
ST32LOBI DS     BL4'0'          NUMBER OF LOB ROWS INSERTED
*                               VALID ONLY FOR SUBTYPE 32       @00A
ST32LOBD DS     BL4'0'          NUMBER OF LOB ROWS DELETED
*                               VALID ONLY FOR SUBTYPE 32       @00A
```

Figure 4-36 SMF record type 85 subtype 32 significant fields (extract from CBRSMF macro) (5 of 5)

Migration and coexistence

Pre-V1R8 systems in the same OAMplex as a V1R8 system must have the fix for APAR OA12683 applied.

Maintenance

The fix for APAR OA17310 must be applied prior to using the OAM RECYCLE function. It fixes an ABEND0C4, a message problem when there are no more volumes to be recycled, and a problem with SMF record type 85 missing new fields.

4.4 GLOBAL display keyword

A GLOBAL keyword has been added to the MODIFY OAM,DISPLAY command.

4.4.1 Usage

You can use the MODIFY OAM,DISPLAY command to display the current settings of the SETOAM, SETOPT, or SETOSMC statements for the OAM address space.

The syntax of this command, from the *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, publication, is as follows:

```
MODIFY OAM,DISPLAY,keyword1,keyword2
```

keyword1 must be one of the options shown in Figure 4-37.

```
SETOPT | SETOAM | SETOSMC
    Specifies the command parameter being displayed. Use SETOAM to display
    values of settings in an object tape environment. SETOPT is used to
    display values of settings in an optical environment. Use SETOSMC to
    display the values of settings for OSMC processing.
```

Figure 4-37 The options for keyword1

keyword2 must be one of the options shown in Figure 4-38.

```
ALL|GLOBAL|storgrp
   Specifies the kind of information that the system is to display. For
   the SETOAM, SETOSMC, and SETOPT parameters, the valid values are as
   follows:

   ALL
      Displays the settings for each valid storage group as well as the
      global default settings. If ALL is specified, the global default,
      if applicable, is displayed as well as the settings for each
      valid storage group in the active SMS configuration.

   GLOBAL
      Displays only the OAM global keywords.

   storgrp
      Displays only the settings for the specified storage group
      name. There can be up to fourteen storage group names
      indicated on a single DISPLAY command.

      Displays the settings for the FIRSTBACKUPGROUP and
      SECONDBACKUPGROUP for the specified storage group name.
```

Figure 4-38 The new GLOBAL keyword2

Figure 4-39 is an example of the new GLOBAL keyword in action when displaying SETOSMC settings.

```
F OAM,DISPLAY,SETOSMC,GLOBAL
CBR1075I GLOBAL value for BACKUP1 is OBJBKP
CBR1075I GLOBAL value for BACKUP2 is -NONE-
CBR1075I GLOBAL value for CYCLEW is STRTONLY
CBR1075I GLOBAL value for MAXRECAL is 2
CBR1075I GLOBAL value for RECALLO is 15, ON
CBR1075I GLOBAL value for RECALLT is 0, ON
CBR1075I GLOBAL value for CLEAROLD is TAPE
```

Figure 4-39 Displaying SETOSMC settings with the GLOBAL keyword

Migration and coexistence

There are no migration or coexistence considerations for use of this new keyword.

4.5 Update from z/OS V1.7

z/OS V1R7 DFSMS Technical Update, SG24-7225, contains a description of various OAM enhancements. One in particular relates to OSMC SINGLE OBJECT RECALL UTILITY. There has been no change to that in z/OS V1R8, but the function was retested because an SMF MACRO error was found in z/OS V1R7.

The SMF error is still present in z/OS V1R8, but it now has an APAR number OA20170 (“OA20170” on page 464).

In z/OS V1R7 DFSMS OAM introduced the ability to recall objects from tape and store them in DB2 to improve performance for subsequent retrievals.

OAM writes SMF record type 85 subtype 38 to document the use of this facility.

We have written a simple program called SMF85TO to scan the SMF records and summarize activity. This was originally made available in *z/OS V1R7 DFSMS Technical Update*, SG24-7225. The example provided here is smaller but has the same function.

The program itself and how to construct it is documented in “SMF record type 85 subtype 38 data display program” on page 484.

In Figure 4-40 we show the JCL to extract the SMF records and run the program.

Note: This JCL assumes that an extract is being taken from the active SMF data set, which is then passed to the SMFT85O program. The JCL could be changed to make a permanent extract of the SMF data, or to read an already created SMF data extract.

```
//MHLRES1O JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SYS1.SC64.MAN1
//OUTDD  DD DSN=&SMFT85,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=5096,
// DISP=(,PASS,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
          OUTDD(OUTDD,TYPE(85(38)))
/*
//  EXEC PGM=SMF85TO
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TO.LOAD
//SYSUDUMP DD SYSOUT=A
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=&SMFT85
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure 4-40 SMF85TO program execution JCL

In Figure 4-41 we show an example of output from the execution of program SMF85TO. In Figure 4-42 on page 131 we show the SMF records that relate to this output.

```
SMF TYPE 85 SUBTYPE 38 RECORDS
COLN/CNID:          OAMTEST.MAR210E                          00000000046
OBJN/SGN/OLEN:      OAMTEST.OBJ0210E                         GROUP00  00000050000
VSN/MT/TKN/VT/BT/FLGS:TST000 06 00000001242                FLG0 ON
```

Figure 4-41 SMF85TO output

The left-hand side of the report contains abbreviations from the SMF records. Record type 85 subtype 38 contains the fields of interest to verify the function of OAM object recall to DB2. In the SMF records, as shown in Figure 4-42, the fields are prefixed with ST38. In the report the fields are shown without the ST38, and have been compressed to have more than one entry per line.

In this example, Collection OAMTEST.MAR210E, which contains object OMATEST.OBJ0210E, was recalled. Object OAMTEST.OBJ0210E is 50000 long. The flags field shown in the fourth line as been interpreted from the SMF field ST38FLG0 as being ON, which indicates that the recall was successful.

Note: You should not use the contents of the CBRSMF macro as presented here. When you assemble the program, or wish to refer to the macro, it will be found in SYS1.MACLIB.

```
ST38     DSECT                  SUBTYPE 38                    @P2A
ST38COLN DS     CL44' '         COLLECTION NAME               @P2A
ST38CNID DS     BL4'0'          COLLECTION ID                 @P2A
ST38OBJN DS     CL44' '         OBJECT NAME                   @P2A
ST38SGN  DS     CL8' '          STORAGE GROUP NAME            @P2A
ST38OLEN DS     BL4'0'          OBJECT LENGTH                 @P2A
ST38VSN  DS     CL6' '          VOLUME SERIAL NUMBER OF OPTICAL
*                               VOLUME OR TAPE VOLUME FROM WHICH
*                               THE COPY OF THE OBJECT
*                               WAS READ.                     @P2A
ST38MT   DS     CL2' '          MEDIA TYPE OF THE VOLUME FROM
*                               WHICH THE COPY OF THE OBJECT
*                               WAS READ.                     @P2A
ST38TKN  DS     BL4'0'          VOLUME LOCATION TOKEN ASSOCIATED
*                               WITH THE COPY OF THE OBJECT
*                               ON THE VOLUME SPECIFIED IN THE
*                               ST39VSN FIELD.                @P2A
ST38VT   DS     CL1' '          VOLUME TYPE                   @P2A
ST38BT   DS     CL1' '          BACKUP TYPE                   @P2A
         DS     CL2' '          RESERVED                      @P2A
ST38FLGS DS     BL4'0'          PROCESSING FLAGS              @P2A
*
ST38FLG0 EQU    X'80'           WHEN ON, OBJECT RECALL
*                               WAS SUCCESSFUL.               @P2A
```

Figure 4-42 SMF record type 85 subtype 38 significant fields (extract from CBRSMF macro)

5

DFSMSdss enhancements

In z/OZ V1R8 a number of enhancements have been incorporated into DFSMSdss. The majority of the enhancements are used by DFSMShsm in support of Fast Replicate.

Users should refer to DFSMShsm for information concerning fast replication.

The interface to Physical Data Set COPY and Physical Data Set DUMP/RESTORE is available for users of DFSMSdss without using DFSMShsm. This document concentrates on this use of DFSMSdss.

DFSMSdss continues to support logical data set operations.

5.1 Defining logical and physical processing

The following information is based on Chapter 3 of *DFSMSdss Storage Administration Guide,* SC26-0423, as it provides a background to the new physical processing in z/OS V1R8.

DFSMSdss can perform two kinds of processing when executing COPY, DUMP, and RESTORE commands:

- *Logical processing* operates against data sets independently of physical device format.
- *Physical processing* moves data at the track-image level and operates against volumes, tracks, and data sets.

Each type of processing offers different capabilities and advantages.

During a restore operation, the data is processed the same way in which it is dumped because physical and logical dump tapes have different formats. If a data set is dumped logically, it is restored logically. If it is dumped physically, it is restored physically. A data set restore operation from a full-volume dump is a physical data set restore operation.

Logical processing

A logical copy, dump, or restore operation treats each data set and its associated information as a logical entity, and processes an entire data set before beginning the next one.

Each data set is moved by tracks from the source device and is potentially written to the target device as a set of data records, allowing data movement between devices with different track and cylinder configurations. Checking of data record consistency is not performed during dump operations.

DFSMSdss performs logical processing if:

- You specify the DATASET keyword with the COPY command. A data set copy is always a logical operation regardless of how or whether you specify input volumes.
- You specify the DATASET keyword with the DUMP command, and either no input volume is specified, or LOGINDDNAME, LOGINDYNAM, or STORGRP is used to specify input volumes.
- The RESTORE command is performed, and the input volume was created by a logical dump.

DFSMSdss uses catalogs or VTOCs to select data sets for logical processing. If you do not specify input volumes, DFSMSdss uses the catalogs to select data sets for copy and dump operations. If you specify input volumes using the LOGINDDNAME, LOGINDYNAM, or STORGRP keywords on the COPY or DUMP command, DFSMSdss uses VTOCs to select data sets for processing.

Note: To copy or dump entire multivolume data sets, you do not need to specify all the volumes in the LOGINDDNAME or LOGINDYNAM volume list. However, you must specify the SELECTMULTI keyword with either the FIRST or ANY subkeywords.

When to use logical processing

Use logical processing for the following situations:

- Data is copied to an unlike device.

  Logical processing is the only way to move data between unlike device types.

- Data that may need to be restored to an unlike device is dumped.

  Data must be restored the same way in which it is dumped. This is particularly important to keep in mind when making backups that you plan to retain for a long period of time (such as vital records backups). If a backup is retained for a long period of time, it is possible that the device type it originally resided on will no longer be in use at your site when you want to restore it. This means that you have to restore it to an unlike device, which can only be done if the backup was made logically.
- Aliases of VSAM user catalogs are to be preserved during the copy and restore functions.

  Aliases are not preserved for physical processing.
- Unmovable data sets or data sets with absolute track allocation are moved to different locations.
- Multivolume data sets are processed.
- VSAM and multivolume data sets are cataloged as part of DFSMSdss processing.
- Data sets are deleted from the source volume after a successful copy or restore operation.
- Non-VSAM and VSAM data sets are renamed after a successful copy or restore operation.
- You want to control the percentage of space allocated on each of the output volumes for copy or restore operations.
- You want to copy and convert a PDS to PDSE or vice versa.
- You want to copy or restore a data set with an undefined DSORG to an unlike device.
- You want to keep together all parts of a VSAM sphere.

Physical processing

Physical processing moves data based on physical track images. Because data movement is carried out at the track level, only target devices with track sizes equal to those of the source device are supported. Physical processing operates on volumes, ranges of tracks, or data sets. For data sets, it relies on volume information (in the VTOC and VVDS) for data set selection, and processes only that part of a data set residing on the specified input volumes.

Notes: VSAM data sets are *not* cataloged during physical processing within SMS or non-SMS environments. The CATALOG keyword is ignored for VSAM data set during physical restore. Use IDCAMS DEFINE RECATALOG to catalog the data sets after the restore.

The RENAME and RENAMEUNCONDITIONAL keywords are ignored for VSAM data sets during physical restore.

DFSMSdss performs physical processing when the following conditions exist:

- You specify the FULL or TRACKS keyword with the COPY or DUMP command.

  This results in a physical volume or physical tracks operation.

  Attention: Take care when invoking the TRACKS keyword with the COPY and RESTORE keywords. The TRACKS keyword should only be used for a data set recovery operation. For example, you can use it to repair a bad track in the VTOC or a data set, or to retrieve data from a damaged data set. You cannot use it in place of a full-volume or a logical data set operation. Doing so could destroy or impair data integrity.

- You specify the DATASET keyword on the COPY or DUMP command and input volumes with the PHYSINDDNAME or PHYSINDYNAM keyword. This produces a physical data set copy or physical data set dump.
- The RESTORE command is executed and the input volume is created by a physical dump operation.

When to use physical processing

Use physical processing when the following conditions exist:

- Backing up system volumes that you might want to restore with a stand-alone DFSMSdss restore operation.

  Stand-Alone DFSMSdss restore only supports physical dump tapes.
- Performance is an issue.

  Generally, the fastest way (measured by elapsed time) to copy or dump an entire volume is with a physical full-volume command. This is primarily because minimal catalog searching is necessary for physical processing.
- Substituting one physical volume for another or recovering an entire volume.

  With a COPY or RESTORE (full-volume or track) command, the volume serial number of the input DASD volume can be copied to the output DASD volume.
- Dealing with I/O errors. Physical processing provides the capability to copy, dump, and restore a specific track or range of tracks.
- Dumping or copying between volumes of the same device type but different capacity.

5.2 Physical and logical data set difference summary

The form that data sets are used by application programs in is what DFSMSdss refers to as the logical form. The data set physical structure and location are provided to the application through the system catalog (usually) and supplementary information from the DASD VTOC and VVDS entries for the data set. For single volume non-VSAM data sets there is not much further information to be obtained, but when the data set becomes multi-volume and has a more complicated structure such as VSAM and may be multi-volume it is necessary to use these sources of data for effective use of the data set. There is a trade off, however, in that obtaining all that information takes time.

For utility programs such as DFSMSdss, whether used in its own right or as support for DFSMShsm, for example, there may be situations where a copy of a data set may be able to be made more rapidly by using a physical process. This is where the DFSMSdss Physical COPY can be used.

5.3 Physical and logical data set copy specification changes

z/OS V1R8 DFSMSdss command specifications have been updated to better emphasize the difference between logical and physical processing.

The INDDNAME/INDYNAM keyword is replaced by LOGINDDNAME/LOGINDYNAM and PHYSINDDNAME/PHYSINDDNAM. LOGINDDNAME/LOGINDYNAM was available prior to z/OS V1R8.

Use of the INDDNAME/INDYNAM keywords is accepted, but they are internally replaced by PHYSINDDNAME/PHYSINDDNAM. Message ADR146I is issued to advise of this once APAR OA20242 has been implemented (as is the case for Physical DUMP). See “OA20242” on page 465 for details.

Physical DUMP operation changes

Although there are code changes in the Physical DUMP part of DFSMSdss to support Physical COPY, to the user there is minimal change. New keywords PHYSINDDNAME and PHYSINDYNAM, if used, make it more obvious that the operations relate to physical operations.

The main keyword difference is the change from the use of the INDDNAME and INDYNAM keywords, which in the past were used in conjunction with the DUMP command to specify physical copies. The new keywords PHYSINDDNAME and PHYSINDYNAM to replace INDDNAME and INDYNAM make it more obvious that the operations relate to physical operations.

You can continue to use existing DUMP jobs using INDDNAME or INDYNAM, and will receive a warning message ADR146I advising that the obsolete keyword has been replaced.

Note: There is no prior system toleration for the new PHYSINDDNAME or PHYSINDYNAM keywords, so jobs should not be changed to use these new keywords until all systems are converted to z/OS V1R8 or later.

In Figure 5-1 we show use of the INDD keyword (short form of INDDNAME) to specify the source volume.

```
//MHLRES1D JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//STEP01  EXEC PGM=ADRDSSU,TIME=1440,REGION=6000K,
//        PARM='UTILMSG=YES'
//SYSPRINT DD  SYSOUT=*
//SBOX38   DD DISP=SHR,UNIT=SYSDA,DSN=MHLRES1.JOBS.JCL
//TEMP     DD  DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN    DD  *
  DUMP DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                  -
   )) -
   SHARE -
   INDD(SBOX38) OUTDD(TEMP)
```

Figure 5-1 DFSMSdss DUMP job using INDD

In Figure 5-2 we show the output from the job shown in Figure 5-1 on page 137, where DFSMSdss has substituted the keyword PHYSINDDNAME for INDD.

```
PAGE 0001    5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES    2007.065 11:40
  DUMP DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                -
   )) -
   SHARE -
   INDD(SBOX38) OUTDD(TEMP)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ADR109I (R/I)-RI01 (01), 2007.065 11:40:49 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR146I (R/I)-RI03 (13), OBSOLETE KEYWORD 'INDDNAME ' SPECIFIED. 'PHYSINDDNAME ' WILL BE USED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2007.065 11:40:49 EXECUTION BEGINS
ADR378I (001)-DTDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME SBOX38
                         MHLRES1.JOBS.JCL
ADR006I (001)-STEND(02), 2007.065 11:40:50 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 2007.065 11:40:50 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 2007.065 11:40:50 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Figure 5-2 DFSMSdss DUMP job output showing message about substitution of PHYSINDDNAME for INDD

5.3.1 Logical copy operations

To make a logical copy of a data set, the COPY command and keywords DATASET and OUTDD are required, and they can be used together with many optional modifiers.

In Figure 5-3 we show the JCL to copy a data set using the system catalog to locate the volume that the data set is on. The example demonstrates the input data set lookup with as few other operands as possible, so the data set is not renamed. As a result, because this is a COPY operation, the copy fails. This situation differs from the case when Physical COPY is used (see 5.3.3, “Physical copy operations” on page 140).

```
//MHLRES1D JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//STEP01  EXEC PGM=ADRDSSU,TIME=1440,REGION=6000K,
//        PARM='UTILMSG=YES'
//SYSPRINT DD  SYSOUT=*
//TEMP     DD  DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN    DD  *
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                -
   )) -
   SHARE -
   OUTDD(TEMP)
```

Figure 5-3 Logical COPY without LOGINDDNAME

In Figure 5-4 we show the output from the job selecting data set by just using the system catalog. As predicted, the copy was not allowed to proceed because no rename keyword was specified.

```
PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.066 15:08
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                  -
   )) -
   SHARE -
   OUTDD(TEMP)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ADR109I (R/I)-RI01 (01), 2007.066 15:08:13 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2007.066 15:08:13 EXECUTION BEGINS
ADR713E (001)-ALLOC(01), UNABLE TO ALLOCATE SMS MANAGED DATA SET MHLRES1.JOBS.JCL BECAUSE NEITHER DELETE NOR RENAMEU WAS SPE
ADR801I (001)-DDDS (01), DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED
                         OTHER REASONS.
ADR455W (001)-DDDS (02), THE FOLLOWING DATA SETS WERE NOT SUCCESSFULLY PROCESSED
                          MHLRES1.JOBS.JCL
ADR006I (001)-STEND(02), 2007.066 15:08:14 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 2007.066 15:08:14 TASK COMPLETED WITH RETURN CODE 0008
ADR012I (SCH)-DSSU (01), 2007.066 15:08:14 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0008 FROM:
                         TASK    001
```

Figure 5-4 Logical COPY output without LOGINDDNAME

If there is a reason to locate a data set other than through the system catalog, for example, because there is a copy on another volume, the LOGINDDNAME keyword can be specified. If the DDNAME specified on LOGINDDNAME refers to the volume that the requested data set is on, the copy is able to proceed.

If, however, the volume referred to was not where the requested data set was, the copy would not be able to proceed.

In Figure 5-5 we show a sample job to make a copy of a data set, illustrating the use of the LOGINDDNAME keyword, but the DDNAME referred to does not contain the data set to be copied, so the job should fail.

```
//MHLRES1D JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//STEP01  EXEC PGM=ADRDSSU,TIME=1440,REGION=6000K,
//        PARM='UTILMSG=YES'
//SYSPRINT DD  SYSOUT=*
//SBOXA8   DD DISP=SHR,UNIT=SYSDA,VOL=SER=SBOXA8
//TEMP     DD  DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN    DD  *
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                  -
   )) -
   SHARE -
   LOGINDDNAME(SBOXA8) OUTDD(TEMP)
```

Figure 5-5 Logical COPY with LOGINDDNAME referring to the wrong volume

In Figure 5-6 we show the output from running the example in Figure 5-5 on page 139.

The volume referred to by LOGINDDNAME(SBOXA8) in the JCL DDNAME(SBOXA8) does not contain data set MHLRES1.JOBS.JCL, so the job fails.

```
PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.066 15:15
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                  -
   )) -
   SHARE -
   LOGINDDNAME(SBOXA8) OUTDD(TEMP)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ADR109I (R/I)-RI01 (01), 2007.066 15:15:53 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2007.066 15:15:53 EXECUTION BEGINS
ADR383W (001)-DDDS (01), DATA SET MHLRES1.JOBS.JCL NOT SELECTED
ADR455W (001)-DDDS (03), THE FOLLOWING DATA SETS WERE NOT SUCCESSFULLY PROCESSED
                          MHLRES1.JOBS.JCL
ADR470W (001)-DDDS (04), NO DATA SETS SELECTED FOR PROCESSING
ADR006I (001)-STEND(02), 2007.066 15:15:54 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 2007.066 15:15:54 TASK COMPLETED WITH RETURN CODE 0004
ADR012I (SCH)-DSSU (01), 2007.066 15:15:54 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0004 FROM:
                         TASK    001
```

Figure 5-6 Logical COPY output with LOGINDDNAME referring to the wrong volume

5.3.2 DSS LOGICAL COPY warning

A VSAM KSDS with data and index components on different volumes may lose its SMS RLSDATA attribute during DSS COPY. Refer to “OA18319” on page 462, for further information.

5.3.3 Physical copy operations

To make a physical copy of a data set, the COPY command and keywords DATASET, OUTDDNAME, and PHYSINDDNAME or PHYSINDYNAM are required, which can be used together with many optional modifiers. The significant difference over a logical COPY is the presence of the PHYSINDDNAME or PHYSINDYNAM keyword. In this respect the syntax is like the version using LOGINDDNAME or LOGINDYNAM.

When using a physical COPY, a data set that is SMS managed can be copied. This is not allowed with a logical copy, as demonstrated in Figure 5-4 on page 139.

The user must then adjust the catalog to refer to the new data set if that is what is intended.

In Figure 5-7 we show sample JCL to make a physical copy of a data set. The input data set is SMS managed (and must therefore be catalogged), and the target volume as specified in OUTDDNAME must also refer to an SMS managed volume.

```
//MHLRES1D JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//STEP01  EXEC PGM=ADRDSSU,TIME=1440,REGION=6000K,
//        PARM='UTILMSG=YES'
//SYSPRINT DD  SYSOUT=*
//SBOX08   DD DISP=SHR,UNIT=SYSDA,DSN=MHLRES1.JOBS.JCL
//TEMP     DD  DISP=SHR,VOL=SER=MLD40C
//SYSIN    DD  *
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                    -
   )) -
   SHARE -
   PHYSINDDNAME(SBOX08) OUTDD(TEMP)
```

Figure 5-7 Physical COPY JCL

In Figure 5-8 we show the output from the job shown in Figure 5-7. There is no specific indication that this is the result of a Physical COPY, but that can be inferred from the fact that the input data set MHLRES1.JOBS.JCL, which is SMS managed, has been copied, which would not have been permitted with a Logical COPY.

```
PAGE 0001    5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES    2007.066 17:22
  COPY DATASET(INCLUDE(-
    MHLRES1.JOBS.JCL                -
   )) -
   SHARE -
   PHYSINDDNAME(SBOX08) OUTDD(TEMP)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ADR109I (R/I)-RI01 (01), 2007.066 17:22:21 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2007.066 17:22:21 EXECUTION BEGINS
ADR396I (001)-PCNVS(01), DATA SET MHLRES1.JOBS.JCL ALLOCATED, ON VOLUME(S): MLD40C
ADR801I (001)-DDDS (01), DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SELECTED: 0 FAILED
SERIALIZATION AND 0 FAILED
                 OTHER REASONS.
ADR454I (001)-DDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
                  MHLRES1.JOBS.JCL
ADR006I (001)-STEND(02), 2007.066 17:22:22 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 2007.066 17:22:22 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 2007.066 17:22:22 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Figure 5-8 Physical COPY job output

In Figure 5-9 we show the results of attempting to display the characteristics of the MHLRES1.JOBS.JCL data set on volume MLD40C. The warning duplicate data set name is displayed in the top right-hand corner of the panel, as highlighted in bold.

```
 Menu  Options  View  Utilities  Compilers  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
DSLIST - Data Sets on volume MLD40C                    Duplicate data set name
Command ===>                                                 Scroll ===> CSR

Command - Enter "/" to select action                Message          Volume
-------------------------------------------------------------------------------
I        MHLRES1.JOBS.JCL                                                MLD40C
****************************** End of Data Set list ***************************
```

Figure 5-9 ISPF 3.4 listing of data set MHLRES1.JOBS.JCL on volume MLD40C

In Figure 5-10 we show the help information about the situation that exists as a result of making a physical copy of the data set and trying to locate it.

```
TUTORIAL ----------------- DUPLICATE DATA SET NAMES ------------------ TUTORIAL
COMMAND ===>

  The data set that you are accessing via ISPF:
        MHLRES1.JOBS.JCL
  is also cataloged on a volume other than MLD40C  .

  Since MLD40C  is an SMS volume, the allocation of the data set
  will use the catalog to find the data set and will allocate the
  data set that is on another volume.  Since this is not the volume
  you specified, the command fails.
```

Figure 5-10 ISPF Help information for the duplicate data set name situation

ISPF prevented any attempt to process the copy of the data set while the original was still catalogged.

The original was renamed so that there was no longer a duplicate catalog situation, but the copied data set could not be cataloged using ISPF as it, being SMS managed, is in an error state.

This copied data set was cataloged by means of the IDCAMS command DEFINE NVSAM(NAME('MHLRES1.jobs.jcl') VOLUMES(Mld40c) DEVT(3390) recatalog) issued under TSO.

6

DFSMShsm enhancements

The DFSMShsm enhancements in DFSMS V1R8 provide improvements in the areas of error handling for alternate duplex tapes, tape recycling, migration of non-VSAM data sets, ARECOVER processing, and altering the priority of queued requests. For details on the DFSMShsm fast replication function in DFSMS V1R8 refer to Chapter 7, “DFSMShsm fast replication” on page 169.

The following topics are covered:

- Error handling on alternate duplex tapes
- Recycle SYNCDEV at intervals
- Migration scratch queue for non-VSAM data set
- Individual data set restore for ARECOVER processing
- New command ALTERPRI

6.1 Error handling on alternate duplex tapes

Before z/OS V1R8.0, when an error was detected on the alternate tape, during the duplex process, this alternate tape was de-mounted and returned to scratch, and DFSMShsm continues to write to the original volume.

It has become problematic for users with high capacity tapes and using duplex tape function to create alternate copies for disaster recovery, due the tape size and the total time the user would be without a valid tape copy.

In z/OS V1R8.0, the duplex tape migration option is improved. A new keyword, ERRORALTERNATE(CONTINUE | MARKFULL), is added to the existing SETSYS DUPLEX(MIGRATION(Y)). If you choose the DUPLEX option, two tapes are created concurrently, and the most common use of that is to keep one on site and the other on a remote tape library.

The new option lets users choose to MARK FULL original and alternate tapes in case of a problem writing on the alternate tape. In this case, both tapes are demounted and the failing data set is retried on a new set of tapes.

The other option on this command is CONTINUE. If you specify CONTINUE, and an error occurs on the alternate volume, then the process continues on the original tape. DFSMShsm creates a TCN record (Tape Copy Need) and a TAPECOPY is immediately attempted after completion of the original tape, or else it will be scheduled to the next auto function window. The default to ERRORALTERNATE is CONTINUE.

This new option is valid when duplexing a tape migration or recycling a tape migration. The syntax of the SETSYS DUPLEX command is shown in Figure 6-1.

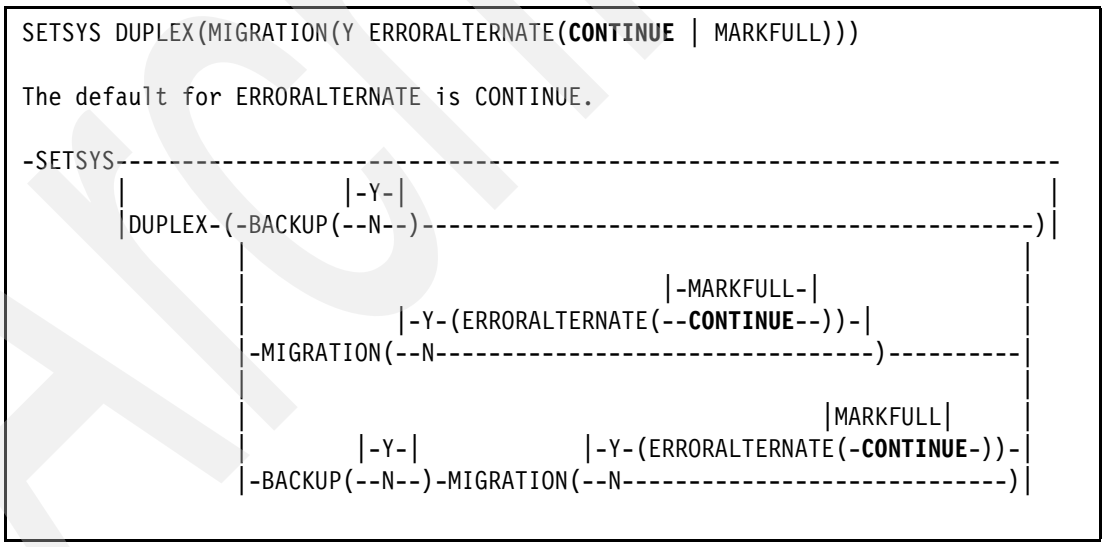


Figure 6-1 New keyword ERRORALTERNATE syntax

Remember that if you want this option working all the time, do not forget to write the command on the DFSMShsm PARMLIB member that you use, as a regular SETSYS command. This causes the command to be issued at DFSMShsm start up. It can be issued dynamically also.

Figure 6-2 shows the result of a dynamic command issue.

```
❶
STC18988   ARC0442I TAPE OUTPUT PROMPT FOR TAPECOPY=NO, DUPLEX
           ARC0442I (CONT.) BACKUP TAPES=NO, DUPLEX MIGRATION TAPES=NO
...
...
-F DFHSM64,SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(MARKFULL))
STC18988   ARC0100I SETSYS COMMAND COMPLETED
...
...
❷
STC18988   ARC0442I TAPE OUTPUT PROMPT FOR TAPECOPY=NO, DUPLEX
           ARC0442I (CONT.) BACKUP TAPES=NO, DUPLEX MIGRATION TAPES=(Y,
           ARC0442I (CONT.) ERRORALTERNATE=MARKFULL)
```

Figure 6-2 Dynamic command output

Figure 6-2 shows how SETSYS was before ❶ and after ❷ the command SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(MARKFULL)).

Figure 6-3 shows the effect of the command to DUPLEX migration tape. One data set is migrated, but two mounts occur due to the DUPLEX(MIGRATION(Y) option.

```
STC18988 00000290  IEC501A M
0B90,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.HMIGTAPE.DATASET
STC18988 00000090  IEC705I TAPE ON
0B90,TST029,SL,COMP,DFHSM64,DFHSM64,HSM.HMIGTAPE.DATASET,MEDIA3
STC18988 00000090 ARC0120I MIGRATION VOLUME TST029 ADDED, RC=0000 ❶, 863
     863 00000090  ARC0120I (CONT.) REAS=0000
STC18988 00000290  IEC501A M
0B91,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.COPY.HMIGTAPE.DATASET
STC18988 00000090  IEC705I TAPE ON
0B91,TST000,SL,COMP,DFHSM64,DFHSM64,HSM.COPY.HMIGTAPE.
               DATASET,MEDIA3
STC18988 00000090  IEC205I SYS00303,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME
LIST, 866
   866 00000090  DSN=HSM.HMIGTAPE.DATASET,VOLS=TST029,TOTALBLOCKS=133
STC18988 00000090  IEC205I SYS00304,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME
LIST, 867
   867 00000090  DSN=HSM.COPY.HMIGTAPE.DATASET,VOLS=TST000,TOTALBLOCKS=133
         00000290  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=DFHSM64 ,
ASID=0076.
STC18988 00000290  IEF234E K 0B90,TST029,PVT,DFHSM64,DFHSM64
STC18988 00000290  IEF234E K 0B91,TST000,PVT,DFHSM64,DFHSM64
```

Figure 6-3 Output of DUPLEX command

Message ARC0120I in Figure 6-3 tells you that, if the return code (RC=) was 0 (zero) ❶, an ADDVOL command has been processed for the type of volume *volser* specified in the message. If the volume record is successfully created or updated, the return code is zero.

When SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(MARKFULL))) has been specified, migration processing ignores the patch provided by APAR OA09928 (refer to

“OA09928” on page 465) that turns off SYNCDEVs as each data set is written to the migration alternate volume.

Note: PATCH OA0998, OW45264, and OW43224 deal with error handling on alternate duplex tapes. The PATCH to turn off the SYNCDEV was suggested to decrease the elapsed recycle processing time. With z/OS V1R8.0, it is no longer beneficial and no longer recommended.

Figure 6-4 shows the PATCH provided by APAR OA09928.

```
...
...
Customers can now suppress syncs on the alternate tape during duplex migration.
Syncs for the migration alternate tape can be turned off via:
PATCH .MCVT.+196 BITS(..1.....)

and back on via:

PATCH .MCVT.+196 BITS(..0.....)

The default is to do syncs on the alternate tapes.

This patch is not recommended for earlier technology tape drives such as the 3490 and
3590.

Prior APAR OW45264 describes a similar patch for turning off alternate tape syncs for
Recycle.  This can also dramatically improve performance with the 3592 -J tape drive.
...
...
```

Figure 6-4 PATCH from APAR OA09928

In this case, with the patch ON (see Figure 6-4) and SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(MARKFULL))), a new PDA trace is added, with text (‘APAR OW43224 is ignored’), so it is easily identified during a problem determination. For more information about APAR OW43224, see “OW43224” on page 466.

Figure 6-5 shows the PATCH provided by APAR OW45264.

```
...
...
Bypassing the SYNCDEV for the duplex alternate tape during Recycle is now
supported. This reduces the time Recycle takes by bypassing the SYNCDEV of the
duplex alternate tape after each dataset. To bypass the alternate tapes SYNCDEV
operation, enter the following PATCH command for your release:

For releases HDZ11D0 AND HDZ11E0:
PATCH .YGCB.+88 BITS(......1.)
.
For releases HDZ11F0, HDZ11G0, HDZ11H0, HDZ11J0 & HDZ11K0:
PATCH .YGCB.+C8 BITS(......1.)
.
To preserve the PATCH when HSM is restarted, place the PATCH command in the HSM
startup PARMLIB member ARCCMDxx.
.
To remove the patch and allow the SYNCDEV to be again occur on the alternate
volume,  enter the following PATCH command for your release and remove the
previous PATCH from ARCCMDxx:
.
For releases HDZ11D0 AND HDZ11E0:
PATCH .YGCB.+88 BITS(......0.)
For releases HDZ11F0, HDZ11G0, HDZ11H0, HDZ11J0 & HDZ11K0:
PATCH .YGCB.+C8 BITS(......0.)
.
If you had the following PATCH installed from APAR OW43224:
PATCH .MCVT.+295 BITS(....1...)
it should be removed at this time.
...
...
```

Figure 6-5 PATCH from APAR OW45264

To simulate a problem on the alternate tape, we issued a V unit,xxxx,FORCE on its unit tape drive that caused an interruption. See Figure 6-6 for log details.

```
00000090 *038 IEE800D CONFIRM VARY FORCE FOR B91 - REPLY NO OR YES
00000290  R 38,YES
00000090  IEE600I REPLY TO 038 IS;YES
00000090  IOS102I DEVICE 0B91 BOXED, OPERATOR REQUEST, WAS ASSIGNED
00000090  IEE793I 0B91     PENDING OFFLINE AND BOXED
00000090  IOS000I 0B91,**,SIM,**,**06,,**,TST000,DFHSM64
00000090  IEC020I 001-3,DFHSM64,DFHSM64,SYS00325,0B91,TST000,
00000090  IEC020I HSM.COPY.HMIGTAPE.DATASET
00000090  IEC020I NON-ACCEPTABLE ERROR
00000090  IEF524I 0B91 PENDING OFFLINE
```

Figure 6-6 VARY OFFLINE FORCE to simulate the tape problem

Figure 6-7 shows the DFSMShsm behavior after the VARY OFFLINE FORCE command.

```
00000090  IEA794I SVC DUMP HAS CAPTURED: 983
00000090  DUMPID=001 REQUESTED BY JOB (DFHSM64 )
00000090  DUMP TITLE=DFSMSDSS-DUMP ABEND=001,COMPID=DF175,ADRFTARR,JOBNAM
00000090             E=DFHSM64
00000090  ARC0421I MIGRATION VOLUME TST029 IS NOW MARKED FULL
00000090  ARC0443I MIGRATION VOLUME TST029 WITH ALTERNATE TST000 985
00000090  ARC0443I (CONT.) IS NOW MARKED FULL
...
...
00000090  IEC205I SYS00324,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME LIST, 990
00000090  DSN=HSM.HMIGTAPE.DATASET,VOLS=TST029,TOTALBLOCKS=11126
00000090  IEC215I 714-0C,IFG0200Z,DFHSM64,DFHSM64,SYS00325,0B91,TST000,
          HSM.COPY.HMIGTAPE.DATASET
00000090  IOS000I 0B91,**,SIM,**,**06,,**,TST000,DFHSM64
00000090  IOS000I 0B91,**,SIM,**,**06,,**,TST000,DFHSM64
00000090  IOS000I 0B91,**,SIM,**,**06,,**,TST000,DFHSM64
00000290  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=MHLRES2 , ASID=005B.
00000290  IEF234E K 0B90,TST029,PVT,DFHSM64,DFHSM64
00000290  IEF196I IGD104I DUMP.D070223.H23.SC64.DFHSM64.S00001
RETAINED,
00000290  IEF196I DDNAME=SYS00011
00000090  IEA611I COMPLETE DUMP ON DUMP.D070223.H23.SC64.DFHSM64.S00001 999
...
...
00000090  INCIDENT TOKEN: SANDBOX  SC64     02/23/2007 23:56:48
00000090   ERROR ID = SEQ00744 CPU00 ASID0074 TIME18.56.48.1
00000290  IEF196I IEF237I 8074 ALLOCATED TO IPCSDDIR
00000290  IEF234E K 0B91,TST000,PVT,DFHSM64,DFHSM64
00000090  IEF244I DFHSM64 DFHSM64 - UNABLE TO ALLOCATE 1 UNIT(S) 002
00000090         AT LEAST 1 OFFLINE UNIT(S) NEEDED.
00000090  IEF877E DFHSM64 NEEDS 1 UNIT(S) 003
00000090  FOR DFHSM64 SYS00333
...
...
STC19100 00000290 ❶IEF234E D 0B91,TST000,,DFHSM64,DFHSM64
STC19100 00000290  IEC501A M
0B90,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.HMIGTAPE.DATASET
...
...
STC19100 00000090  IEC705I TAPE ON❷
0B90,TST001,SL,COMP,DFHSM64,DFHSM64,HSM.HMIGTAPE.DATAS
                ET,MEDIA3
STC19100 00000090  ARC0120I MIGRATION VOLUME TST001 ADDED, RC=0000, 150
    150 00000090  ARC0120I (CONT.) REAS=0000
STC19100 00000290  IEC501A M
0B91,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.COPY.HMIGTAPE.DATASE
                T
STC19100 00000090  IEC705I TAPE ON
0B91,TST002,SL,COMP,DFHSM64,DFHSM64,HSM.COPY.HMIGTAPE.
                  DATASET,MEDIA3
```

Figure 6-7 Output of DFSMShsm behavior after VARY OFFLINE FORCE

In the sequence in Figure 6-7 on page 148 ❶, tape - TST000 - is demounted. What happens with the tape depends on what TAPE manager you are using in your environment and the respective policy you have adopted. After that, on ❷, DFSMShsm requests two new tapes to continue from where it had a problem.

Because of this new option on SETSYS DUPLEX, an additional keyword was added on the LIST TTOC command so that you can produce a list with only tapes that were closed due to the ERRORALTERNATE option. Figure 6-8 shows some optional parameters on LIST TTOC command.

```
 TTOC SELECT Optional Parameters:

_ _________________ __
  |_EMPTY___________|
  |_FULL____________|
  |_NOTFULL_________|
  |_ASSOCIATED______|
  |_NOTASSOCIATED___|
  |_ERRORALTERNATE_| ←   new option
```

Figure 6-8 Optional parameters from TTOC

Figure 6-9 is a sample of the LIST TTOC command with the new option ERRORALTERNATE.

```
LIST TTOC SELECT(ERRORALTERNATE) ODS('MHLRES4.HSM.OUTPUT')

- DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 17:05:15 ON 07/03/02 FOR SYSTEM=SC64

VOLSER UNIT    VOL  REUSE    VALID    PCT   VOL    RACF PREV   SUCC   NUM ONE  ALT   LIB   STORAGE
       NAME    TYPE CAPACITY BLKS     VALID STATUS      VOL    VOL    REC FILE VOL         GROUP
TST001 3590-1 ML2  00197300 00197270 100   FULL   NO   *NONE* *NONE* 001 YES  TST002 LIB1 SGLIB1
----- END OF - TAPE VOLUME TTOC  - LISTING -----
```

Figure 6-9 Output of LIST TTOC command

This enhancement only applies to DFSMShsm at the V1.8 or later level that have specified the option SETSYS DUPLEX with the ERRORALTERNATE(MARKFULL) option. If DFSMShsm is not at the V1.8 level, DFSMShsm continues to work as it did previously.

Message change

Message ARC0442I changed because of this improvement, including the new keywords. See Figure 6-10 for details.

```
 ARC0442I TAPE OUTPUT PROMPT FOR  TAPECOPY = x, DUPLEX BACKUP TAPES =x,
DUPLEX MIGRATION  TAPES = x,  ERRORALTERNATE= {CONTINUE | MARKFULL}
```

Figure 6-10 Output of changed message ARC0442I

For more information about DFSMShsm messages, refer to *z/OS MVS System Messages Vol 2 (ARC-ASA),* SA22-7632.

Coexistence and migration

Table 6-1 details coexistence and migration for SESTYS DUPLEX with the new ERRORALTERNATE option.

Table 6-1 Coexistence table for SESTYS DUPLEX with new option ERRORALTERNATE.

Element or feature	DFSMShsm
When change was introduced	z/OS V1R8.0
Applies to migration from	None
Timing	After the first IPL of z/OS V1R8.0
Target system hardware requirements	None
Target system software requirements	None
Other system (coexistence or fallback) requirements	None
Restrictions	None
System impacts	None

6.2 Recycle SYNCDEV at intervals

Prior to z/OS V1.8, using the option SETSYS DUPLEX caused DFSMShsm to issue a SYNCDEV after each successful processed data set for both primary and alternate tapes. This is a time-consuming operation. To avoid the SYNCDEV operation and improve performance, a patch was provided to turn it off.

With z/OS V1.8 the recycle process issues the SYNCDEV at intervals. It occurs at about each 20 data sets or 500 Mb, whichever comes first. So, even if you are duplexing, performance should improve on your process. After the last data set on the recycle input volume has been moved to the output tape volumes, a SYNCDEV is done regardless of any counts. This guarantees that all data from the input tape has been written to the recycle output tape (both original and alternate).

With this new function implemented, DFSMShsm continues to honor the existing patch to turn off all SYNCDEV, except in the case where DFSMShsm recycles a migration tape and SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(MARKFULL))) is specified. In this case, DFSMShsm issues a PDA trace entry with text ('APAR OW43224 is ignored') to identify this condition.

Note: The PATCH to turn off the SYNCDEV was suggested to decrease the elapsed recycle processing time. On z/OS V1R8.0 it is no longer beneficial and no longer recommended.

The specifics of how Recycle SYNCDEV at intervals works are:

- If you are working with NO duplexing:
  - Standard: SYNCDEV executes after 20 data sets or 500 Mb, whichever comes first.
  - Error: If an error occurs on the output volume, it is marked full, a new output volume is selected, and the process continues starting on the first data set after the last successful SYNCDEV on the previous tape.
- If you are working with duplexing and recycling a BACKUP volume:
  - Standard: SYNCDEV executes after each 20 data sets or 500 Mb, whichever comes first.

- Original volume error: If an output error occurs on the original volume, the result is that both original and alternate are marked full, a new set of tapes is selected, and recycle continues after the first data set after the last successful SYNCDEV on the previous tape.
- Alternate volume error: If an error occurs on the alternate volume, DFSMShsm continues to write on the original volume tape and the alternate is demounted. When it has completed writing the original tape, it marks the tape full and schedules a TAPECOPY to create a new alternate tape.

- If you are working with duplexing and recycling a MIGRATE volume:
  - Standard: SYNCDEV executes after each 20 data sets or 500 Mb, whichever comes first.
  - Original volume error: If an output error occurs on the original volume, the result is that both original and alternate are marked full, a new set of tapes is selected, and recycle continues after the first data set after the last successful SYNCDEV on the previous tape.
  - Alternate volume error: If an error occurs on the alternate volume, DFSMShsm works as defined on the ERRORALTERNATE parameter of the SETSYS DUPLEX(MIGRATON(Y)) option.
    - If ERRORALTERNATE(CONTINUE) is specified, the alternate tape is demounted and DFSMShsm continues write on the original tape. When DFSMShsm finishes writing the original tape, it schedules a TAPECOPY to create a new alternate volume.
    - If ERRORALTERNATE(MARKFULL) is specified, both the original and alternate tape are marked FULL. DFSMShsm recycle requests two new volumes and continues writing with the first data set after the last SYNCDEV on the previous tape.

Important: If you currently do calculations to track recycle times, take into consideration that now SYNCDEV works with a *block* of 20 data sets or 500 Mb, whichever comes first.

Coexistence and migration

The enhancement on SYNCDEV on z/OS V1.R8 made the patch to bypass it no longer beneficial. See Table 6-2 for details.

Table 6-2 Coexistence table for SYNCDEV enhancement

Element or feature	DFSMShsm
When change was introduced	z/OS V1R8.0
Applies to migration from	None
Timing	After the first IPL of z/OS V1R8.0
Is the migration action required?	No, but recommended because the patch is no longer beneficial
Target system hardware requirements	None
Target system software requirements	None
Other system (coexistence or fallback) requirements	None
Restrictions	None

Element or feature	DFSMShsm
System impacts	None

6.3 Migration scratch queue for non-VSAM data set

Prior to z/OS V1.8, during the migration of non-VSAM data sets, DFSMShsm used considerable time doing the scratch operation from the L0 volume.

With z/OS V1R8 DFSMShsm is able to create a new task under the migration queue when migrating from L0 to tape that does the scratch activity (from scratch queue requests), so the performance is improved, as the requests are now handled asynchronously. If you have multiple migrate tasks, you have this new task to each one.

This new process is applied when migrating conditions are:

- From L0 volume
- On command volume migration
- On primary space management
- On interval migration

And are not applied when:

- You are migrating a single data set.
- During L1 to L2 migration - secondary space management.

6.3.1 Individual data set restore for ARECOVER processing

Aggregate Backup and Recovery Support (ABARS) facilitates a point-in-time backup of a collection of related data in a consistent manner. This group of related data is defined to ABARS as an aggregate. The backup copies are created in a device-independent format. During backup processing, the data is packaged as a single entity in preparation for taking it off-site. This enables the recovery of individual applications in user-priority sequence.

ABARS has three components:

- Aggregate group definition: The aggregate group lists the selection data set names, instruction data set name, and the management class for the aggregate. This component is implemented through the Interactive Storage Management Facility (ISMF) and requires the storage management subsystem (SMS) address space to be active.

- Aggregate backup (ABACKUP): The ABACKUP command backs up a user-defined group (aggregate group) of data sets at that moment in time. This DFSMShsm function also requires the SMS address space to be active. ABACKUP is capable to get data sets from user tapes, ML1, ML2, SMS, and Non-SMS volumes. See a simplified Aggregate backup (ABACKUP) flow in Figure 6-11.

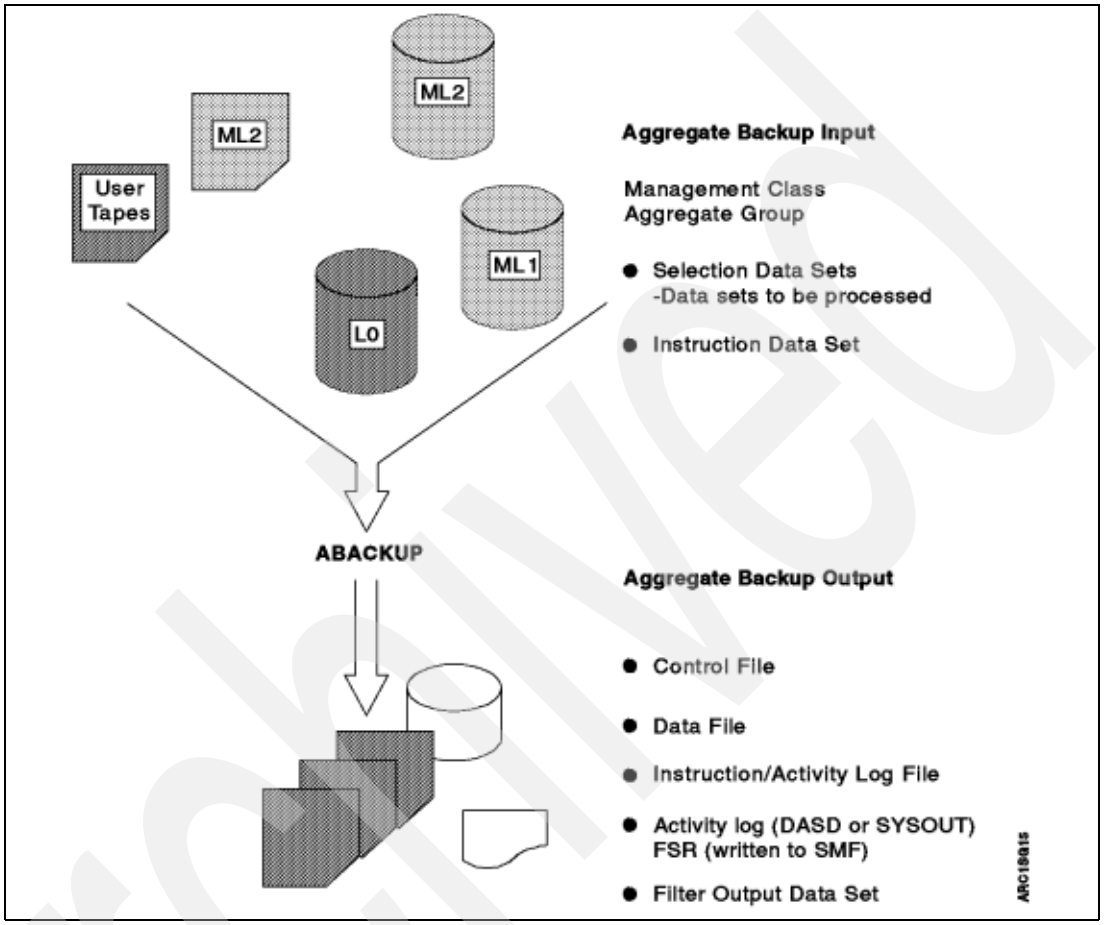


Figure 6-11 Simplified aggregate backup flow

- Aggregate recovery (ARECOVER): The ARECOVER command recovers data sets that were previously backed up by an aggregate backup. The SMS address space is not required, but it is highly recommended to simplify the recovery process. This address space is required to recover VSAM data sets in the ALLOCATE list and any extended format data sets. See a simplified aggregate backup (ARECOVER) flow in Figure 6-12.

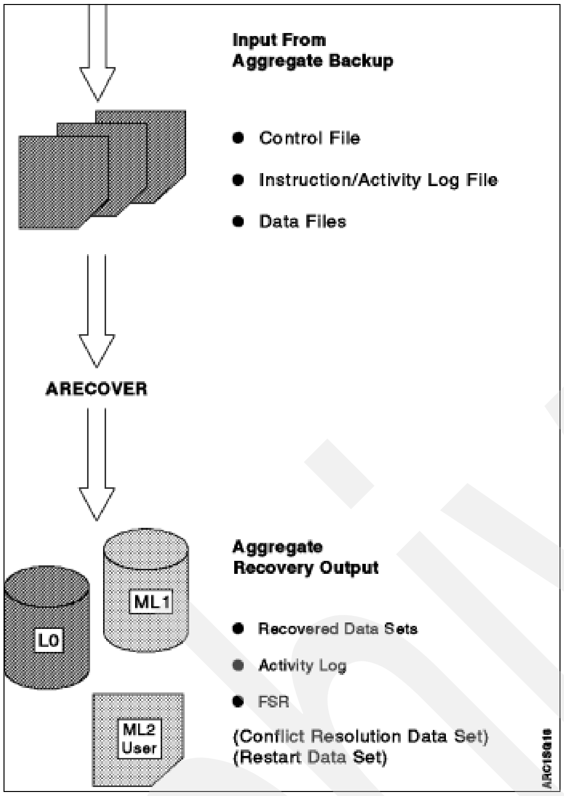


Figure 6-12 Simplified aggregate recover flow

Prior to z/OS V1R8 we were not able to select one single data set to recover. Now, we have additional keywords ONLYDATASET and LISTOFNAMES that make it possible.

Figure 6-13 shows the keywords that were added to ARECOVER in DFSMShsm 1.8.0.

```
>>_ARECOVER...
    ________________________________________________________________... ><
          |                                                          |
          |___ONLYDATASET___(_____NAME(dsname)___________________)_|
            |_ONLYDS______|    |                                |
                               |___LISTOFNAMES(listdsname)___|
                                 |_LON______________________|
```

Figure 6-13 ARECOVER new keywords

The optional ONLYDATASET parameter allows you to recover individual data sets from an aggregate backup version. You can specify a single fully qualified data set name using the NAME(dsname) parameter of ONLYDATASET to recover a particular data set.

To recover multiple data sets, you can supply the name of a data set that contains a list of data set names to be recovered on the LISTOFNAMES(listdsname) parameter of ONLYDATASET. The LISTOFNAMES data set must be a sequential data set, fixed-block,

record size of 80, and be cataloged. The LISTOFNAMES data set should contain one fully qualified data set name per record.

The abbreviation for ONLYDATASET is ONLYDS and the abbreviation for LISTOFNAMES is LON.

For more details about these new keywords, refer to *DFSMShsm Storage Administration Guide,* SC26-0421**.**

First you need a valid ABACKUP file. So, to accomplish that, you should have an ABARS environment prepared to do backup and recovery. We do not show in this book how to set up the ABARS function.

To reach ABARS panels under ISMF, go to the ISMF main panel and choose **9 Aggregate Group**. Check whether you have storage administrator privileges (on the main panel select option 0, then option 0, and then option 2 for Storage Admin) or you will be able to list only. After that, choose option 5, Abackup - Backup an Aggregate Group. Figure 6-14 and Figure 6-15 on page 156 show the AGGREGATE GROUP BACKUP panels used to do this preparation.

```
 Panel  Utilities  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                             AGGREGATE GROUP BACKUP                Page 1 of 2
 Command ===>

 CDS Name : ACTIVE

   Aggregate Group Name  . . MHLRES4

   Unit Name . . . . . .
❶  Processing Option . . . . 1                 (1=Verify, 2=Execute)
❷  Wait for Completion . . . N                 (Y or N)
   Stack / Nostack . . . . .                   (S=STACK, N=NOSTACK or blank)
   Optimize  . . . . . . . . 3                 (1 to 4 or blank)
   Delete Data Sets After Abackup  . . N       (Y or N)

   Filter Output Data Set Name (1 to 44 Characters)
❸  ===> 'MHLRES4.ABARS.OUTPUT'

   Enter "/" to select option
   Process only           ❹ /  LO        ML1        ML2        USERTAPE

 Use ENTER to Continue;
```

Figure 6-14 Output 1 of 2 about creating a AGGREGATE GROUP BACKUP

To prepare our backup, we choose in the first panel (Figure 6-14 on page 155) the parameters:

- ❶ and ❷, which are self-explanitory.
- ❸, which will be a file that will receive our output results.
- ❹, which is a slash before L0, so this specific backup selects from the filter list only data sets from L0.

At anytime, using the panel, you can choose Help (usually pre-set as PF1).

Attention: If you are a first-time TSO user, issue a PFSHOW ON/OFF or FKA ON/OFF to check your predefined PF Keys.

```
Panel  Utilities  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                            AGGREGATE  GROUP  BACKUP           Page  2 of  2
Command ===>


DFSMShsm Command and Processing Option:
NOWAIT ABACKUP MHLRES4 VERIFY
FILTEROUTPUTDATASET('MHLRES4.ABARS.OUTPUT')
OPTIMIZE(3) PONLY(L0)









Enter 1 to Submit DFSMShsm ABACKUP  COMMAND
Enter 2 to Save Generated ABACKUP  PARAMETERS

  Select Option  . . 1                          (1=SUBMIT, 2=SAVE)

Use ENTER to Perform Selection;
```

Figure 6-15 Output 2 of 2 about creating a AGGREGATE GROUP BACKUP

It is a good practice to use the VERIFY option, as shown in Figure 6-15. After the VERIFY, check the results and do any required adjustments. Run the VERIFY option until you get a RC 00 as a result. After that, change the option to EXECUTE (2). You can use the EXECUTE option directly.

Figure 6-16 on page 157 shows where you are going to find the ACTIVITY LOG from ABACKUP function.

```
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP
ARC6054I (CONT.) MHLRES4, AT 19:16:35,
ARC6054I (CONT.)  STARTED TASK = DFHSMABR.ABAR0135
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP MHLRES4 WILL BE
ARC6030I (CONT.) ROUTED TO
ARC6030I (CONT.) HSMACT.H2.ABACKUP.MHLRES4.D07051.T191635
COMMAND REQUEST 00000018 SENT TO DFSMSHSM
```

Figure 6-16 Support information where your dynamically allocated ACTIVITY LOG file is placed

Figure 6-17 shows you a sample of the ACTIVITY LOG from ABARS.

```
PAGE 0001  Z/OS DFSMSHSM 1.8.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER
07.051  19:16
ARC6000I ABACKUP MHLRES4 EXECUTE FILTEROUTPUTDATASET('MHLRES4.ABARS.OUTPUT')
OPTIMIZE(3) PONLY(LO )
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP MHLRES4, AT 19:16:35,
STARTED TASK = DFHSMABR.ABAR0135
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP MHLRES4 WILL BE ROUTED TO
HSMACT.H2.ABACKUP.MHLRES4.D07051.T191635
ARC6379I THE MANAGEMENT CLASS CONSTRUCTS USED IN THE AGGREGATE GROUP, MHLRES4,
ARE:
CLASS NAME : MCDB22
DESCRIPTION:
MGMT CLASS FOR DB2 TS
EXPIRATION ATTRIBUTES
 EXPIRE AFTER DAYS NON-USAGE: NOLIMIT
  EXPIRE AFTER DATE/DAYS     : NOLIMIT
...
...
ARC6004I 0079 ABACKUP DATA SET FILTERING IS COMPLETE. 18 OF 18 DATA SETS WERE
SELECTED: 0 FAILED SERIALIZATION AND 0
ARC6004I 0079 ABACKUP FAILED FOR
PAGE 0002  Z/OS DFSMSHSM 1.8.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER
07.051  19:16
ARC6004I 0079 ABACKUP                OTHER REASONS.
ARC6004I 0079 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE
SUCCESSFULLY PROCESSED  ❶
ARC6004I 0079 ABACKUP              MHLRES4.SC64.SPFLOG2.LIST
ARC6004I 0079 ABACKUP                MHLRES4.JCL.CNTL
ARC6004I 0079 ABACKUP MHLRES4.SC63.ISPF42.ISPPROF
...
...
ARC6071I VOLUMES USED FOR INSTRUCTION/ACTIVITY LOG FILE
MHLRES4.ABARS.OUTPUT.I.C01V0001 DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP MHLRES4 ARE:
 TST022 ❷
ARC6051I AN INSTRUCTION DATA SET WAS NOT SPECIFIED FOR AGGREGATE GROUP MHLRES4
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP MHLRES4, AT
19:18:08, RETCODE = 000
```

Figure 6-17 ACTIVITY LOG containing information about the current ABACKUP execution

In this ACTIVITY LOG (Figure 6-17 on page 157) you will find ❶ files that you saved at this time and the tape you used ❷, among other information.

The output data set that you chose on the first panel on the ABACKUP function (see Figure 6-14 on page 155, ❸) is similar to the one shown in Figure 6-18.

```
000001 FILTEROUTPUTDATASET LIST FOR AGGREGATE GROUP MHLRES4,
000002 VERSION 0001, CREATED ON 2007/02/20 AT 19:16:35
000003
000004 THE FOLLOWING DATA SETS WERE SPECIFIED IN THE INCLUDE LIST:
000005 MHLRES4.ABARS.OUTPUT                          VOLSER= MLD30C
000006 MHLRES4.ABARS.SELECT                          VOLSER= MLD30C
000007 MHLRES4.BRODCAST                              VOLSER= SBOX88
000008 MHLRES4.JCL.CNTL                              VOLSER= MLD20C
000009 MHLRES4.LIST                                  VOLSER= MLD50C
000010 MHLRES4.LOG.MISC                              VOLSER= SBOX88
000011 MHLRES4.SC63.ISPF42.ISPPROF                   VOLSER= MLD20C
000012 MHLRES4.SC64.SPFLOG1.LIST                     VOLSER= MLD20C
000013 MHLRES4.SC64.SPFLOG2.LIST                     VOLSER= MLD00C
000014 MHLRES4.SC64.SPFLOG3.LIST                     VOLSER= MLD30C
000015 MHLRES4.SC64.SPFTEMP2.CNTL                    VOLSER= SBOX88
000010 MHLRES4.SC63.ISPF42.ISPPROF                   VOLSER= MLD20C
...
...
000022 MHLRES4.TCPIP.PROFILE                         VOLSER= SBOX88
000023 A TOTAL OF 000018 DATA SETS SPECIFIED IN THE INCLUDE LIST WERE SELECTED.
```

Figure 6-18 Print panel from MHLRES4.ABARS.OUTPUT used on out test

Note: Not all panels were used from all available. On some panels, pressing DOWN, you have more options.

Now, let us do the ARECOVER choosing just one data set. Note that the first panel has a new line:

```
Recover Individual Data Sets. Y      (Y or N)
```

See ❹ in Figure 6-19 on page 159.

To get this panel, go to the ISMF main panel, then choose option **9. Aggregate Group** and after that option **6. Arecover - Recover an Aggregate Group**.

Figure 6-19 shows panel 1 of 8 from AGGREGATE GROUP RECOVER.

```
 Panel  Utilities  Scroll  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                           AGGREGATE GROUP RECOVER               Page 1 of 8
Command ===>

❶Abackup Control Dataset  . . 'MHLRES4.ABARS.OUTPUT.C.C01V0001'
                                        (1 to 44 Characters)
    Xmit . . . . . . . . . . . N        (Y or N)
    Stack / Nostack  . . . . .          (S=STACK, N=NOSTACK or blank)
  Aggregate Group Name . . . .
    Date . . . . . . . . . . .          (yyyy/mm/dd)
    Version  . . . . . . . . .          (1 to 9999)

❷Processing Option  . . . . . 2         (1=Prepare, 2=Verify, 3=Execute)
❸Wait for Completion  . . . . N         (Y or N)
  Target GDG Data Set Status            (A=ACTIVE, D=DEFERRED, R=ROLLEDOFF,
                                        S=SOURCE or blank)
  Volume Count . . . . . . . . .        (A=ANY or blank)
  Recover Instruction Data Set  . . N   (Y or N)
  Recover Activity Log  . . . . . . N   (Y or N)
❹Recover Individual Data Sets  . . Y    (Y or N)

Use ENTER to Continue; Use DOWN to View Additional Options;
```

Figure 6-19 ARECOVER panel 01 of 08

In Figure 6-19, we used the ABACKUP CONTROL DATASET ❶field to specify the name of the aggregate group backup control file to be used for aggregate recovery.

Other fields we choose on this test were:

- ❷ In a first step, we did a verify and after all checked, we choose **Execute**
- ❸ Usually we choose N here. If you choose Y for any reason, be advised that you will do a synchronous operation, so your session will be locked up until the end of the process.
- ❹ To demonstrate the new option, we choose Y.

Figure 6-20 shows panel 2 of 8 from AGGREGATE GROUP RECOVER.

```
 Panel  Utilities  Scroll  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                            AGGREGATE GROUP RECOVER               Page 2 of 8
Command ===>

ABACKUP CONTROL DATASET:  'MHLRES4.ABARS.OUTPUT.C.C01V0001'

Model Entity . . . .
                                        (1 to 44 Characters)
  Target Unit Name . .

  Target Utilization Percentage . .     (1 to 100)

❶Backup Migrated Data Sets . . . . Y    (Y or N)
❷New Migration Level . . . . . . . 3    (1=ML1, 2=ML2, 3=SOURCELEVEL)

Rename All to High Level Qualifier  . .          (1 to 8 characters)
❸Rename Selected Data Sets . . . . . . N         (Y or N)

❹Specify Conflict Resolution Option  . . N       (Y or N)

Use ENTER to Continue; Use UP to View Previous Options;
```

Figure 6-20 ARECOVER panel 2 of 8

In Figure 6-20 we choose ❶ Y for the Backup Migrated Data Sets field to indicate whether you want to cause a backup of all data sets that were successfully recovered to ML1 DASD volumes the next time automatic backup is run. ❷For New Migration Level choose 3 to indicate that migrated data sets are to be recovered to the same migration level as the original data set from the aggregate backup site. Specify N for ❸ Rename Selected Data Sets so on this test, your recovered data set is not renamed. Specify N ❹ for the Specify Conflict Resolution Option field to indicate whether you want to specify the action to be taken when a like-named data set conflict occurs.

Figure 6-21 shows panel 3 of 8 from AGGREGATE GROUP RECOVER. It lets you choose your unit name ❶ and tape volume ❷.

```
Panel  Utilities  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                            AGGREGATE GROUP RECOVER                Page 3 of 8
Command ===>

Abackup Control Dataset: 'MHLRES4.ABARS.OUTPUT.C.C01V0001'

  Unit Name  . . 3590-1 ❶

  Volumes   ===> TST022 ❷===>         ===>         ===>
            ===>         ===>         ===>         ===>
            ===>         ===>         ===>         ===>
            ===>         ===>         ===>









Use ENTER to Continue;
```

Figure 6-21 ARECOVER panel 03 of 08

In Figure 6-17 on page 157 you have information about the volume that was used to save your data. This information is duplicated here ❶ in Figure 6-22.

```
...
...
ARC6071I VOLUMES USED FOR INSTRUCTION/ACTIVITY LOG FILE
MHLRES4.ABARS.OUTPUT.I.C01V0001 DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP MHLRES4 ARE:
❶ TST022
...
...
```

Figure 6-22 The tape volume that was used in our demonstration

We jump to panel 7 of 8, but you have more options on panels that were not shown. It is your decision as to whether you need these extra options. To complete our test, we do not.

Figure 6-23 shows you panel 7 of 8 from AGGREGATE GROUP RECOVER. Here you can choose one specific data set to restore.

```
 Panel  Utilities  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              AGGREGATE GROUP RECOVER                Page 7 of 8
 Command ===>

 ABACKUP CONTROL DATASET:
 'MHLRES4.ABARS.OUTPUT.C.C01V0001'

 Specify Data Set Name for Individual Recovery:

❶ Single Data Set Name . . . . 'MHLRES4.SC64.SPFTEMP2.CNTL'
                                                    (1 to 44 Characters)
❷ List Of Names Data Set . . .
                                                    (1 to 44 Characters)









 Use ENTER to Continue;
```

Figure 6-23 ARECOVER panel 7 of 8

In Figure 6-23, you can choose from ❶ and specify one data set to recover (this is our test at this time) or from ❷ and specify the name of the data set that consists of names in the aggregate group to be used for recovery.

For the List Of Names Data Set you can specify at ❷ (Figure 6-23), there are the following rules:

- Must be a sequential file
- Can be fixed format, block size=80 and length=80 or fixed blocked format, any multiple of 80, length=80
- Must be cataloged
- Only fully qualified data set names accepted
- PDS member name not accepted
- GDG data sets - only fully qualified accepted (no relative generation here)
- Aliases not supported

Panel 8 of 8 from AGGREGATE GROUP RECOVER on Figure 6-24 shows you the set of commands that DFSMShsm executes through the ABARS task.

```
 Panel  Utilities  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                             AGGREGATE  GROUP  RECOVER            Page  8 of  8
Command ===>


DFSMShsm Command and Processing Option:
NOWAIT ARECOVER
EXECUTE
DATASETNAME('MHLRES4.ABARS.OUTPUT.C.C01V0001')
MIGRATEDDATA(SOURCELEVEL)
UNIT(3590-1) VOLUMES(...)
ONLYDATASET(NAME('MHLRES4.SC64.SPFTEMP2.CNTL'))




Enter 1 to Submit DFSMShsm ARECOVER COMMAND
Enter 2 to Save Generated ARECOVER PARAMETERS

  Select Option  . . 1                          (1=SUBMIT, 2=SAVE)

Use ENTER to Perform Selection;
```

Figure 6-24 ARECOVER panel 8 of 8

After this recovery, you see a message like the one in Figure 6-25 telling you about your success by RETCODE = 000. It will be shown on your current session and will be recorded in the recover log. For details about how the recover log name is created, see Table 6-3 on page 164.

```
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE
ARC6103I (CONT.) GROUP MHLRES4,
ARC6103I (CONT.)  USING CONTROL FILE DATA SET
ARC6103I (CONT.) MHLRES4.ABARS.OUTPUT.C.C01V0001,
ARC6103I (CONT.)  AT 20:24:37, RETCODE = 000
ARC1000I MHLRES4.ABARS.OUTPUT.C.C01V0001 ARECOVER PROCESSING ENDED
```

Figure 6-25 Message indicating ARECOVER PROCESSING ENDED

Note: Only the main panels from the ARECOVER were shown here. You can navigate there and see all other options.

You can receive other messages, indicating a failure or a non-ZERO return code. In this case, you can search on your environment the high level qualifier HSMACT. This is the default used by DFSMShsm DASD activity logs. It may be the case that in your installation, it had been changed using a patch. Check with your DFSMShsm administrator about that. In our case, the recover log was recorded in data set HSMACT.H2.ARECOVER.MHLRES4.D07051.T202329. See the Table 6-3 on page 164 about the conventions used for activity log names. If you want more details regarding this

point, see the manual *DFSMShsm Implementation and Customization Guide,* SC35-0418*,* under the topic “Controlling the Device Type for the Activity Logs.”

Table 6-3 provides details about the activity log data sets names conventions.

Table 6-3 Type of activity log name

ABARS	**mcvtactn.Hmcvthost.function.agname.Dyyddd.Thhmmss**
All other types	mcvtactn.Hmcvthost.function.Dyyddd.Thhmmss
mcvtactn	Activity log high-level qualifier (in our test, HSMACT (This is the default.))
H, D, and T Constants	
mcvthost	Identifier for the DFSMShsm host that creates these activity logs
function	ABACKUP, ARECOVER, CMDLOG, BAKLOG, DMPLOG, or MIGLOG
agname	Aggregate group name
yyddd	Year and day of allocation
hhmmss	Hour, minute, and second of allocation

Messages

With z/OS DFSMS V1.8 there are updated messages and a new one, ARC6411E. See details on Table 6-4.

Table 6-4 Changed messages

ARC6173E	Added LOCATE (indicating the data set not cataloged) and OBTAIN (indicating the data set received an OBTAIN error).
ARC6309E	Added LISTOFNAMES keyword and explanation.
ARC6310E	Added LISTOFNAMES keyword and explanation.
ARC6311I	Added LISTOFNAMES keyword and explanation and change the type code from “E” to “I”.
ARC6312I	Added LISTOFNAMES keyword and explanation and change the type code from “E” to “I”.
I ARC6411E	Data set data-set-name1 specified on the ONLYDATASET with the NAME parameter on ARECOVER command is not identified as being part of the aggregate group being recovered.

For more details about messages in Table 6-4, refer to *z/OS MVS System Messages Vol 2 (ARC-ASA),* SA22-7632.

Coexistence and migration

The new algorithm of ARECOVER on z/OS V1.8 made it possible to specify a single fully qualified data set name to recover from an AGGREGATE GROUP. See Table 6-5 for details.

Table 6-5 Coexistence table for ARECOVER new option

Element or feature	**DFSMShsm**
When change was introduced	z/OS V1R8.0
Applies to migration from	None

Element or feature	DFSMShsm
Timing	After the first IPL of z/OS V1R8.0
Is the migration action required?	None
Target system hardware requirements	None
Target system software requirements	None
Other system (coexistence or fallback) requirements	None
Restrictions	None
System impacts	None

6.4 New command ALTERPRI

There is a new DFSMShsm command, ALTERPRI, to alter the priority of queued requests. You can alter the priority of the following request types:

- ABACKUP
- ARECOVER
- BACKDS
- BACKVOL
- DELETE
- FRBACKUP
- FREEVOL
- FRRECOV
- MIGRATE
- RECALL
- RECOVER

You cannot alter the priority of BACKVOL CDS commands and requests that have already been selected for processing. You should use the ALTERPRI command to alter the priority of queued requests on an as-needed basis. You should not use this command as the primary means of assigning priority values to new requests.

There are two options of priority here:

- The HIGH parameter, which is the default, alters the specified request so that it has the highest priority on its respective queue.
- Conversely, the LOW parameter alters the request so that it has the lowest priority on its respective queue.

The mutually exclusive REQUEST, USER, or DATASETNAME parameters indicate which requests DFSMShsm should re-prioritize. Use the QUERY REQUEST command to determine the request number to issue on the ALTERPRI command. DFSMShsm re-prioritizes all queued requests that match the REQUEST, USERID, or DATASETNAME criteria specified on the ALTERPRI command.

To re-prioritize a recall request on the Common Recall Queue, issue the ALTERPRI command on the same host that originated the recall request.

Figure 6-26 shows ALTERPRI syntax.

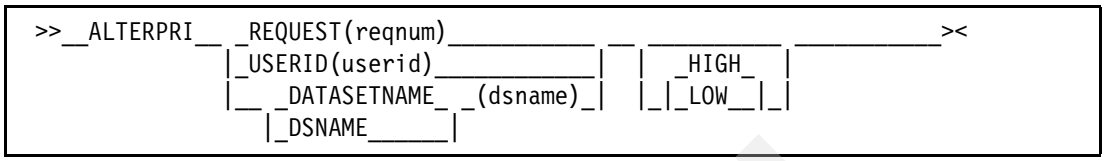


Figure 6-26 Output of ALTERPRI command keywords

The following command alters all requests for a particular data set so that they have the highest priority on their respective queues:

```
ALTERPRI DATASETNAME(dsname)
```

The following command alters all requests with a particular request number so that they have the highest priority on their respective queue:

```
ALTERPRI REQUEST(reqnum) HIGH
```

The following command alters all requests issued by a particular user so that they have the lowest priority on their respective queues:

```
ALTERPRI USERID(userid) LOW
```

Figure 6-27 shows the result of a query command, See that MHLRES4.TCPIP.PROFILE ❶ is in the last position in the recall queue.

```
 -F DFHSM64,Q REQUEST
STC20554   ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#1,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000211, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000000 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#2,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000212, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000001 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#3,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000213, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000002 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.TCPIP.PROFILE ❶,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000214, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000003 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
```

Figure 6-27 Output of command F DFHSM64,Q REQUEST before ALTERPRI

To demonstrate our test, we issue an ALTERPRI command and move MHLRES4.TCPIP.PROFILE ❶ (see Figure 6-27 on page 166) to the highest position. Figure 6-28 shows the results of the ALTERPRI command.

```
00000290  F DFHSM64,ALTERPRI REQUEST(00000214) HIGH
00000090  ARC0980I ALTERPRI REQUEST COMMAND STARTING
00000090  ARC0982I RECALL MWE FOR DATA SET MHLRES4.TCPIP.PROFILE 546
00000090  ARC0982I (CONT.) FOR USER MHLRES4, REQUEST 00000214, REPRIORITIZED TO
00000090  ARC0982I (CONT.) HIGH
00000090  ARC0981I ALTERPRI REQUEST COMMAND COMPLETED, RC=0000
```

Figure 6-28 Output of F DFHSM64,ALTERPRI command

After issuing the ALTERPRI command the file MHLRES4.TCPIP.PROFILE ❶ is now on the top.

Figure 6-29 shows the result of a QUERY command. Note that MHLRES4.TCPIP.PROFILE ❶ is the first one to recall.

```
 -F DFHSM64,Q REQUEST
STC20554   ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.TCPIP.PROFILE ❶,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000214, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000000 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#1,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000211, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000001 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#2,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000212, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000002 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#3,
           ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000213, WAITING TO BE
           ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000003 MWES AHEAD OF
           ARC1543I (CONT.) THIS ONE
STC20554   ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=2
```

Figure 6-29 Output of F DFHSM64,Q REQUEST after the ALTERPRI command

The first recall results after changing the queue priority are shown in Figure 6-30.

```
 ARC1000I MHLRES4.TCPIP.PROFILE  RECALL PROCESSING ENDED
 ARC0612I VOLUME MOUNT ISSUED FOR RECALL OR RECOVER OF MHLRES4.DUMP.OUT.#1
```

Figure 6-30 Output of first results after change the recall priority

For more details about this command, you can refer to *z/OS V1R8.0 DFSMS Storage Administration Reference (for DFSMShsm, DFSMSdss, DFSMSdfp)*.

Protecting ALTERPRI command

Each storage administrator command can be protected through the following fully qualified discrete FACILITY class profile:

```
STGADMIN.ARC.command
```

In this case, a security administrator can create the fully qualified, discrete profile to authorize this command to storage administrators:

```
STGADMIN.ARC.ALTERPRI
```

You can find more details like the entire command list and specific RACF profiles in the manual *DFSMShsm Implementation and Customization Guide,* SC35-0418.

Tip: Search for ALTERPRI in the *DFSMShsm Implementation and Customization Guide,* SC35-0418. You will find an item about protecting the DFSMShsm FACILITY.

Coexistence and migration

See Table 6-6 for details.

Table 6-6 Coexistence table for ALTERPRI new command

Element or feature	DFSMShsm
When change was introduced	z/OS V1R8.0
Applies to migration from	None
Timing	After the first IPL of z/OS V1R8.0
Is the migration action required?	None
Target system hardware requirements	None
Target system software requirements	None
Other system (coexistence or fallback) requirements	None
Restrictions	None
System impacts	None

7

DFSMShsm fast replication

The DFSMShsm fast replication function in DFSMS V1R8 allows you to dump fast replication backup copies to tape (through operator commands or automatic dump processing), and allows you to restore individual data sets from copy pool backup copies.

The following topics are covered:

- Fast replication review
- Backup and recovery of copy pools
- Tape support
- Data set recovery
- Reporting on the DFSMShsm fast replication environment
- Security for DFSMShsm fast replication
- Audit and error recovery

7.1 Fast replication overview

Point in time is the appearance of an almost instantaneous volume copy.

Fast Data Replication occurs so fast because it builds a map, with pointers, to the source volume tracks or extents. There is no longer a need to wait for the physical copy to complete before applications can resume their access to the data. Both the source and target data are available for read/write access almost immediately, while the copy process continues in the background. This process guarantees that the contents of the target volume are an exact duplicate of the source volume at that point in time. You can back up, recover to, and resume processing from that point in time.

Point-in-time copy is a mirroring technology that provides an instant copy of the data. It is a duplication of source files by means of mirrored disks. In other words, all data residing on disk is mirrored to another disk residing on the same storage subsystem. Most 24x7 database operations require a mirror solution. It has no impact on application performance, and provides immediate access to a copy of the mirrored data. This access to a copy of mirrored data, which can be obtained by various techniques with a minimal impact on mirroring, is one of the main returns on investment that many companies practice.

Point-in-time copy is also known as a T0 (time-zero) copy. After the copy is finished, the connection between source and mirror is split off. The mirror can be mounted to another host and can be used for backup and recovery purposes.

A Split of a Continuous Mirror is caused by the creation and maintenance of a mirror relationship between the source and the target volumes. This relationship occurs on-site, as opposed to a remote environment. The target volume is not accessible until a split or detach of the mirror relationship is initiated. A Split of a Continuous Mirror guarantees that the contents of the target volume are an exact duplicate of the source volume at the point in time where the split occurs. After the split is complete, you can back up, recover to, and resume processing from that point in time.

In the past, volume level point-in-time copies have been used almost exclusively as disaster recovery backups. With new software services that are available, these copies, clones, or mirrors can be used for job restart, data mining, and application testing.

Point-in-time hardware support

The DFSMShsm fast replication line item supports the volume-level FlashCopy® function of the IBM System Storage™ DS8000™, IBM TotalStorage Enterprise Storage Server® (ESS) DASD, SnapShot feature of the IBM RAMAC® Virtual Array (RVA) and STK Shared Virtual Array™ (SVA), and any other DASD that support FlashCopy APIs.

Note: Appropriate microcode levels may be required to activate these features on the storage device.

The process of creating a point-in-time copy has two phases. The first phase completes in a few seconds by constructing a map to describe the source volume.

At the completion of this first phase, the data has been cloned and both the source and target volumes are available for read and write access. From the user's perspective, the source and target volumes' contents are an exact duplicate at this point in time, even though actual copying of data has not actually been initiated.

The second phase consists of copying the physical source volume data to the target volume. The point-in-time copy is maintained and preserved by an on-demand copy of the data to the

target volume. When an update request has been issued for either a source or target volume data set that has not yet been copied to the target volume, the Fast Data Replication tool immediately copies the data from the source volume to the target volume before the update is applied. The possible situations are:

- Read request to the source volume

  The data is read from the source volume.

- Read request to the target volume

  If the data has already been copied from the source volume to the target volume, the data is read from the target volume. If the data has not yet been copied to the target volume, the data is read from the source volume.

- Write request to the source volume

  If the data has already been copied from the source volume to the target volume, this results in a normal write to source volume. However, if the data has not yet been copied from the source volume to the target volume, the data will be copied from the source volume to the target volume before the source volume update occurs.

- Write request to the target volume

  If the data has already been copied from the source volume to the target volume, the write request results in a normal write to the target volume. However, if the data has not yet been copied from the source volume to the target volume, the data will be copied to the target volume before the target volume update occurs.

7.1.1 FlashCopy

FlashCopy provides a point-in-time copy of a logical volume, with almost instant availability for the application of both the source and target volumes. Only a minimal interruption is required for the FlashCopy relationship to be established, so the copy operation can be initiated. The copy is then created in the background by the IBM TotalStorage ESS, with minimal impact on other ESS activities.

Note: FlashCopy is an optional feature that must be enabled in the DS8000.

FlashCopy: background copy

By default, FlashCopy performs a background copy. The background copy task makes a physical copy of all tracks from the source volume to the target volume. De-staging algorithms are used to efficiently manage the background ESS copy process. The background copy task runs at a lower priority than normal I/O on the ESS, so as not to affect the normal application I/O processing.

The ESS, using the metadata structure that was created during establish, keeps track of which data has been copied from the source to the target and manages the integrity of both copies. If an application wants to read some data from the target that has not yet been copied, the data is read from the source. Otherwise, the read can be satisfied from the target volume.

Before updating a not-yet-copied track on the source volume, the ESS performs an on demand copy of the track to the target volume. Subsequent reads to this track on the target volume are satisfied from the target volume.

Before updating a not-yet-copied track on the target volume, the ESS performs an on demand copy of this track to the target volume. This on demand activity is done

asynchronously after the host has received I/O completion (after the data is written in cache and NVS), so host I/O is not delayed.

After some time, when all tracks have been copied to the target volume, the FlashCopy relationship automatically ends unless the persistent FlashCopy option was specified.

As Figure 7-1 illustrates, a FlashCopy relationship goes through three stages:

- Establishing the relationship
- Copying the data
- Terminating the relationship

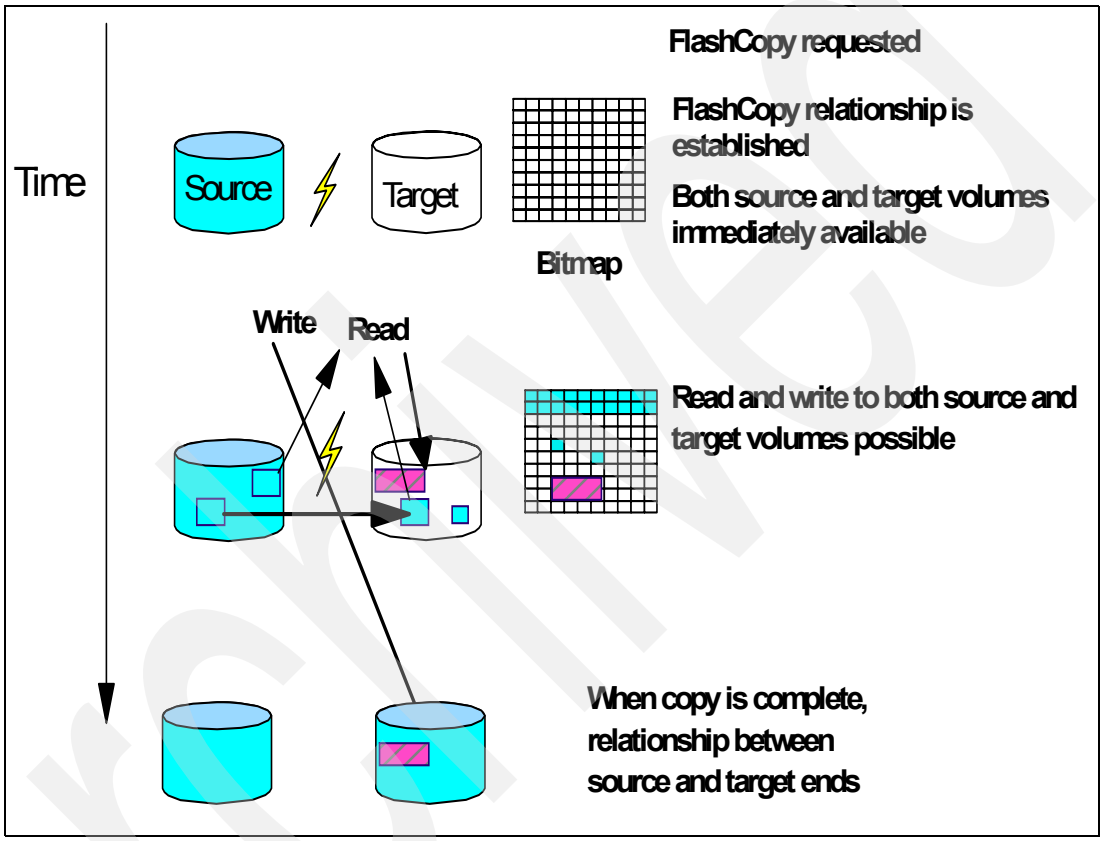


Figure 7-1 FlashCopy with background copy

FlashCopy: no background copy

When selecting not to perform the background copy, the relationship is established but the background copy task — of all source volume tracks — is not initiated. Only the source tracks that receive application updates are copied to the target. Before updating a track on the source volume, the ESS performs an on demand copy of the track to the target volume, thus preserving the T0 copy. Similarly, before updating a track on the target volume, the ESS performs an on demand copy of this track to the target volume.

A FlashCopy relationship that was established selecting no-background remains active until one of the following occurs:

- An explicit FlashCopy withdraw is done to terminate the relationship.
- All source device tracks have been copied on the target device because they were all updated.
- All target device tracks have been updated by user applications.

When a no-background copy FlashCopy relationship is terminated, the target volume is left in an indeterminate state. Some of the tracks on the volume may contain data from the source volume, while other tracks may contain residual data that was on the target volume before the copy. The volume should not be used in these conditions unless it is reformatted or used for another copy operation.

FlashCopy Version 1

FlashCopy V1 is invoked at volume level. The following considerations apply:

- The source and target volumes must have the same track format.
- The target volume must be as large as the source volume.
- The source and target volumes must be within the same ESS logical subsystem (LSS).
- A source and a target volume can only be involved in one FlashCopy relationship at a time.

As soon as a FlashCopy establish command is issued (either invoked by a TSO command, the DFSMSdss utility, the ANTRQST macro, or by means of the ESS Copy Services Web user interface (WUI)), the ESS establishes a FlashCopy relationship between the target volume and the source volume.

FlashCopy Version 2

FlashCopy Version 2 supports all of the FlashCopy V1 functions plus these enhancements:

- FlashCopy V2 can be used for data set copies, as well as volume copies.
- The source and target of a FlashCopy can be on different LSSs within a DS8000 or an ESS.
- Multiple FlashCopy relationships are allowed.
- Incremental copies are possible.
- Inband commands can be sent over PPRC links to a remote site.
- FlashCopy consistency groups can be created.

In addition, there has been a reduction in the FlashCopy establish times. Figure 7-2 compares the features of FlashCopy Version 1 and FlashCopy Version 2.

Restriction: When you are using FlashCopy Version 1, any subsequent FRBACKUP COPYPOOL command that is issued against the same copy pool or against another copy pool with a common storage group before all of the background copies are complete causes the command to fail. This restriction does not exist for FlashCopy Version 2 because a source volume can be in multiple concurrent relationships.

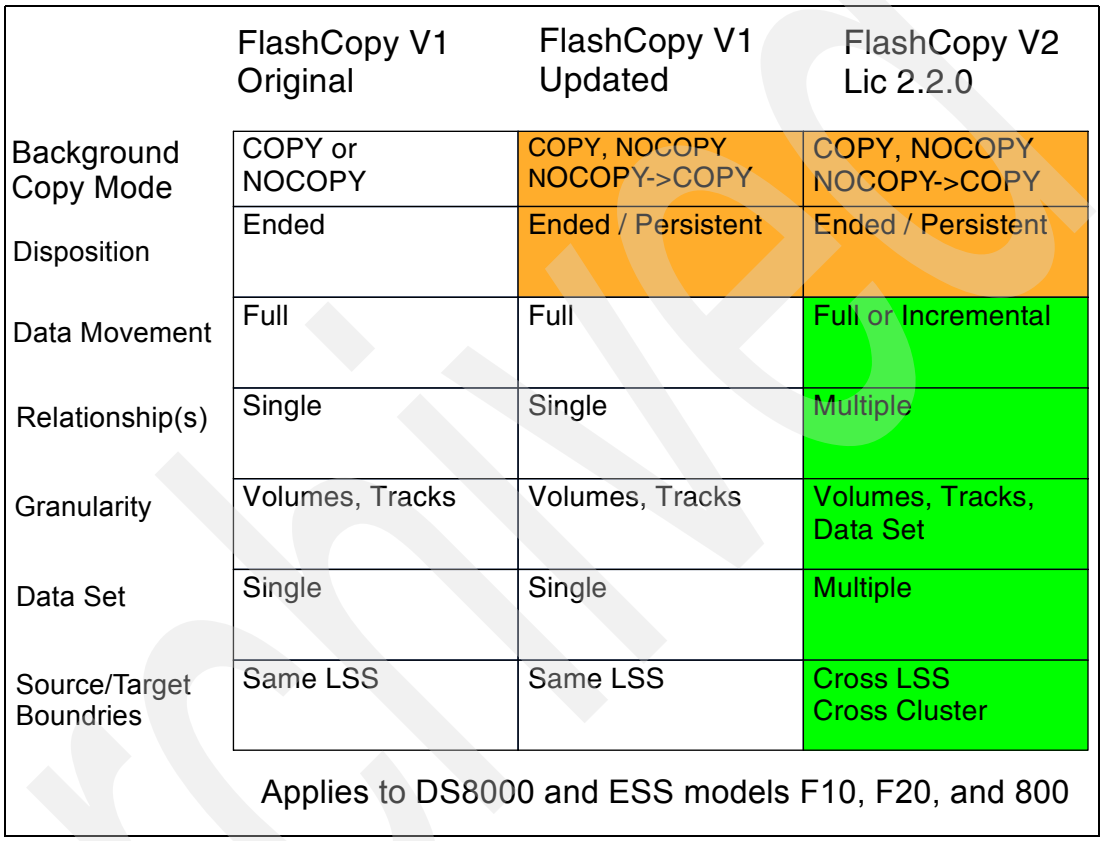


Figure 7-2 FlashCopy V1 and FlashCopy V2 comparison

7.1.2 DFSMShsm fast replication

DFSMShsm was enhanced in z/OS DFSMS V1.5 to manage full volume fast replication backup versions. DFSMShsm fast replication supports both FlashCopy Version 1 and FlashCopy Version 2.

Copy pool constructs

Copy pools are defined through a new SMS construct named copy pool. This construct enables customers to define which storage group should be processed collectively for fast replication functions and specifies how many backup versions of the pool DFSMShsm should maintain. When the maximum number of versions has been reached, the oldest version is rolled off.

SMS also introduced a copy pool backup storage group type that is used to define which volumes DFSMShsm may use as the target volumes of the fast replication backup versions.

DFSMShsm invokes the DFSMSdss COPY FULL function to perform a full volume copy of the data from the source devices in a copy pool to the target devices in a copy pool backup storage group.

DFSMShsm invokes the DFSMSdss physical data set COPY function to perform a fast replication recovery at the data set level from the target devices in a copy pool backup storage group to the source devices in a copy pool.

Related DFSMShsm commands

Three new DFSMShsm commands were being introduced in z/OS DFSMS V1.5 to support this new function:

- FRBACKUP

  Create a fast replication backup version for each volume in a specified copy pool. This command is not supported as part of automatic backup or dump. It is invoked by entering the command at the operator console, HSEND TSO command, and batch or by a program through the ARCHSEND macro interface.

- FRDELETE

  Delete one or more unneeded fast replication backup versions.

- FRRECOV

  Use fast replication to recover a data set, single volume, or a pool of volumes from the managed backup versions.

> **Note:** Recovery can be performed at the volume or copy pool level, and since z/OS DFSMS V1.8 at the data set level.

ISMF is updated to support these enhancements to SMS. The DFSMShsm LIST and QUERY commands and the ARCXTRCT macro are also modified to aid you in the use and monitoring of the fast replication backup versions.

DFSMSdss support

DFSMShsm invokes DFSMSdss to invoke fast replication support. The control is returned to DFHSMhsm after the fast replication has been initiated for all of the volumes (within minutes). If a volume failed, DFSMShsm messages are issued.

DFSMSdss was enhanced to support the following functions:

- Individual data set restore for extended format VSAM data sets, including extended format linear striped data sets, which are often used by DB2 customers.
- Provides a new FASTREPLICATION(REQUIRED) keyword.
- Callers of DFSMShsm may specify the new DFSMShsm keyword NOVTOCENQ to indicate no volume serialization (no reserve on the VTOC) to DFSMSdss.

> **Note:** The NOVTOCENQ must only be used when another utility, such as DB2, is providing the serialization. If it is used improperly, the results can be disastrous.

- Bypass security verification options during full volume copy.

What is a copy pool

A copy pool is an SMS construct that defines which storage groups should be processed collectively by the DFSMShsm function. A copy pool can contain the names of up to 256

storage groups that should be processed collectively for fast replication. An optional parameter can be used to specify how many backup versions of the pool DFSMShsm should maintain.

The name of this new construct may be up to 30 characters in length. The first character of the name can be an uppercase alphabetic or national character. Any of the remaining characters can be uppercase alphabetic, national, or numeric.

When defining a copy pool, the individual source pool storage group names are recorded. The actual volumes that are associated with each storage group are retrieved during subsequent processing. Consequently, the volumes that are processed during a function, such as fast replication backup, may or may not be the same volumes that are associated with the storage group at the time the copy pool was established. For example, if one or more volumes are added to a storage group within a copy pool, then those volumes are processed as part of the next fast replication backup automatically. There is no need to update the copy pool definition unless storage groups are to be added or removed from the pool.

An individual storage group may be associated with more than one copy pool. Due to implementation limitations, DFSMShsm must limit to 50 the number of copy pools with which a storage group may be associated.

As many as 85 backup versions may be maintained for each copy pool. Each version that is maintained requires a unique target volume for each source volume. If specifying five backup versions, then five target volumes must be available for each source volume that is associated with the copy pool. All target volumes must be available when the backup command is issued.

With ESS FlashCopy Version 1, the source and target volumes must be in the same Logical Subsystem (LSS), which is limited to a maximum of 256 total volumes.

With ESS FlashCopy Version 2, the source and target volumes may be in separate LSSs.

Restriction: An individual storage group can be associated with more than one copy pool, but with no more than 50. Although you can include a storage group in multiple copy pools, IBM recommends against doing so when FlashCopy Version 1 is used as the fast replication utility. Each individual copy pool with its associated background copies for all of the volumes in the common storage group must process completely before the next copy pool can be processed. This processing can take several hours and prevents the volumes in the non-common storage groups in those copy pools from having a timely backup created. (FlashCopy Version 2 and SnapShot do not have this restriction.)

Restriction: You can maintain as many as 85 backup versions for each copy pool. Each maintained version requires a unique target volume for each source volume. If you specify five dedicated versions, there must be five target volumes that are available for each source volume that is associated with the copy pool. All target volumes must be available when you issue the FRBACKUP COPYPOOL(cpname) PREPARE command. If the PREPARE function is not performed, the target volumes need not be available until the actual backup version is created.

Restriction: With ESS FlashCopy Version 1, the source and target volumes must be in the same LSS, which is limited to 256 total volumes. Thus, for three unique versions, you could have up to 64 source volumes, leaving 192 volumes available as target volumes. The source volumes within a storage group can span logical and physical subsystems.

What is a copy pool backup storage group

SMS storage group types consist of Pool, VIO, Dummy, Tape, Object, and Object Backup. A new storage group type, copy pool backup, is introduced to specify candidate target volumes where the copies of the pool storage groups are kept.

Volumes that are associated with the new copy pool backup storage group are for DFSMShsm use. SMS SCDS validation fails if a copy pool backup storage group is assigned for allocation by the SMS ACS storage group selection routines.

The SMS status of storage group and volume are used during data set allocation and end-of-volume extend. The SMS status of a copy pool backup storage group and volume are not used by DFSMShsm during a fast replication request. (For example, a status of Disable New or Disable All does not prevent DFSMShsm from processing the volume.) You may restrict DFSMShsm access to a volume in a copy pool backup storage group by varying the volume off-line to MVS.

A new field is provided in the storage group definition for POOL storage groups to specify the name of the associated copy pool backup storage group. Only pool storage groups may define associated copy pool backup storage groups. A pool storage group may specify its own copy pool backup storage group or specify a copy pool backup storage group that is shared by multiple pool storage groups.

For each source volume in a storage group to be copied, there must exist enough eligible target volumes in the copy pool backup storage group to satisfy the needs of the number of specified backup versions.

An eligible target volume must:

- Have the same track format as the source volume.
- Be the exact size of the source volume.
- For FlashCopy:

  Not also be a primary or secondary volume in an XRC or PPRC volume pair.

  - For Version 1, reside in the same LSS as the source volume.
  - For Version 1, at the time of the backup, not be in a FlashCopy relationship.
- For SnapShot:

  Reside in the same RVA/SVA as the source volume.

ISMF panels

The Interactive Storage Management Facility (ISMF) provides a series of applications for storage administrators to define and manage the SMS configuration. Figure 7-3 shows the ISMF primary option menu for storage administrators.

This primary option menu differs from the one that users see. In addition to the options found on the ISMF Primary Option Menu for users, the primary option menu for storage administrators includes:

- Storage Group
- Automatic Class Selection
- Control Data Set
- Library Management
- Data Collection

```
 Panel  Help
------------------------------------------------------------------------------
                 ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R8
Enter Selection or Command ===>

Select one of the following options and press Enter:
0  ISMF Profile             - Specify ISMF User Profile
1  Data Set                 - Perform Functions Against Data Sets
2  Volume                   - Perform Functions Against Volumes
3  Management Class         - Specify Data Set Backup and Migration Criteria
4  Data Class               - Specify Data Set Allocation Parameters
5  Storage Class            - Specify Data Set Performance and Availability
6  Storage Group            - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set         - Specify System Names and Default Criteria
9  Aggregate Group          - Specify Data Set Recovery Parameters
10 Library Management       - Specify Library and Drive Configurations
11 Enhanced ACS Management  - Perform Enhanced Test/Configuration Management
C  Data Collection          - Process Data Collection Function
L  List                     - Perform Functions Against Saved ISMF Lists
P  Copy Pool                - Specify Pool Storage Groups for Copies
R  Removable Media Manager  - Perform Functions Against Removable Media
X  Exit                     - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 7-3 ISMF Primary Option Menu

7.1.3 Preparing for fast replication

Your storage administrator must define copy pools and associated source and backup storage groups in order to use the DFSMShsm fast replication function.

The following steps should be used to define the necessary copy pool construct, copy pool backup storage group, and associate existing pool storage groups for fast replication support.

Defining copy pools

Define your copy pool definitions by selecting option **P Copy Pool** from the ISMF Primary Option Menu shown in Figure 7-3 on page 178.

The Copy Pool Application Selection Panel appears as shown in Figure 7-4. You can work with existing copy pools or define a new one by entering the copy pool name and selecting option **3 Define a Copy Pool**.

```
 Panel  Utilities  Help
------------------------------------------------------------------------------
                       COPY POOL APPLICATION SELECTION
Command ===>

To perform Copy Pool Operations, Specify:
  CDS Name . . . . SYS1.SMS.SCDS
                                (1 to 44 character data set name or 'Active' )
  Copy Pool Name   CP1                              (For Copy Pool List, fully
                                              or partially specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Copy Pools
     2. Display - Display a Copy Pool
     3. Define  - Define a Copy Pool
     4. Alter   - Alter a Copy Pool

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 7-4 Copy Pool Application Selection panel

The Copy Pool Define panel appears, as shown in Figure 7-5. Specify the dump classes to be used with this copy pool, and you can specify the number of backup versions that you want to keep of this copy pool. The default is two copies.

Note: DFSMShsm can keep up to 85 backup versions for each copy pool. We recommend keeping at least two versions. Before a new version is created, the oldest version is invalidated and the target volumes from that version are used as targets for the new version. If the new version fails for some reason, then only *n*-1 valid versions are available, where *n* is the number of requested versions. Maintaining two versions ensures that at least one valid version is always available.

We recommend that if *n* backups are required, *n*+1 should be kept.

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              COPY POOL DEFINE                 Page 1 of 4
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : CP1

To DEFINE Copy Pool, Specify:
  Description ==> COPY POOL 1
              ==>

  Auto Dump . . . N   (Y or N)        Dump Sys/Sys Group Name . . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .

  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 2       (0 to 85 or blank)



Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 7-5 Copy Pool Define panel (page 1 of 2)

Scroll down to reach the next page (shown in Figure 7-6) in order to specify the Storage Groups to include in the copy pool. You can specify up to 256 Storage Groups in a copy pool. Scroll down to pages 3 and 4 if needed.

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              COPY POOL DEFINE                  Page 2 of 4
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : CP1

To DEFINE Copy Pool, Specify:
  Storage Group Names:  (specify 1 to 256 names)
   ==> SG1
   ==>
   ==>
   ==>
   ==>
   ==>
   ==>
   ==>
   ==>
   ==>

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 7-6 Copy Pool Define panel (page 2 of 2)

Note: You can specify Storage Groups not yet defined in the SCDS, but this makes the SCDS invalid for activation. The Storage Groups specified must be defined in the SCDS before activating the configuration.

Defining copy pool backup storage groups

You now have to define a copy pool backup storage group. This storage group is designated as the target where the copies of the Pool Storage Groups will be kept. Start to define your copy pool backup definitions by selecting option **6 Storage Group** from the ISMF Primary Option Menu shown in Figure 7-3 on page 178. Figure 7-7 shows the Storage Group Application Selection panel.

```
 Panel  Utilities  Help
------------------------------------------------------------------------------
                     STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . SYS1.SMS.SCDS
                              (1 to 44 character data set name or 'Active' )
  Storage Group Name    CPBSG1            (For Storage Group List, fully or
                                          partially specified or * for all)
  Storage Group Type    COPY POOL BACKUP  (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                          OBJECT, OBJECT BACKUP, or TAPE)

Select one of the following options  :
  2  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 7-7 Storage Group Application Selection panel

The Storage Group Application Selection panel enables you to:

- Specify the storage group name for the copy pool backup.
- Specify the storage group type of copy pool backup.
- Associate your existing source pool storage groups with the copy pool backup storage group where you want to keep your copies.

After entering the chosen storage group name and storage group type of copy pool backup, select option **2 Define - Define a Storage Group**.

When the copy pool backup storage group is defined you should also define volumes in it. Here you need to know which pool storage group should be associated with this copy pool backup storage group and the number of versions that are to be kept for the pool storage group. The number of versions is defined in the related copy pool definition. For example, if your pool storage group contains 10 volumes and you want to keep two backup versions, then the copy pool backup storage group must contain 20 target volumes.

If the pool storage group has extended or overflow storage groups defined, you must also define target volumes for those. For example, if the pool storage group with 10 volumes has an overflow storage group with 5 volumes, you must define 30 target volumes in the copy pool backup storage group for two backup versions.

After defining the copy pool backup storage group, you must alter your existing pool storage groups to define the copy pool backup storage group where you want to keep their copies.

Start by selecting option **6 Storage Group** from the ISMF Primary Option Menu, as shown in Figure 7-3 on page 178. The Storage Group Application Selection Panel appears. Specify the desired storage group name and storage group type. Then select option **3 Alter** (Figure 7-8).

```
 Panel  Utilities  Help
------------------------------------------------------------------------------
                     STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . SYS1.SMS.SCDS
                              (1 to 44 character data set name or 'Active' )
  Storage Group Name    SG1               (For Storage Group List, fully or
                                          partially specified or * for all)
  Storage Group Type    POOL              (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                          OBJECT, OBJECT BACKUP, or TAPE)

Select one of the following options  :
  3  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 7-8 Storage Group Application Selection panel: Alter a Storage Group

The Pool Storage Group Alter panel appears. Specify the desired copy pool backup in the copy pool backup SG Name field (Figure 7-9).

```
 Panel  Utilities  Help
------------------------------------------------------------------------------
                          POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : SG1
To ALTER Storage Group, Specify:
 Description ==>
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . .
                                      Copy Pool Backup SG Name . . . CPBSG1
 Dump Class . . .                     (1 to 8 characters)
 Dump Class . . .                     Dump Class . . .
 Dump Class . . .                     Dump Class . . .
 Allocation/migration Threshold: High . . 85  (1-99)     Low  . .     (0-99)
 Guaranteed Backup Frequency  . . . . . .          (1 to 9999 or NOLIMIT)

 ALTER     SMS Storage Group Status . . . N   (Y or N)
Use ENTER to Perform Verification and Selection;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 7-9 Pool Storage Group Alter panel

A pool storage group can have only one copy pool backup storage group associated with it. Many pool storage groups can be associated with the same copy pool backup storage group, so we can have different versions of different pool storage groups in a copy pool backup storage group. DFSMShsm keeps control of the copies that you have.

From the Pool Storage Group Alter panel, you can also alter the SMS storage group status by typing Y in the ALTER SMS Storage Group Status field. There is also a change related to copy pool backup storage group SMS status — it can only have an SMS status of ENABLE or NOTCON.

The modifications in the source CDS must be made in a z/OS V1.5 or later system.

7.2 Backup and recovery of copy pools

We used for our tests an environment with one storage group per copy pool. The copy pool CP1 was made up of storage group SG1, which was linked with copy pool backup storage group CPBSG1, as shown in Figure 7-10.

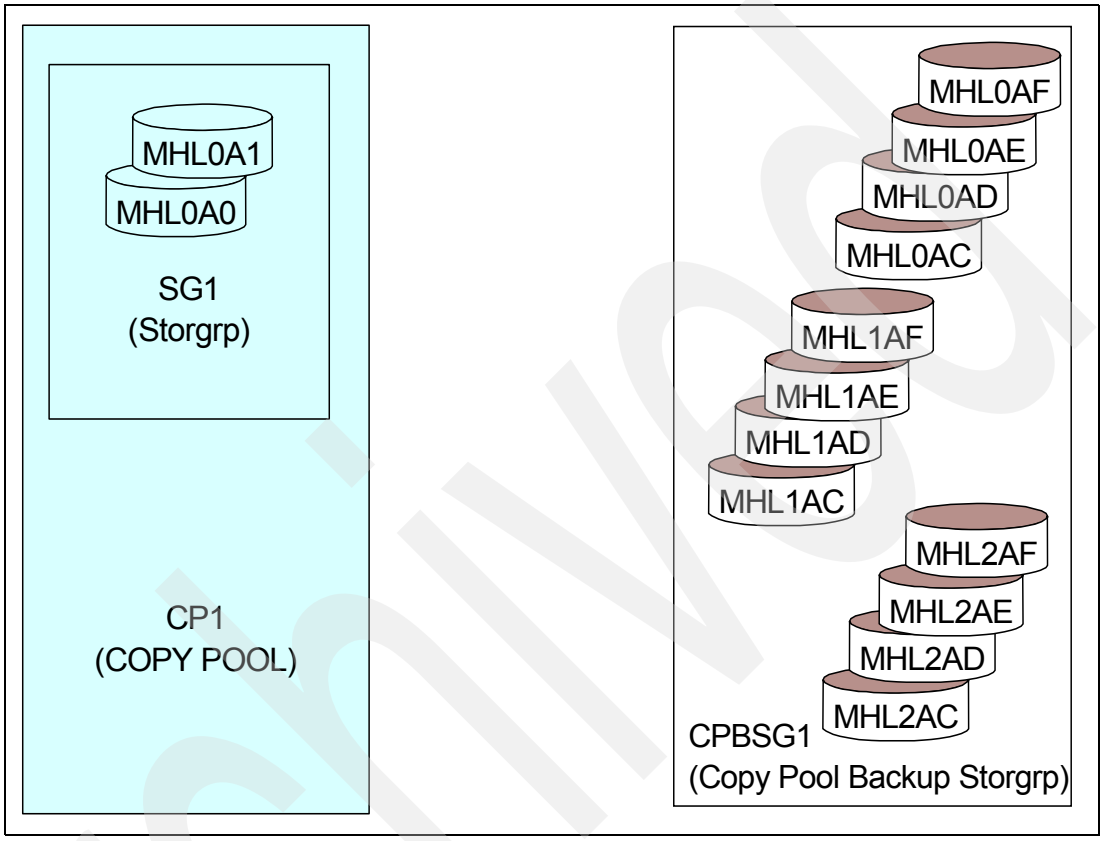


Figure 7-10 Lab environment

The copy pool was defined (see Figure 7-15 on page 192) to hold a maximum of two fast replication backup versions and to use dump class DCREDB18 for full volume dumps of target volumes.

In order to show all DFSMShsm messages we used a setting of SETSYS MSGLEVEL(FULL).

7.2.1 Creating a fast replication backup copy

Now we created a fast replication backup version of copy pool CP1 by using the command shown in Example 7-1.

Example 7-1 Requesting a new version of a fast replication backup

```
FRBACKUP COPYPOOL(CP1) EXECUTE TOKEN(EX1)
```

The DFSMShsm backup log contained the messages shown in Example 7-2. There were no messages in the dump log because full volume dump was not involved at this point.

Example 7-2 Backup log for fast replication of copy pool CP1 without copy to tape

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL CP1, AT 14:49:20 ON 2007/02/19, TOKEN='EX1'
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 14:49
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
 ARC0640I ARCFRTM -  COPY IDY(MHLOA0) ODY(MHLOAF) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(MHLOA1) ODY(MHL1AC) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2007.050 14:49:20 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2007.050 14:49:20 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
 ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2007.050 14:49:20 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHLOA1 WAS COPIED USING A FAST REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2007.050 14:49:20 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2007.050 14:49:20 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2007.050 14:49:20 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (002)-TOMI (02), VOLUME MHLOA0 WAS COPIED USING A FAST REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2007.050 14:49:21 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2007.050 14:49:21 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2007.050 14:49:21 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
 ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL CP1
 ARC1805I (CONT.) MHLOA0
 ARC1805I (CONT.) MHLOA1
 ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL CP1, AT 14:49:21 ON 2007/02/19, FUNCTION RC=0000,
  MAXIMUM VOLUME RC=0000
```

The resulting fast replication backup version can be shown by using the DFSMShsm LIST command (see Example 7-3).

Example 7-3 LIST command to show copy pool backup versions

```
HSEND LIST COPYPOOL(CP1)
```

The output of the LIST command shows a DUMPSTATE of NONE, which means that there is at this point neither a full volume dump in creation, nor available for this fast replication backup version (see Figure). For a detailed description of the possible contents of the new DUMPSTATE field see the explanations related to Figure on page 211.

```
1-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 16:34:37 ON
07/02/22 FOR SYSTEM=SC64

0COPYPOOL=CP1

 VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
   003       Y      2007/02/22      16:31:22   RECOVERABLE           NONE
 TOKEN(C)=C'EX3'
 TOKEN(H)=X'C5E7F3'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
```

Figure 7-11 Output of the LIST COPYPOOL(CP1) command

Note: The output of the LIST COPYPOOL command has changed significantly since DFSMShsm V1.7 and there are new keywords available in order to request more or less information for display (see Figure 7-17 on page 196 and Figure 7-18 on page 197 for more details).

There is a new column heading FASTREPLICATIONSTATE that has replaced the former VALID column to provide more detailed information about whether this version can be used for recovery by the FRRECOV command.

Possible contents of the field FASTREPLICATIONSTATE are:

RECOVERABLE The copy pool has a valid DASD copy version that can be used for recovery.

DUMPONLY The copy pool was defined with a number of 0 replicate backup versions, so the DASD copies that are currently available are only kept until the dump copies of a particular version have completed successfully.

FAILED The copy pool cannot be recovered by using the DASD copy of this version because the FRBACKUP did not complete successfully due to a failure or a WITHDRAW being done before all volumes in the copy pool were successfully copied.

NONE A DASD copy does not exist. For example, this status applies if only a dump copy exists for a particular backup version of a copy pool.

So far, we have created a fast replication backup version of copy pool CP1, which copied the contents of volumes MHL0A0 and MHL0A1 to volumes MHL0AF and MHL1AC (see Figure 7-12).

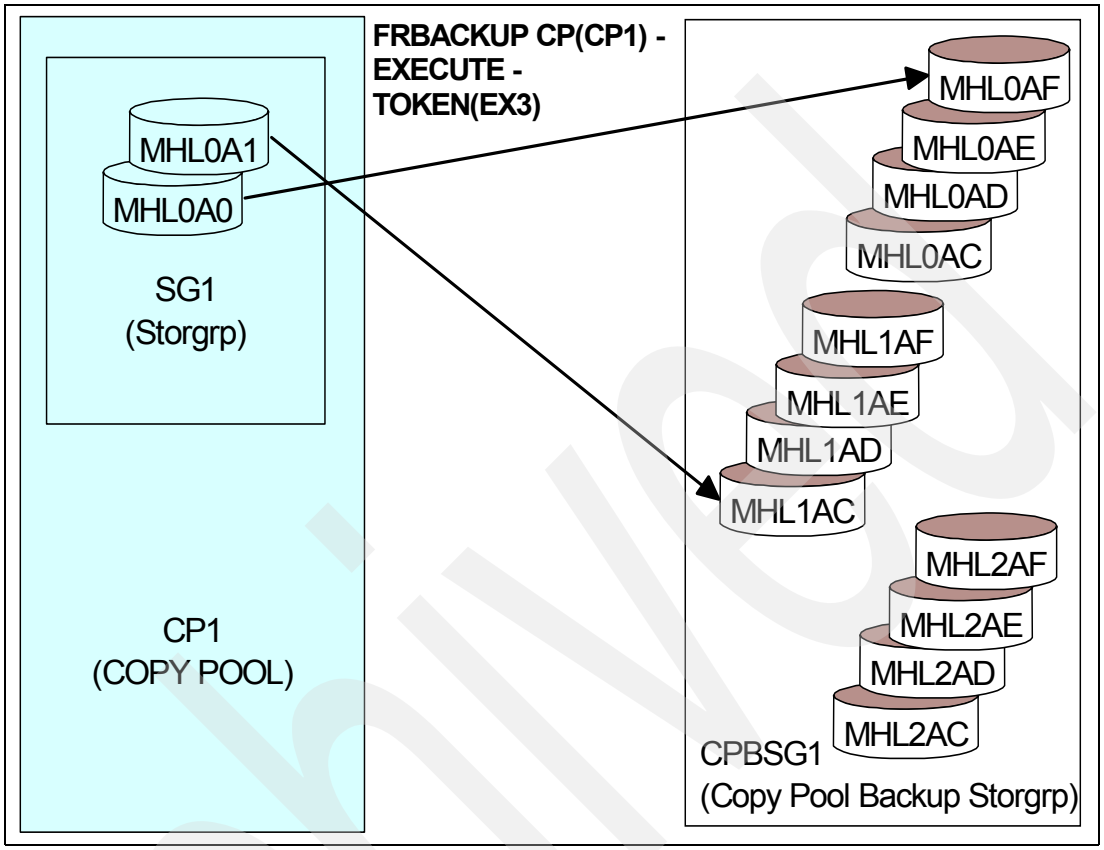


Figure 7-12 FRBACKUP without dump processing

7.2.2 Using fast replication backups for recovery of copy pools and volumes

You can use the FRRECOV command to recover copy pools or volumes from the fast replication backup versions.

Recovery of an entire copy pool is very easy to request and runs very quickly (at a similar speed as fast replication backup). Usually, you start with a command (as shown in Example 7-4) to make sure that no source volumes in the copy pool are in an existing FlashCopy relationship. If one or more volumes are in an existing FlashCopy relationship, the recovery fails.

Example 7-4 FRRECOV command that works at the copy pool level

```
HSEND FRRECOV CP(CP1) VERIFY(Y) GENERATION(0) FROMDASD
```

If one of the volumes cannot be serialized by DFSMShsm, the recovery of that particular volume fails. A very common reason for this kind of failure is an open ICF catalog. You can close the catalog by using the F CATALOG command, as shown in Example 7-5.

Example 7-5 MODIFY CATALOG,UNALLOCATE to close an open ICF catalog

```
F CATALOG,UNALLOCATE(UCAT.TESTFR)
```

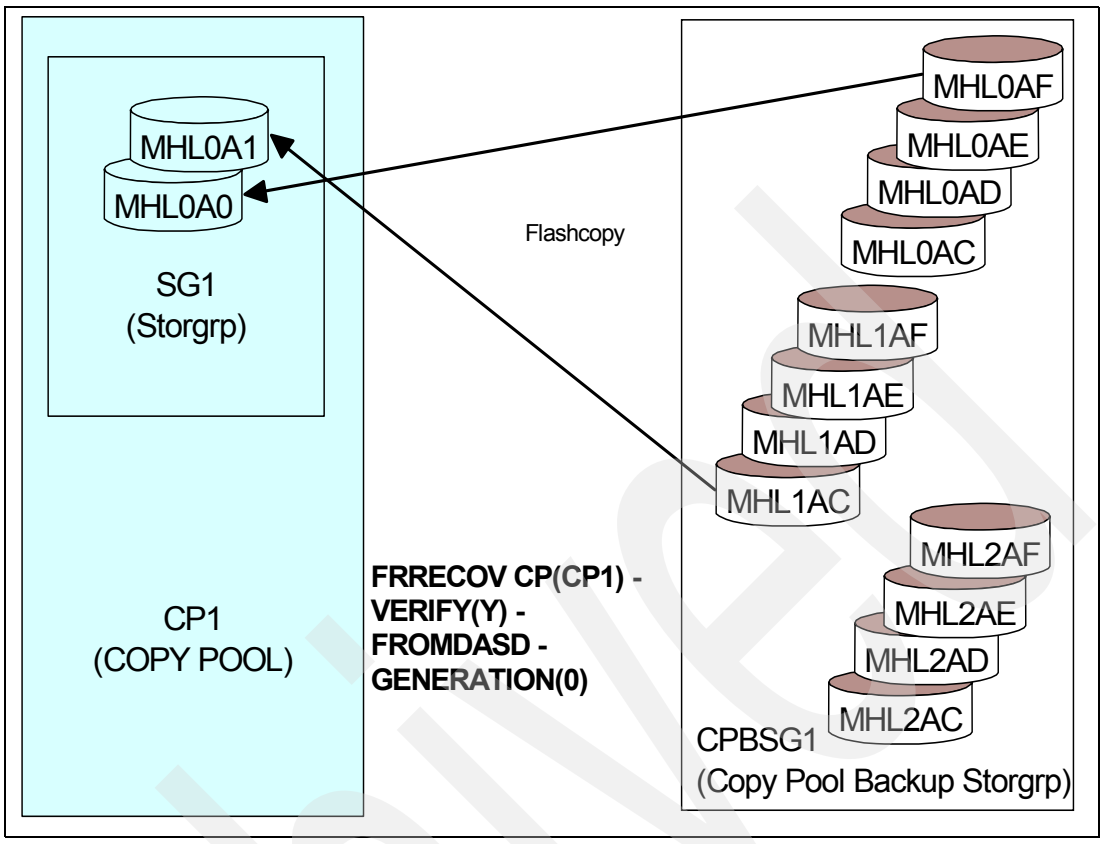


Figure 7-13 FRRECOV command at the copy pool level

The process during execution of command FRRECOV CP(CP1) FROMDASD is exactly the same as during execution of command FRBACKUP CP(CP1), but source and target volumes have swapped places with each other. An example for accompanying messages in the JESMSGLG output of DFSMShsm is shown in Figure 7-14.

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR  350
ARC1801I (CONT.) COPY POOL CP1, AT 19:05:10 ON 2007/03/16
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE  351
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST REPLICATION RECOVERY
ARC1805I (CONT.) OF COPY POOL CP1
ARC1805I (CONT.) MHL0A0
ARC1805I (CONT.) MHL0A1
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR  354
ARC1802I (CONT.) COPY POOL CP1, AT 19:05:11 ON 2007/03/16, FUNCTION
ARC1802I (CONT.) RC=0000, MAXIMUM VOLUME RC=0000
```

Figure 7-14 Messages during execution of FRRECOV CP(CP1) FROMDASD

If there is an open ICF catalog on one of the volumes, the resulting messages are as shown in Example 7-6.

Example 7-6 Excerpt from the backup log during execution of a copy pool recovery

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR COPY POOL CP1, AT 18:41:29 ON 2007/03/16
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.075 18:41
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
```

```
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
ARC0640I ARCFRTM -  COPY IDY(MHL0AF) ODY(MHL0A0) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM -  COPY IDY(MHL1AC) ODY(MHL0A1) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2007.075 18:41:30 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
 2007.075 18:41:30 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2007.075 18:41:30 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR306E (002)-SBRTN(01),
 UNABLE TO COPY THE VOLUME BECAUSE OUTPUT VOLUME MHL0A0 IS IN USE. TASK IS TERMINATED
ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2007.075 18:41:30 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2007.075 18:41:30 TASK COMPLETED WITH RETURN CODE 0008
ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2007.075 18:41:30 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHL1AC WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2007.075 18:41:30 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2007.075 18:41:30 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01),
 2007.075 18:41:30 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0008 FROM:
ARC0640I ARCFRTM -                          TASK    002
ARC0400I VOLUME MHL0A0 IS 00% FREE, 00000031 FREE TRACK(S), 000008 FREE CYLINDER(S), FRAG .198
ARC0401I LARGEST EXTENTS FOR MHL0A0 ARE CYLINDERS       8, TRACKS      120
ARC0402I VTOC FOR MHL0A0 IS 0090 TRACKS(04500 DSCBS), 04487 FREE DSCBS(99% OF TOTAL)
ARC0400I VOLUME MHL0A1 IS 00% FREE, 00000000 FREE TRACK(S), 000000 FREE CYLINDER(S), FRAG .000
ARC0401I LARGEST EXTENTS FOR MHL0A1 ARE CYLINDERS       0, TRACKS        0
ARC0402I VTOC FOR MHL0A1 IS 0090 TRACKS(04500 DSCBS), 04488 FREE DSCBS(99% OF TOTAL)
ARC1803E THE FOLLOWING 00001 VOLUME(S) FAILED DURING FAST REPLICATION RECOVERY OF COPY POOL CP1
ARC1803E (CONT.) MHL0A0
ARC1805I THE FOLLOWING 00001 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION RECOVERY OF COPY POOL CP1
ARC1805I (CONT.) MHL0A1
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR COPY POOL CP1, AT 18:41:30 ON 2007/03/16, FUNCTION RC=0008,
 MAXIMUM VOLUME RC=0008
```

After close of the catalog by using the command shown in Example 7-5 on page 188, the recovery process is resumed by a request at the volume level (see Example 7-7). We recommend this method because a new FRECOVER command at the copy pool level either fails due to existing FlashCopy relationships of some volumes or causes excessive and unnecessary copy activities for many volumes that were already successfully recovered (see the explanation of message ARC1803E in *z/OS MVS System Messages Vol 2 (ARC-ASA),* SA22-7632).

Example 7-7 FRRECOV command at the volume level

```
HSEND FRRECOV TOVOLUME(MHL0A0) FROMCOPYPOOL(CP1)
```

If the volume needed for recovery is currently associated with more than one copy pool, the FRRECOV TOVOLUME command must specify the FROMCOPYPOOL keyword in order to reference the appropriate copy pool's fast replication backup version. Example 7-8 shows the messages issued during volume level fast replication recovery in the DFSMShsm backup log.

Example 7-8 Excerpt from the backup log during execution of a fast replication recovery at the volume level

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR VOLUME MHL0A0, AT 18:52:47 ON 2007/03/16
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.075 18:52
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  COPY IDY(MHL0AF) ODY(MHL0A0) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2007.075 18:52:47 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
```

```
ARC0640I ARCFRTM - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR006I (001)-STEND(01), 2007.075 18:52:47 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (001)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (001)-TOMI (02), VOLUME MHL0AF WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (001)-STEND(02), 2007.075 18:52:48 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (001)-CLTSK(01), 2007.075 18:52:48 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2007.075 18:52:48 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC0400I VOLUME MHL0A0 IS 00% FREE, 00000018 FREE TRACK(S), 000000 FREE CYLINDER(S), FRAG .823
ARC0401I LARGEST EXTENTS FOR MHL0A0 ARE CYLINDERS       0, TRACKS        6
ARC0402I VTOC FOR MHL0A0 IS 0090 TRACKS(04500 DSCBS), 04487 FREE DSCBS(99% OF TOTAL)
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR VOLUME MHL0A0, AT 18:52:48 ON 2007/03/16, FUNCTION RC=0000,
 MAXIMUM VOLUME RC=0000
```

7.3 Tape support

Tape support of fast replication enhances DFSMShsm to manage full volume dumps of the DASD volumes that are holding the replication backup data. The full volume dumps can be created:

- Automatically as part of the automatic dump process
- By command as part of the fast replication backup process
- By command in a separate process at any time after a replication backup version of a copy pool was created

The resulting dump copies provide the same look and feel as dump generations that are being created by dumping the volumes of the storage groups of a copy pool directly.

There is also available a request for fast replication backups that are processed by using the NOCOPY option of FlashCopy. A DASD backup version created by this option cannot be used for recovery of the copy pool and is only maintained until the target volumes have been dumped to tape. By using this option you might need fewer volumes in the backup copy pool storage groups.

7.3.1 Dump of target volumes during autodump

Automatic dump was enhanced with z/OS DFSMShsm 1.8 to process storage groups defined to automatic dump enabled copy pools.

Prerequisites

In order to enable a copy pool for automatic dump processing, Auto Dump must be set to Y and at least one dump class must be specified (see Figure 7-15).

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              COPY POOL ALTER                  Page 1 of 4
 Command ===>

 SCDS Name  . . : SYS1.SMS.SCDS
 Copy Pool Name : CP1

 To ALTER Copy Pool, Specify:
   Description ==> COPY POOL 1
               ==>

   Auto Dump . . . Y   (Y or N)        Dump Sys/Sys Group Name . . .
   Dump Class  . . DCREDB18            Dump Class  . .
   Dump Class  . .                     Dump Class  . .
   Dump Class  . .

   Number of Recoverable DASD Fast
    Replicate Backup Versions  . . . . 2      (0 to 85 or blank)



 Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 7-15 Definition of copy pool CP1

You can request system affinity as with storage groups by entering the name of a system or a system group specified in the System or Sysgrp fields in the base configuration of the SMS configuration into the field Dump Sys/Sys Group Name.

You can use any dump class definition for copy pools. For our tests we have defined a dump class by using the command shown in Example 7-9.

Example 7-9 DEFINE DUMPCLASS for use with fast replication tape support

```
HSEND DEFINE DUMPCLASS(DCREDB18 AUTOREUSE +
                               DATASETRESTORE FREQUENCY(7) +
                               FRDUMP(REQUIRED) +
                               FRRECOV(AFM(YES)) +
                               RETPD(356) +
                               UNIT(3590-1) +
                               STACK(30) +
                               VTOCCOPIES(3) +
                     DISPOSITION('FASTR TARGET TO TAPE'))
```

VTOC copies are not required to recover a data set from a dump tape generated from a fast replication backup copy. Also, DATASETRESTORE is not required to be specified in the dump class. If a dump class is only used for dump copies created from fast replication backup copies, and if you plan to use the FRRECOV for individual data sets, then you should

consider creating VTOCCOPIES to enhance your options for recovery (see “Recovery of data sets from a fast replication dump copy” on page 224).

While not being required for any technical reason, it might be a good idea to create new dump classes for fast replication dump processing. Unique dump classes for copy pools ensure that copy pool volumes are dumped as a set, as each dump volume will always contain only dump copies that were created using the same dump class during dump volume stacking.

There are new keywords available for definition of a dump class that is intended for use with copy pools:

FRDUMP(REQUIRED) The default setting of FRDUMP(REQUIRED) indicates that every volume in a copy pool must have been successfully dumped before the DASD backup version that is being dumped is replaced with a more recent backup copy.

FRDUMP(OPTIONAL) The alternative setting allows you to replace the DASD backup version of a copy pool even if one or more volumes were not successfully dumped.

FRRECOV(AFM(YES)) The default setting of FASTREPLICATIONRECOVER specifies that a dump class copy is available for recovery without needing to be specified on the FRRECOV command.

FRRECOV(AFM(NO)) The alternative setting specifies that a dump class copy must be explicitly referenced by a FRRECOV command in order to be used.

Automatic dump (phase 2)

This phase is often referred to as volume function in contrast to phases 1 (deletion of expired dump copies) and 3 (deletion of excess dump VTOC copies), which are so-called level functions because they are executed on a primary host only. A complete description of what happens in these phases is contained in the manual *DFSMShsm Storage Administration Guide,* SC26-0421. In this section we emphasize only the changes that are related to fast replication backup tape support.

1. At the beginning of this phase, DFSMShsm obtains volume lists from storage groups defined to automatic dump enabled copy pools.
2. Volumes of copy pools are considered candidates if the frequency requirement for the dump class to which the copy pool belongs has been met.
3. Volumes of copy pools are considered candidates *even* if they were processed by automatic dump within the last 14 hours.
4. If a DASD fast replication backup copy generation 0 exists and has not previously been successfully dumped, automatic dump uses the dump classes specified in the copy pool to determine whether the frequency requirement is being met. If the generation 0 dump copy is partially complete, automatic dump skips all volumes that were successfully processed in the dump classes previously attempted.

   Note: Automatic dump never takes into account other fast replication backup copies than generation 0. In case you need to create a dump copy of generation 1 to 84 consider using the FRBACKUP CP(..) DUMPONLY command.

5. The automatic dump settings (Y/N) of the source volumes’ storage groups are ignored by automatic dump when it is processing a copy pool.

6. Volumes associated with the same set of eligible dump classes, of which at least one enables dump stacking (parameter STACK(n) in dump class definition requests n > 1) will be always processed at first. Within those volumes, the order is:
   a. Affinity copy pool volumes
   b. Non-affinity copy pool volumes
   c. Affinity SMS volumes
   d. Non-affinity SMS volumes
   e. Non-SMS volumes
7. Volumes associated with copy pools are not stacked with non-copy pool volumes.

   Note: If you want to make sure that only volumes of one copy pool will be stacked on the same tape volume you should use different dump classes for each copy pool.

8. After all volumes eligible for stacking have been dumped, the remaining volumes are dumped, with DFSMShsm selecting the order of processing as listed above.

In Example 7-10 you can see the messages that were issued in the JESMSGLG data set of DFSMShsm during automatic dump. The new messages ARC1841I and ARC1842I are issued for each copy pool being processed during automatic dump.

Example 7-10 Excerpts from the JESMSGLG output of DFSMShsm during automatic dump

```
17.10.00 STC18988  ARC0620I AUTOMATIC DUMP STARTING
 17.10.00 STC18988  ARC0570I AUTOMATIC DUMP FOR ALL SMS MANAGED VOLUME(S)  555
    555             ARC0570I (CONT.) TERMINATED, RC=17 REASON=0
 17.10.00 STC18988  ARC1841I AUTOMATIC DUMP STARTING FOR COPY POOL CP1
 17.10.00 STC18988  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  557
    557             ARC0622I (CONT.) MHLOA1(SMS) AT 17:10:00 ON 2007/02/22, SYSTEM SC64,
    557             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 17.10.00 STC18988  ARC0728I VTOC FOR VOLUME MHLOA1 COPIED TO DATA SET  558
    558             ARC0728I (CONT.) HSM.DUMPVTOC.T223116.VMHLOA1.D07053 ON VOLUME SBXHS4
 17.10.00 STC18988  IEC501A M 0B90,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHLOA1.D07053.T223116
 17.10.40 STC18988  IEC705I TAPE ON 0B90,TST025,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHLOA1.D07053.T223116,MEDIA3
 17.10.41 STC18988  ARC0120I DUMP VOLUME TST025 ADDED, RC= 0, REAS= 0
 17.15.47 STC18988  IEC205I SYS00023,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME LIST,  599
    599             DSN=HSM.DMP.DCREDB18.VMHLOA1.D07053.T223116,VOLS=TST025,
    599             TOTALBLOCKS=50048
 17.15.47 STC18988  ARC0637I DUMP COPY OF VOLUME MHLOA1 COMPLETE,  601
    601             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/13, DISPOSITION= FASTR
    601             ARC0637I (CONT.) TARGET TO TAPE
 17.15.47 STC18988  ARC0623I FULL VOLUME DUMP OF VOLUME MHLOA1 ENDING AT  602
    602             ARC0623I (CONT.) 17:15:47, PROCESSING SUCCESSFUL
 17.15.47 STC18988  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  603
    603             ARC0622I (CONT.) MHLOA0(SMS) AT 17:15:47 ON 2007/02/22, SYSTEM SC64,
    603             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 17.15.48 STC18988  ARC0728I VTOC FOR VOLUME MHLOA0 COPIED TO DATA SET  604
    604             ARC0728I (CONT.) HSM.DUMPVTOC.T223116.VMHLOA0.D07053 ON VOLUME SBXHS4
 17.21.16 STC18988  IEC205I SYS00023,DFHSM64,DFHSM64,FILESEQ=2, COMPLETE VOLUME LIST,  624
    624             DSN=HSM.DMP.DCREDB18.VMHLOA0.D07053.T223116,VOLS=TST025,
    624             TOTALBLOCKS=49976
 17.21.17 STC18988  ARC1842I AUTO DUMP HAS COMPLETED FOR COPY POOL CP1, AT  626
    626             ARC1842I (CONT.) 17:21:17 ON 2007/02/22, MAXIMUM VOLUME RC=0000
 17.21.17 STC18988  IEF234E K 0B90,TST025,PVT,DFHSM64,DFHSM64
 17.21.18 STC18988  ARC0637I DUMP COPY OF VOLUME MHLOA0 COMPLETE,  628
    628             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/13, DISPOSITION= FASTR
    628             ARC0637I (CONT.) TARGET TO TAPE
 17.21.18 STC18988  ARC0623I FULL VOLUME DUMP OF VOLUME MHLOA0 ENDING AT  629
    629             ARC0623I (CONT.) 17:21:18, PROCESSING SUCCESSFUL
 17.21.18 STC18988  ARC0621I AUTOMATIC DUMP ENDING
```

Refer to Figure 7-16 to see what happens with copy pool CP1 in our test environment during autmatic dump. As DFSMShsm always uses the DUMPCONDITIONING keyword when calling DFSMSdss for the full volume copies during fast replication backup processing, the dump generations appear to be made directly from the source volumes.

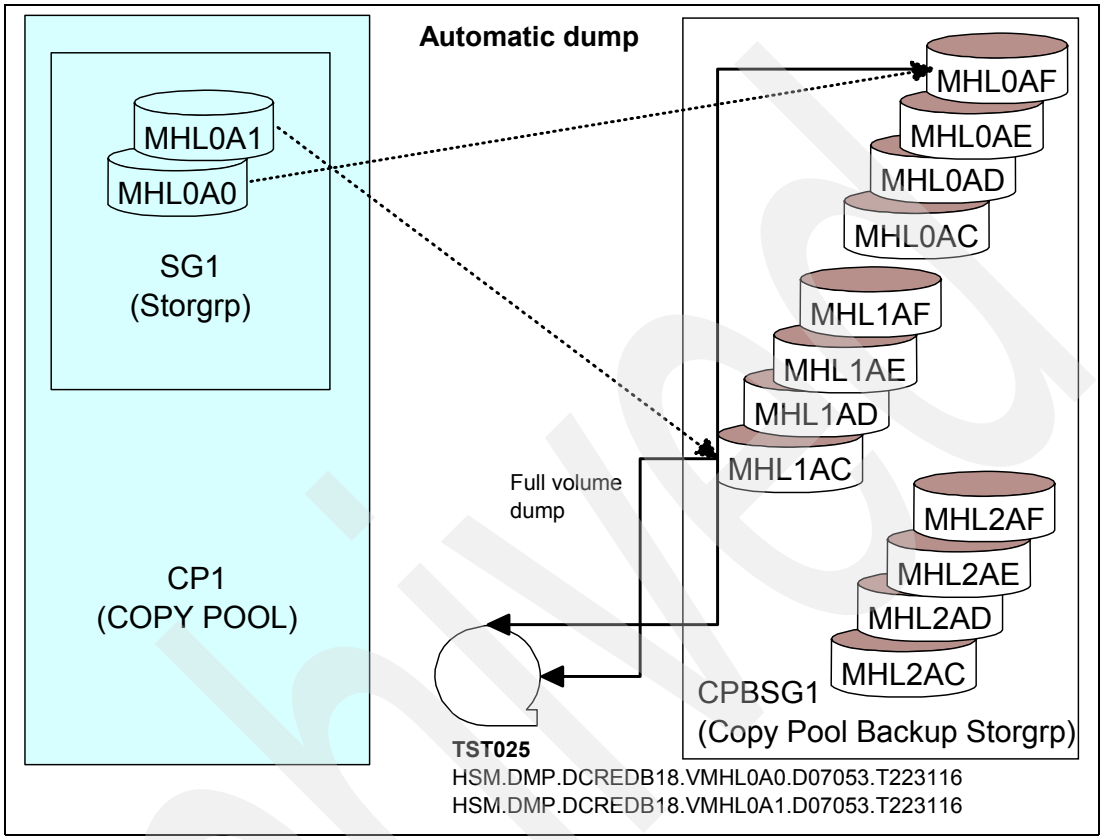


Figure 7-16 Automatic dump of copy pool CP1

The DFSMShsm dump log contained the messages shown in Example 7-11.

Example 7-11 Dump log messages issued during automatic dump

```
 DFSMSHSM DUMP LOG, TIME 16:45:45,  DATE 07/02/22
ARC0620I AUTOMATIC DUMP STARTING
ARC0570I AUTOMATIC DUMP FOR ALL SMS MANAGED VOLUME(S) TERMINATED, RC=17 REASON=0
ARC1841I AUTOMATIC DUMP STARTING FOR COPY POOL CP1
ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHL0A1(SMS) AT 17:10:00 ON 2007/02/22, SYSTEM SC64, TASK ID=ARCDVOL1 ,
 TO DUMP CLASS(ES)=   DCREDB18
ARC0728I VTOC FOR VOLUME MHL0A1 COPIED TO DATA SET HSM.DUMPVTOC.T223116.VMHL0A1.D07053 ON VOLUME SBXHS4
ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.053 17:10
ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00022)          -
ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00023) -
ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.053 17:10:00 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.053 17:10:00 EXECUTION BEGINS
ARC0120I DUMP VOLUME TST025 ADDED, RC= 0, REAS= 0
ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.053 17:15:47 EXECUTION ENDS
ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.053 17:15:47 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.053 17:15:47 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC0637I DUMP COPY OF VOLUME MHL0A1 COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/13, DISPOSITION= FASTR TARGET TO TAPE
ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A1 ENDING AT 17:15:47, PROCESSING SUCCESSFUL
ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHL0A0(SMS) AT 17:15:47 ON 2007/02/22, SYSTEM SC64, TASK ID=ARCDVOL1 ,
 TO DUMP CLASS(ES)=   DCREDB18
ARC0728I VTOC FOR VOLUME MHL0A0 COPIED TO DATA SET HSM.DUMPVTOC.T223116.VMHL0A0.D07053 ON VOLUME SBXHS4
```

```
ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.053 17:15
ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00025)          -
ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00023) -
ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.053 17:15:48 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.053 17:15:48 EXECUTION BEGINS
ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.053 17:21:16 EXECUTION ENDS
ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.053 17:21:16 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.053 17:21:16 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC1842I AUTO DUMP HAS COMPLETED FOR COPY POOL CP1, AT 17:21:17 ON 2007/02/22, MAXIMUM VOLUME RC=0000
ARC0637I DUMP COPY OF VOLUME MHLOAO COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/13, DISPOSITION= FASTR TARGET TO TAPE
ARC0623I FULL VOLUME DUMP OF VOLUME MHLOAO ENDING AT 17:21:18, PROCESSING SUCCESSFUL
ARC0621I AUTOMATIC DUMP ENDING
```

When the new dump copy of copy pool CP1 in dump class DCREDB18 is created successfully you can request the execution of a LIST COPYPOOL command, as shown in Example 7-12, in order to see the results of the automatic dump as being stored in the BCDS.

Example 7-12 LIST COPYPOOL command to show all information about generation 0

```
HSEND LI CP(CP1) ALLVOLS
```

The output of the command is shown in Figure 7-19 on page 198.

Note the changed contents in field DUMPSTATE, which switched from NONE to ALLCOMPLETE.

Since there is more information related with the fast replication backups of a copy pool if full volume dumps of the target volumes are being requested, the LIST COPYPOOL command was enhanced with z/OS DFSMShsm V1.8 by supplying additional keywords.

The new required parameters of the LIST COPYPOOL command are shown in Figure 7-17.

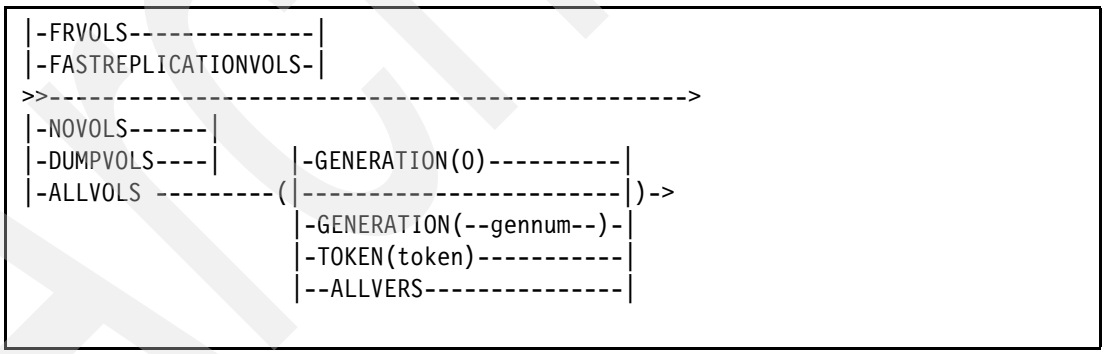


Figure 7-17 New required parameters of the LIST COPYPOOL command

The parameters are:

- FRVOLS

  Requests a list of source and target pairs (default).

- NOVOLS

  Requests a list of dump class information, if a dump copy version exists.

- DUMPVOLS

  Requests a list of source and dump volumes for all dump versions.

- ALLVOLS

  Shows all available information including dump classes as well as DASD and tape volumes in use for backups and dumps.
- ALLVOLS(GENERATION(gennum))

  A particular generation (generation 0 is the default).
- ALLVOLS(TOKEN(token))

  A backup version that was created by using this particular token.
- ALLVOLS(ALLVERS)

  All versions.

Note: While being introduced as required parameters, you need not really specify one of these. If you do not specify any of the keywords above, FRVOLS is used as the default setting.

The new optional parameters of the LIST COPYPOOL command are shown in Figure 7-18.

FASTREPLICATIONSTATE allows you to limit the display to fast replication backup versions based on the following parameters:

RECOVERABLE Available on DASD (copy pool requests #versions > 0)

DUMPONLY Still available on DASD (copy pool requests #versions = 0)

FAILED Not available for recovery of the copy pool due to a failure during the FRBACKUP process or a request for WITHDRAW

NONE Only available on tape

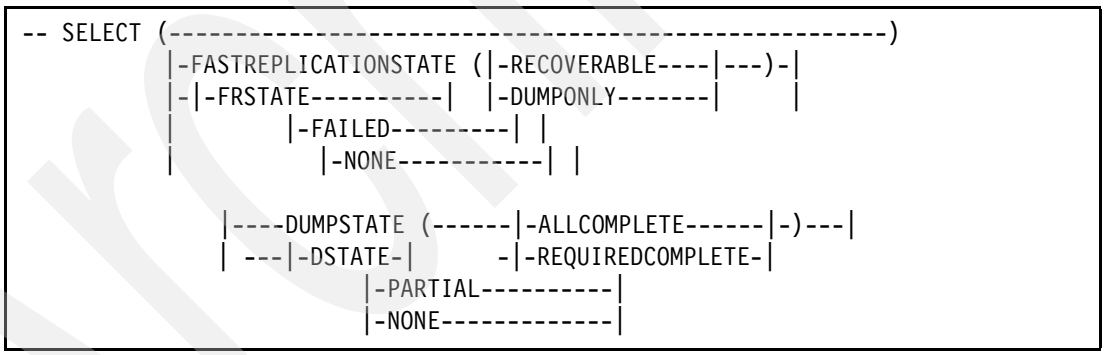


Figure 7-18 New optional parameters of the LIST COPYPOOL command

DUMPSTATE allows you to limit the display to fast replication backup versions based on the following parameters being specified:

ALLCOMPLETE Have all volumes dumped to all dump classes specified.

REQUIREDCOMPLETE Have all volumes dumped to at least the required dump classes.

Note: A fast replication backup version is only marked with a DUMPSTATE of REQUIREDCOMPLETE if at least one dump class is specified that is not required (see Example 7-9 on page 192) and if not all volumes have been dumped to one of these non-required dump classes. If all the specified dump classes are required, DUMPSTATE is either PARTIAL or ALLCOMPLETE.

PARTIAL	Have at least one volume dumped to a dump class specified
FAILED	Have no volume dumped to any dump class specified
NONE	Not required for dump because there is no relationship between the copy pool and any dump class

```
 -- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 17:25:03 ON
07/02/22 FOR SYSTEM=SC64

 COPYPOOL=CP1

 VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
   003       Y      2007/02/22      16:31:22   RECOVERABLE           ALLCOMPLETE
 TOKEN(C)=C'EX3'
 TOKEN(H)=X'C5E7F3'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
0
 DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
 DCREDB18       Y       COMPLETE     00002     2008/02/13      Y

             HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
             NO      NONE     *********  ************************************

 SOURCE   DUMPVOLS                                          DEVICE TYPE
 MHL0A0   TST025                                              3590-1
 FILE SEQ=02, DSNAME=HSM.DMP.DCREDB18.VMHL0A0.D07053.T223116
 MHL0A1   TST025                                              3590-1
 FILE SEQ=01, DSNAME=HSM.DMP.DCREDB18.VMHL0A1.D07053.T223116
```

Figure 7-19 Output of LIST CP(CP1) ALLVOLS

The possible contents of the field DUMPSTATE at the version level are:

ALLCOMPLETE	All DASD volumes have been successfully dumped to the dump classes specified.
REQUIRED COMPLETE	All DASD volumes have been successfully dumped to at least the required dump classes.
PARTIAL	For at least one required dump class the DUMPSTATE at the dump class level is PARTIAL.
NONE	No dumps are associated with the copy pool.
FAILED	The creation of a dump copy is being requested but so far not a single volume has been successfully dumped.

The possible contents of the field DUMPSTATE at the dump class level are:

COMPLETE	All DASD volumes have been successfully dumped to this particular dump class.
PARTIAL	At least one volume of the copy pool has not yet been successfully dumped to this particular dump class.
FAILED	A dump copy to this particular dump class is being requested but so far not a single volume has been successfully dumped.

Automatic dump (phases 1 and 3)

These phases are executed in a primary host only.

During deletion of expired dump copies the expiration of dump copies of copy pool volumes is determined at the dump class level. As with dump copies of non-copy pool volumes, DFSMShsm does not automatically delete the last and only remaining copy of a source volume. If you want to delete such copies you must do this with one of the following commands:

- You can use the DELVOL ... DUMP command, which must include the keywords LASTCOPY and COPYPOOLCOPY for dumps of copy pools.
- You can use the FRDELETE command.

Basically, there is no difference in processing of phase 1 between copy pool and no-copy pool volumes.

Deletion of excess dump VTOC copy data sets (phase 3) works exactly the same way for all kind of volumes.

7.3.2 FRBACKUP DUMP

In case you want the dump copies to be created as soon as possible after a successful creation of a new fast replication backup version you can request the dump copy with the FRBACKUP command by including the DUMP parameter (see Example 7-13).

Example 7-13 FRBACKUP command together with the DUMP parameter

```
HSEND FRBACKUP COPYPOOL(CP1) EXECUTE DUMP
```

The whole process is also called a fast replication dump. During execution of this command there are two phases:

1. Fast replication backup - See Figure 7-20 on page 200.
2. Copy pool dump - See Figure 7-23 on page 201.

Copy pool fast replication backup

As soon as the fast replication backup completes successfully for the copy pool, the next phase is started.

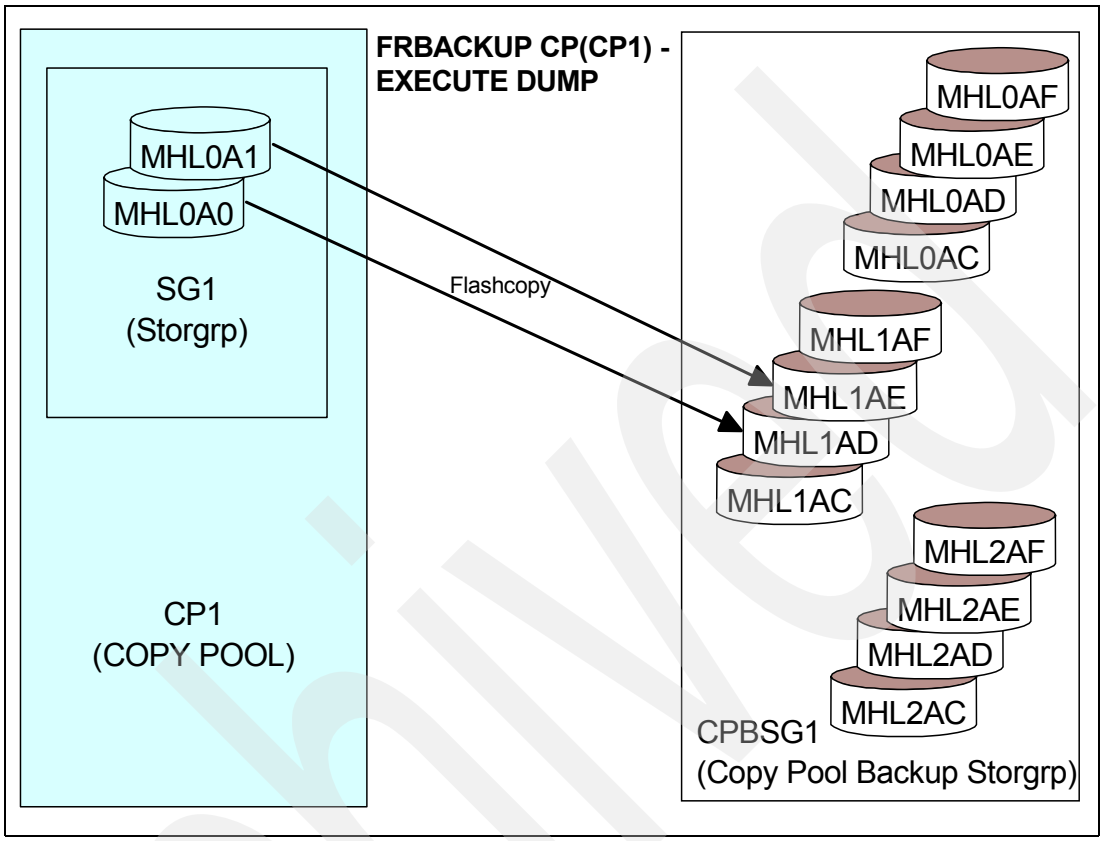


Figure 7-20 Phase 1 of FRBACKUP with imbedded dump

```
 -- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 19:00:29 ON
07/02/19 FOR SYSTEM=SC64

 COPYPOOL=CP1

 VERSION  VTOCENQ     DATE           TIME      FASTREPLICATIONSTATE  DUMPSTATE
   002       Y      2007/02/19      18:56:14   RECOVERABLE           FAILED
 TOKEN(C)=C'EX2'
 TOKEN(H)=X'C5E7F2'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL1AD  MHL0A1 - MHL1AE

0----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-21 Output of a LIST CP(CP1) SELECT(DUMPSTATE(FAILED))

Successful completion means that a FASTREPLICATIONSTATE of RECOVERABLE or DUMPONLY is displayed by the LIST COPYPOOL command (see Figure 7-21).

This does not necessarily imply that the internal FlashCopy process came to an end before the start of phase 2, as we can see in Figure 7-22, which is the output of a QUERY command that was requested at 18:57 (Example 7-14).

Example 7-14 QERY COPYPOOL

```
HSEND QUERY COPYPOOL(CP1)
```

```
ARC1820I THE FOLLOWING VOLUMES IN COPY POOL CP1, VERSION 002, HAVE AN ACTIVE
ARC1820I (CONT.) FLASHCOPY BACKGROUND COPY
ARC1820I (CONT.) SGNAME   FR-PRIMARY FR-BACKUP
ARC1820I (CONT.) SG1      MHL0A0     MHL1AD
ARC1820I (CONT.) SG1      MHL0A1     MHL1AE
ARC1821I NONE OF THE VOLUMES IN COPY POOL CP1, VERSION 001, HAVE AN ACTIVE
ARC1821I (CONT.) FLASHCOPY BACKGROUND COPY
```

Figure 7-22 Output of QUERY COPYPOOL command

This shows clearly that the background FlashCopy process is still going on while the copy pool dump process started at 18:56:15, as we can see in Example 7-15 on page 202, which is an excerpt of the JESMSGLG output of the HSM address space.

Copy pool dump

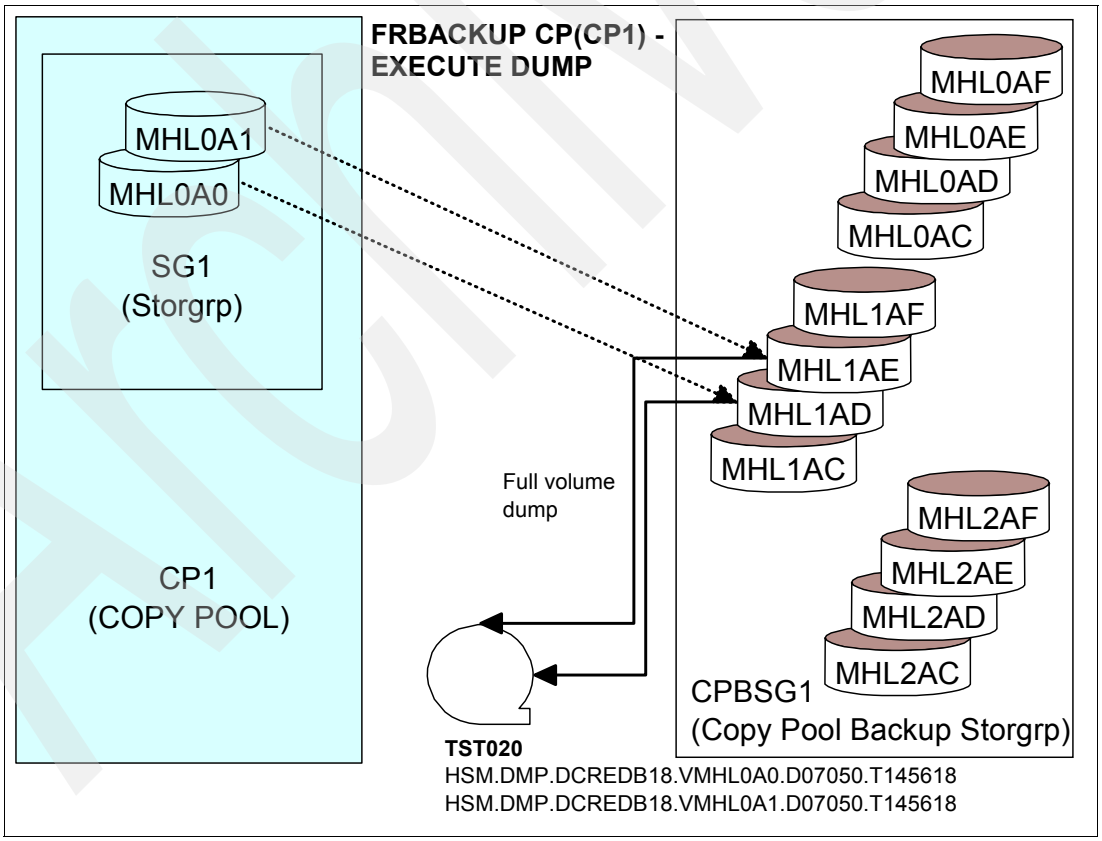


Figure 7-23 Phase 2 of FRBACKUP with imbedded dump

The full volume dump is performed to dump class DCREDB18, as we did not specify any dump class with the FRBACKUP command (see Example 7-13 on page 199). Alternatively, you can request a dump class with the command by using the DUMPCLASS parameter. In

this case all other dump classes associated with the copy pool are ignored during execution of this FRBACKUP command.

Example 7-15 JESMSGLG of HSM address space (excerpt)

```
18.56.14 STC18151  ARC1801I FAST REPLICATION DUMP IS STARTING FOR COPY  519
    519             ARC1801I (CONT.) POOL CP1, AT 18:56:14 ON 2007/02/19, TOKEN='EX2'
 18.56.15 STC18151  ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE  527
    527             ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF
    527             ARC1805I (CONT.) COPY POOL CP1
 18.56.15 STC18151  ARC1805I (CONT.) MHL0A0
 18.56.15 STC18151  ARC1805I (CONT.) MHL0A1
 18.56.15 STC18151  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  530
    530             ARC0622I (CONT.) MHL0A0(SMS) AT 18:56:15 ON 2007/02/19, SYSTEM SC64,
    530             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 18.56.16 STC18151  IEC501A M 0B91,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618
 18.57.17 STC18151  IEC705I TAPE ON 0B91,TST020,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618,MEDIA3
 18.57.17 STC18151  ARC0120I DUMP VOLUME TST020 ADDED, RC= 0, REAS= 0
 19.02.25 STC18151  IEC205I SYS00105,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME LIST,  539
    539             DSN=HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618,VOLS=TST020,
    539             TOTALBLOCKS=47476
 19.02.25 STC18151  ARC0637I DUMP COPY OF VOLUME MHL0A0 COMPLETE,  541
    541             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR
    541             ARC0637I (CONT.) TARGET TO TAPE
 19.02.25 STC18151  ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A0 ENDING AT  542
    542             ARC0623I (CONT.) 19:02:25, PROCESSING SUCCESSFUL
 19.02.25 STC18151  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  543
    543             ARC0622I (CONT.) MHL0A1(SMS) AT 19:02:25 ON 2007/02/19, SYSTEM SC64,
    543             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 19.05.00 STC18151  ARC0405I HOST 2 UPDATING SPACE INFORMATION ON ALL  546
    546             ARC0405I (CONT.) VOLUMES
 19.06.07 STC18151  IEC205I SYS00105,DFHSM64,DFHSM64,FILESEQ=2, COMPLETE VOLUME LIST,  547
    547             DSN=HSM.DMP.DCREDB18.VMHL0A1.D07050.T145618,VOLS=TST020,
    547             TOTALBLOCKS=47447
 19.06.07 STC18151  ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED  549
    549             ARC1802I (CONT.) FOR COPY POOL CP1, AT 19:06:07 ON 2007/02/19,
    549             ARC1802I (CONT.) FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
 19.06.08 STC18151  IEF234E K 0B91,TST020,PVT,DFHSM64,DFHSM64
 19.06.08 STC18151  ARC0637I DUMP COPY OF VOLUME MHL0A1 COMPLETE,  551
    551             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR
    551             ARC0637I (CONT.) TARGET TO TAPE
 19.06.08 STC18151  ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A1 ENDING AT  552
    552             ARC0623I (CONT.) 19:06:08, PROCESSING SUCCESSFUL
```

Another execution of the QUERY COPYPOOL command (as shown in Example 7-14 on page 201) was requested at 19:01:00. As Figure 7-24 shows, the background copy is now complete and the FlashCopy relationship of the volume pairs has ended.

```
ARC1821I NONE OF THE VOLUMES IN COPY POOL CP1, VERSION 002, HAVE AN ACTIVE
ARC1821I (CONT.) FLASHCOPY BACKGROUND COPY
ARC1821I NONE OF THE VOLUMES IN COPY POOL CP1, VERSION 001, HAVE AN ACTIVE
ARC1821I (CONT.) FLASHCOPY BACKGROUND COPY
```

Figure 7-24 Output of QUERY COPYPOOL command

```
 ARC1801I FAST REPLICATION DUMP IS STARTING FOR COPY POOL CP1, AT 18:56:14 ON 2007/02/19, TOKEN='EX2'
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 18:56
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
 ARC0640I ARCFRTM -  COPY IDY(MHLOA0) ODY(MHL1AD) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(MHLOA1) ODY(MHL1AE) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2007.050 18:56:15 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2007.050 18:56:15 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
 ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2007.050 18:56:15 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS
OVERLAID
 ARC0640I ARCFRTM - ADR806I (002)-TOMI (02), VOLUME MHLOA0 WAS COPIED USING A FAST REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2007.050 18:56:15 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2007.050 18:56:15 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2007.050 18:56:15 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS
OVERLAID
 ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHLOA1 WAS COPIED USING A FAST REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2007.050 18:56:15 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2007.050 18:56:15 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2007.050 18:56:15 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE
IS 0000
 ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL CP1
 ARC1805I (CONT.) MHLOA0
 ARC1805I (CONT.) MHLOA1
 ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED FOR COPY POOL CP1, AT 19:06:07 ON 2007/02/19, FUNCTION
RC=0000,
  MAXIMUM VOLUME RC=0000
```

Figure 7-25 DFSMShsm backup log messages with fast replication dump

The processing of the fast replication dump is reported in the backup log of DFSMShsm, as shown in Figure 7-25 on page 203. Note that the process is documented in messages ARC1801I and ARC1802I as FAST REPLICATION DUMP. The end of the process is reported after the last full volume dump has ended successfully.

```
ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=2
ARC0144I AUDIT=NOT HELD AND INACTIVE, LIST=NOT HELD AND INACTIVE, RECYCLE=NOT
ARC0144I (CONT.) HELD AND INACTIVE, REPORT=NOT HELD AND INACTIVE
ARC0160I MIGRATION=NOT HELD, AUTOMIGRATION=HELD, RECALL=NOT HELD,
ARC0160I (CONT.) TAPERECALL=NOT HELD, DATA SET MIGRATION=INACTIVE, VOLUME
ARC0160I (CONT.) MIGRATION=INACTIVE, DATA SET RECALL=INACTIVE
ARC0163I BACKUP=NOT HELD, AUTOBACKUP=HELD, RECOVERY=NOT HELD,
ARC0163I (CONT.) TAPEDATASETRECOVERY=NOT HELD, DATA SET BACKUP=NOT HELD, VOLUME
ARC0163I (CONT.) BACKUP=INACTIVE, DATA SET RECOVERY=INACTIVE, VOLUME
ARC0163I (CONT.) RECOVERY=INACTIVE
ARC0276I DATA SET BACKUP=INACTIVE, DATA SET BACKUP ACTUAL IDLETASKS=(ALLOC=00,
ARC0276I (CONT.) MAX=00)
ARC1826I FRBACKUP=NOT HELD AND INACTIVE,FRRECOV=NOT HELD AND INACTIVE,FRBACKUP
ARC1826I (CONT.) DUMP=NOT HELD AND ACTIVE,FRRECOV(TAPE)=NOT HELD AND INACTIVE,
ARC1826I (CONT.) FRRECOV(DATASET)=NOT HELD AND INACTIVE
ARC0642I DUMP=NOT HELD, AUTODUMP=NOT HELD, VOLUME DUMP=INACTIVE, VOLUME
ARC0642I (CONT.) RESTORE=INACTIVE, DATA SET RESTORE=INACTIVE
ARC1822I FRBACKUP DUMP OR DUMPONLY OF COPY POOL CP1 FOR USER MHLRES2, REQUEST
ARC1822I (CONT.) 59 ON HOST 2 IS IN PROGRESS: NOT PROCESSED = 2, TOTAL = 2
ARC0161I FRBACKUP DUMP OF VOLUME MHLOA0,COPY POOL=CP1 FOR USER MHLRES2,
ARC0161I (CONT.) REQUEST 00000059
ARC0437I - TAPECOPY NOT HELD AND INACTIVE
ARC0437I - TAPEREPL NOT HELD AND INACTIVE
ARC0415I EXPIREBV=NOT HELD AND INACTIVE, LAST STORED BACKUP VERSION KEY=, LAST
ARC0415I (CONT.) STORED ABARS VERSION KEY=, LAST PLANNED END KEY=
ARC0460I PRIVATE AREA LIMIT=8168K, UNALLOCATED=5560K, LARGEST FREE AREAS=5516K,
ARC0460I (CONT.)  40K
ARC0460I EXTENDED PRIVATE AREA LIMIT=1466M, UNALLOCATED=1436M, LARGEST FREE
ARC0460I (CONT.) AREAS=1436M, 56K
ARC6018I AGGREGATE BACKUP/RECOVERY = INACTIVE
ARC6019I AGGREGATE BACKUP = NOT HELD, AGGREGATE RECOVERY = NOT HELD
ARC1540I COMMON RECALL QUEUE PLACEMENT FACTORS: CONNECTION STATUS=CONNECTED,
ARC1540I (CONT.) CRQPLEX HOLD STATUS=NONE,HOST COMMONQUEUE HOLD STATUS=NONE,
ARC1540I (CONT.) STRUCTURE ENTRIES=000% FULL,STRUCTURE ELEMENTS=000% FULL
ARC1541I COMMON RECALL QUEUE SELECTION FACTORS: CONNECTION STATUS=CONNECTED,
ARC1541I (CONT.) HOST RECALL HOLD STATUS=NONE,HOST COMMONQUEUE HOLD STATUS=NONE
ARC0101I QUERY ACTIVE COMMAND COMPLETED ON HOST=2
```

Figure 7-26 Output of QUERY ACTIVE command during processing of a fast replication dump

When you look at the output of a QUERY ACTIVE command during execution of a fast replication dump process you will find the FRBACKUP DUMP to be reported as active (see Figure 7-26 on page 204).

```
 ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHLOAO(SMS) AT 18:56:15 ON 2007/02/19, SYSTEM SC64, TASK ID=ARCDVOL1
,
  TO DUMP CLASS(ES)=   DCREDB18
 ARC0728I VTOC FOR VOLUME MHLOAO COPIED TO DATA SET HSM.DUMPVTOC.T145618.VMHLOAO.DO7050 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 18:56
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00104)          -
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00105) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.050 18:56:16 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.050 18:56:16 EXECUTION BEGINS
 ARC0120I DUMP VOLUME TST020 ADDED, RC= 0, REAS= 0
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.050 19:02:25 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.050 19:02:25 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.050 19:02:25 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE
IS 0000
 ARC0637I DUMP COPY OF VOLUME MHLOAO COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR TARGET TO
TAPE
 ARC0623I FULL VOLUME DUMP OF VOLUME MHLOAO ENDING AT 19:02:25, PROCESSING SUCCESSFUL
 ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHLOA1(SMS) AT 19:02:25 ON 2007/02/19, SYSTEM SC64, TASK ID=ARCDVOL1
,
  TO DUMP CLASS(ES)=   DCREDB18
 ARC0728I VTOC FOR VOLUME MHLOA1 COPIED TO DATA SET HSM.DUMPVTOC.T145618.VMHLOA1.DO7050 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 19:02
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00108)          -
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00105) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.050 19:02:25 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.050 19:02:25 EXECUTION BEGINS
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.050 19:06:07 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.050 19:06:07 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.050 19:06:07 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE
IS 0000
 ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED FOR COPY POOL CP1, AT 19:06:07 ON 2007/02/19, FUNCTION
RC=0000,
  MAXIMUM VOLUME RC=0000
 ARC0637I DUMP COPY OF VOLUME MHLOA1 COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR TARGET TO
TAPE
 ARC0623I FULL VOLUME DUMP OF VOLUME MHLOA1 ENDING AT 19:06:08, PROCESSING SUCCESSFUL
  DFSMSHSM DUMP LOG, TIME 19:06:08,  DATE 07/02/19
```

Figure 7-27 DFSMShsm dump log messages with fast replication dump

The processing of the fast replication dump is reported in the dump log of DFSMShsm, as shown in Figure 7-27 on page 205. We can see similar messages as during full volume dumps of level0 volumes. The end of the process is reported by message ARC1802I as FAST REPLICATION DUMP after the last full volume dump has ended successfully.

```
 -- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 19:13:20 ON
07/02/19 FOR SYSTEM=SC64

 COPYPOOL=CP1

 VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
   002       Y      2007/02/19      18:56:14   RECOVERABLE           ALLCOMPLETE
 TOKEN(C)=C'EX2'
 TOKEN(H)=X'C5E7F2'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL1AD  MHL0A1 - MHL1AE

DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
 DCREDB18       Y       COMPLETE     00002     2008/02/10      Y

             HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
             NO      NONE     *********  ************************************

 SOURCE   DUMPVOLS                                        DEVICE TYPE
 MHL0A0   TST020                                            3590-1
 FILE SEQ=01, DSNAME=HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618
 MHL0A1   TST020                                            3590-1
 FILE SEQ=02, DSNAME=HSM.DMP.DCREDB18.VMHL0A1.D07050.T145618
0----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-28 Output of the LI COPYPOOL command after the end of the fast replication dump

After the end of the fast replication dump process you can see all the details by executing a LIST command, as shown in Example 7-16.

Example 7-16 LIST COPYPOOL(CP1) ALLVOLS

```
HSEND LI CP(CP1) ALLVOLS(TOKEN(EX2))
```

The output (see Figure 7-28) shows a DUMPSTATE of ALLCOMPLETE at the version level, which means that all outstanding dump copies for this version ended successfully and are reported at the dump class level with a DUMPSTATE of COMPLETE.

The volume serial numbers of all the dump volumes that contain dump copies and the names of all dump files are included as well.

To determine the contents of such a dump copy we can use the LIST command, as shown in Example 7-22 on page 211.

```
1-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 19:21:17 ON 07/02/19 FOR SYSTEM=SC64

ODUMP   VOL    UNIT     FILE SOURCE              DUMPED     DUMPED                          PCT  HW ENC C SET OF
DUMP
 VOLSER STATUS TYPE     SEQ  VOLSER SMS CLASS    DATE       TIME      EXP DATE IDRC LIBRARY FULL        P VOLSERS
0TST020 UNEXP  3590-1                 DCREDB18                        2008/02/10 Y  LIB1      04  N  *** Y
                       01   MHL0A0 Y            2007/02/19 18:56:14                                      TST020

        ENCTYPE    RSAKEY/KPWD
        *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618
1CONTENTS OF VTOC COPY FOR SOURCE VOLUME  MHL0A0
0DATASET NAME                              ORG  MULTI  CREATED   REFERENCED  EXP DATE  RACF  PSWD   CHANGED
0MHLRES2.DCOLLECT.D997                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 MHLRES2.DCOLLECT.D998                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     NO
 MHLRES2.DCOLLECT.D999                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 SYS1.VTOCIX.MHL0A0                        PS    ***   03/10/20  00/00/00    00/00/00  NO    NO     NO
 SYS1.VVDS.VMHL0A0                         VS    ***   03/11/07  00/00/00    00/00/00  ***   ***    NO
 YYY.CNTL.JCL                              PO    NO    07/02/19  07/02/19    00/00/00  NO    NO     YES
                       02   MHL0A1 Y            2007/02/19 18:56:14                                      TST020

        ENCTYPE    RSAKEY/KPWD
        *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0A1.D07050.T145618
0----- END OF - DUMP VOLUME - LISTING -----

1-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 19:21:41 ON 07/02/19 FOR SYSTEM=SC64

ODUMP   VOL    UNIT     FILE SOURCE              DUMPED     DUMPED                          PCT  HW ENC C SET OF
DUMP
 VOLSER STATUS TYPE     SEQ  VOLSER SMS CLASS    DATE       TIME      EXP DATE IDRC LIBRARY FULL        P VOLSERS
0TST020 UNEXP  3590-1                 DCREDB18                        2008/02/10 Y  LIB1      04  N  *** Y
                       01   MHL0A0 Y            2007/02/19 18:56:14                                      TST020

        ENCTYPE    RSAKEY/KPWD
        *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0A0.D07050.T145618
                       02   MHL0A1 Y            2007/02/19 18:56:14                                      TST020

        ENCTYPE    RSAKEY/KPWD
        *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0A1.D07050.T145618
1CONTENTS OF VTOC COPY FOR SOURCE VOLUME  MHL0A1
0DATASET NAME                              ORG  MULTI  CREATED   REFERENCED  EXP DATE  RACF  PSWD   CHANGED
0MHLRES2.DCOLLECT.D987                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 MHLRES2.DCOLLECT.D988                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     NO
 MHLRES2.DCOLLECT.D989                     PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 SYS1.VTOCIX.MHL0A1                        PS    ***   03/10/20  00/00/00    00/00/00  NO    NO     NO
 SYS1.VVDS.VMHL0A1                         VS    ***   03/11/07  00/00/00    00/00/00  ***   ***    NO
 YYY.CMD.CLIST                             PE    NO    07/02/19  07/02/19    00/00/00  NO    NO     YES
0----- END OF - DUMP VOLUME - LISTING -----
```

Figure 7-29 Output of LIST DUMPVOLUME DCONTENTS(MHL0A1)

The output (see Figure 7-29) shows the same look and feel as with other dump volumes with the exception of column CP, which contains Y to indicate that the contents of this dump volume were created while processing at least one target volume of a fast replication backup.

7.3.3 FRBACKUP DUMPONLY

If you want to create another dump copy of a copy pool or complete a dump copy for that version (DUMPSTATE is PARTIAL) at any time (as long as the DASD copy you want to dump exists) you can use the DUMPONLY parameter of the FRBACKUP command.

Suppose that there was a Version 1 DASD copy (token EX1) of copy pool CP1 created some time ago (see Figure 7-30).

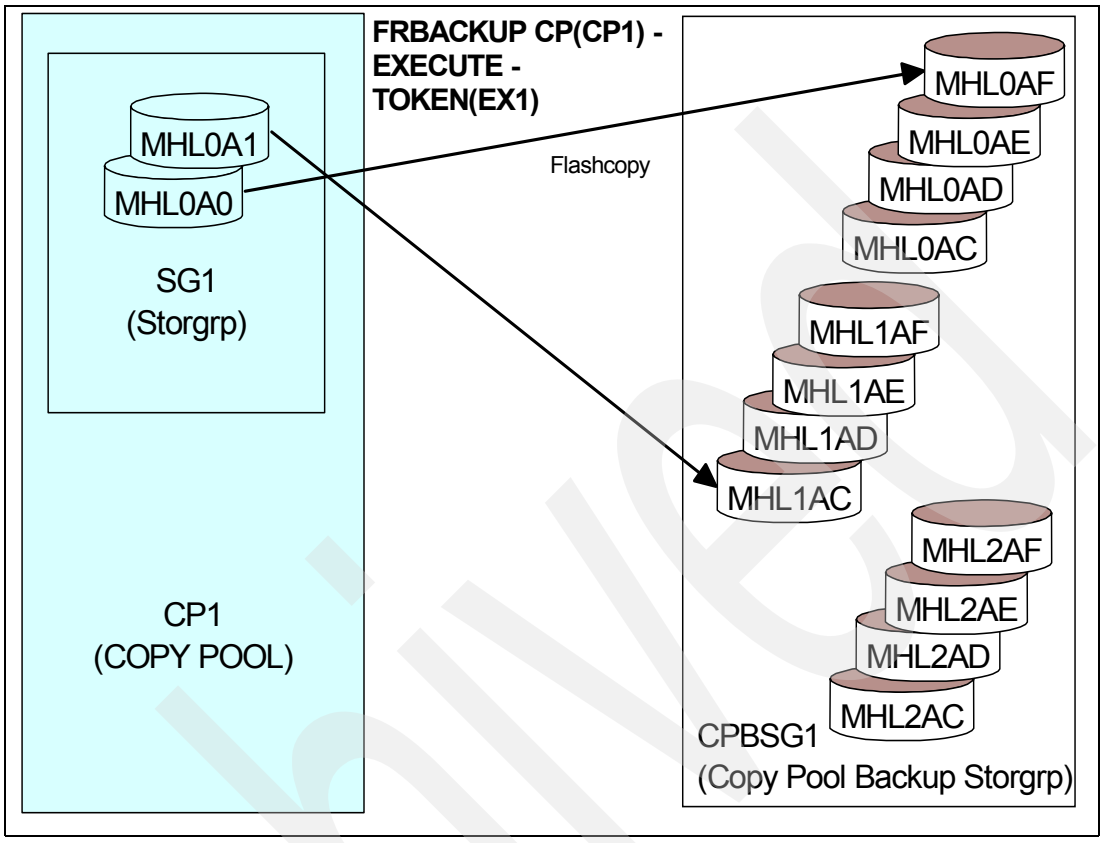


Figure 7-30 Creation of Version 1 DASD copy of copy pool CP1

If it is still available, we can display its properties by using the LIST COPYPOOL command, as shown in Example 7-17.

Example 7-17 LIST COPYPOOL command to verify the existence of a DASD copy

```
HSEND LIST CP(CP1) ALLVOLS(TOKEN(EX1))
```

The output of the LIST command is shown in Figure 7-31 on page 209. The FASTREPLICATIONSTATE of RECOVERABLE proves that the DASD copy is available and complete. The DUMPSTATE of NONE shows that there is no dump copy available so far.

When we want to dump this particular fast replication backup version we can use the command shown in Example 7-18.

Example 7-18 FRBACKUP COPYPOOL(CP1) DUMPONLY

```
HSEND FRBACKUP COPYPOOL(CP1) DUMPONLY(TOKEN(EX1))
```

If you do not specify a dump class with the FRBACKUP command DFSMShsm looks up the copy pool definition and uses the dump classes specified with the copy pool.

```
 -- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 15:05:04 ON
07/02/19 FOR SYSTEM=SC64

 COPYPOOL=CP1

 VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
   001       Y      2007/02/19      14:49:20   RECOVERABLE           NONE
 TOKEN(C)=C'EX1'
 TOKEN(H)=X'C5E7F1'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC

0----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-31 Output of the LIST COPYPOOL command to verify the existence of a DASD copy

The resulting messages in the JESMSGLG data set of the HSM address space are shown in Example 7-19. Note the time stamp of the dump data set. While the full volume dump process actually starts at 17:42:50, the low-level qualifier of the dump data set name (T204914, format Tssmmhh) reflects the time stamp of the fast replication backup as reported by the LIST COPYPOOL command (see Figure on page 187).

Example 7-19 Messages in JESMSGLG for FRBACKUP COPYPOOL(SG1) DUMPONLY

```
17.42.49 STC18151  ARC1801I FAST REPLICATION DUMPONLY IS STARTING FOR  261
    261             ARC1801I (CONT.) COPY POOL CP1, AT 17:42:49 ON 2007/02/19
 17.42.50 STC18151  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  262
    262             ARC0622I (CONT.) MHL0A0(SMS) AT 17:42:50 ON 2007/02/19, SYSTEM SC64,
    262             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 17.42.50 STC18151  IEC501A M 0B90,PRIVAT,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHL0A0.D07050.T204914
 17.43.48 STC18151  IEC705I TAPE ON 0B90,TST019,SL,COMP,DFHSM64,DFHSM64,HSM.DMP.DCREDB18.VMHL0A0.D07050.T204914,MEDIA3
 17.43.49 STC18151  ARC0120I DUMP VOLUME TST019 ADDED, RC= 0, REAS= 0
 17.47.28 STC18151  IEC205I SYS00073,DFHSM64,DFHSM64,FILESEQ=1, COMPLETE VOLUME LIST,  267
    267             DSN=HSM.DMP.DCREDB18.VMHL0A0.D07050.T204914,VOLS=TST019,
    267             TOTALBLOCKS=47437
 17.47.28 STC18151  ARC0637I DUMP COPY OF VOLUME MHL0A0 COMPLETE,  269
    269             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR
    269             ARC0637I (CONT.) TARGET TO TAPE
 17.47.28 STC18151  ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A0 ENDING AT  270
    270             ARC0623I (CONT.) 17:47:28, PROCESSING SUCCESSFUL
 17.47.28 STC18151  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME  271
    271             ARC0622I (CONT.) MHL0A1(SMS) AT 17:47:28 ON 2007/02/19, SYSTEM SC64,
    271             ARC0622I (CONT.) TASK ID=ARCDVOL1 , TO DUMP CLASS(ES)=   DCREDB18
 17.51.09 STC18151  IEC205I SYS00073,DFHSM64,DFHSM64,FILESEQ=2, COMPLETE VOLUME LIST,  280
    280             DSN=HSM.DMP.DCREDB18.VMHL0A1.D07050.T204914,VOLS=TST019,
    280             TOTALBLOCKS=47437
 17.51.10 STC18151  ARC1802I FAST REPLICATION BACKUP DUMPONLY HAS  282
    282             ARC1802I (CONT.) COMPLETED FOR COPY POOL CP1, AT 17:51:10 ON
    282             ARC1802I (CONT.) 2007/02/19, FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
 17.51.10 STC18151  IEF234E K 0B90,TST019,PVT,DFHSM64,DFHSM64
 17.51.11 STC18151  ARC0637I DUMP COPY OF VOLUME MHL0A1 COMPLETE,  284
    284             ARC0637I (CONT.) DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR
    284             ARC0637I (CONT.) TARGET TO TAPE
 17.51.11 STC18151  ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A1 ENDING AT  285
    285             ARC0623I (CONT.) 17:51:11, PROCESSING SUCCESSFUL
```

The fast replication DUMPONLY processing is documented by messages in the backup log (see Figure 7-32), as well as in the dump log (see Example 7-20).

```
 ARC1801I FAST REPLICATION DUMPONLY IS STARTING FOR COPY POOL CP1, AT 17:42:49 ON 2007/02/19
 ARC1802I FAST REPLICATION BACKUP DUMPONLY HAS COMPLETED FOR COPY POOL CP1, AT 17:51:10 ON 2007/02/19, FUNCTION
RC=0000,
  MAXIMUM VOLUME RC=0000
```

Figure 7-32 Messages in the DFSMShsm backup log for FRBACKUP COPYPOOL(SG1) DUMPONLY

The messages in the DFSMShsm dump log look very similar to what we can see at any time when a full volume dump takes place under control of DFSMShsm. What we cannot see as part of the dump log is which volumes stand behind the references that are used with the INDDNAME statements.

There is one hint available that this was not a full volume dump of MHL0A0 as usual. Look at the time stamp of the DUMPVTOC data set and you can once more realize that this does not represent the current time of full volume dump processing (which started at 17:42:50), but instead T204914 (format Tssmmhh) represents exactly the time (14:49:20) when fast replication backup of volume MHL0A0 was started (see Figure on page 211).

Example 7-20 Messages in the DFSMShsm dump log for FRBACKUP COPYPOOL(SG1) DUMPONLY

```
DFSMSHSM DUMP LOG, TIME 15:26:04,  DATE 07/02/19
 ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHL0A0(SMS) AT 17:42:50 ON 2007/02/19, SYSTEM SC64, TASK ID=ARCDVOL1 ,
  TO DUMP CLASS(ES)=   DCREDB18
 ARC0728I VTOC FOR VOLUME MHL0A0 COPIED TO DATA SET HSM.DUMPVTOC.T204914.VMHL0A0.D07050 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 17:42
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00072)          -
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00073) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.050 17:42:50 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.050 17:42:50 EXECUTION BEGINS
 ARC0120I DUMP VOLUME TST019 ADDED, RC= 0, REAS= 0
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.050 17:47:28 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.050 17:47:28 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.050 17:47:28 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS
0000
 ARC0637I DUMP COPY OF VOLUME MHL0A0 COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR TARGET TO TAPE
 ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A0 ENDING AT 17:47:28, PROCESSING SUCCESSFUL
 ARC0622I FULL VOLUME DUMP STARTING ON VOLUME MHL0A1(SMS) AT 17:47:28 ON 2007/02/19, SYSTEM SC64, TASK ID=ARCDVOL1 ,
  TO DUMP CLASS(ES)=   DCREDB18
 ARC0728I VTOC FOR VOLUME MHL0A1 COPIED TO DATA SET HSM.DUMPVTOC.T204914.VMHL0A1.D07050 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.050 17:47
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS00075)          -
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS00073) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2007.050 17:47:28 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2007.050 17:47:28 EXECUTION BEGINS
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2007.050 17:51:09 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2007.050 17:51:09 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2007.050 17:51:09 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS
0000
 ARC1802I FAST REPLICATION BACKUP DUMPONLY HAS COMPLETED FOR COPY POOL CP1, AT 17:51:10 ON 2007/02/19, FUNCTION RC=0000,
```

```
 MAXIMUM VOLUME RC=0000
ARC0637I DUMP COPY OF VOLUME MHL0A1 COMPLETE, DCLASS=DCREDB18, EXPDT=2008/02/10, DISPOSITION= FASTR TARGET TO TAPE
ARC0623I FULL VOLUME DUMP OF VOLUME MHL0A1 ENDING AT 17:51:11, PROCESSING SUCCESSFUL
 DFSMSHSM DUMP LOG, TIME 17:51:11,  DATE 07/02/19
```

When a LIST COPYPOOL(CP1) command (as shown in Example 7-21) is performed after all related full volume dumps have completed successfully, the DUMPSTATE indicates ALLCOMPLETE (see Figure).

Example 7-21 Syntax of LIST COPYPOOL command

```
LIST COPYPOOL(CP1) ALLVOLS(TOKEN(EX1))
```

Use the ALLVOLS parameter to request the display of all available information (volume pairs, dump classes, dump volumes, names of dump copy data sets), as shown in Figure 7-33.

```
 -- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 17:52:36 ON
07/02/19 FOR SYSTEM=SC64

 COPYPOOL=CP1

 VERSION  VTOCENQ     DATE          TIME     FASTREPLICATIONSTATE  DUMPSTATE
   001      Y      2007/02/19     14:49:20   RECOVERABLE           ALLCOMPLETE
 TOKEN(C)=C'EX1'
 TOKEN(H)=X'C5E7F1'
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
0
 DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
 DCREDB18      Y        COMPLETE     00002     2008/02/10      Y

             HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
             NO      NONE     ********* ****************************************

 SOURCE   DUMPVOLS                                          DEVICE TYPE
 MHL0A0   TST019                                             3590-1
 FILE SEQ=01, DSNAME=HSM.DMP.DCREDB18.VMHL0A0.D07050.T204914
 MHL0A1   TST019                                             3590-1
 FILE SEQ=02, DSNAME=HSM.DMP.DCREDB18.VMHL0A1.D07050.T204914
0----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-33 Output of the LIST COPYPOOL(CP1) ALLVOLS

To determine the contents of such a dump copy we can use the LIST command, as shown in Example 7-22.

Example 7-22 LIST DUMPVOLUME command including DCONTENTS keyword

```
LIST DUMPVOLUME TST019 DCONTENTS(MHL0A1)
```

The output (Example 7-23) shows the same look and feel as other dump volumes with the exception of column CP, which contains Y to indicate that the contents of this dump volume were created while processing at least one target volume of a fast replication backup.

Example 7-23 Output of LIST DUMPVOLUME DCONTENTS(MHL0A1)

```
1-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 18:08:58 ON 07/02/19 FOR SYSTEM=SC64

0DUMP   VOL    UNIT     FILE SOURCE              DUMPED     DUMPED                          PCT  HW ENC C SET OF DUMP
 VOLSER STATUS TYPE     SEQ  VOLSER SMS CLASS    DATE       TIME      EXP DATE IDRC LIBRARY FULL        P VOLSERS
0TST019 UNEXP  3590-1                  DCREDB18                      2008/02/10 Y  LIB1      04  N  *** Y
                       01   MHL0AO Y            2007/02/19 14:49:20                                      TST019

        ENCTYPE    RSAKEY/KPWD
        *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0AO.D07050.T204914
                      02   MHL0A1 Y            2007/02/19 14:49:20                                      TST019

       ENCTYPE    RSAKEY/KPWD
       *********  ****************************************************************

 DUMP COPY DATA SET NAME = HSM.DMP.DCREDB18.VMHL0A1.D07050.T204914
1CONTENTS OF VTOC COPY FOR SOURCE VOLUME  MHL0A1
0DATASET NAME                                ORG  MULTI  CREATED   REFERENCED  EXP DATE  RACF  PSWD   CHANGED
0MHLRES2.DCOLLECT.D987                       PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 MHLRES2.DCOLLECT.D988                       PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 MHLRES2.DCOLLECT.D989                       PS    NO    07/02/16  07/02/16    00/00/00  NO    NO     YES
 SYS1.VTOCIX.MHL0A1                          PS    ***   03/10/20  00/00/00    00/00/00  NO    NO     NO
 SYS1.VVDS.VMHL0A1                           VS    ***   03/11/07  00/00/00    00/00/00  ***   ***    NO
0----- END OF - DUMP VOLUME - LISTING -----
```

By specifying the DUMPONLY parameter of the FRBACKUP command, a full volume dump is requested for all target volumes of a particular fast replication backup version of a copy pool (see Figure 7-34).

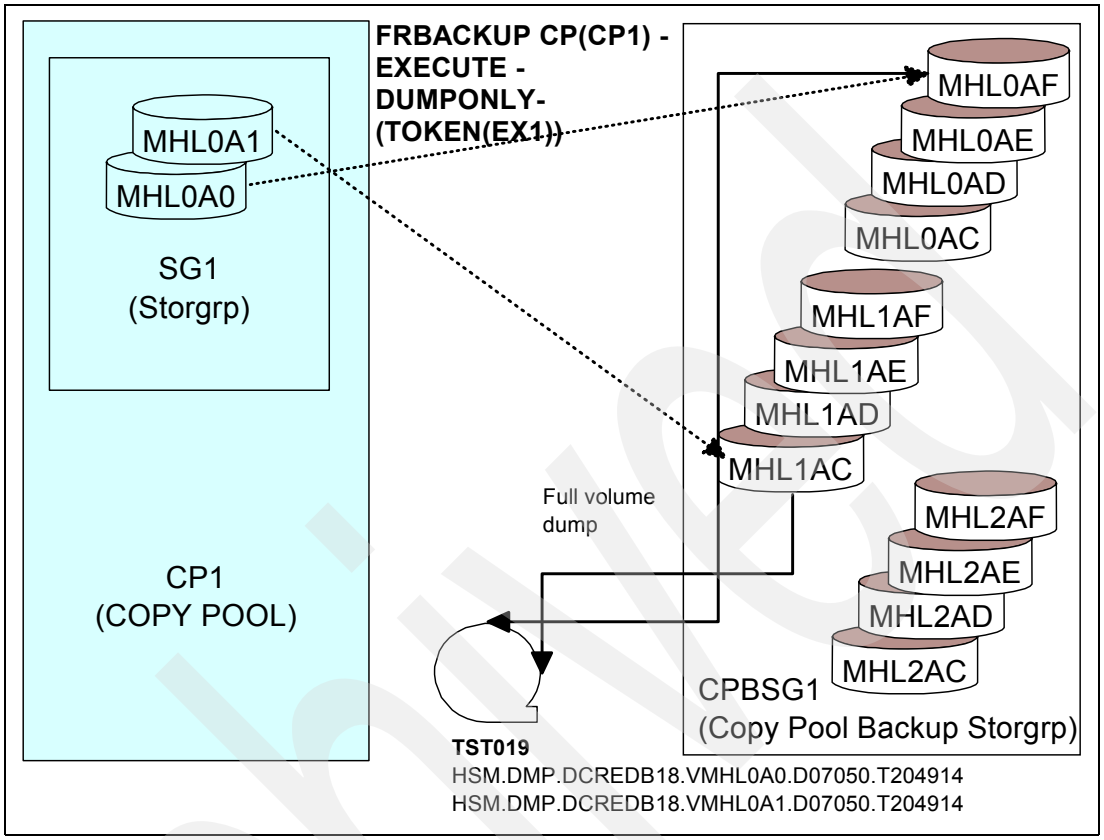


Figure 7-34 FRBACKUP with DUMPONLY parameter

The DUMPONLY parameter of the FRBACKUP command can be used to request any dump class even if it is not assigned at the copy pool level. By using up to five different dump classes you can have up to five dump copies for each fast replication backup generation. This allows for a maximum of 425 dump copies per copy pool.

7.3.4 Copy pools that request NOCOPY type FlashCopy processing

If you are interested in a full volume dump of a volume pool and if you can quiesce the applications that are using these data only for a very short period of time, you should consider using fast replication. When you are short on volumes to hold the fast replication backups, you can request the FlashCopy process with NOCOPY and to maintain these DASD copies only until a dump copy is successfully created.

Copy pool for NOCOPY processing

You must define a copy pool that contains 0 (zero) in the field number of recoverable DASD fast replicate backup versions (see Figure 7-35).

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              COPY POOL DEFINE                 Page 1 of 4
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : CPO

To DEFINE Copy Pool, Specify:
  Description ==> DUMPONLY COPY POOL
              ==>

  Auto Dump . . . Y   (Y or N)        Dump Sys/Sys Group Name . . .
  Dump Class  . . CRYPCOPY            Dump Class  . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .

  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 0       (0 to 85 or blank)



Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 7-35 Copy pool definition for NOCOPY processing during fast replication backup

Depending on your needs you can specify Y or N in the field auto dump. If you specify Y you must specify at least one dump class, as shown in Example 7-24.

Example 7-24 FRBACKUP command for NOCOPY

```
HSEND FRBACKUP CP(CPO) DUMP(DCLASS(CRYPCOPY))
```

When you request a command (as shown in Example 7-24 on page 214) DFSMShsm requests a fast replication backup for CP0 followed by a full volume dump of all target volumes that were paired with volumes of copy pool CP0 during fast replication backup (see Figure 7-36).

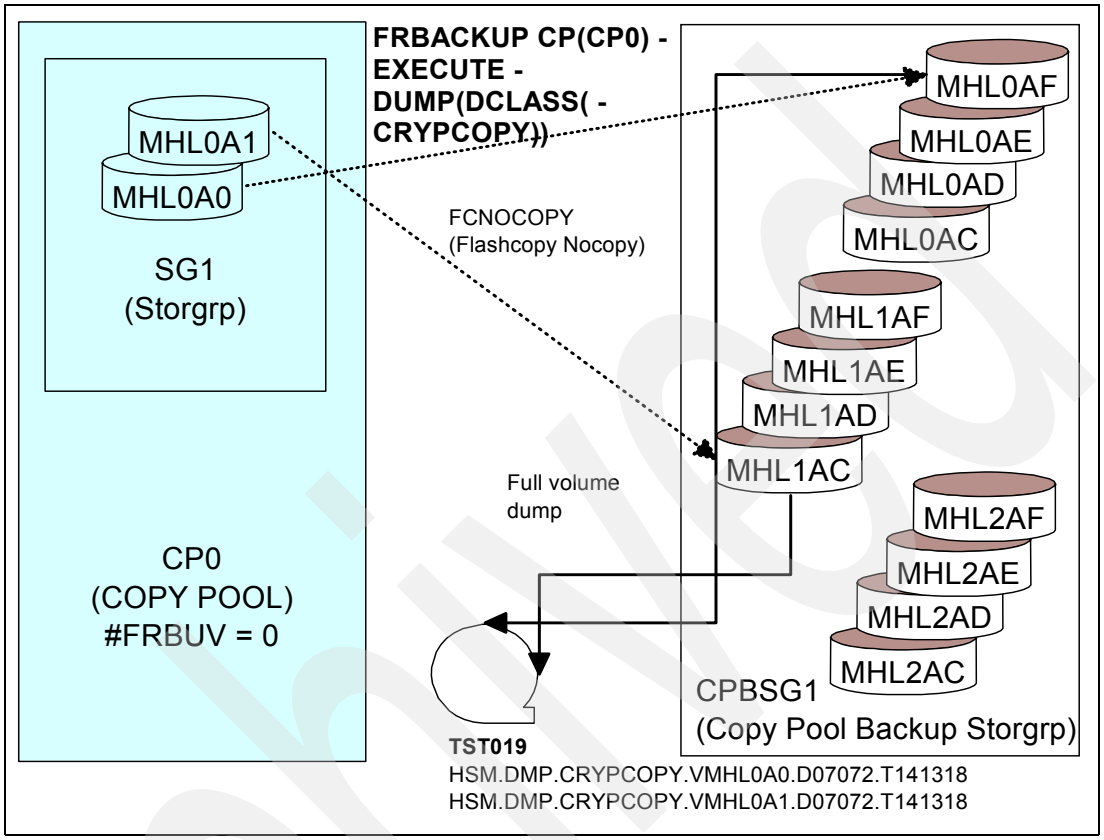


Figure 7-36 Fast replication dump with NOCOPY

When you look at the backup log of DFSMShsm (see Example 7-25) while executing this request, you can see that the copy commands for execution of the fast replication backup phase of the process include FCNC keywords.

Example 7-25 DFSMShsm backup log while executing a fast replication dump with NOCOPY

```
DFSMSHSM BACKUP LOG, TIME 18:05:18,  DATE 07/03/12
ARC1801I FAST REPLICATION DUMP IS STARTING FOR COPY POOL CP0, AT 18:13:14 ON 2007/03/13
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.072 18:13
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
 ARC0640I ARCFRTM -  COPY IDY(MHL0A0) ODY(MHL0AF) DUMPCOND FR(REQ) PUR ALLX ALLD(*) FCNC
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(MHL0A1) ODY(MHL1AC) DUMPCOND FR(REQ) PUR ALLX ALLD(*) FCNC
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2007.072 18:13:14 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2007.072 18:13:14 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
 ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2007.072 18:13:14 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (002)-TOMI (02), VOLUME MHL0A0 WAS COPIED USING A FAST REPLICATION FUNCTION
```

```
ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2007.072 18:13:14 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2007.072 18:13:14 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2007.072 18:13:14 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHL0A1 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2007.072 18:13:14 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2007.072 18:13:14 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2007.072 18:13:14 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL CP0
ARC1805I (CONT.) MHL0A0
ARC1805I (CONT.) MHL0A1
ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED FOR COPY POOL CP0, AT 18:43:45 ON 2007/03/13, FUNCTION RC=0000,
 MAXIMUM VOLUME RC=0000
```

Execution of a LIST command (as shown in Example 7-26) at the beginning of the process of a fast replication dump shows a FASTREPLICATIONSTATE of DUMPONLY (see Figure 7-37), which means that you cannot use this fast replication backup for recovery of the copy pool.

Example 7-26 LIST CP(CP0) command

```
HSEND LIST CP(CP0) ALLVOLS
```

The DUMPSTATE is FAILED on the version level as well as on the dump class level because the full volume dumps of the target volumes are still in progress (see Figure 7-37).

```
1-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:14:40 ON 07/0

0COPYPOOL=CP0

 VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
   001       Y      2007/03/13      18:13:14   DUMPONLY              FAILED
 TOKEN(C)=C''
 TOKEN(H)=X''
 TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
0
 DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
 CRYPCOPY       N       FAILED       *****     2008/03/03      N


 SOURCE   DUMPVOLS                                          DEVICE TYPE
 MHL0A0   ******                                             ********
 FILE SEQ=**, DSNAME=**
 MHL0A1   ******                                             ********
 FILE SEQ=**, DSNAME=**

0----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-37 Output of LIST CP(CP0) command while DUMPSTATE was FAILED

When we repeated the same LIST command, after some minutes we saw messages as shown in Figure 7-38.

Since at least one copy pool volume was successfully dumped to tape, the DUMPSTATE switched from FAILED to PARTIAL.

The target volume that was successfully dumped so far is no longer reported as a member of a volume pair.

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:43:36 ON 07/03

COPYPOOL=CP0

VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
  001       Y      2007/03/13      18:13:14   DUMPONLY              PARTIAL
TOKEN(C)=C''
TOKEN(H)=X''
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHL0A0 -         MHL0A1 - MHL1AC

DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
CRYPCOPY       N       PARTIAL      00001     2008/03/03      N

            HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
            NO      KEYPW    CLRAES128  PASSWORD

SOURCE   DUMPVOLS                                          DEVICE TYPE
MHL0A0   TST019                                             3590-1
FILE SEQ=01, DSNAME=HSM.DMP.CRYPCOPY.VMHL0A0.D07072.T141318
MHL0A1   ******                                             ********
FILE SEQ=**, DSNAME=**

----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-38 Output of LIST CP(CP0) command while DUMPSTATE was PARTIAL

When all copy pool volumes were successfully dumped to tape, the same LIST command shows output as shown in Figure 7-39:

- DUMPSTATE was changing to ALLCOMPLETE / COMPLETE.
- FASTREPLICATIONSTATE changed from DUMPONLY to NONE.
- The former shown target volumes of the volume pairs are no longer presented.

Volumes MHL0AF and MHL1AC can be reused by fast replication for any other FlashCopy relationships.

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:44:36 ON 07/03

COPYPOOL=CP0

VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
  001       Y      2007/03/13      18:13:14   NONE                  ALLCOMPLETE
TOKEN(C)=C''
TOKEN(H)=X''
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHL0A0 - ******  MHL0A1 - ******

DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
CRYPCOPY       N       COMPLETE     00002     2008/03/03      N

            HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
            NO      KEYPW    CLRAES128  PASSWORD

SOURCE   DUMPVOLS                                          DEVICE TYPE
MHL0A0   TST019                                              3590-1
FILE SEQ=01, DSNAME=HSM.DMP.CRYPCOPY.VMHL0A0.D07072.T141318
MHL0A1   TST019                                              3590-1
FILE SEQ=02, DSNAME=HSM.DMP.CRYPCOPY.VMHL0A1.D07072.T141318

----- END OF -- COPY POOL -- LISTING -----
```

Figure 7-39 Output of LIST CP(CP0) command when DUMPSTATE was ALLCOMPLETE

7.3.5 Two kinds of dump copies

DFSMShsm can create dump copies of Level0 volumes by using one of the following methods:

- Automatic dump
- BACKVOL command with keyword DUMP
- FRBACKUP command with keyword DUMP or keyword DUMPONLY

In all cases a dump generation record (DGN) in the BCDS describes each dump copy and a dump volume record (DVL) in the BCDS describes each dump volume.

When we use a list command, as shown in Example 7-27, we can list the associated dump copies for each level 0 volume.

Example 7-27 Request for a list of associated dump copies of a level 0 volume

```
HSEND LI PVOL(MHL0A0) ALLDUMPS BCDS
```

An example of the output of the list command is shown in Example 7-28.

Example 7-28 Output of LI PVOL(MHL0A0) ALLDUMPS BCDS command

```
1- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-BCDS--- ALLDUMPS----- AT 19:56:38 ON 07/02/28 FOR SYSTEM=SC64

 SOURCE                                          SET OF DUMP
 VOLSER GEN SMS  DUMPED     TIME    CLASS     EXP DATE   VOLSERS ❶

 MHL0A0  00 YES 07/02/28  15:52:42  MHLRES    07/03/07  TST015 ****** ****** ****** ****** ****** ****** ****** ******
 FAST REPLICATION ASSOCIATED DATA FOR VOLUME MHL0A0:
 COPY POOL                      GEN   DUMPED     TIME    CLASS     EXP DATE   SET OF DUMP VOLSERS ❷

 CP1                             00 2007/02/27 17:03:06 DCREDB18  2008/02/18  TST028 ****** ****** ****** ****** ******
                                                        PLUSCOPY  2008/02/18  TST024 ****** ****** ****** ****** ******
                                 01 2007/02/26 14:00:35 DCREDB18  2008/02/17  TST005 ****** ****** ****** ****** ******
                                 02 2007/02/22 16:31:22 DCREDB18  2008/02/13  TST025 ****** ****** ****** ****** ******
                                 03 2007/02/19 18:56:14 DCREDB18  2008/02/10  TST020 ****** ****** ****** ****** ******
                                 04 2007/02/19 14:49:20 DCREDB18  2008/02/10  TST019 ****** ****** ****** ****** ******
----- END OF - PRIMARY VOLUME - LISTING -----
```

As you can see, there are two lists of generations of dump copies presented in the list:

- The first list to be seen (❶) results from full volume dumps of volume MHL0A0.
- The second list (❷) results from full volume dumps of target volumes that appeared to be volume MHL0A0 (same contents but with different volume label in cylinder 0 track 0).

The basic difference between the traditional full volume dump copies and the fast replication associated full volume dump copies is that the fast replication associated dump copies are created from volumes that were the target of a FULL COPY DUMPCONDITIONING operation. This results in an incorrect volume serial number (that of the target volume) being dumped from record three of cylinder 0 track 0. You do not need to worry about that at any time because during any reuse of a dump copy like this for full volume recovery purposes DFSMSdss remembers that this is a dump copy of a conditioned volume and will take appropriate action to restore the volume label as of the source volume's (see Figure 7-40).

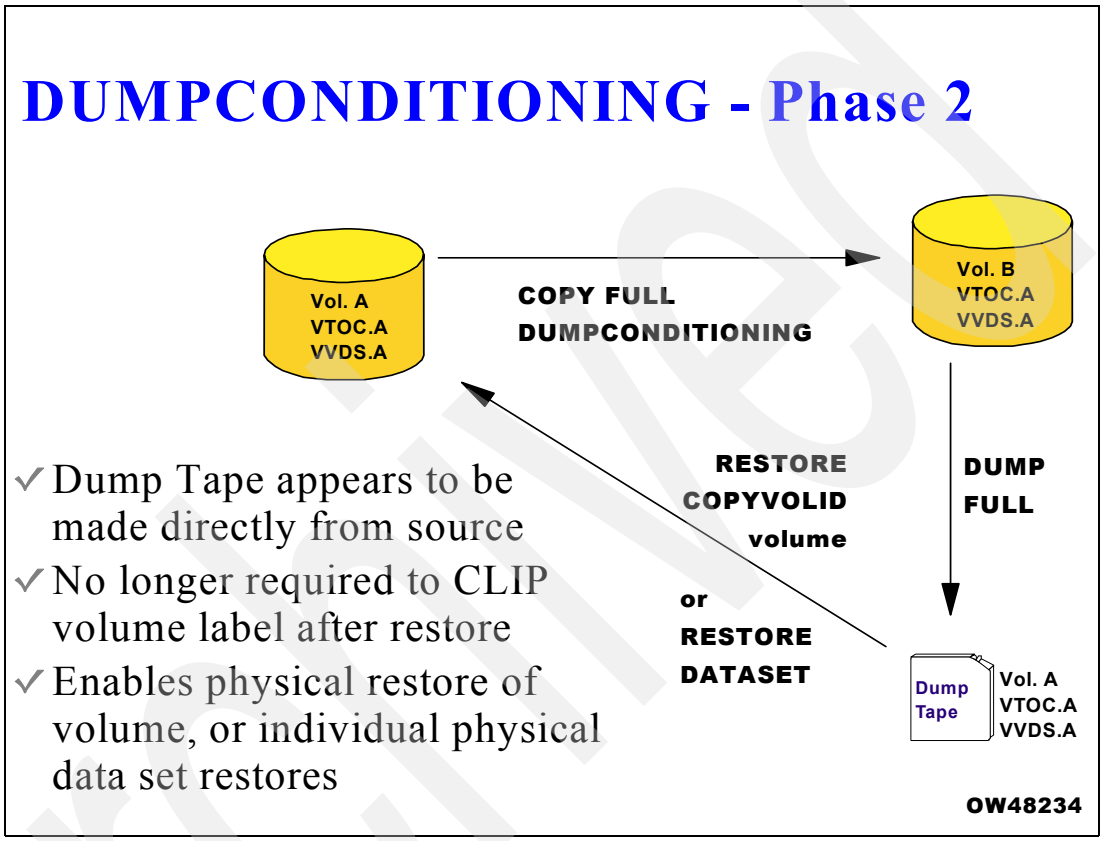


Figure 7-40 DUMPCONDITIONING

To learn more about DUMPCONDITIONING see the manuals:

- *DFSMSdss Storage Administration Guide,* SC26-0423
- *DFSMSdfp Storage Administration Reference,* SC26-7402
- APAR OW48234.

The two types of full volume dump copies look very similar but must be treated in different ways by DFSMShsm.

Dump generation records of dump copies that are not associated with fast replication backup are referenced by the BCDS eligible volume record (MCP) of the related volumes and can only be used by the commands:

- HRECOVER dsname
- HSEND RECOVER dsname FROMDUMP
- HSEND RECOVER * TOVOLUME(volser) FROMDUMP

In case of a fast replication dump copy (that was created from a volume of a copy pool backup storage group), the dump generation records are not referenced by the BCDS eligible

volume record (MCP). Such dump generations are referenced by the new fast replication records (FRx®) and are exclusively available for recovery by the FRREVCOV command.

See Table 7-1 for a detailed comparison between traditional dump copies that DFSMShsm has been providing for the last 20 years and the new dump copies created from fast replication target volumes that are available since z/OS DFSMShsm V1.8.

Table 7-1 Comparison of fast replication dump copies and other dump copies

	Dump of pool storage group volumes	**Dump of copy pool backup storage group volumes**
Created by command	BACKVOL volser DUMP	FRBACKUP CP(cpname) {DUMP / DUMPONLY}
Created during Autodump	Yes	Yes
Described by DGN record	Yes	Yes
Described by DVL record	Yes	Yes
Referenced by MCP record	Yes	No
Available for use with RECOVER dsn command	Yes	No
Available for use with RECOVER * command	Yes	No
Available for use with command HRECOVER dsn	Yes	No
Available for use with FRRECOV command	No	Yes
Reported with LIST DVOL command	Yes	Yes
Reported with command LIST PVOL(volser) ALLDUMPS BCDS	Yes	Yes
Reported with command LIST COPYPOOL DUMPVOLS	No	Yes
Reported with command LIST COPYPOOL ALLVOLS	No	Yes
Can be used by DFSMSdss in batch directly for recovery of volumes and data sets	Yes	Yes
Warning in case of one of the volumes of a storage group was not successfully dumped	No	Yes
Maximum number of dump copies	100 generations up to 5 copies per generation	85 generations up to 5 copies per generation
Maximum number of parallel processes during creation	SETSYS MAXDUMPTASKS(32)	SETSYS MAXDUMPTASKS(32)
Maximum number of parallel processes during recovery of volumes by DFSMShsm	1	1

	Dump of pool storage group volumes	Dump of copy pool backup storage group volumes
Maximum number of parallel processes during recovery of data sets	SETSYS MAXDSTAPERECOVERTASKS(64)	SETSYS MAXDSTAPERECOVERTASKS(64)

7.3.6 Recovery from fast replication dumps

Fast replication recovery processing was enhanced with z/OS DFSMS 1.8 to support recovery from dumps of copy pool volumes.

The FRRECOV command allows request for recovery of a single volume or data sets when you refer to a replication dump copy.

Note: Recovery of an entire copy pool by using just one command can only be performed from the DASD backup copies.

Recovery of a single volume from a fast replication dump copy

Using the command shown in Example 7-29 selects the latest version (GEN(0)) of the set of replication dump copies of copy pool CP1 for the recovery of volume MHL0A0. The FROMCOPYPOOL is an optional keyword that must be used if the volume that is being recovered resides within a storage group that is shared by multiple copy pools.

Example 7-29 FRRECOV command to recover a single volume from a dump copy

```
FRRECOV TOVOLUME(MHL0A1) FROMCOPYPOOL(CP1) FROMDUMP
```

If a version other than the current version is to be recovered, then use the GENERATION, VERSION, DATE, or TOKEN keyword. There is no option to perform a recovery at the storage group level.

As with FRBACKUP, DFSMShsm can process up to 64 concurrent invocations of DFSMSdss, with 15 being the default.

Messages that were issued during execution of the FRRECOV command are in the dump activity log (see Example 7-30).

Example 7-30 Excerpt from the dump log while processing a fast replication recovery at the volume level from dump

```
DFSMSHSM DUMP LOG, TIME 17:57:52,  DATE 07/03/12
 ARC0622I FULL VOLUME RESTORE STARTING ON VOLUME MHL0A0 AT 17:59:16 ON 2007/03/12, SYSTEM SC64, TASK ID=ARCGRVOL
 ARC0640I ARCGRVOL - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.071 17:59
 ARC0640I ARCGRVOL - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I ARCGRVOL -  RESTORE FULL INDDNAME(SYS00332) OUTDDNAME(SYS00331) -
 ARC0640I ARCGRVOL -  PURGE COPYVOLID CANCELERROR
 ARC0640I ARCGRVOL - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
 ARC0640I ARCGRVOL - ADR109I (R/I)-RI01 (01), 2007.071 17:59:16 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
 ARC0640I ARCGRVOL - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I ARCGRVOL - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I ARCGRVOL - ADR006I (001)-STEND(01), 2007.071 17:59:16 EXECUTION BEGINS
 ARC0640I ARCGRVOL - ADR780I (001)-TDFP (01),
  THE INPUT DUMP DATA SET BEING PROCESSED IS IN FULL VOLUME FORMAT AND WAS CREATED BY DFSMSDSS VERSION 1
 ARC0640I ARCGRVOL -                          RELEASE 8 MODIFICATION LEVEL 0
 ARC0640I ARCGRVOL - ADR808I (001)-TDFP (01),
  THE INPUT DUMP DATA SET BEING PROCESSED WAS CREATED FROM A CONDITIONED VOLUME
 ARC0640I ARCGRVOL - ADR460I (001)-UTMSG(01), UTILITY GENERATED MESSAGES FOLLOW FOR VOLUME MHL0A0
 ARC0640I ARCGRVOL -  ICKDSF - MVS/ESA    DEVICE SUPPORT FACILITIES 17.0               TIME: 18:04:57        03/12/07
```

```
    PAGE   1
 ARC0640I ARCGRVOL -
 ARC0640I ARCGRVOL -       BUILDIX DDNAME(SYS00331) IXVTOC
 ARC0640I ARCGRVOL -  ICK01502I BUILDIX FUNCTION STARTED
 ARC0640I ARCGRVOL -  ICK00700I DEVICE INFORMATION FOR 8005 IS CURRENTLY AS FOLLOWS:
 ARC0640I ARCGRVOL -            PHYSICAL DEVICE = 3390
 ARC0640I ARCGRVOL -            STORAGE CONTROLLER = 2105
 ARC0640I ARCGRVOL -            STORAGE CONTROL DESCRIPTOR = E8
 ARC0640I ARCGRVOL -            DEVICE DESCRIPTOR = 0A
 ARC0640I ARCGRVOL -            ADDITIONAL DEVICE INFORMATION = 4A000035
 ARC0640I ARCGRVOL -            TRKS/CYL = 15, # PRIMARY CYLS = 3339
 ARC0640I ARCGRVOL -  ICK04000I DEVICE IS IN SIMPLEX STATE
 ARC0640I ARCGRVOL -  ICK01503I 8005 REQUEST RECEIVED TO CONVERT VTOC TO IXFORMAT
 ARC0640I ARCGRVOL -  ICK01504I 8005 VTOC FORMAT IS CURRENTLY OSFORMAT, REQUEST ACCEPTED
 ARC0640I ARCGRVOL -  ICK01513I 8005  BUILDIX PROCESSING COMPLETED:  VTOC IS NOW IN IXFORMAT
 ARC0640I ARCGRVOL -  ICK01317I VTOC-INDEX IS LOCATED AT CCHH=X'0001 0000' AND IS    30 TRACKS.
 ARC0640I ARCGRVOL -            18:05:05    03/12/07
 ARC0640I ARCGRVOL -
 ARC0640I ARCGRVOL -  ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
 ARC0640I ARCGRVOL - ADR006I (001)-STEND(02), 2007.071 18:05:16 EXECUTION ENDS
 ARC0640I ARCGRVOL - PAGE 0002     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.071 17:59
 ARC0640I ARCGRVOL - ADR013I (001)-CLTSK(01), 2007.071 18:05:16 TASK COMPLETED WITH RETURN CODE 0000
 ARC0640I ARCGRVOL - ADR012I (SCH)-DSSU (01), 2007.071 18:05:16 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS
0000
 ARC0400I VOLUME MHLOA0 IS 26% FREE, 00000018 FREE TRACK(S), 000900 FREE CYLINDER(S), FRAG .008
 ARC0401I LARGEST EXTENTS FOR MHLOA0 ARE CYLINDERS     890, TRACKS    1335
 ARC0402I VTOC FOR MHLOA0 IS 0090 TRACKS(04500 DSCBS), 04489 FREE DSCBS(99% OF TOTAL)
 ARC0623I FULL VOLUME RESTORE OF VOLUME MHLOA0 ENDING AT 18:05:18, DCLASS=DCREDB18, DGEN=000, DATE=2007/03/12,
  PROCESSING SUCCESSFUL
 ARC1802I FAST REPLICATION RECOVERY FROM DUMP HAS COMPLETED FOR VOLUME MHLOA0, AT 18:05:18 ON 2007/03/12,
  FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
  DFSMSHSM DUMP LOG, TIME 18:05:18,  DATE 07/03/12
```

Volume restore FROMDUMP(APPLYINCREMENTAL)

You can request a recovery from a fast replication backup copy on tape that includes APPLYINCREMENTAL processing by using a command, as shown in Example 7-31.

Example 7-31 FRRECOV command that includes FROMDUMP(APPLYINCREMENTAL)

```
HSEND FRRECOV TOVOLUME(MHLOA0) FROMCP(CP1) FROMDUMP(APPLYINCREMENTAL) GEN(0)
```

This process is well-known from the command:

```
RECOVER * TOVOLUME(...) FROMDUMP(APPLYINCREMENTAL)
```

It works exactly the same way. After successful restore of a DASD volume from a full volume dump copy, an incremental volume recovery process follows on. A data set will be selected for processing during the incremental recovery phase if the data set was backed up by command or during volume backup after the creation of the fast replication backup.

In Figure 7-41 you can see the messages from the backup log when we executed the command as shown in Example 7-31 on page 223. Note that there was an ICF catalog on volume MHL0A0.

```
ARC1801I FAST REPLICATION RECOVERY FROM DUMP IS STARTING FOR VOLUME MHL0A0, AT 17:59:16 ON 2007/03/12
 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
 IDC0604I DATA SET BEING IMPORTED WAS EXPORTED ON   03/12/07 AT 17:53:45
 IGD01010I SG ACS GETS CONTROL &ACSENVIR=RECOVER
 IGD01010I SG ACS GETS CONTROL &ACSENVIR=RECOVER
 IDC0181I MANAGEMENTCLASS USED IS MCDB22
 IDC0181I STORAGECLASS USED IS HSMFR
 IDC0508I DATA ALLOCATION STATUS FOR VOLUME MHL0A0 IS 0
 IDC0509I INDEX ALLOCATION STATUS FOR VOLUME MHL0A0 IS 0
 IDC01654I ALIASES FROM THE PORTABLE DATA SET WERE NOT DEFINED
           TESTFR
 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
 IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
 ARC0778I DATA SET UCAT.TESTFR WAS RECOVERED FROM A BACKUP MADE AT 17:53:45 ON 2007/03/12 WITHOUT SERIALIZATION
 ARC0734I ACTION=RECOVER FRVOL=SBXHS5 TOVOL=MHL0A0 TRACKS=    16 RC=   0, REASON=    0, AGE= ***, DSN=UCAT.TESTFR
 ARC0734I ACTION=RECOVER FRVOL=****** TOVOL=MHL0A0 TRACKS=   *** RC=  43, REASON=   18, AGE= ***,
DSN=SYS1.VVDS.VMHL0A0
 ARC0734I ACTION=RECOVER FRVOL=****** TOVOL=MHL0A0 TRACKS=   *** RC=  58, REASON=   36, AGE= ***,
  DSN=TESTFR.DCOLLECT.ESDS01
 ARC0734I ACTION=RECOVER FRVOL=****** TOVOL=MHL0A0 TRACKS=   *** RC=  58, REASON=   36, AGE= ***,
  DSN=TESTFR.MHLRES2.MVOL.DATA
 ARC0734I ACTION=RCVSCHD FRVOL=****** TOVOL=MHL0A0 TRACKS=   *** RC=   0, REASON=    0, AGE= ***,
DSN=TESTFR.CNTL.JCL
 ARC0734I ACTION=RCVSCHD FRVOL=****** TOVOL=MHL0A0 TRACKS=   *** RC=   0, REASON=    0, AGE= ***,
  DSN=TESTFR.DCOLLECT.ESDS0
ARC0778I DATA SET TESTFR.CNTL.JCL WAS RECOVERED FROM A BACKUP MADE AT 17:53:19 ON 2007/03/12
 ARC0734I ACTION=RECOVER FRVOL=SBXHS4 TOVOL=MHL0A0 TRACKS=   150 RC=   0, REASON=    0, AGE= ***,
DSN=TESTFR.CNTL.JCL
 ARC0778I DATA SET TESTFR.DCOLLECT.ESDS0 WAS RECOVERED FROM A BACKUP MADE AT 17:55:21 ON 2007/03/12
 ARC0734I ACTION=RECOVER FRVOL=SBXHS5 TOVOL=MHL0A0 TRACKS= 13350 RC=   0, REASON=    0, AGE= ***,
  DSN=TESTFR.DCOLLECT.ESDS0
 ARC0773I RECOVERY ENDED ON VOLUME MHL0A0 TIME 18:06:18
```

Figure 7-41 Backup log messages from fast replication recovery including APPLYINCREMENTAL

Additional messages can be found in the DFSMShsm dump log, as shown in Example 7-30 on page 222.

Recovery of data sets from a fast replication dump copy

This is covered in 7.4, “Data set recovery” on page 224.

7.4 Data set recovery

Starting with z/OS DFSMShsm V1.8, you can use the FRROCOV command to recover an individual data set from either DASD or tape fast replication backup copies. Since DFSMShsm does not record any information at the data set level in the BCDS during fast replication backup, the current catalog entry of the data set is looked up by DFSMShsm in order to see which volumes are involved in the recovery process. DFSMShsm always riles on the catalog entry, and the FRRECOV command fails either any of the following is true:

- The catalog entry is not available.
- The catalog entry points to different volumes as when the fast replication backup was created.

Note: To be eligible for recovery by using the FRRECOV command, a data set must be cataloged and the catalog entry must point to the same volumes on which the data set resided when the backup copy was created.

The keywords that are available to request a data set recovery from a fast replication backup are shown in Figure 7-42.

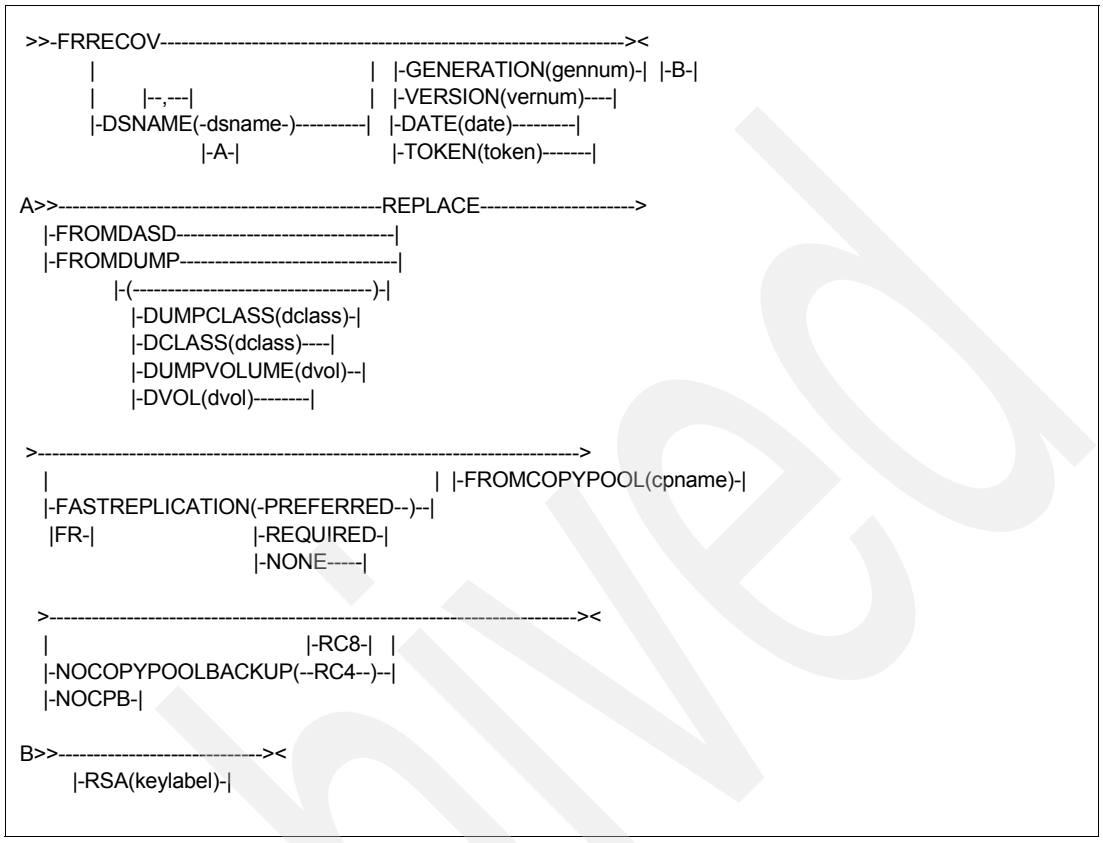


Figure 7-42 FRRECOV command keywords for data set recovery

For dsname you can substitute one or more fully or partially specified data set names. For partially specified data set names you can use wild cards of the form:

%	To represent a single character
*****	To represent a single qualifier or parts of a qualifier
****	To represent anything (even more than one qualifier)

The standard search order for catalogs applies. You may not specify ** as the high-level qualifier or as the only qualifier.

7.4.1 Data set filtering

You can request, for example, the recovery of two data sets by using the command as shown in Example 7-32.

Example 7-32 FRRECOV DSNAME command to recover two data sets

```
HSEND FRRECOV DSNAME(TESTFR.CMD.CLIST TESTFR.CNTL.JCL) REPLACE FROMCP(CP1)
```

Note that the specification of FROMCP(CP1) is required because the volumes where the data sets reside are assigned to more than one copy pool (CP0 and CP1). If you omit FROMCP you end up with message ARC1866I, as shown in Figure 7-43.

```
ARC1866I FAST REPLICATION RECOVERY HAS FAILED FOR DATA SET TESTFR.CMD.CLIST,
ARC1866I (CONT.) RC=0020
```

Figure 7-43 ARC1866I RC=0020

Messages issued during the data set recovery process can be found in the:

- JESMSGLG (Example 7-33)
- Backup log (Example 7-34)
- Dump log (Example 7-35)

Example 7-33 JESMSGLG during FRRECOV processing of two data sets

```
16.06.15 STC21322  ARC1861I THE FOLLOWING 0002 DATA SET(S) WERE  870
   870             ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
   870             ARC1861I (CONT.) SET RECOVERY:
16.06.15 STC21322  ARC1861I (CONT.) TESTFR.CMD.CLIST, COPYPOOL=CP1, DEVTYPE=DASD
16.06.15 STC21322  ARC1861I (CONT.) TESTFR.CNTL.JCL, COPYPOOL=CP1, DEVTYPE=DASD
16.06.15 STC21322  ARC1802I FAST REPLICATION DATA SET RECOVERY HAS  873
   873             ARC1802I (CONT.) COMPLETED FOR DATA SET TESTFR.CMD.CLIST, ***, AT
   873             ARC1802I (CONT.) 16:06:15 ON 2007/03/19, FUNCTION RC=0000, MAXIMUM
   873             ARC1802I (CONT.) DATA SET RC=0000
```

The catalog search found entries for both data sets, so both of them are selected for further processing.

Example 7-34 Backup log during FRRECOV processing of two data sets

```
ARC1801I FAST REPLICATION DATA SET RECOVERY IS STARTING FOR DATA SET TESTFR.CMD.CLIST, ***, AT 16:06:14 ON 2007/03/19
 ARC1861I THE FOLLOWING 0002 DATA SET(S) WERE SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA SET RECOVERY:
 ARC1861I (CONT.) TESTFR.CMD.CLIST, COPYPOOL=CP1, DEVTYPE=DASD
 ARC1861I (CONT.) TESTFR.CNTL.JCL, COPYPOOL=CP1, DEVTYPE=DASD
 ARC1802I FAST REPLICATION DATA SET RECOVERY HAS COMPLETED FOR DATA SET TESTFR.CMD.CLIST, ***, AT 16:06:15 ON 2007/03/19,
  FUNCTION RC=0000, MAXIMUM DATA SET RC=0000
```

The dump log shows the keywords that are generated for execution of the data set copy. It is a physical data set operation (keyword PHYSINDY) that requests the same management class, storage class, and volume as was found in the catalog entry of each data set. While BYPASSACS is requested as well, this does not really change the catalog entry of the data sets.

In this particular case, preallocated data sets were found to be in place, which fitted exactly the needs of the data sets that were to be recovered.

Example 7-35 Dump log during FRRECOV processing of two data sets

```
DFSMSHSM DUMP LOG, TIME 18:43:46,  DATE 07/03/13
 ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.078 16:06
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I GDSN02 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.078 16:06
 ARC0640I GDSN02 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
 ARC0640I GDSN01 -  COPY DS(INC(TESTFR.CMD.CLIST                          )) -
 ARC0640I GDSN02 -  COPY DS(INC(TESTFR.CNTL.JCL                           )) -
 ARC0640I GDSN01 -  PHYSINDY(MHL1AC) OUTDYNAM(MHLOA1) -
 ARC0640I GDSN02 -  PHYSINDY(MHLOAF) OUTDYNAM(MHLOA0) -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.CMD.CLIST                         ) -
 ARC0640I GDSN02 -  BYPASSACS(TESTFR.CNTL.JCL                          ) -
 ARC0640I GDSN01 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN02 -  FASTREPLICATION(PREFERRED) -
```

```
ARC0640I GDSNO1 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
ARC0640I GDSNO2 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
ARC0640I GDSNO1 -  TGTGDS( DEFERRED) -
ARC0640I GDSNO2 -  TGTGDS( DEFERRED) -
ARC0640I GDSNO1 -  FORCECP(0) PROCESS(SYS1) -
ARC0640I GDSNO2 -  FORCECP(0) PROCESS(SYS1) -
ARC0640I GDSNO1 -  STORCLAS(HSMFR                      ) -
ARC0640I GDSNO2 -  STORCLAS(HSMFR                      ) -
ARC0640I GDSNO1 -  MGMTCLAS(MCDB22                     )
ARC0640I GDSNO1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I GDSNO2 -  MGMTCLAS(HFS                        )
ARC0640I GDSNO1 - ADR109I (R/I)-RI01 (01), 2007.078 16:06:14 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I GDSNO2 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I GDSNO2 - ADR109I (R/I)-RI01 (01), 2007.078 16:06:14 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I GDSNO1 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I GDSNO1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I GDSNO2 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I GDSNO2 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I GDSNO1 - ADR006I (001)-STEND(01), 2007.078 16:06:14 EXECUTION BEGINS
ARC0640I GDSNO2 - ADR006I (001)-STEND(01), 2007.078 16:06:14 EXECUTION BEGINS
ARC0640I GDSNO1 - ADR442I (001)-PPRNV(01), DATA SET TESTFR.CMD.CLIST PREALLOCATED, ON VOLUME(S): MHL0A1
ARC0640I GDSNO2 - ADR442I (001)-PPRNV(01), DATA SET TESTFR.CNTL.JCL PREALLOCATED, ON VOLUME(S): MHL0A0
ARC0640I GDSNO1 - ADR806I (001)-TOMI (01), DATA SET TESTFR.CMD.CLIST COPIED USING A FAST REPLICATION FUNCTION
ARC0640I GDSNO2 - ADR806I (001)-TOMI (01), DATA SET TESTFR.CNTL.JCL COPIED USING A FAST REPLICATION FUNCTION
ARC0640I GDSNO1 - ADR454I (001)-DDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC0640I GDSNO1 -                           TESTFR.CMD.CLIST
ARC0640I GDSNO1 - ADR006I (001)-STEND(02), 2007.078 16:06:14 EXECUTION ENDS
ARC0640I GDSNO2 - ADR454I (001)-DDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC0640I GDSNO2 -                           TESTFR.CNTL.JCL
ARC0640I GDSNO2 - ADR006I (001)-STEND(02), 2007.078 16:06:14 EXECUTION ENDS
ARC0640I GDSNO1 - ADR013I (001)-CLTSK(01), 2007.078 16:06:14 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I GDSNO1 - ADR012I (SCH)-DSSU (01), 2007.078 16:06:14 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC0640I GDSNO2 - ADR013I (001)-CLTSK(01), 2007.078 16:06:14 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I GDSNO2 - ADR012I (SCH)-DSSU (01), 2007.078 16:06:14 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Let us look at another example that requests data set filtering by some masking characters (Example 7-36).

Example 7-36 FRRECOV DSNAME command to recover a group of data sets

```
HSEND FRRECOV DSNAME(TESTFR.C*.* TESTFR.N*.*) REPLACE FROMCP(CP1)
```

When we did a catalog search by using the PDF utility DSLIST (3.4) providing the same data set name masks, we got a display as shown in Figure 7-44.

```
 Menu  Options  View  Utilities  Compilers  Help
------------------------------------------------------------------------------
DSLIST - Data Sets Matching TESTFR.C*                          2 new append(s)
Command ===>                                                  Scroll ===> CSR

Command - Enter "/" to select action                Message           Volume
------------------------------------------------------------------------------
        TESTFR.CMD.CLIST                                                MHL0A1
        TESTFR.CNTL.JCL                                                 MHL0A0
        TESTFR.NONFRBA.TEST1                                            SBOX1E
        TESTFR.NONFRBA.TEST2                                            SBOX1D
***************************** End of Data Set list ****************************
```

Figure 7-44 DSLIST display for data set name masks as used with FRRECOV

When you compare the results of data set filtering by using Figure 7-44 on page 227, Figure 7-45, and Example 7-37, you can verify that during processing of the FRRECOV command the same data sets are selected as when looking up the catalog by using DSLIST (3.4).

```
ARC1001I TESTFR.NONFRBA.TEST1 RECOVER FAILED, RC=0093, REAS=0066-0024
ARC1193I FAST REPLICATION RECOVERY FAILED
ARC1001I TESTFR.NONFRBA.TEST2 RECOVER FAILED, RC=0093, REAS=0066-0024
ARC1193I FAST REPLICATION RECOVERY FAILED
ARC1000I TESTFR.CMD.CLIST RECOVER PROCESSING ENDED
ARC1001I TESTFR.C*.*, *** FRRECOV FAILED, RC=0008, REAS=0000
ARC1808E ONE OR MORE FAILURES OCCURRED DURING FAST REPLICATION RECOVERY OF
ARC1808E (CONT.) DATA SET TESTFR.C*.*, ***
ARC1000I TESTFR.CNTL.JCL RECOVER PROCESSING ENDED
***
```

Figure 7-45 Messages returned to TSO session in case of processing a group of data sets

If some of the matching data set names cannot be recovered because there are no fast replication backups available, recovery fails for these data sets but command processing continues (see Figure 7-45 and Example 7-37) and all data sets currently cataloged on a recoverable copy pool volume will be recovered.

Example 7-37 Messages issued in backup log in case of processing a group of data sets

```
ARC1801I FAST REPLICATION DATA SET RECOVERY IS STARTING FOR DATA SET TESTFR.C*.*, ***, AT 18:12:53 ON 2007/03/19
 ARC1861I THE FOLLOWING 0002 DATA SET(S) WERE SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA SET RECOVERY:
 ARC1861I (CONT.) TESTFR.CMD.CLIST, COPYPOOL=CP1, DEVTYPE=DASD
 ARC1861I (CONT.) TESTFR.CNTL.JCL, COPYPOOL=CP1, DEVTYPE=DASD
ARC1862I THE FOLLOWING 0002 DATA SET(S) WERE NOT SELECTED FOR FAST REPLICATION DATA SET RECOVERY PROCESSING:
 ARC1862I (CONT.) TESTFR.NONFRBA.TEST1
 ARC1862I (CONT.) TESTFR.NONFRBA.TEST2
 ARC1802I FAST REPLICATION DATA SET RECOVERY HAS COMPLETED FOR DATA SET TESTFR.C*.*, ***, AT 18:12:53 ON 2007/03/19,
  FUNCTION RC=0008, MAXIMUM DATA SET RC=0066
```

7.4.2 Multi-volume data sets

Multi-volume data sets are supported by FRRECOV DSNAME command processing. But you should always keep in mind that recovering a multi-volume data set from full volume copies or dumps may fail. Even if the recovery process ends without any error messages you might end up in a corrupted data set because the data set's catalog entry is from than at the time of fast replication backup.

Suppose that we want to recover data set TESTFR.DCOLLECT ESDS01, which is a multi-volume VSAM data set. Its catalog entry is shown in Figure 7-46.

```
CLUSTER ------- TESTFR.DCOLLECT.ESDS01
     IN-CAT --- UCAT.TESTFR
     HISTORY
DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
      SMSDATA
        STORAGECLASS ------HSMFR     MANAGEMENTCLASS---MCDB22
        DATACLASS --------(NULL)     LBACKUP ---0000.000.0000
        BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
        BWO---------------(NULL)
      RLSDATA
        LOG ----------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG -------
----(NULL)
        VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
        LOGSTREAMID-----------------------------(NULL)
        RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
        RECOVERY TIMESTAMP GMT-------X'0000000000000000'
    DATA ------- TESTFR.DCOLLECT.ESDS01.DATA
      IN-CAT --- UCAT.TESTFR
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
        ACCOUNT-INFO-----------------------------------(NULL)
      VOLUMES
        VOLSER------------MHL0A0     DEVTYPE------X'3010200F'
        VOLSER------------MHL0A1     DEVTYPE------X'3010200F'
```

Figure 7-46 Output of LISTC command for TESTFR.DCOLLECT.ESDS01

Recovery when volume pointers in ICF are unchanged

The fast replication backup version we use for recovery is listed inFigure 7-47.

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 13:16:07 ON 07/03

COPYPOOL=CP1

VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
  006       Y      2007/03/12      16:01:45   RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C'EX6'
TOKEN(H)=X'C5E7F6'
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
```

Figure 7-47 LIST CP of the fast replication backup version to be used for recovery

The catalog entry of the data set did not change since the fast replication backup was made.

Execution of the command as shown in Example 7-38 leads to a perfectly recovered data set.

Example 7-38 FRRECOV DSNAME command

```
FRRECOV DSNAME(TESTFR.DCOLLECT.ESDS01) REPLACE FROMCP(CP1)
```

The messages in the dump log of DFSMShsm (see Example 7-39) show clearly that two copy commands are being executed internally by DFSMShsm:

- One for the data set's components on volume MHL0A0
- One for the data set's components on volume MHL0A1

Example 7-39 DFSMShsm dump log while processing FRRECOV for TESTFR.DCOLLECT.ESDS01

```
ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.081
18:45
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 -  COPY DS(INC(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  PHYSINDY(MHL0AF) OUTDYNAM(MHL0A0) -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                    ) -
 ARC0640I GDSN02 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.081
18:45
 ARC0640I GDSN01 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN02 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
 ARC0640I GDSN01 -  FORCECP(0) PROCESS(SYS1) -
 ARC0640I GDSN02 -  COPY DS(INC(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  STORCLAS(HSMFR                      ) -
 ARC0640I GDSN02 -  PHYSINDY(MHL1AC) OUTDYNAM(MHL0A1) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                     )
 ARC0640I GDSN02 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                    ) -
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I GDSN02 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2007.081 18:45:50 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN02 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
 ARC0640I GDSN02 -  FORCECP(0) PROCESS(SYS1) -
 ARC0640I GDSN02 -  STORCLAS(HSMFR                      ) -
 ARC0640I GDSN02 -  MGMTCLAS(MCDB22                     )
 ARC0640I GDSN02 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I GDSN02 - ADR109I (R/I)-RI01 (01), 2007.081 18:45:50 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN01 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN02 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
 ARC0640I GDSN02 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2007.081 18:45:50 EXECUTION BEGINS
 ARC0640I GDSN02 - ADR006I (001)-STEND(01), 2007.081 18:45:50 EXECUTION BEGINS
 ARC0640I GDSN01 - ADR442I (001)-PPRVS(01), DATA SET TESTFR.DCOLLECT.ESDS01 PREALLOCATED, IN
CATALOG UCAT.TESTFR,
  ON VOLUME(S): MHL0A0
 ARC0640I GDSN02 - ADR442I (001)-PPRVS(01), DATA SET TESTFR.DCOLLECT.ESDS01 PREALLOCATED, IN
CATALOG UCAT.TESTFR,
```

```
  ON VOLUME(S): MHL0A1
 ARC0640I GDSN02 - ADR806I (001)-TOMI (03), DATA SET TESTFR.DCOLLECT.ESDS01 COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I GDSN02 - ADR454I (001)-DDDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
 ARC0640I GDSN02 -                           CLUSTER NAME   TESTFR.DCOLLECT.ESDS01
 ARC0640I GDSN02 -                           COMPONENT NAME TESTFR.DCOLLECT.ESDS01.DATA
 ARC0640I GDSN02 - ADR006I (001)-STEND(02), 2007.081 18:45:51 EXECUTION ENDS
 ARC0640I GDSN02 - ADR013I (001)-CLTSK(01), 2007.081 18:45:51 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN02 - ADR012I (SCH)-DSSU (01), 2007.081 18:45:51 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
 ARC0640I GDSN01 - ADR806I (001)-TOMI (03), DATA SET TESTFR.DCOLLECT.ESDS01 COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I GDSN01 - ADR454I (001)-DDDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
 ARC0640I GDSN01 -                           CLUSTER NAME   TESTFR.DCOLLECT.ESDS01
 ARC0640I GDSN01 -                           COMPONENT NAME TESTFR.DCOLLECT.ESDS01.DATA
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2007.081 18:45:51 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2007.081 18:45:51 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2007.081 18:45:51 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
```

Recovery after volume count has been reduced

Now we have reorganized the data set and it is no longer made up of pieces spread among several volumes (see the catalog entry in Figure 7-48).

```
CLUSTER ------- TESTFR.DCOLLECT.ESDS01
     IN-CAT --- UCAT.TESTFR
     HISTORY
DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
      SMSDATA
        STORAGECLASS ------HSMFR     MANAGEMENTCLASS---MCDB22
        DATACLASS --------(NULL)     LBACKUP ---0000.000.0000
        BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
        BWO---------------(NULL)
      RLSDATA
        LOG ---------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG -------
----(NULL)
        VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
        LOGSTREAMID----------------------------(NULL)
        RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
        RECOVERY TIMESTAMP GMT-------X'0000000000000000'
    DATA ------- TESTFR.DCOLLECT.ESDS01.DATA
      IN-CAT --- UCAT.TESTFR
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
        ACCOUNT-INFO-----------------------------------(NULL)
      VOLUMES
        VOLSER------------MHL0A0     DEVTYPE------X'3010200F'
        VOLSER------------*          DEVTYPE------X'3010200F'
```

Figure 7-48 Output of LISTC command for TESTFR.DCOLLECT.ESDS01

The backup copy we use for recovery is still the same that we last used (see Figure 7-47 on page 229). Remember that during creation of this backup the data set was spread among the two volumes.

The recovery of the data set seems to work fine, as the messages in Example 7-40 lead us to believe.

Example 7-40 DFSMShsm dump log while processing FRRECOV for TESTFR.DCOLLECT.ESDS01

```
ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.081
19:36
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 -  COPY DS(INC(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  PHYSINDY(MHL0AF) OUTDYNAM(MHL0A0) -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                     ) -
 ARC0640I GDSN01 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN01 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
 ARC0640I GDSN01 -  FORCECP(0) PROCESS(SYS1) -
 ARC0640I GDSN01 -  STORCLAS(HSMFR                        ) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                       )
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2007.081 19:36:17 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN01 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2007.081 19:36:17 EXECUTION BEGINS
 ARC0640I GDSN01 - ADR442I (001)-PPRVS(01), DATA SET TESTFR.DCOLLECT.ESDS01 PREALLOCATED, IN
CATALOG UCAT.TESTFR,
  ON VOLUME(S): MHL0A0
 ARC0640I GDSN01 - ADR390I (001)-PPRVS(01),
  DATA SET TESTFR.DCOLLECT.ESDS01 WILL BE SCRATCHED FROM MHL0A0 BECAUSE OF UNMATCHED SIZE. IT
WILL BE
 ARC0640I GDSN01 -                              REALLOCATED
 ARC0640I GDSN01 - ADR431I (001)-DYNA (02), DATA SET TESTFR.DCOLLECT.ESDS01.DATA HAS BEEN
DELETED
 ARC0640I GDSN01 - ADR396I (001)-PCVSM(01),
  DATA SET CLUSTER TESTFR.DCOLLECT.ESDS01 COMPONENT TESTFR.DCOLLECT.ESDS01.DATA ALLOCATED, ON
VOLUME(S):
 ARC0640I GDSN01 -                              MHL0A0
 ARC0640I GDSN01 - ADR806I (001)-TOMI (03), DATA SET TESTFR.DCOLLECT.ESDS01 COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I GDSN01 - ADR454I (001)-DDDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
 ARC0640I GDSN01 -                               CLUSTER NAME   TESTFR.DCOLLECT.ESDS01
 ARC0640I GDSN01 -                               COMPONENT NAME TESTFR.DCOLLECT.ESDS01.DATA
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2007.081 19:36:17 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2007.081 19:36:17 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2007.081 19:36:17 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
```

But when we compare the contents of the data set after the FRRECOV process with the one before we did the recover, we see that there is some missing data — the piece of data that was backed up from MHL0A1.

The likelihood for a decrease in volume count is not as high as an increase in volume count (it depends on the installation's behavior), but this is an issue that you should keep in mind.

Note: Be very careful when you recover a multi-volume data set from a fast replication backup and when the volume count of the data set has decreased since the backup was made.

Recovery after volume count has been increased

We tested the option where the volume count has been increased. The data set is spread among two volumes at the time of recovery because of a volume overflow that took place some time since the last fast replication backup.

Now we use a different fast replication backup version (see Figure 7-49).

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 13:16:07 ON 07/03

COPYPOOL=CP1

VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
  007       Y      2007/03/22      19:39:37   RECOVERABLE           NONE
TOKEN(C)=C''
TOKEN(H)=X''
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHLOA0 - MHL1AD  MHLOA1 - MHL1AE
```

Figure 7-49 LIST CP of the fast replication backup version to be used for recovery

TESTFR.DCOLLECT.ESDS01 was, at time of backup, a single volume data set (see Figure 7-50).

```
CLUSTER ------- TESTFR.DCOLLECT.ESDS01
     IN-CAT --- UCAT.TESTFR
     HISTORY
DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
      SMSDATA
        STORAGECLASS ------HSMFR     MANAGEMENTCLASS---MCDB22
        DATACLASS --------(NULL)     LBACKUP ---0000.000.0000
        BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
        BWO---------------(NULL)
      RLSDATA
        LOG ----------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG -------
----(NULL)
        VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
        LOGSTREAMID-----------------------------(NULL)
        RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
        RECOVERY TIMESTAMP GMT-------X'0000000000000000'
    DATA ------- TESTFR.DCOLLECT.ESDS01.DATA
      IN-CAT --- UCAT.TESTFR
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
        ACCOUNT-INFO-----------------------------------(NULL)
      VOLUMES
        VOLSER------------MHL0A0     DEVTYPE------X'3010200F'
        VOLSER------------*          DEVTYPE------X'3010200F'
```

Figure 7-50 Output of LISTC command for TESTFR.DCOLLECT.ESDS01

The data set was successfully recovered by the process and the catalog entry was changed accordingly to show just a candidate volume instead of the second volume pointer.

As you can see in the dump log of DFSMShsm (see Example 7-41), a physical data set copy from MHL1AE to MHL0A1 was tried (GDSN02) but did not end successfully because the data set was not found on volume MHL1AE (remember that at backup time the data set had no extents on any volume other than MHL0A0).

Example 7-41 DFSMShsm dump log while processing FRRECOV for TESTFR.DCOLLECT.ESDS01

```
ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.082
12:41
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 -  COPY DS(INC(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  PHYSINDY(MHL1AD) OUTDYNAM(MHL0A0) -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                     ) -
 ARC0640I GDSN01 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN01 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
 ARC0640I GDSN01 -  FORCECP(0) PROCESS(SYS1) -
 ARC0640I GDSN01 -  STORCLAS(HSMFR                          ) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                         )
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
```

```
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2007.082 12:41:45 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN01 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2007.082 12:41:45 EXECUTION BEGINS
 ARC0640I GDSN02 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.082
12:41
 ARC0640I GDSN02 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN02 -  COPY DS(INC(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN02 -  PHYSINDY(MHL1AE) OUTDYNAM(MHL0A1) -
 ARC0640I GDSN02 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                    ) -
 ARC0640I GDSN02 -  FASTREPLICATION(PREFERRED) -
 ARC0640I GDSN02 -  ALLDATA(*) ALLEXCP REPLACEU CANCELERROR DEBUG(FRMSG(DTL)) -
 ARC0640I GDSN02 -  FORCECP(0) PROCESS(SYS1) -
 ARC0640I GDSN02 -  STORCLAS(HSMFR                        ) -
 ARC0640I GDSN02 -  MGMTCLAS(MCDB22                       )
 ARC0640I GDSN02 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ARC0640I GDSN02 - ADR109I (R/I)-RI01 (01), 2007.082 12:41:45 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN02 - ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
 ARC0640I GDSN02 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN02 - ADR006I (001)-STEND(01), 2007.082 12:41:45 EXECUTION BEGINS
 ARC0640I GDSN02 - ADR801I (001)-DDDS (01),
  DATA SET FILTERING IS COMPLETE. 0 OF 0 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0
FAILED FOR
 ARC0640I GDSN02 -                          OTHER REASONS.
 ARC0640I GDSN02 - ADR383W (001)-DDDS (01), DATA SET TESTFR.DCOLLECT.ESDS01 NOT SELECTED
 ARC0640I GDSN02 - ADR470W (001)-DDDS (04), NO DATA SETS SELECTED FOR PROCESSING
 ARC0640I GDSN02 - ADR006I (001)-STEND(02), 2007.082 12:41:45 EXECUTION ENDS
 ARC0640I GDSN02 - ADR013I (001)-CLTSK(01), 2007.082 12:41:45 TASK COMPLETED WITH RETURN CODE
0004
 ARC0640I GDSN02 - ADR012I (SCH)-DSSU (01),
  2007.082 12:41:45 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0004 FROM:
 ARC0640I GDSN02 -                          TASK    001
 ARC0640I GDSN01 - ADR442I (001)-PPRVS(01), DATA SET TESTFR.DCOLLECT.ESDS01 PREALLOCATED, IN
CATALOG UCAT.TESTFR,
  ON VOLUME(S): MHL0A0
 ARC0640I GDSN01 - ADR390I (001)-PPRVS(01),
  DATA SET TESTFR.DCOLLECT.ESDS01 WILL BE SCRATCHED FROM MHL0A0 BECAUSE OF UNMATCHED SIZE. IT
WILL BE
 ARC0640I GDSN01 -                          REALLOCATED
 ARC0640I GDSN01 - ADR431I (001)-DYNA (02), DATA SET TESTFR.DCOLLECT.ESDS01.DATA HAS BEEN
DELETED
 ARC0640I GDSN01 - ADR396I (001)-PCVSM(01),
  DATA SET CLUSTER TESTFR.DCOLLECT.ESDS01 COMPONENT TESTFR.DCOLLECT.ESDS01.DATA ALLOCATED, ON
VOLUME(S):
 ARC0640I GDSN01 -                          MHL0A0
 ARC0640I GDSN01 - ADR806I (001)-TOMI (03), DATA SET TESTFR.DCOLLECT.ESDS01 COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I GDSN01 - ADR454I (001)-DDDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
 ARC0640I GDSN01 -                           CLUSTER NAME   TESTFR.DCOLLECT.ESDS01
 ARC0640I GDSN01 -                           COMPONENT NAME TESTFR.DCOLLECT.ESDS01.DATA
```

```
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2007.082 12:41:46 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2007.082 12:41:46 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2007.082 12:41:46 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
```

During further processing the current data set was deleted, reallocated, and restored from the copy that was found on MHL1AD, which is the target volume of MHL0A0.

Conclusion

If you recover multi-volume data sets you should check the results of FRRECOV.

But how can you remember the catalog entry of a data set at backup time? Basically, you need some kind of VTOC copies that you can provide easily as part of the fast replication dump process. Another option is to create your own meta data by using IDCAMS DCOLLECT. Go to 7.4.4, “Hints and tips for the recovery of data sets that no longer exist” on page 239, for more information.

7.4.3 Data set recovery from fast replication dumps

Fast replication recovery at the data set level is initiated by a command, as shown in Example 7-42.

Example 7-42 FRRECOV DSNAME command to request recovery from fast replication dump

```
HSEND FRRECOV DSNAME(TESTFR.DCOLLECT.ESDS01) REPLACE FROMCP(CP1) GEN(1) FROMDUMP
```

The recovery was done successfully based on the catalog entry, as shown in Figure 7-51.

```
CLUSTER ------- TESTFR.DCOLLECT.ESDS01
     IN-CAT --- UCAT.TESTFR
     HISTORY
DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
      SMSDATA
        STORAGECLASS ------HSMFR     MANAGEMENTCLASS---MCDB22
        DATACLASS --------(NULL)     LBACKUP ---0000.000.0000
        BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
        BWO---------------(NULL)
      RLSDATA
        LOG ----------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG -------
----(NULL)
        VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
        LOGSTREAMID-----------------------------(NULL)
        RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
        RECOVERY TIMESTAMP GMT-------X'0000000000000000'
    DATA ------- TESTFR.DCOLLECT.ESDS01.DATA
      IN-CAT --- UCAT.TESTFR
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2007.058
        RELEASE----------------2     EXPIRATION------0000.000
        ACCOUNT-INFO-----------------------------------(NULL)
      VOLUMES
        VOLSER------------MHL0A0     DEVTYPE------X'3010200F'
        VOLSER------------MHL0A1     DEVTYPE------X'3010200F'
```

Figure 7-51 Output of LISTC command for TESTFR.DCOLLECT.ESDS01

The fast replication dump that we requested is listed in Figure 7-52.

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:02:44 ON 07/03

COPYPOOL=CP1

VERSION  VTOCENQ     DATE           TIME     FASTREPLICATIONSTATE  DUMPSTATE
  006      Y      2007/03/12      16:01:45   RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C'EX6'
TOKEN(H)=X'C5E7F6'
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC

DUMPCLASS   REQUIRED   DUMPSTATE   VOLSSUC     EXPDATE    AVAILABLE
DCREDB18       Y       COMPLETE     00002     2008/03/02      Y

            HWCOMP  ENCRYPT  ENCTYPE    RSAKEY/KPWD
            NO      NONE     *********  ************************************

SOURCE   DUMPVOLS                                        DEVICE TYPE
MHL0A0   TST007                                           3590-1
FILE SEQ=01, DSNAME=HSM.DMP.DCREDB18.VMHL0A0.D07071.T450116
MHL0A1   TST007                                           3590-1
FILE SEQ=02, DSNAME=HSM.DMP.DCREDB18.VMHL0A1.D07071.T450116
```

Figure 7-52 LIST CP of the fast replication backup version to be used for recovery

The excerpt from the DFSMShsm dump log (see Example 7-43) shows that now restore commands instead of copy commands are being executed. The processing is at the physical data set level as well as with copy. The entire process is driven by the catalog entry of the data set being requested for recovery. As you can see in Example 7-43, there are two restore processes being executed according to the volume pointers in the catalog entry (MHL0A0 and MHL0A1).

Example 7-43 DFSMShsm dump log while processing FRRECOV for TESTFR.DCOLLECT.ESDS01

```
ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.082
17:40
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I GDSN01 -  RESTORE DS(INCLUDE(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  INDDNAME(SYS03708) OUTDDNAME(SYS03709) REPLACE CANCELERROR -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                    ) -
 ARC0640I GDSN01 -  STORCLAS(STANDARD                    )         FORCECP(0) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                      )
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2007.082 17:40:34 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN01 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2007.082 17:40:34 EXECUTION BEGINS
 ARC0640I GDSN01 - ADR780I (001)-TDDS (01),
```

```
  THE INPUT DUMP DATA SET BEING PROCESSED IS IN FULL VOLUME FORMAT AND WAS CREATED BY DFSMSDSS
VERSION 1
 ARC0640I GDSN01 -                          RELEASE 8 MODIFICATION LEVEL 0
 ARC0640I GDSN01 - ADR378I (001)-TDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
FROM VOLUME MHLOA0
 ARC0640I GDSN01 -                           TESTFR.DCOLLECT.ESDS01
TESTFR.DCOLLECT.ESDS01.DATA
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2007.082 17:46:36 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2007.082 17:46:36 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2007.082 17:46:36 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
 ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R08.0 DATA SET SERVICES     2007.082
17:46
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
 ARC0640I GDSN01 -  RESTORE DS(INCLUDE(TESTFR.DCOLLECT.ESDS01                    )) -
 ARC0640I GDSN01 -  INDDNAME(SYS03708) OUTDDNAME(SYS03712) REPLACE CANCELERROR -
 ARC0640I GDSN01 -  BYPASSACS(TESTFR.DCOLLECT.ESDS01                          ) -
 ARC0640I GDSN01 -  STORCLAS(STANDARD                      )       FORCECP(0) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                        )
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2007.082 17:46:36 INITIAL SCAN OF USER CONTROL
STATEMENTS COMPLETED.
 ARC0640I GDSN01 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2007.082 17:46:36 EXECUTION BEGINS
 ARC0640I GDSN01 - ADR780I (001)-TDDS (01),
  THE INPUT DUMP DATA SET BEING PROCESSED IS IN FULL VOLUME FORMAT AND WAS CREATED BY DFSMSDSS
VERSION 1
 ARC0640I GDSN01 -                          RELEASE 8 MODIFICATION LEVEL 0
 ARC0640I GDSN01 - ADR378I (001)-TDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
FROM VOLUME MHLOA1
 ARC0640I GDSN01 -                           TESTFR.DCOLLECT.ESDS01
TESTFR.DCOLLECT.ESDS01.DATA
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2007.082 17:49:39 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2007.082 17:49:39 TASK COMPLETED WITH RETURN CODE
0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2007.082 17:49:39 DFSMSDSS PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
```

7.4.4 Hints and tips for the recovery of data sets that no longer exist

If the data set that you need to recover was deleted and you want to recover it from a fast replication backup, you need at least a catalog entry for the data set. The most simple and common way to meet this requirement is to allocate an empty dummy data set on the volumes.

You do not need to remember the exact DCB or size because if it does not fit, physical copy deletes and reallocates the data set according to the actual needs. There are two important things that you must know:

- VSAM (including type, like KSDS, ESDS, and so on) or non-VSAM
- Volumes where the data set resided

If you recover a data set from a fast replication dump copy you could find the information in a VTOCCOPY data set that was created under control of DFSMShsm (see Figure 7-53).

If you want to recover from a DASD fast replication copy a VTOCCOPY data set is not available.

You will find all this information in DCOLLECT output (see Figure 7-55 on page 241) of the volumes that are being backed up by using fast replication.

```
 BROWSE    HSM.DUMPVTOC.T450116.VMHL0A0.D07071        Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> CSR
******************************** Top of Data ********************************
SYS1.VTOCIX.MHL0A0                          ..Å.........  ...h....{.............
TESTFR.DCOLLECT.D997                        ..,.....,...  ..Ø{....Ø.............
SYS1.VVDS.VMHL0A0                           ..Å..........Ø.Ø....Ø.............
TESTFR.DCOLLECT.D999                        ..,.....,...  ..Ø{....Ø.............
TESTFR.DCOLLECT.ESDS0.DATA                  ..,.....,.....Ø.{....Ø.............
TESTFR.CNTL.JCL                             ..,.....,......ØØ....µ.............
TESTFR.DCOLLECT.ESDS01.DATA                 ..,.....,.....Ø.{....Ø.............
TESTFR.SRCHFOR.LIST                         ..,.....,...  ..Ø&..Ã.Ø.............
TESTFR.MHLRES2.MVOL.DATA                    ..,.....,...  ..ØØ..Ã.Ø.............
UCAT.TESTFR                                 ..,.....,.å...{.{....Ø.............
UCAT.TESTFR.CATINDEX                        ..,...........{.h....Ø.............
****************************** Bottom of Data ******************************
```

Figure 7-53 Dump VTOCCOPY data set created by DFSMShsm based on dump class setting

When you run a job (Figure 7-54) right before you start a fast replication for a copy pool, you can preserve the volume pointers as well as the information of the type of VSAM for future use.

```
//STEP1  EXEC  PGM=IDCAMS
//*
//*
//SYSPRINT  DD  SYSOUT=*
//DCOUT    DD    DSN=MHLRES2.DCOLLECT.SG1,DISP=(MOD,CATLG),
//             SPACE=(CYL,(1,1),RLSE),RECFM=VB,LRECL=644,
//             STORCLAS=HSMFR
//SYSIN    DD    *
    DCOLLECT -
          OUTFILE(DCOUT) -
          STORAGEGROUP( -
                SG1 -
                 ) -

/* END OF DCOLLECT COMMAND
```

Figure 7-54 Example JCL for execution of a DCOLLECT run

We recommend reusing a data set for output that is allocated on one of the copy pool volumes. This makes sure that you never become confused about which data set provides accurate information for a particular fast replication backup version that you intend to use for recovery.

You must be prepared to not be able to open a copy of this data set from one of the target volumes. So, if you need to access more than the current content of such a data set, start

FRRECOV for this particular data set from the desired fast replication backup or dump generation. This should never be a problem if you really reuse (not reallocate) the data set when you run the next DCOLLECT since the volume pointers in the catalog entry should not change.

```
 BROWSE    MHLRES2.DCOLLECT.SG1                          Line 00000016 Col 005 084
 Command ===>                                              Scroll ===> CSR
D ..SC64.(.........TESTFR.DCOLLECT.ESDS01.DATA                . ......{.MHLOAO
A ..SC64.(.........TESTFR.DCOLLECT.ESDS01.DATA                TESTFR.DCOLLECT.
D ..SC64.(.........UCAT.TESTFR                                .â¦.....{.MHLOAO
A ..SC64.(.........UCAT.TESTFR                                UCAT.TESTFR
D ..SC64.(.........SYS1.VTOCIX.MHLOAO                         . ±... .Ø.MHLOAO
D ..SC64.(.........TESTFR.MHLRES2.MVOL.DATA                   . ±... .&.MHLOAO
V ..SC64.(.........MHLOA1X..........¢...¢..............h...?3390    Ø.....SG
D ..SC64.(.........SYS1.VVDS.VMHLOA1                          . ........MHLOA1
A ..SC64.(.........SYS1.VVDS.VMHLOA1                          SYS1.VVDS.VMHLOA
D ..SC64.(.........TESTFR.DCOLLECT.D987                       .à.... .&.MHLOA1
D ..SC64.(.........TESTFR.DCOLLECT.D989                       .à.... .&.MHLOA1
D ..SC64.(.........TESTFR.CMD.CLIST                           .è......°.MHLOA1
D ..SC64.(.........TESTFR.MHLRES2.DUMMY                       . .... .&.MHLOA1
D ..SC64.(.........TESTFR.MHLRES2.MVOL.DATA                   . .... .&.MHLOA1
D ..SC64.(.........TESTFR.DCOLLECT.ESDS1.DATA                 . ......{.MHLOA1
A ..SC64.(.........TESTFR.DCOLLECT.ESDS1.DATA                 TESTFR.DCOLLECT.
D ..SC64.(.........TESTFR.SUPERC.LIST                         . .... .m.MHLOA1
D ..SC64.(.........TESTFR.DCOLLECT.ESDS01.DATA                . ......{.MHLOA1
A ..SC64.(.........TESTFR.DCOLLECT.ESDS01.DATA                TESTFR.DCOLLECT.
D ..SC64.(.........SYS1.VTOCIX.MHLOA1                         . ±... .Ø.MHLOA1
```

Figure 7-55 Output of DCOLLECT STORAGEGROUP(SG1)

In order to find information about the data set names that you are interested in it is easiest to browse the DCOLLECT data set and to use the find command.

When all required information is available you can allocate a dummy data set. You should consider using a storage class that was defined with the guaranteed space attribute in order to make sure that the dummy data set is allocated on the same volumes as it was at time of backup. Do not forget to alter the storage class after allocation of the data set. During FRRECOV DSNAME processing there is no call for ACS routine services, so the storage class is not redetermined automatically.

After you have finished the preparation activities you can try the FRRECOV DSNAME command for the data set. While the physical data set copy is a physical process that works at the tracks level to copy a data set, you never need to worry about the physical location of a data set's extents. DFSMSdss is able to use any free extents on a volume for output of the data set copy process. You might get into trouble if the current free space on the volumes is not sufficient to restore the data set. If this is the case, recovery fails, so you must retry the FRRECOV DSNAME command after you have moved some data sets off the volumes.

7.5 Reporting on the DFSMShsm fast replication environment

In this section we discuss reporting in the fast replication environment.

7.5.1 Statistic records

DFSMShsm writes its own statistic records. This is not a statistic record created by a z/OS function such as dynamic allocation. The command SETSYS SMF or NOSMF tells DFSMShsm to write or not write SMF records.

The optional parameters SMF(smfid) and NOSMF are mutually exclusive. They specify whether DFSMShsm writes SMF records that contain DFSMShsm statistics.

- SMF(smfid) specifies that DFSMShsm write SMF records in the SYS1.MANX or SYS1.MANY system data sets.

  For smfid, substitute a record identification. Use SMF user codes for the record identification in the range of 128 to 255.

  If you specify smfid, DFSMShsm writes records with SMF identifications of smfid and smfid+1. Records with an identification of smfid contain daily statistics (DSR) and volume statistics (VSR). Records with an identification of smfid+1 contain function statistics (FSR) and ABARS function statistics (WWFSR).

  Note: When a DFSMShsm function executes, only selected fields within the FSR record are set. The fields that are set depend on the function that is being performed and the method that is used to request the function. The fields that are set are required by the DFSMShsm REPORT command.

- NOSMF specifies that no SMF records are to be written.

7.5.2 REPORT command

The REPORT command has been updated with z/OS DFSMS V1.5 in support of DFSMShsm fast replication.

REPORT DAILY command

The REPORT DAILY command has been updated with z/OS DFSMS 1.5 to report the total number of volumes for which a fast replication backup or recovery has been requested. The total number of failures is also reported. The two lines in Example 7-44 have been added to the REPORT DAILY heading.

There were no updates with z/OS DFSMS 1.8. Processing of dumps of copy pool volumes is counted in the existing fields:

- FULL VOLUME DUMPS=
- DUMP COPIES=
- FULL VOLUME RESTORES=

Example 7-44 Updates to the REPORT DAILY heading with z/OS DFSMS V1.5

```
FAST REPLICATION VOLUME BACKUPS = 00000004 REQUESTED, 00000000 FAILED
FAST REPLICATION VOLUME RECOVERS = 00000001 REQUESTED, 00000000 FAILED
```

REPORT VOLUMES command

The REPORT VOLUMES command has been updated with z/OS DFSMS V1.5 to report the total number times that a fast replication backup was successfully created for the volume and the number of failed attempts for that volume. Recover statistics are also reported. The two lines in Example 7-45 on page 243 have been added to the REPORT VOLUMES heading.

There were no updates with z/OS DFSMS V1.8. Processing of dumps of copy pool volumes is reported in the existing fields:

- VOLUME DUMP=
- DUMP COPIES=
- FULL VOLUME RESTORES=

Note that dumps of copy pool volumes are created by processing volumes of copy pool backup storage groups, but you will never find statistics about volumes of copy pool backup storage groups in DFSMShsm.

Example 7-45 Updates to the REPORT VOLUMES heading with z/OS DFSMS 1.5

```
FAST REPLICATION BACKUP = 0000 REQUESTED, 0000 FAILED
FAST REPLICATION RECOVER = 0000 REQUESTED, 0000 FAILED
```

REPORT statistics

The two REPORT options show the current statistics (DSR and VSR) for the DFSMShsm address space. The statistics are accumulated in the DFSMShsm work space and are written to the MCDS and to SMF under the following conditions:

- For DSR
  - At the start of a new hour
  - When a REPORT command is entered
  - When a QUERY STATISTICS command is entered
  - When DFSMShsm shuts down
  - At the start of a new day (the previous day's record is written)

  Note: DFSMShsm activity (recalls, migrations, backups, recovers, dumps, recycles, restores, and deletions of migrated data sets) must occur in order for DFSMShsm to recognize the start of a new hour or day.

- For VSR
  - Whenever a daily statistics record is written
  - At the end of secondary space management
  - After space management has been performed on a volume
- For FSR
  - The FSR is both one of the DFSMShsm statistic records and a control block that contains statistics for a particular function that is performed. It is maintained in the DFSMShsm workspace until a function such as FRBACKUP has completed.
  - Upon completion of the function, the record is written to the DFSMShsm log and accumulated by category into the daily and volume statistics records in the migration control data set (MCDS).
  - The FSR is also written to the SMF data sets (SYS1.MANx or SYS1.MANy) if SETSYS SMF is specified.

  Tip: If statistics are needed for a week or several days, they may be extracted more easily from the SMF data, not only out of the address space or MCDS. It is also easier to use REXX™ or other programs to analyze them.

7.5.3 DSR records

There are no updates in the DSR records with z/OS DFSMS V1.8.

7.5.4 Updated VSR records

There are no updates in the VSR records with z/OS DFSMS V1.8.

7.5.5 New fields in FSR records

Fields related to fast replication are shown in bold in Table 7-2.

Table 7-2 Function statistics record extract of the new and updated fields

Offsets	Type	Length	Name	Description
42 (2A)	FIXED	1	FSRTYPE	DFSMShsm function type. The function types are as follows: 1=Primary to level 1 migration 2=Level 1 to level 2 migration, or level 1 to level 1 migration or level 2 to level 2 migration 3=Primary to level 2 migration 4=Recall from level 1 to primary 5=Recall from level 2 to primary 6=Delete a migrated data set 7=Daily backup 8=Spill backup 9=Recovery 10=Recycle backup volume 11=Data set deletion by age 12=Recycle migration volume 13=Full volume dump 14=Volume or data set restore 15=ABACKUP function (see WWFSR control block) 16=ARECOVER function (see WWFSR control block) 17=Expire primary or migrated data sets 18=Partrel function 19=Expire or roll off incremental backup version 20=(H)BDELETE an incremental backup version 21=Fast replication backup function 22=Fast replication recover function 23=Fast replication delete function
131 (83)	BITSTRING	1	FSRMFLGS	Flags from the MWE.
	1...		FSRFRTRY	When set to 1, the backup copy was made during a retry, after the first try failed because the data set was in use.
	.1..		FSRF_REMOTE	When set to 1, this request completed successfully on a remote system.

Offsets	Type	Length	Name	Description
	..1.		FSRFPIGB	When set to 1, the request was completed using a tape already mounted.
	...1		FSRF_REMOTE_ HOST_ PROCESSED	When set to 1, MWE processed by remote host.
	.... 1...		FSRF_DASD	When set to 1, the DASD copy of the version was deleted.
	.... .1..		FSRF_DUMPCPY	When set to 1, the dump class of the copy pool dump was deleted.
	.... ..1.		FSRF_DUMPVER	When set to 1, the entire dump version of the copy pool version was deleted.
	1		FSRF_COPYPOOL _FRDUMP	A value of 1 indicates a fast replication dump or restore.
191 (BF)	FIXED	1	FSR_COPYMETHOD	Requested method of fast replication. The valid methods are as follows: 1=None 2=Preferred 3=Required
223 (DF)	BITSTRING	1	FSRFLG4	Flags.
	1...		FSRF_FRRECOV_ DSNAME	When set to 1, fast replication recovery was requested for a data set through the FRRECOV DSNAME command.
	.1..		FSRF_FRRECOV_ FROMDISK	When set to 1, fast replication recovery will be performed from disk. This flag is set only when FSRF_FRRECOV_DSNAME is set to 1.
	..1.		FSRF_MULT_ DSNAMES	When set to 1, the fast replication recovery request specified more than one data set name.
	...1		FSRF_MULTI VOLUME	When set to 1, the fast replication recovery request was performed on part of a multi-volume data set. This flag is set only when FSRF_FRRECOV_DSNAME is set to 1.
	.... 1...		FSRF_ALTER PRI	When set to 1, the priority of this request was altered through the ALTERPRI command.
	.... .1..		FSRF_ALTER PRI_HI	When set to 1, the HIGH keyword was specified on the ALTERPRI command. When set to 0, the LOW keyword was specified. This flag applies only when FSRF_ALTERPRI is set to 1.
	xx		*	Reserved.

Offsets	Type	Length	Name	Description
248 (F8)	CHARACTER	40	FSR_CPNAME	Copy pool name. This flag is set only when FSRF_FRRECOV_DSNAME is set to 1.
248 (F8)	CHARACTER	8(5)	FSRDCLAS	A 5-element array consisting of 8-byte fields containing the names of dump classes.
292 (124)	FIXED	4	FSR_FR_REAS	Fast replication return code.

The FSR SMF record number depends on the setting of SETSYS SMF(smfid). If smfid is specified as 240, then the FSRs will be 241 SMF records. The FSR SMF record type is always equal to smfid+1.

For more information about SMF records or DFSMShsm FSR records, refer to the following publications:

- *z/OS MVS System Management Facilities (SMF)*, SA22-7630
- *DFSMShsm Diagnosis*, GC52-1083
- *DFSMShsm Implementation and Customization Guide*, SC35-0418

Gathering the data

Prior to producing reports for DFSMShsm fast replication, gather the appropriate data. The first step is to determine the SMF ID that is related to DFSMShsm Functional Statistics Records. You can do this several ways:

- Method 1: Use the QUERY SETSYS command. The output in the ARC0150I message contains the SMFID, as shown in Example 7-46.

Example 7-46 SMFID in QUERY SETSYS output

```
ARC0150I JOURNAL={NONE | SPEED | RECOVERY}, LOG={YES | NO | HELD}, TRACE={YES | NO},
SMFID={smfid | NONE}, DEBUG={YES | NO}, EMERG={YES | NO}, JES={2 | 3}, SYS1DUMP={YES | NO},
RACFIND={YES | NO}, ERASEONSCRATCH={YES | NO}
```

- Method 2: Review the system ARCCMDxx member of SYS1.PARMLIB.

  Look for SETSYS SMFID(smfid) and use it to determine whether SMF records are being generated, and if so which one to use:

  - DSR and VSR = smfid.
  - FSR and WWFSR = smfid + 1.
    (WWFSRs are the ABARSs function statistics)

    For example, an SMFID=240 means that FSR records are type 241.
  - SMFID=NONE means there are no records being collected.

All FSR types are shown in Figure 7-56:

- Function types 1 through 14 and 17 through 23 are FSR records.
- Function types 15 and 16 are ABARS WWFSR records.

For details, see Section 1.2.1 in the manual *DFSMShsm Implementation and Customization Guide*, SC35-0418.

```
DFSMShsm function type (FSRTYPE). The function types are:
1=Primary to level 1 migration
2=Level 1 to level 2 migration,
   or level 1 to level 1 migration
   or level 2 to level 2 migration
3=Primary to level 2 migration
4=Recall from level 1 to primary
5=Recall from level 2 to primary
6=Delete a migrated data set
7=Daily backup
8=Spill backup
9=Recovery
10=Recycle backup volume
11=Data set deletion by age
12=Recycle migration volume
13=Full volume dump
14=Volume or data set restore
15=ABACKUP function
16=ARECOVER function
17=Expire primary or migrated data sets
18=Partrel function
19=Expire or roll off incremental backup version
20=(H)BDELETE an incremental backup version
21=Fast replication backup function
22=Fast replication recover function
23=Fast replication delete function
```

Figure 7-56 FSR type records

Preparing for analysis

To begin analyzing the data, first combine SMF data for all of the systems that share the same DFSMShsm Control Data Sets (MCDS, BCDS, OCDS) into a single file. We eventually want to convert the SMF data to variable-blocked (VB) format to be processed by a REXX exec.

The SMF dump program (IFASMFDP) is used to transfer the contents of the SMF data set to another data set and to reset the status of the dumped data set to empty so that SMF can use it again for recording data. The SMF dump program dumps the contents of multiple VSAM or QSAM data sets to sequential data sets on either tape or direct-access devices. The SMF dump program enables the installation to route different records to separate files and produce a summary activity report.

The IFASMFDP program typically produces data as RECFM=VBS. In order for the REXX tool FSRSTAT to process the IFASMFDP output, you must convert the format to RECFM=VB. You can use IDCAMS REPRO to convert the DFSMShsm SMF data to RECFM=VB, LRECL=4096. Do not be tempted to use the IFASMFDP program to create RECFM=VB, LRECL=4096 data directly.

Sample JCL is provided in Example 7-47 and in Example 7-49.

Example 7-47 IFASMFDP JCL example

```
//STEP1    EXEC PGM=IFASMFDP
   //INDD1    DD DSN=SYS1.MANX,DISP=SHR
   //OUTDD1   DD DSN=MHLRES2.DATA,UNIT=3390,VOL=SER=WORK01,
   //            DISP=(NEW,CATLG,KEEP),SPACE=(CYL,(10,1))
   //SYSPRINT DD    SYSOUT=A
   //SYSIN    DD    *
          INDD(INDD1,OPTIONS(DUMP))
          OUTDD(OUTDD1,TYPE(241))
          DATE(2003120,2003126)
   /*

use TYPE(241) - select this to match FSR records
or use TYPE(240:241) - select this to match DSR,VSR and all FSR records
DATE(2003120,2003126) - select a period of 7 or 14 days,
             using YYYYddd Julian date format.
             2003020 is Year 2003, 120th day
```

The output data set that is specified in the OUTDD1 DD statement in Example 7-47 contains the summary activity report. Example 7-48 shows a sample report.

Example 7-48 SMF dump summary activity report

```
                                SUMMARY ACTIVITY REPORT
START DATE-TIME  10/22/2003-13:10:01                      END DATE-TIME  10/22/2003-13:35:08
RECORD       RECORDS         PERCENT      AVG. RECORD   MIN. RECORD   MAX. RECORD       RECORDS
  TYPE          READ        OF TOTAL          LENGTH        LENGTH        LENGTH       WRITTEN
   240             7           .15 %        1,036.00         1,036         1,036             7
   241             4           .09 %          300.00           300           300             4
TOTAL          4,591           100 %          888.24            18        32,720            13
NUMBER OF RECORDS IN ERROR            0
```

Example 7-49 IDCAMS REPRO example

```
//* INPUT FILE IS RECFM=VBS FROM IFASMFDP
     //STEP1    EXEC PGM=IDCAMS
     //SYSPRINT DD SYSOUT=A
     //INDATA   DD DISP=OLD,DSNAME=MHLRES2.DATA
     //OUTDATA  DD DSNAME=MHLRES2.FSR,UNIT=3390,
     //            DISP=(NEW,CATLG),VOL=SER=WORK01,
     //            SPACE=(CYL,(1,1)),RECFM=VB,LRECL=4096
     //SYSIN    DD   *
          REPRO -
               INFILE(INDATA) -
               OUTFILE(OUTDATA)
     /*
```

Sample REXX execs

The FSRSTAT program is a REXX sample program that reads DFSMShsm FSR records and generates a statistical summary report. The FRSTAT program is shipped with DFSMShsm in SYS1.SAMPLIB member ARCTOOLS. It has to be modified so that it only selects fast replication FSRTYPE records.

The installation of DFSMShsm places a member called ARCTOOLS in the SYS1.SAMPLIB data set. When you supply a valid jobcard and run the ARCTOOLS program, it creates the following partitioned data sets:

HSM.SAMPLE.TOOL	Sample REXX execs and JCL to manage your DFSMShsm data
HSM.ABARUTIL.JCL	JCL used by ABARS utilities
HSM.ABARUTIL.PROCLIB	JCL PROCs used by ABARS utilities
HSM.ABARUTIL.DOCS	Documentation for ABARS utilities

The HSM.SAMPLE.TOOL data set contains the FSRSTAT PGM along with other tools. FRSTAT is written in REXX, so the following considerations apply:

- Does not require any special programs or languages (such as SAS or MICS)
- Can be modified easily and customized to meet your needs
- Can be slow, so consider running it in batch using PGM=IKJEFT01
- Requires input data to be converted to RECFM=VB format

Creating reports

You can run the FSRSTAT exec in batch using PGM=IKJEFT01. A sample JCL is shown in Example 7-50, where:

- The SYSEXEC DD statement references the location of the FSRSTAT REXX program.
- The output file name is the same as the input file name with the added low-level qualifier of FSRSTAT. For example:
  - Input file name: MHLRES2.FSR
  - Output file name: MHLRES2.FSR.FSRSTAT

Example 7-50 IKJEFT01 FSRSTAT report JCLI

```
//STEP1 EXEC PGM=IKJEFT01
//SYSEXEC  DD DSN=HSM.SAMPLE.TOOL,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  FSRSTAT 'MHLRES2.FSR' FSRID(241)
/*
```

FSRSTAT report output is shown in Example 7-51.

Example 7-51 FSRSTAT report output

```
Data Set Analyzed: 'MHLRES2.FSR'
Number of lines read: 13

FSR records by type
021               4

FSR records by host
SC65              4    100.00%    100.00%

FSR records by Return Code
0000              4    100.00%    100.00%

FSR records by Date
0103295F          4
```

Customizing reports

You can use DFSORT to produce customized reports. In our example, we first extract the FSR type 21, 22, and 23 records in separate data sets. In the SYSIN statements:

- X'15' is the value for FSR record type 21.
- X'16' is the value for FSR record type 22.
- X'17' is the value for FSR record type 23.

Example 7-52 shows a DFSORT job.

Example 7-52 DFSORT example jcl

```
//STEPNN   EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//PLDTRACE DD DUMMY
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DISP=SHR,DSN=MHLRES2.FSR
//SORTOF1  DD LIKE=MHLRES2.FSR,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         DSN=MHLRES2.FSR.TYPE21
//SORTOF2  DD LIKE=MHLRES2.FSR,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         DSN=MHLRES2.FSR.TYPE22
//SORTOF3  DD LIKE=MHLRES2.FSR,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         DSN=MHLRES2.FSR.TYPE23
//SYSIN    DD *
  OPTION COPY,VLSHRT
  OUTFIL FILES=1,
         INCLUDE=(43,1,BI,EQ,X'15')
  OUTFIL FILES=2,
         INCLUDE=(43,1,BI,EQ,X'16')
  OUTFIL FILES=3,
         INCLUDE=(43,1,BI,EQ,X'17')
```

You can use an ICETOOL job, as shown in Example 7-53, to generate a report about specific data that you want to analyze. A good field for performance reports might be the FSRCPU value.

Example 7-53 ICETOOL sample to extract FSR data

```
//MHLRES2Y JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=K,
// NOTIFY=&SYSUID,TIME=1440,MSGLEVEL=(1,1)
/*JOBPARMBL=999,SYSAFF=*
//JOBLIB DD DSN=SYS1.SICELINK,DISP=SHR
//ICETL2  EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//LIST1    DD  SYSOUT=*
//DFSPARM  DD *
//IN01     DD DSN=MHLRES2.FSR.TYPE21,DISP=SHR
//OUT01    DD DSN=MHLRES2.FSR.TYPE21.CREATE,DISP=SHR
//*           DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//*           SPACE=(CYL,(5,1),RLSE),
//*           DCB=(*.IN01)
//TOOLIN   DD *
  COPY FROM(IN01) USING(CPY1)
  DISPLAY FROM(OUT01) LIST(LIST1) BLANK            -
          TITLE('FSRTYPES           ') -
          ON(1,4,CH)
```

```
//CPY1CNTL  DD *
  INCLUDE COND=(43,1,BI,EQ,X'15')
  OUTFIL FNAMES=OUT01,
  OUTREC=(1,4,C'FSRTYPE',43,1,FI,44,1,HEX,
         45,44,
         89,6,
         99,6,
        105,4,HEX,
        179,2,HEX,
        181,4,HEX)
/*
```

Example 7-54 shows the output that is generated from the job in the previous example.

Example 7-54 Output of an ICETOOL job

```
 BROWSE    MHLRES2.FSR.TYPE21.CREATE
 Command ===>
****************************************************** Top of Data ******************
FSRTYPE 21 00DSN$DB8B$DB                                MHL213MHL1250000000030000000001
FSRTYPE 21 00DSN$DB8B$DB                                MHL214MHL1260000000030000000000
FSRTYPE 21 00DSN$DB8B$LG                                MHL225MHL0370000000030000000000
FSRTYPE 21 00DSN$DB8B$LG                                MHL226MHL1470000000030000000000
FSRTYPE 21 00CP1                                        MHL0ACMHL0A0000000003000000000
FSRTYPE 21 00CP1                                        MHL0ADMHL0A1000000003000000000
FSRTYPE 21 00CP2                                        MHL1ACMHL1A0000000003000000000
FSRTYPE 21 00CP2                                        MHL1ADMHL1A1000000003000000000
FSRTYPE 21 00CP3                                        MHL2AEMHL1A0000000003000000000
FSRTYPE 21 00CP3                                        MHL2AFMHL1A1000000003000000000
FSRTYPE 21 00DSN$DB8B$DB                                MHL013MHL1250000000030000000001
FSRTYPE 21 00DSN$DB8B$DB                                MHL014MHL1260000000030000000001
FSRTYPE 21 00DSN$DB8B$LG                                MHL025MHL0370000000030000000000
FSRTYPE 21 00DSN$DB8B$LG                                MHL026MHL1470000000030000000000
FSRTYPE 21 00CP1                                        MHL0AEMHL0A0000000003000000001
```

7.6 Security for DFSMShsm fast replication

This section describes the RACF support for DFSMShsm commands and the RACF Facility Class profiles that are used to protect the DFSMShsm fast replication function.

This support is available since DFSMShsm V1.5. For systems prior to z/OS V1.5, the level of authorization was defined using the AUTH command in the DFSMShsm PARMLIB member.

DFSMShsm provides two levels of commands to be protected by RACF Facility Class profiles:

- Storage administrator commands
- End-user commands

User commands can be issued by any TSO user if that user has appropriate RACF authority to the data sets to be processed.

Storage administrator commands can only be issued by a DFSMShsm authorized user. No RACF checking is performed to confirm your authority to access the data sets to be processed. In many installations, the storage administrator is different from the security administrator. The security administrator would like to control and monitor the activities against all sensitive data.

A storage administrator who only needs to issue DFSMShsm configuration type commands, such as ADDVOL, can also issue other commands, such as RECOVER datasetname NEWNAME. This user might not be allowed by RACF to access this data set, but he can copy or rename it to a new data set name to which he has read or alter authority. He can use DFSMShsm storage administrator commands to access or delete data that is restricted to him.

As with storage administrator commands, users have access to all user commands and parameters. There is no granularity at the command level.

The RACF FACILITY class

Table 7-3 shows the base set of FACILITY class profiles for DFSMShsm command protection.

Note: The resource names in the RACF FACILITY class for protection of the DFSMShsm fast replication commands has changed with DFSMShsm V1.8 (see Table 7-4 on page 252 for the new format).

Table 7-3 Base STGADMIN profiles

Profile	Protects
STGADMIN.*	All DFSMS commands including all DFSMShsm commands
STGADMIN.ARC.*	All DFSMShsm commands
STGADMIN.ARC.cmd	Specific storage administrator command
STGADMIN.ARC.cmd.prm	Specific storage administrator command with specific parameter
STGADMIN.ARC.ENDUSER.*	All user commands
STGADMIN.ARC.ENDUSER.h_cmd	Specific user command
STGADMIN.ARC.ENDUSER.h_cmd.prm	Specific user command with specific parameter

Note that the STGADMIN.* profile in this table is a general profile that can be used by other applications, so deleting it may affect other applications.

Table 7-4 shows the RACF profiles that are required to protect the DFSMShsm fast replication commands FRBACKUP, FRRECOV, and FRDELETE.

Table 7-4 STGADMIN profiles for fast replication

Profile	Protects
STGADMIN.ARC.FB.*	Any FRBACKUP command
STGADMIN.ARC.FB.cpname	FRBACKUP COPYPOOL only for cpname
STGADMIN.ARC.FR.*	Any FRRECOV command
STGADMIN.ARC.FR.cpname	FRRECOV COPYPOOL only for cpname
STGADMIN.ARC.FD.*	Any FRDELETE command
STGADMIN.ARC.FD.cpname	FRDELETE COPYPOOL only for cpname
STGADMIN.ARC.LC.*	Any LIST COPYPOOL and LIST DSNAME(dsname) COPYPOOL command

Profile	Protects
STGADMIN.ARC.LC.cpname	LIST COPYPOOL(cpname) only for cpname

For a complete listing of DFSMShsm commands that can be protected, refer to the manual *DFSMShsm Implementation and Customization Guide,* SC35-0418.

7.7 Audit and error recovery

This section discusses auditing and basic error recovery in the fast replication environment.

7.7.1 FRDELETE considerations

This section contains considerations for using the DFSMShsm Fast Replication FRDELETE command.

The FRDELETE command was enhanced by new parameters with DFSMShsm 1.8:

DASDONLY Delete only a DASD backup copy (works like pre-DFSMShsm 1.8).

DUMPONLY(DCLASS(...) Delete only a dump copy (of a particular dump class).

BOTH Delete DASD backup copy and dump tape copy of that particular version (this is the default).

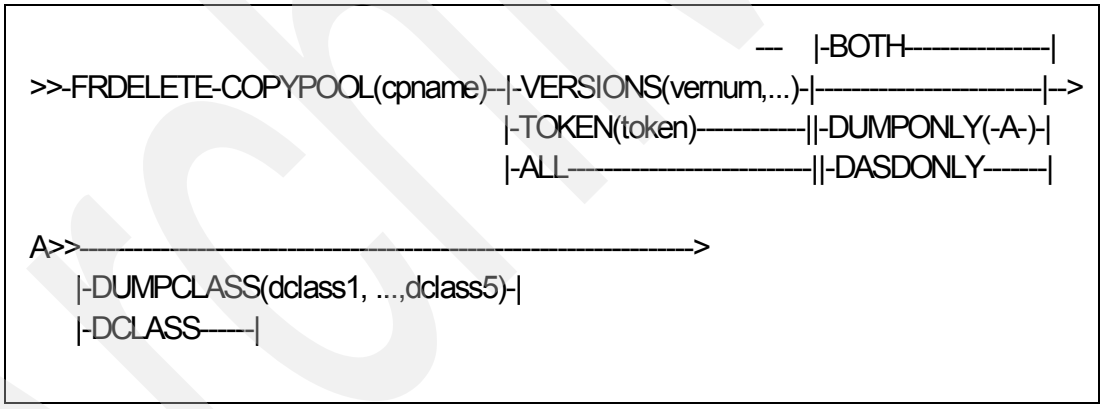


Figure 7-57 FRDELETE command syntax

When you use an FRDELETE command, as shown in Example 7-55, you receive some output in the backup log, as shown in Figure 7-58 on page 254.

Example 7-55 FRDELETE DUMPONLY command

```
HSEND FRDELETE COPYPOOL(CP1) VERSION(1) DUMPONLY
```

```
 DFSMSHSM DUMP LOG, TIME 15:38:38,  DATE 07/03/12
ARC0639I DUMP VTOC COPY SCRATCHED, DATA SET NAME=HSM.DUMPVTOC.T204914.VMHL0A0.D07050
ARC0262I DUMP COPY INVALIDATED FOR VOLUME MHL0A0, CREATION DATE=07/02/19
ARC0262I DUMP GENERATION INVALIDATED FOR VOLUME MHL0A0, CREATION DATE=07/02/19
ARC0639I DUMP VTOC COPY SCRATCHED, DATA SET NAME=HSM.DUMPVTOC.T204914.VMHL0A1.D07050
ARC0261I TAPE VOLUME TST019 NEEDS TO BE REINITIALIZED
ARC0263I DUMP VOLUME TST019 DELETED, VOLUME STATUS=PURGED
ARC0262I DUMP COPY INVALIDATED FOR VOLUME MHL0A1, CREATION DATE=07/02/19
ARC0262I DUMP GENERATION INVALIDATED FOR VOLUME MHL0A1, CREATION DATE=07/02/19
```

Figure 7-58 Output of FRDELETE COPYPOOL DUMPONLY command

FRDELETE is used to delete unneeded fast replication backup or dump versions. New DASD backup copy versions replace older backup copy versions through normal processing. Dump copy versions are deleted based on dump class settings during automatic dump.

Note: Depending on your settings you may end up in a different number of fast replication DASD backup generations and fast replication tape dump generations. This is because DASD backup generations are strictly rolled off by generation processing (based on the number of replicate backup version setting in the copy pool construct), while tape dump generations are primarily rolled off by retention period (based on dump class setting).

There are some instances, other than normal roll off, that cause unneeded copy versions to be deleted:

- The number of fast replication backup versions (as being reported by the LIST CP(...) ALLVOLS(ALLVERS) command) reaches the maximum of 85 versions. If this happens, during creation of a new generation by using the FRBACKUP command, the gen(84) dump copy is rolled off (see Figure 7-59).

Generation	Version	DASD	Tape (Dump)
+1	586	yes	not yet
0	585	yes	yes
1	584	yes	yes
2	583	yes	yes
3	582	no	yes
...			
83	502	no	yes
~~84~~	~~501~~	~~no~~	~~yes~~

Figure 7-59 Roll-out of dump copies because of reaching the architectural limit of 85 generations

- The number of versions specified is decreased by modifying the SMS copy pool definition.

  This causes subsequent FRBACKUP commands to remove unneeded DASD copy versions.

- When a copy pool is renamed and is no longer needed, it must be deleted by the storage administrator. Use the FRDELETE COPYPOOL command and:
  - Specify the ALL keyword when all copies of the version are to be deleted.
  - Specify the VERSIONS keyword when an individual or group of individual versions is to be deleted.
  - Specify the TOKEN keyword to delete the version that is associated with the token. If the token is not unique to an individual version, the delete fails.
  - If none of the above are specified, the delete fails.

Before a version is deleted, any outstanding FlashCopy relationships are withdrawn. *Never* withdraw the relationship outside of DFSMShsm. This causes the backup version to be invalidated, but DFSMShsm thinks that it is valid. It also could result in a data integrity exposure.

7.7.2 Decreasing copy pool backup volumes

When decreasing the number of volumes in a copy pool backup storage group, take the following into consideration:

- You must take special care when removing a volume from a copy pool backup storage group. Before doing so, you must ensure that the volume is not the target of a valid DFSMShsm backup version.
- You can verify this by examining the output from the DFSMShsm command LIST COPYPOOLBACKUPSTORAGEGROUP.
- If a target volume must be removed from a valid source to target pairing, you should use the FRDELETE command to delete the DFSMShsm backup version that the volume is a part of before you remove the volume.

7.7.3 Reuse of invalid backup versions

New DASD copy versions replace invalid DASD copy versions through the next FRBACKUP processing.

7.7.4 FIXCDS display and repair

Five new record types have been introduced by fast replication within the DFSMShsm BCDS: F, H, I, J, and K.

New MCH keys

There are five control record types in use in order to describe fast replication backups:

X'23' = FRD	Code K (new with DFSMShsm 1.8)
X'25' = FRSV	Code J (new with DFSMShsm 1.5)
X'2D' = FRTV	Code I (new with DFSMShsm 1.5)
X'2E' = FRB	Code F (new with DFSMShsm 1.5)
X'2F' = FRVP	Code H (new with DFSMShsm 1.5)

K is the fast replication dump record (FRD)

The key for a type K FRD record is the name of the copy pool and the version number the dumps were made for.

Example 7-56 FIXCDS output of a K (FRD) record

```
HSEND FIXCDS K CP1.........................005
MCH=   01DC2300 C04E338A 3CC35B0E 00000000 00000000
 *         C$                      *
+0000  00010000 00000000 C4C3D9C5 C4C2F1F8 00010000 0108049F 00028000 00000000
 *        DCREDB18                 *
+0020  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0040  00000000 D7D3E4E2 C3D6D7E8 00010000 0108049F 00020000 00000000 00000000
 *    PLUSCOPY                     *
+0060  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0080  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+00A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+00C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+00E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0100  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0120  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0140  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*                                *
+0160  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+0180  00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
ARC0197I TYPE K, KEY CP1.........................005, FIXCDS DISPLAY
ARC0197I (CONT.) SUCCESSFUL
```

F is the fast replication backup record (FRB)

The key for a type F FRB record is the name of the copy pool that the record represents. Example 7-57 shows an example of the key that is used with a type F FRB record.

Example 7-57 FIXCDS output of a F (FRB) record

```
FIXCDS F DSN$DB8B$DB ODS(MHLRES2.FIXCDS.CB)
MCH=   01782E00 BA2CE223 4A4F5E4C 00000000 00000000                              *      S
+0000  00000004 00000004 00000000 00000000 00000000 00000000 00000000 00000000   *
+0020  00000000 00000000 00000004 00000000 D4C8D3D9 C5E2F160 E3C5E2E3 00000000   *            MHLRES1 TEST
+0040  00000000 00000000 00000000 00000000 00000000 00000000 14224253 0103288F   *
+0060  00010100 00000000 00000000 00000003 00000000 D4C8D3D9 C5E2F160 E3C5E2E3   *            MHLRES1 TEST
+0080  00000000 00000000 00000000 00000000 00000000 00000000 00000000 11564119   *
+00A0  0103288F 00010100 00000000 00000000 00000002 00000000 00000000 00000000   *
+00C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+00E0  22101134 0103287F 00010100 00000000 00000000 00000005 00000000 00000000   *
+0100  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0120  00000000 00000000 00000000 00000400 00000000 00000000                      *
ARC0197I TYPE F, KEY DSN$DB8B$DB, FIXCDS DISPLAY SUCCESSFUL
```

H is the fast replication volume pairs record (FRVP)

Table 7-5 shows the fields that make up a key for a type H FRVP backup record.

Table 7-5 Type H record key fields

Characters	
1 to 30	The copy pool name suffixed with periods if the copy pool name is less than 30 characters in length.
31	The type of record. P if the record is a prepare record. B if the record is a backup record.
32 and 33	The version number in hexadecimal format.
34 to 41	The storage group name suffixed with periods if the storage group name is less than eight characters in length.
42 and 43	The extent number.

Example 7-58 shows the key (in hexadecimal) that is used for a type H FRVP backup record. In this example, we use a key that consists of:

- A copy pool name of COPYPL1 (followed by periods to fill the field)
- A B for a backup record request
- A version number of 1
- A storage group name of SRCSG1 (followed by periods to fill the field)
- An extent number of 0 (zero)

Example 7-58 FIXCDS output of a H (FRVP) record

```
FIXCDS H X'C3D7F14B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4BC20008E2C7F14B4B4B4B0000'
ODS('MHLRES2.DISPLAY.H.OUT')
MCH=   00942F00 00000000 00000000 00000000 00000000                            *
+0000  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  *
+0020  D4C8D3F0 C1F0D4C8 D3F0C1C5 80000000 00000000 00000000 0000D4C8 D3F0C1F1  *MHL0AOMHL0AE MHL0A1
+0040  D4C8D3F0 C1C6A000 00000000 00000000 00000000                            *MHL0AF
ARC0197I TYPE H, KEY CP1.......................B..SG1......., FIXCDS DISPLAY SUCCESSFUL H
```

Example 7-59 shows a sample IDCAMS PRINT job to print all H records. It can be used for printing the other records as well.

Example 7-59 Sample IDCAMS PRINT job to print all H records

```
//PRINT  EXEC PGM=IDCAMS
//*********************************************************
//*                                                       *
//*  PRINT new Fast replication records                   *
//* X'25' = FRSV (Code J)                                 *
//* X'2D' = FRTV (Code I)                                 *
//* X'2E' = FRB (Code F)                                  *
//* X'2F' = FRVP (Code H)                                 *
//*********************************************************
//SYSPRINT  DD SYSOUT=*
//DD1   DD DSN=HSM.BCDS,
//         DISP=SHR
//DD2   DD DSN=MHLRES3.BCDS.PRINT3,DISP=(,CATLG,DELETE),
//         SPACE=(TRK,5),DCB=(LRECL=250,RECFM=VB),UNIT=3390
//SYSIN DD *
 PRINT -
    INFILE(DD1)  OUTFILE(DD2) FROMKEY(X'2F') TOKEY(X'2F')
/*
```

Output from the IDCAMS job is shown in Example 7-60.

Example 7-60 Output from IDCAMS PRINT job to print all H records

```
IDCAMS  SYSTEM SERVICES                                          TIME: 21:02:52       10/20/03     PAGE      1
LISTING OF DATA SET -HSM.BCDS
KEY OF RECORD - 2FC4E2D55BC4C2F8C25BC4C24B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4BC20009C4C2F8C24B4B4B4B0001
000000  2FC4E2D5 5BC4C2F8 C25BC4C2 4B4B4B4B   4B4B4B4B 4B4B4B4B 4B4B4B4B 4B4B4BC2
                                                                  *.DSN$DB8B$DB..................B
000020  0009C4C2 F8C24B4B 4B4B0001 00942F00   00000000 00000000 00000000 00000000
                                                                  *..DB8B.........................
000040  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
                                                                  *...............................
000060  D4C8D3F1 F2F5D4C8 D3F0F1F3 80000000   00000000 00000000 0000D4C8 D3F1F2F6
                                                                  *MHL125MHL013.............MHL126
000080  D4C8D3F0 F1F4A000 00000000 00000000   00000000
                                                                  *MHL014.............
KEY OF RECORD - 2FC4E2D55BC4C2F8C25BC4C24B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4BC2000AC4C2F8C24B4B4B4B0001
000000  2FC4E2D5 5BC4C2F8 C25BC4C2 4B4B4B4B   4B4B4B4B 4B4B4B4B 4B4B4B4B 4B4B4BC2
                                                                  *.DSN$DB8B$DB..................B
000020  000AC4C2 F8C24B4B 4B4B0001 00942F00   00000000 00000000 00000000 00000000
                                                                  *..DB8B.........................
000040  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *...............................
000060  D4C8D3F1 F2F5D4C8 D3F2F1F3 80000000   00000000 00000000 0000D4C8 D3F1F2F6
                                                                  *MHL125MHL213.............MHL126
000080  D4C8D3F2 F1F4A000 00000000 00000000   00000000
                                                                  *MHL214.............
KEY OF RECORD - 2FC4E2D55BC4C2F8C25BD3C74B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4BC20009C4C2F8C2D3D6C7F10001
000000  2FC4E2D5 5BC4C2F8 C25BD3C7 4B4B4B4B   4B4B4B4B 4B4B4B4B 4B4B4B4B 4B4B4BC2
                                                                  *.DSN$DB8B$LG..................B
000020  0009C4C2 F8C2D3D6 C7F10001 007A2F00   00000000 00000000 00000000 00000000
                                                                  *..DB8BLOG1...:..................
000040  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *...............................
000060  D4C8D3F0 F3F7D4C8 D3F0F2F5 A0000000   00000000 00000000 0000
                                                                  *MHL037MHL025.............
KEY OF RECORD - 2FC4E2D55BC4C2F8C25BD3C74B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4BC20009C4C2F8C2D3D6C7F20001
000000  2FC4E2D5 5BC4C2F8 C25BD3C7 4B4B4B4B   4B4B4B4B 4B4B4B4B 4B4B4B4B 4B4B4BC2
                                                                  *.DSN$DB8B$LG..................B
000020  0009C4C2 F8C2D3D6 C7F20001 007A2F00   00000000 00000000 00000000 00000000
                                                                  *..DB8BLOG2...:..................
000040  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
                                                                  *...............................
000060  D4C8D3F1 F4F7D4C8 D3F0F2F6 A0000000   00000000 00000000 0000
                                                                  *MHL147MHL026.............
```

I is the fast replication target volume record (FRTV)

The key for a type I FRTV record is the volume serial number. Example 7-61 shows an example of the key that is used with a type I FRTV record.

Example 7-61 FIXCDS output of an I (FRTV) record

```
FIXCDS I MHL214  ODS(MHLRES2.FIXCDS.CB)
MCH=   00662D00 BA2C090A 71FB6688 BA2C090A 71FB6688                   *
+0000  D4C8D3F1 F2F60000 00000000 00000000 00000000 00000000 00000000 00000000   *MHL126
+0020  00000000 00000000                                                      *
ARC0197I TYPE I, KEY MHL214, FIXCDS DISPLAY SUCCESSFUL
```

J is the fast replication source volume record (FRSV)

The key for a type J FRSV record is the volume serial number. An example of the key that is used with a type J FRSV record is shown in Example 7-62.

Example 7-62 FIXCDS output of a J (FRSV) record

```
FIXCDS J MHL126  ODS(MHLRES2.FIXCDS.CB)
MCH=   07AC2500 BA2CE221 E11F1B08 B9CC2CCD 05D66242                       *   S      O
+0000  C4E2D55B C4C2F8C2 5BC4C240 40404040 40404040 40404040 40404040 40400B03   *DSN$DB8B$DB
+0020  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0040  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0060  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0080  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
```

```
+00A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+00C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+00E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0100  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
...
+0720  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0740  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *
+0760  00000000 00000000 00000000                                               *
ARC0197I TYPE J, KEY MHL126, FIXCDS DISPLAY SUCCESSFUL
```

7.7.5 AUDIT COPYPOOLCONTROLS

AUDIT was enhanced in order to check the fast replication CDS record relationships. You can use a command, as shown in Example 7-63.

Example 7-63 AUDIT COPYPOOLCONTROLS command

```
HSEND AUDIT COPYPOOLCONTROLS(CP1) ODS(MHLRES2.AUDITCP.LIST)
```

If you do not specify the name of a copy pool, DFSMShsm checks the fast replication CDS record relationships of all copypools.

Note: AUDIT COPYPOOLCONTROLS only provides detection support, so there is no option available to specify the FIX parameter.

Figure 7-60 shows an example for the output of the AUDIT COPYPOOLCONTROLS. We deleted two fast replication CDS records in order to demonstrate a simple fix process by using your own-built FIXCDS commands.

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 18:08:13 ON 07/03/16 FOR
COMMAND ENTERED:
AUDIT COPYPOOLCONTROLS(CP1)

/* ERR 177 I RECORD WAS EXPECTED FOR VOLUME MHLOAF AND WAS NOT FOUND
/* ERR 177 I RECORD WAS EXPECTED FOR VOLUME MHL1AC AND WAS NOT FOUND
- END OF -     ENHANCED AUDIT - LISTING -
```

Figure 7-60 Output of AUDIT COPYPOOLCONTROLS command

You can find a complete description of all AUDIT error types in *DFSMSdfp Storage Administration Reference,* SC26-7402. In Table 7-6 we listed those error types that are related with fast replication CDS records.

Table 7-6 AUDIT COPYPOOLCONTROLS error types

Description	Troubleshooting Hints
*ERR 170 SPECIFIED COPY POOL cpname DOES NOT EXIST	
*ERR 171 F RECORD INDICATES x DUMP VERSIONS, ONLY y VERSIONS WERE FOUND	
*ERR 172 H RECORD INDICATES x DASD VERSIONS, ONLY y VERSIONS WERE FOUND	

Description	Troubleshooting Hints
*ERR 173 F (FRB) RECORD INDICATES x H (FRVP) RECORDS FOR COPY POOL cpname, ONLY y RECORDS WERE FOUND	
*ERR 174 F (FRB) RECORD FOR COPY POOL cpname WAS NOT FOUND, ORPHANED H (FRVP) RECORDS EXIST	Use the FIXCDS DELETE command to remove the orphaned entries from the BCDS.
*ERR 175 H (FRVP) RECORD INDICATES IT IS A 'PREPARE' RECORD, THE CORRESPONDING F (FRB) RECORD DOES NOT	► You need to determine which indicator is correct. Use the FIXCDS PATCH command to correct the problem. ► You can use the LIST COPYPOOL command to help you determine which indicator is correct. ► Either the 'Prepare' flag should be on in the F (FRB) record or the FRVP records are orphan entries and should be deleted using the FIXCDS DELETE command.
*ERR 176 H (FRVP) RECORD FOR VERSION x EXISTS, BUT VERSION CANNOT BE FOUND IN CORRESPONDING F (FRB) RECORD FOR COPY POOL cpname	An FRVP record may be an orphan and should be deleted using the FIXCDS DELETE command.
ERR 177 I (FRTV) RECORD WAS EXPECTED FOR VOLUME volser AND WAS NOT FOUND A F (FRB) record indicates this is a valid version. The volume represented by this FRTV record needs to be made available and an FRTV record should be created to represent it or the fast replication backup of this copy pool will fail	The FIXCDS CREATE command can be used to recreate the FRTV record.
*ERR 178 I (FRTV) RECORD INDICATED BY H (FRVP) RECORD CANNOT BE FOUND. NEEDS TO EXIST FOR COPY POOL cpname VERSION vsn	
*ERR 179 H (FRVP) RECORD HAS WRONG TARGET VOLUME tvolser, OR I (FRTV) RECORD HAS WRONG SOURCE VOLUME svolser	Determine which one is incorrect and correct the record using the FIXCDS PATCH command.
*ERR 180 J (FRSV) RECORD NOT FOUND FOR VOLUME volser, WHICH BELONGS TO COPY POOL cpname, VERSION vsn	
*ERR 181 F (FRB) RECORD FOR cpname NOT FOUND, POSSIBLY ORPHANED J (FRSV) RECORD FOR VOLUME volser	► Remove cpname from the FRSV cpname array. ► If there are no cpnames left in the array, delete the FRSV record for VOLUME volser.
*ERR 182 NO P (MCP) RECORD FOUND FOR VOLUME volser, IS THIS AN ERROR?	
*ERR 183 VOLUME volser FROM THE J (FRSV) RECORD L0 FIELD COULD NOT BE FOUND IN A CORRESPONDING H (FRVP) RECORD	H (FRVP) record was not found or volser was not found in the corresponding H (FRVP) records.
*ERR 184 ORPHANED J (FRSV) RECORD DETECTED	Delete the record using the FIXCDS DELETE command.

Description	Troubleshooting Hints
*ERR 185 COPY POOL cpname COULD NOT BE FOUND IN THE J (FRSV) RECORD COPY POOL ARRAY FOR VOLUME volser	You need to use the FIXCDS PATCH command to patch the FRSV record with the cpname, or if the FRSV record contains no other copy pool names, delete it with the FIXCDS DELETE command.
*ERR 186 ORPHANED K (FRD) RECORD FOUND, COPY POOL cpname NO LONGER EXISTS F (FRB) record could not be found for copy pool cpname.	
*ERR 187 K (FRD) RECORD EXISTS, F (FRB) RECORD INDICATES NO DUMP COPY The FRB record indicates that there is no dump copy for this copy pool version but an FRD record exists for it.	Either the FRD record should be deleted using the FIXCDS DELETE command or the FRD_EXISTS flag in the FRB_BVI should be patched on.
*ERR 188 K (FRD) RECORD NOT FOUND, F (FRB) RECORD FOR COPY POOL cpname INDICATES THAT IT DOES EXIST The FRB record indicates a dump for this copy pool exists, but the record representing the dump cannot be found.	
*ERR 189 H (FRVP) RECORD NOT FOUND COPY POOL cpname During the AUDIT VOLUMECONTROLS (RECOVERABLE), an error was detected.	You should consider running AUDIT COPYPOOLCONTROLS (cpname) to learn more regarding this discrepancy.
*ERR 190 VOLUME volser COULD NOT BE FOUND IN ANY ASSOCIATED H (FRVP) RECORD FOR COPY POOL cpname	You should consider running AUDIT COPYPOOLCONTROLS (cpname) to learn more regarding this discrepancy.
*ERR 191 P RECORD FOUND FOR VOLUME volser, IS NOT INDICATED AS A COPY POOL VOLUME	
*ERR 192 F RECORD FOUND FOR COPYPOOL cpname, WHICH VOLUME volser BELONGS TO	Audit the specified copy pool to determine the recoverability of this volume.
*ERR 193 J RECORD NOT FOUND FOR VOLUME volser	► The MCP or DGN record for volume volser indicates that an FRSV record should exist for this volume. ► Expect this error to be issued for each dumped version of this volume. ► The MCPF_COPYPOOL bit should be patched off if it is determined that this is no longer a copypool volume, as is indicated by the missing FRSV record. ► FIXCDS P volser PATCH(X'9' BITS(.....0)).

In order to correct the error type that was reported in Figure 7-60 on page 259, we find the following description in Table 7-6 on page 259:

“ERR 177 I (FRTV) RECORD WAS EXPECTED FOR VOLUME volser AND WAS NOT FOUND

A FRB record indicates that this is a valid version. The volume represented by this FRTV record needs to be made available and an FRTV record should be created to represent it or the fast replication backup of this copy pool fails.

The troubleshooting hint says:

```
The FIXCDS CREATE command can be used to recreate the FRTV record.
```

The structure of a FRB record (see Example 7-61 on page 258) is rather simple. We just need to learn which are the source volumes of the volumes mentioned in the messages of error type 177. This can be done by a command, as shown in Example 7-64.

Example 7-64 LIST COPYPOOL command

```
HSEND LI CP(CP1)
```

The output shown in Figure 7-61 gives a quick answer and allows to create the commands as shown in Example 7-65.

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 15:32:31 ON 07/03

COPYPOOL=CP1

VERSION  VTOCENQ     DATE          TIME     FASTREPLICATIONSTATE  DUMPSTATE
  006      Y      2007/03/12      16:01:45   RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C'EX6'
TOKEN(H)=X'C5E7F6'
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SG1      MHL0A0 - MHL0AF  MHL0A1 - MHL1AC
```

Figure 7-61 Output of LIST CP(CP1)

Example 7-65 FIXCDS commands to create FRTV records

```
HSEND FIXCDS I MHL1AC CREATE(0 MHL0A1)
HSEND FIXCDS I MHL0AF CREATE(0 MHL0A0)
```

```
DATA BEFORE PATCH
+0000  00000000 00000000
 *                              *
DATA AFTER PATCH
+0000  D4C8D3F0 C1F10000
 *MHL0A1                        *
```

Figure 7-62 Messages in response to FIXCDS I MHL1AC CREATE command

8

DFSMSrmm enhancements

The DFSMSrmm enhancements in DFSMS V1.8 provide improvements in the areas of enterprise level interface, UTC implementation, tape data set authorization, vital record specification policy management simplification, and usability items.

The following topics are covered:

- Support true e-mail address for the RMM NOTIFY function
- Setting up DFSMSrmm common time support
- DFSMSrmm VRS policy management simplification
  - Separation of Data Set Name Mask from the Policy
  - Release options applied if VRS matched
  - Special ABEND and OPEN via DSNAME match
  - Find unused VRSs
  - Incomplete VRS chains - dummy VRS *broken*
  - Toleration and removal of old functions
  - Conversion to DFSMSrmm from other tape management systems
- DFSMSrmm usability items
  - Updates to RMM TSO SEARCHVOLUME subcommand
  - ISPF lists show retention information
  - SELECT primary command in RMM dialog search results
  - ISPF lists show retention information
  - Rexx variable constraint relief
- Enabling ISPF Data Set List (DSLIST) support
- Prepare for future releases
  - Set a DFSMSrmm control data set ID
  - Re-allocate your DFSMSrmm control data set
  - Update LRECL for REPORT, BACKUP, and JRNLBKUP DD
  - Migrate from VRSEL(OLD) to VRSEL(NEW)

8.1 Support true e-mail address for the RMM NOTIFY function

The established DFSMSrmm notification feature is extended to optionally send mails to Internet addresses. E-mail notification is provoked in the same way as the known notifications. There is no difference in the current processing, only the result (e-mail instead of the established messages) is different.

Prerequisites for e-mail notification:

- SMTP server available in the customer environment and known to DFSMSrmm
- Owner attribute, e-mail address is set

If you specified an e-mail address, you must have an SMTP server configured and started.

Information about how to set up an SMTP server can be found in the manual *z/OS Communications Server: IP Configuration Guide*, SC31-8775.

By default, DFSMSrmm tries to use an SMTP server called SMTP on the current JES node. To tell DFSMSrmm to use any other SMTP server, use the restricted owner SMTP.

Restriction: The owner SMTP is now a reserved owner name value that you can use to configure the node name and SMTP server address space or machine name.

You use the NODE operand to identify the node that runs the SMTP server, and the USERID operand to identify the SMTP server.

Figure 8-1 shows the difference between the use of a notification message and a true e-mail address.

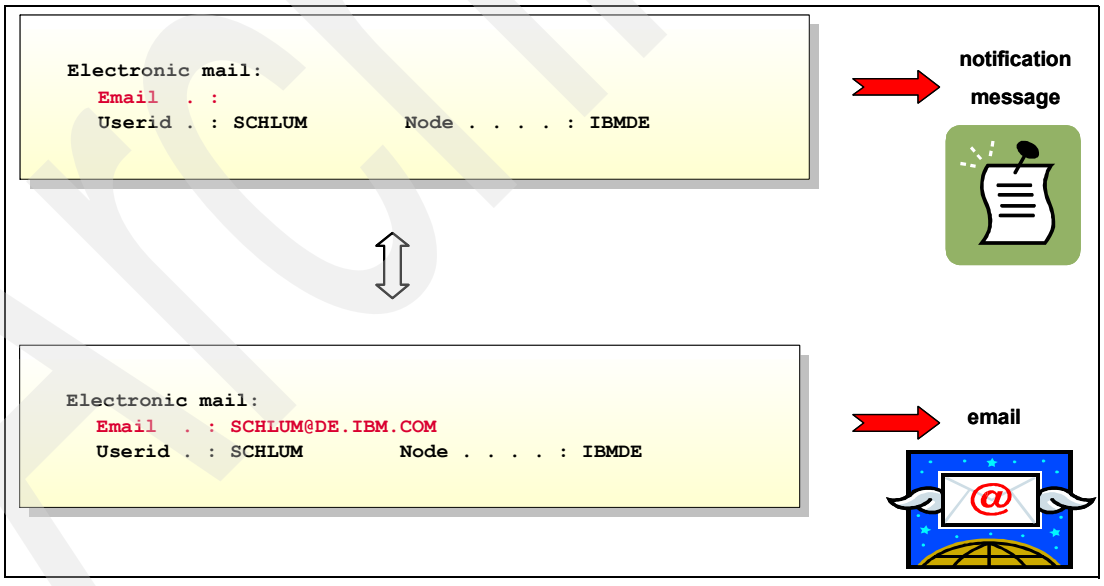


Figure 8-1 Difference between notification message and true e-mail address

The trigger for using e-mail notification is the owner's e-mail address. As long as there is no e-mail address defined to DFSMSrmm, DFSMSrmm will send the notification to node/user ID. If an e-mail address is defined, DFSMSrmm sends the notification to this address, whether a node/user ID is defined or not.

8.2 Support true e-mail address implementation

To set up e-mail notification, five topics need to be addressed.

8.2.1 Basic setup

Basic setup items that are common for normal notification and e-mail notification are:

- PARMLIB option NOTIFY(YES)

  Specifies whether DFSMSrmm should automatically notify volume owners when the volumes they own become eligible for release or when software product volumes are added.

- Volume RELEASEACTION(NOTIFY)

  Must be specified as part of the GETVOLUME, ADDVOLUME, or CHANGEVOLUME subcommands to get notifications sent, when a volume is released.

- DFSMSrmm under JES2 or JES3

  You must be running DFSMSrmm under the JES2 or JES3 subsystem and not the master subsystem.

8.2.2 Defining an owner's e-mail address

The e-mail address of an owner can be defined using the DFSMSrmm dialog, as shown in Figure 8-2, or using the appropriate TSO subcommands, like ADDOWNER or CHANGEOWNER.

```
  Panel  Help
 ------------------------------------------------------------------------------
                    DFSMSrmm Add Owner Details - SCHLUM
Command ===>


  Owner id . . SCHLUM
  Surname  . . Schlumberger
  Forenames    Norbert
  Department   INTERNATIONALE PROJEKTE (6304)

  Address:
    Line 1 . . IBM DEUTSCHLAND INFORMATIONSSYSTEME GMBH
    Line 2 . . AM KELTENWALD 1
    Line 3 . . 71139 EHNINGEN (GERMANY)

  Telephone:
    Internal   919-3579     External . . . +49-(0)7056-964522

  Electronic mail:
    Email  . . SCHLUM§DE.IBM.COM
    Userid . . SCHLUM       Node . . . . . IBMDE

Press ENTER to ADD owner details, or END command to CANCEL.
```

Figure 8-2 Specifying a true e-mail address using the add owner dialog panel

Use Example 8-1 to add a new owner using the RMM TSO ADDOWNER or CHANGE OWNER subcommand.

Example 8-1 Specifying a true e-mail address using the RMM TSO subcommands

```
            RMM ADDOWNER SCHLUM +
                ADDR1('IBM DEUTSCHLAND INFORMATIONSSYSTEME GMBH') +
                ADDR2('AM KELTENWALD 1') +
                ADDR3('71139 EHNINGEN (GERMANY)') +
                DEPARTMENT('INTERNATIONALE PROJEKTE (6304)') +
                EMAIL('SCHLUM§DE.IBM.COM') +
                EXTEL('07034-15-3579') +
                FNAME('NORBERT') +
                INTEL('919-3579') +
                NODE(IBMDE) +
                SNAME('SCHLUMBERGER') +
                USER(SCHLUM)

  or

            RMM CHANGEOWNER SCHLUM +
                EMAIL('SCHLUM§DE.IBM.COM')
```

If you specified an e-mail address, you must have an SMTP server configured and started.

Information about how to set up an SMTP server can be found in the manual *z/OS Communications Server: IP Configuration Guide*.

By default, DFSMSrmm tries to use an SMTP server called SMTP on the current JES node.

8.2.3 SMTP server

To tell DFSMSrmm to use any other SMTP server, use the restricted owner named SMTP. The Owner SMTP is now a reserved owner name value that you can use to configure the node name and SMTP server address space or machine name.

You use the NODE operand to identify the node that runs the SMTP server, and the USERID operand to identify the SMTP server. The default used by DFSMSrmm is:

node Current JES node
name SMTP

Use the restricted owner SMTP to change the default values:

node The node that runs the SMTP server
userid The SMTP server task name

Note: If you add the owner SMPT you must specify both values.

Use the TSO NETSTAT command, as shown in Example 8-2, to get the server task name of your SMTP.

Example 8-2 Sample NETSTAT command

```
  TSO NETSTAT
```

Figure 8-3 shows you the result of the NETSTAT command.

```
EZZ2350I MVS TCP/IP NETSTAT CS V1R7       TCPIP Name: TCPIP           13:32:28
EZZ2585I User Id  Conn     Local Socket           Foreign Socket         State
EZZ2586I -------  ----     ------------           --------------         -----
EZZ2587I BPXOINIT 00000044 0.0.0.0..10007         0.0.0.0..0             Listen
EZZ2587I CNMPEGZB 00013534 0.0.0.0..4080          0.0.0.0..0             Listen
EZZ2587I CNMPEGZB 00013536 0.0.0.0..4020          0.0.0.0..0             Listen
EZZ2587I CNMPEGZB 00013535 0.0.0.0..4092          0.0.0.0..0             Listen
EZZ2587I DB2BDIST 000048EF 0.0.0.0..5022          0.0.0.0..0             Listen
EZZ2587I DB2BDIST 000048F2 0.0.0.0..5021          0.0.0.0..0             Listen
EZZ2587I DFSKERN  00000031 0.0.0.0..139           0.0.0.0..0             Listen
EZZ2587I INETD1   00000057 0.0.0.0..923           0.0.0.0..0             Listen
EZZ2587I INETD1   00000054 0.0.0.0..512           0.0.0.0..0             Listen
EZZ2587I INETD1   00000056 0.0.0.0..514           0.0.0.0..0             Listen
EZZ2587I INETD1   00000055 0.0.0.0..513           0.0.0.0..0             Listen
EZZ2587I NPM      00000019 0.0.0.0..1965          0.0.0.0..0             Listen
EZZ2587I OMPROUTE 00000033 127.0.0.1..1026        127.0.0.1..1027        Establ
EZZ2587I TAPEMAN  00000046 9.149.157.65..35043    0.0.0.0..0             Listen
EZZ2587I TCPFTPH1 00000039 0.0.0.0..21            0.0.0.0..0             Listen
EZZ2587I TCPIP    00000016 127.0.0.1..1024        127.0.0.1..1025        Establ
EZZ2587I TCPIP    0000001F 0.0.0.0..423           0.0.0.0..0             Listen
EZZ2587I TCPIP    00000015 127.0.0.1..1025        127.0.0.1..1024        Establ
EZZ2587I TCPIP    000186C1 9.149.157.65..23       9.157.16.221..2005     Establ
EZZ2587I TCPIP    00000012 127.0.0.1..1024        0.0.0.0..0             Listen
EZZ2587I TCPIP    00018178 9.149.157.65..23       9.143.10.29..1695      Establ
EZZ2587I TCPIP    0000001E 0.0.0.0..623           0.0.0.0..0             Listen
EZZ2587I TCPIP    00000020 0.0.0.0..23            0.0.0.0..0             Listen
EZZ2587I TCPIP    0000002E 127.0.0.1..1027        127.0.0.1..1026        Establ
EZZ2587I TCPSMTP  0000003E 0.0.0.0..25            0.0.0.0..0             Listen
EZZ2587I WWW      000186F3 9.149.157.65..80       9.157.16.217..2796     FinWai
EZZ2587I WWW      00017ED9 0.0.0.0..80            0.0.0.0..0             Listen
EZZ2587I DFSKERN  00000030 0.0.0.0..137           *..*                   UDP
EZZ2587I DFSKERN  00000032 0.0.0.0..138           *..*                   UDP
EZZ2587I OMPROUTE 000186F5 0.0.0.0..60380         *..*                   UDP
EZZ2587I TCPSMTP  0000003F 0.0.0.0..1039          *..*                   UDP
EZZ2587I VTAM     0000000D 9.149.157.65..12003    *..*                   UDP
EZZ2587I VTAM     0000000B 9.149.157.65..12001    *..*                   UDP
EZZ2587I VTAM     0000000E 9.149.157.65..12004    *..*                   UDP
EZZ2587I VTAM     0000000C 9.149.157.65..12002    *..*                   UDP
```

Figure 8-3 NETSTAT output

Note: Normally, port TCP 25 is reserved for SMTP. Verify that the name of the member containing the SMTP cataloged procedure has been added to the PORT statement in hlq.PROFILE.TCPIP.

To get the node name use the display JES2 Network Environment command, as shown in Example 8-3.

Example 8-3 Display the JES2 Network Environment

```
/$D NJEDEF
or
/$D NJEDEF
```

Figure 8-4 shows you the result of the JES2 Network Environment display command.

```
 RESPONSE=SC70
  $HASP831 NJEDEF
  $HASP831 NJEDEF  OWNNAME=WTSCPLX2,OWNNODE=1,DELAY=300,
  $HASP831         HDRBUF=(LIMIT=100,WARN=80,FREE=100),
  $HASP831         JRNUM=1,JTNUM=1,SRNUM=7,STNUM=7,
  $HASP831         LINENUM=40,MAILMSG=YES,MAXHOP=0,
  $HASP831         NODENUM=999,PATH=1,RESTMAX=0,
  $HASP831         RESTNODE=100,RESTTOL=0,TIMETOL=0
```

Figure 8-4 Display JES2 Network Environment result

8.2.4 E-mail message configuration

DFSMSrmm supports two notification tasks:

- Volume release notification - Notify the owner when the owner's volumes are eligible for release.
- Product notification - Notify the designated product owner when a program product volume is added.

For each task EDGMTAB contains two different sets of messages for normal and e-mail notification. This enables you to design two completely independent notification messages, one for the known notification way and one for e-mails. Figure 8-5 shows you a sample e-mail notification if a volume is changed to status pending release.

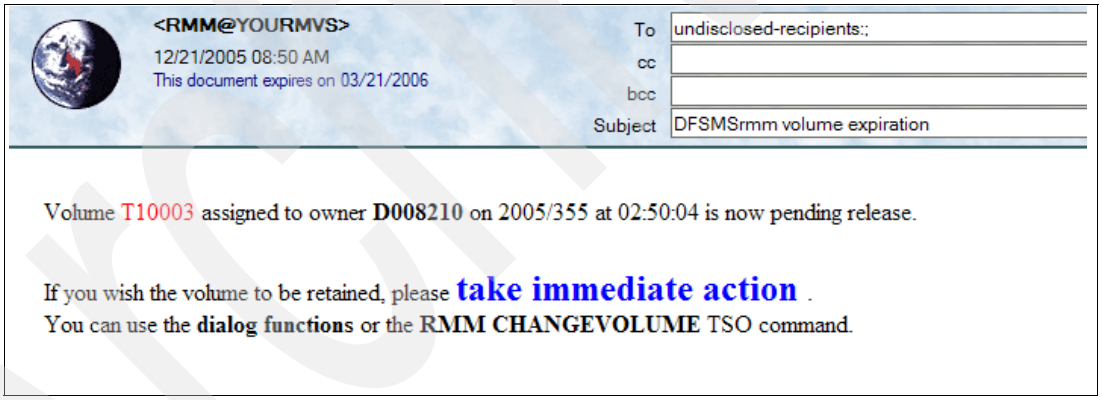


Figure 8-5 Sample volume release notification e-mail

The default messages (2450-2463) delivered in EDGMTAB create an e-mail as shown above if a volume is pending release. The established messages (2405–2409) generate plain text, as shown in Figure 8-6.

```
Subject: DFSMSrmm volume expiration
Volume T10003 assigned to owner D008210  on 2005/355   at 02:50:04
is now pending release.
If you wish the volume to be retained, please take immediate action.
You can use the dialog functions or the RMM CHANGEVOLUME TSO command.
```

Figure 8-6 Sample volume release notification message

To prevent that, DFSMSrmm changes the volume status to the status SCRATCH and retains the volume for an additional 90 days using the command shown in Example 8-4.

Example 8-4 Use the RMM TSO CHANGEVOLUME subcommand to change the expiration date

```
   RMM CHANGEVOLUME T10003 RETPD(90)
or
   RMM CV T10003 RETPD(90)
```

```
   ....
   EDGMSGB 2456,TYPE=I,MOD=YES,MSGID=NO,
                  ('Volume <FONT COLOR=RED>¢¢¢¢¢</FONT> ',
                   'assigned to owner <b>¢¢¢¢¢¢¢</b> ')
   SPACE 2
   EDGMSGB 2457,TYPE=I,MOD=YES,MSGID=NO,
                   'on ¢¢¢¢¢¢¢¢¢ at ¢¢:¢¢:¢¢'
   SPACE 2
   EDGMSGB 2458,TYPE=I,MOD=NO,MSGID=NO,
                  ('is now pending release.<p>',
                   'If you wish the volume to be retained, please ')
   SPACE 2
   EDGMSGB 2459,TYPE=I,MOD=NO,MSGID=NO,
                  ('<font size="+2" color=blue><b>take immediate action ‘,
                   '</font></b>.<br>')
   SPACE 2
   EDGMSGB 2460,TYPE=I,MOD=NO,MSGID=NO,
                   ('You can use the <b>dialog functions</b> ',
                    'or the <b>RMM CHANGEVOLUME</b>')
   SPACE 2
   EDGMSGB 2461,TYPE=I,MOD=NO,MSGID=NO,
                   ' TSO command.'
   ....
```

Figure 8-7 Messages 2456 - 2461 in EDGMTAB

These messages can contain HTML code to format the text, as shown in Example 8-5.

Example 8-5 Sample HTML code

```
   ('<font size="+2" color=blue><b>take immediate action '</font></b>.<br>')
```

Figure 8-8 shows you a sample e-mail notification if a new software product volume is added.

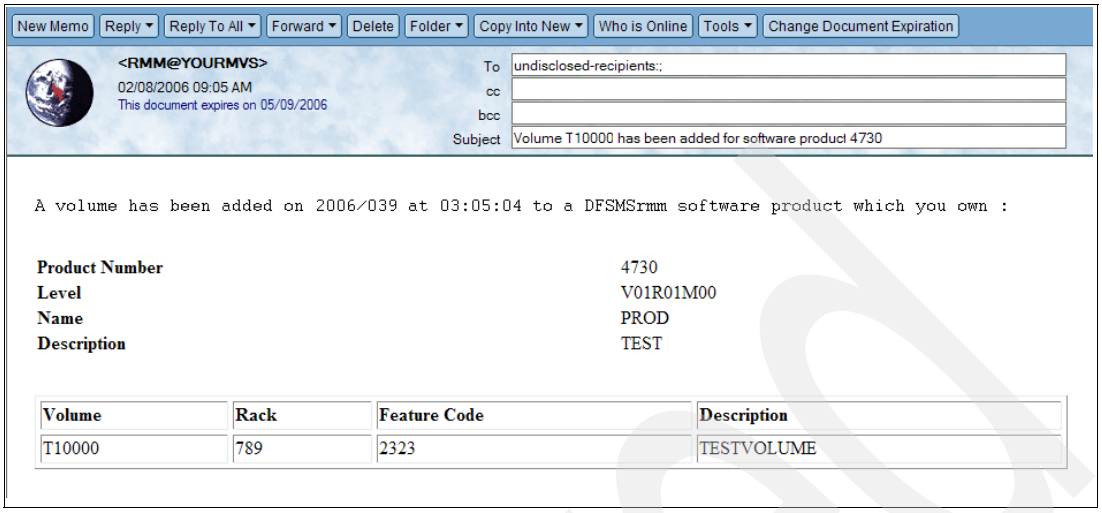


Figure 8-8 Sample add software product volume notification e-mail

The default messages (2720–2739) delivered in EDGMTAB create an e-mail, as shown above, if a program product volume is added. The established messages (2700–2713) generate plain text, as shown in Figure 8-9.

```
Subject: Volume T10000 has been added for software product 4730
A volume has been added on 2005/355 at 02:51:06 to a DFSMSrmm
software product which you own:
  Product Number  = 4730      Level = V01R01M00
  Name                = PROD
  Description         = TEST

  Volume   Rack     Feature Code   Description
  ------        ------      ------------          ------------------------------
  T10000    789       2323              TESTVOLUME
```

Figure 8-9 Sample add software product volume notification message

8.3 Setting up DFSMSrmm common time support

Before DFSMSrmm common time support (UTC), also known as GMT, is enabled, all dates and times are stored in the DFSMSrmm control data set in local time. When the control data set is shared, and the sharing systems are set to run in different time zones, the local dates and times in the control data set may be from any of your systems. When you display information or extract records, you need to be aware of how the records were created, on which system, and where they may have been updated in order to interpret the dates and times shown. The same consideration also applies for records created or updated prior to enabling common time support because DFSMSrmm assumes that they are times local to the system running the DFSMSrmm subsystem and converts the values based on that assumption.

When you enable common time support, DFSMSrmm maintains the records in the control data set in common time. Most date and time fields are paired together to enable an accurate conversion to and from common time and between different time zones. In some cases, DFSMSrmm has date fields in control data set records, and there is no associated time field.

For these date fields, DFSMSrmm uses an internal algorithm that approximates conversion between time zones based on the time zone offsets involved.

Warning: Using the SET system command with either the DATE or the TIME keyword, or both, or replying to message IEA888A to run the system on future or past dates can affect the way that DFSMSrmm calculates local times. In order to get the correct results from DFSMSrmm processing when you need to test with future or past dates, you should alter the TOD clock and keep the time zone offset as before.

8.3.1 DFSMSrmm - using the date and time

Dates and times in DFSMSrmm are used mostly for reference, but also to calculate retention periods. Examples are:

- Journal record prefix

  Contains date and time that the journal records were written, compared for forward recovery.

- Record creation date and time

  In any record stored in the DFSMSrmm control data set.

- Expiration date

  In volume and data set records. Shows the date that the entity is expired, but not the time.

- Tokens

  In control data set records. Used for various purposes but usually as a unique value to associate multi-volumes multi-files, or logical volumes and stacked volumes together.

- Reference dates

  In control data set records.

Note: Dates and times are always displayed to the user in the same format that they were recorded, in local time.

All current uses of date and time in DFSMSrmm use the TIME DEC option of the TIME macro. This retrieves the local date and time. DFSMSrmm assumes that all systems that share a control data set (CDS), and therefore the JOURNAL as well, run on a synchronized time source.

Although in theory there is supposed to be a common date and time function available in DFSMSrmm and used for all date conversions, in practice it is used only when a date conversion is required or to get the current date in DFSMSrmm internal format. It is not used where the time is wanted or when the date is wanted not in internal format.

DFSMSrmm uses an internal date format that is derived via TIME DEC from the TOD — a 4-byte, 7-character packed decimal number YYYYDDDs. Using this value DFSMSrmm has no known limit to the dates that can be handled. The internal time is a seven-character packed decimal number also, HHMMSSTs. DFSMSrmm does not need to upgrade to using a 128-bit timer value, nor have any problems with the TOD clock rolling over in 2043.

8.3.2 Date and time in a DFSMSrmm client server environment

All dates and times used on the client system are local time. All dates and times used on the server are in local time. When data is retrieved from the DFSMSrmm control data set (CDS)

and displayed by subcommand processing, the dates and times used are exactly as stored in the DFSMSrmm CDS. No conversion from server time zone to client time zone is performed. Figure 8-10 shows an example of when there is a need to use the DFSMSrmm common time support.

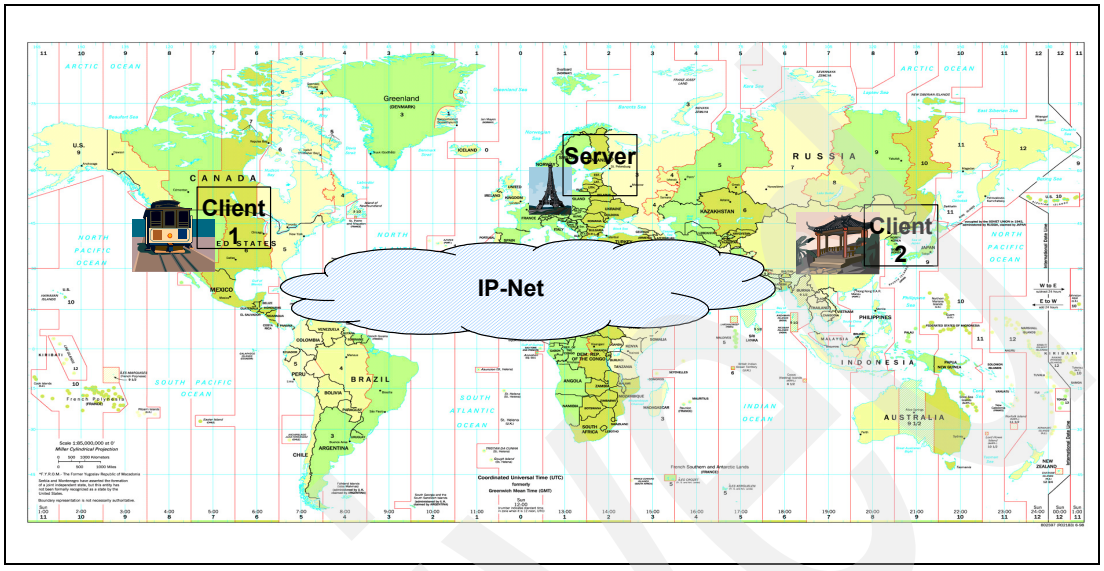


Figure 8-10 DFSMSrmm client server implementation

8.3.3 Enable common time support

Before you can enable the DFSMSrmm common time support you must check that the (time of Day (TOD) clocks of all systems in the RMMPLEX are set to Universal Time, Coordinated (UTC).

To enable common time support:

1. Ensure that all systems in the RMMplex have toleration maintenance installed or are at z/OS V1R8 or later, and all applications dependent on the correct date and time information from DFSMSrmm are updated to support the new time zone support if required. DFSMSrmm subcommand output remains in local time, so most applications do not need to change unless they are to exploit the availability of the time zone offset.
2. Ensure that the system time of day clock is set to GMT on all systems in the RMMplex. It is common practice for the system to use local time based either on the TIMEZONE value in the CLOCKxx member of PARMLIB or from an external time source.
3. Run the EDGUTIL utility with UPDATE with the UTC(YES) operand on the CONTROL statement of the SYSIN file to enable common time support.

Important: You cannot disable common time support once it is enabled.

UTC(YES) enables DFSMSrmm common time support. Prior to enabling this support all systems in the RMMplex should have toleration maintenance installed or be z/OS V1.8 or later, and applications dependent on the correct date and time information from DFSMSrmm should be updated to support the new time zone support if required.

Figure 8-11 shows you the result of a RMM TSO LISTCONTOL if you do not have DFSMSrmm common time support enabled.

```
Control record:
Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                 Update date = 2007/082    Update time = 17:49:39
Journal: Utilization =  17% (75% threshold)     STATUS: = ENABLED
CDS:     Utilization =  81%
Exit status:                              Options:
  EDGUX100 = ENABLED                        Stacked Volumes         = NONE
  EDGUX200 = NONE                           Extended Bin            = DISABLED
                                            Common Time             = DISABLED
                                            CDSID ENQ name          = ENABLED
Last backup:                              Last expiration processing:
  Date = 2007/079    Time = 13:25:42        Date = 2007/079    Time = 13:25:39
Last journal backup:                      Last store update:
  Date = 2007/079    Time = 13:25:42        Date = 2007/072    Time = 18:30:39
Last report extract:                      Last VRS processing:
  Date = 2007/072    Time = 18:30:39        Date = 2007/079    Time = 13:25:39
Last scratch procedure:                   Last Catalog synchronize:
  Date =             Time =                 Date =             Time =
Rack numbers       = 6258                 Empty racks        = 4212
LOCAL store bins   = 0                    Empty LOCAL bins   = 0
DISTANT store bins = 0                    Empty DISTANT bins = 0
REMOTE store bins  = 0                    Empty REMOTE bins  = 0
Control functions in progress:
Backup         = N  Restore        = N
Verify         = N  Expiration     = N
Report Extract = N  Disaster Store = N
VRS            = N  Synchronize    = N
Client/Server:
  host name  =
  IP address =
```

Figure 8-11 Common time support disabled

Use Example 8-6 to enable DFSMSrmm common time support.

Example 8-6 Enable UTC support

```
//UTIL     EXEC PGM=EDGUTIL,PARM='UPDATE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.CONTROL.DSET
//SYSIN    DD   *
 CONTROL UTC(YES) CDSID(SC70)
/*
```

Where:

CDSID Specifies one-to-eight alphanumeric characters that identify the control data set by name. There is no default. A CDSID is required in z/OS 1.9 or later.

UTC Enables DFSMSrmm common time support.

Once you enable common time support DFSMSrmm starts to record CDS record dates and times in common time and converts existing values, as required, from local times to common time. Figure 8-12 shows you an example of a display data set.

Each user can set his own common time zone values to display the date and time information correctly. Example 8-8 on page 284 shows you the DFSMSrmm dialog user option selection where the time zone can be specified.

```
  Panel  Help
------------------------------------------------------------------------------
EDGP§OP1                DFSMSrmm Dialog User Options
Command ===>

Date format . . . . . JULIAN          ( American, European, Iso or Julian )
Time zone . . . . . . MST   -04:00:00  ( zone  offsetHH:MM:SS )

Confirm deletes . . . YES             ( Yes or No )

Processing option . . F                F - Foreground, B - Background

Eject option  . . . . C                C - Convenience, B - Bulk

Variable reuse  . . . Y                Y - Yes, N - No

Job statement information:-

  ===> //RMMJOB   JOB ,RMM,NOTIFY=&SYSUID,
  ===> //         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
  ===> //*
  ===> //*

      Enter END command to save changes, or CANCEL to end without saving.
```

Figure 8-12 Set your own common time zone value

All dates and times displayed or entered in the DFSMSrmm dialog are values local to this time zone. To change the time zone you must specify an offset value and a text string to identify that zone to you. The offset value is the time that your selected time zone is ahead of or behind universal time (UTC/GMT). Changes you make only affect future dialog actions and displays.

The report extract data set contains date and time values in the local time of the running system. The extract header record includes a field that lists the time zone offset, as shown in Figure 8-13.

```
H          2007/072  183039SC64                                    JN-04:00:00
```

Figure 8-13 Report extract data set time zone offset

Recommendation: We recommend that you have the system TOD clock set to GMT and enable DFSMSrmm to use UTC. You do this by using the EDGUTIL utility with the UPDATE parameter. Once you do this, any newly recorded dates and time will be stored in common time and existing records are converted to be in all common time as they are updated. You will continue to see dates and times presented in local time because DFSMSrmm handles the conversion from common time to your local time.

Set the TIMEZONE in the CLOCKxx PARMLIB member

TIMEZONE d.hh.mm.ss specifies the difference between the local time and the Coordinated Universal Time (UTC). If ETRMODE YES and ETRZONE YES are specified (and an operational Sysplex Timer® is available), the system ignores the TIMEZONE parameter.

d Specifies the direction from UTC.
Value Range: E for east of UTC or W for west of UTC.
Default: W.

hh.mm.ss Specifies the number of hours (hh), minutes (mm), and seconds (ss) that the local time differs from the UTC.

Value Range: The value for hh must be between 00 and 15. The value for mm and ss must be between 00 and 59. mm.ss values are optional.

In addition, the combined hh.mm.ss value must be within the range 00:00:00–15:00:00. This means that a value like 15.59.59 is not valid because it is outside the range, even though the hh portion is between 00 and 15 and the mm and ss portions are between 00 and 59. If the mm portion or ss portion or both are omitted, a default value of 00 is applied to the omitted portion, and appears in message IEA598I at IPL time. For example, if CLOCKxx contains TIMEZONE W.15, then at IPL time, message IEA598I indicates:
IEA598I TIME ZONE = W.15.00.00
Default: 00.00.00

See Example 8-7 for how you can use the DISPLAY T command to display the local time of day and date and the UTC of day and date.

Example 8-7 Displaying the local and coordinated universal time and date

```
D T
```

The local time of day and date and the coordinated universal time of day and date are to be displayed (message IEE136I), as shown in Figure 8-14.

```
RESPONSE=SC70      IEE136I LOCAL: TIME=12.56.19 DATE=2007.068  UTC:
RESPONSE=TIME=17.56.19 DATE=2007.068
```

Figure 8-14 Result of the displaying the local and coordinated universal time and date

8.3.4 Potential problems using local time

Some potential problems are:

- Systems sharing the CDS but in different time zones create records using their own local date and time values.
- Journal records might appear to not be in sequence. However, the VSI count and record numbers show the correct sequence. Tokens vary but will at least always increment.
- Users see a date and time and believe it to be in their local time zone, but it could reflect that used on another system.
- Data sets or volumes could be released early.

8.4 DFSMSrmm VRS policy management simplification

In this new release the VRS processing changed and the RMM TSO subcommand and the reporting are enhanced as follows:

1. Separation of Data Set Name Mask from the Policy.
2. Release options applied if VRS matched.
3. Special ABEND and OPEN via DSNAME match.
4. Find unused VRSs.
5. Incomplete VRS chains - dummy VRS *broken*.
6. Toleration and removal of old functions.
7. Conversion to DFSMSrmm from other tape management systems.

Attention: The DFSMSrmm PARMLIB option VRSEL(OLD) will be removed in a future release. We recommend migrating from VRSEL(OLD) to VRSEL(NEW) before moving to z/OS V1.8. Each time you run VRSEL processing and VRSEL(OLD) is in use, the new message EDG2317E and a minimum return code of 4 are issued.

8.4.1 Separation of Data Set Name Mask from the Policy

Figure 8-15 shows the structure of a DSNAME VRS and how the concept of separating the data set name mask from the policy itself is put into effect.

The data set name VRS now allows a COUNT of zero so that this first VRS in a chain has no retention specification. The NEXTVRS in the chain and subsequent VRSs now specify the entire policy.

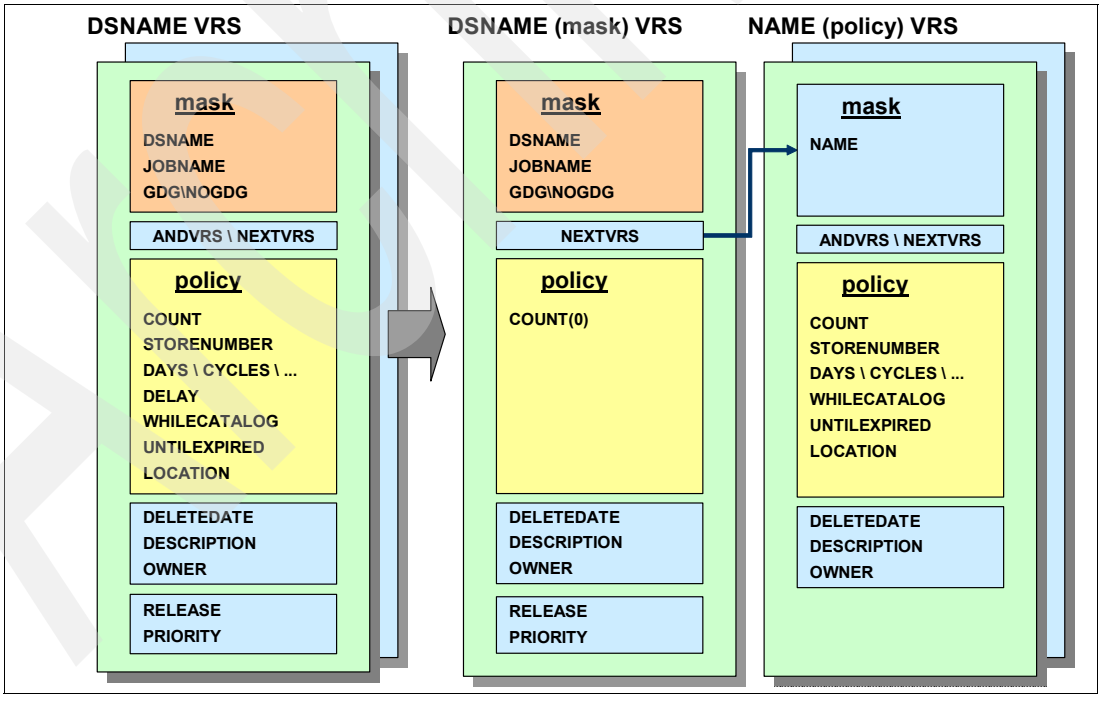


Figure 8-15 DSNAME VRS concept

Setting up DFSMSrmm retention and movement policies can require large numbers of VRSs to be created because each data set name mask VRS also contains the initial retention and movement information.

Separating the data set name mask from the policy itself, as shown in Figure 8-16, enables clear and well-defined service levels to be set up for tape management.

These policy/service-level VRSs can then be easily modified as required without changing the filters that select them.

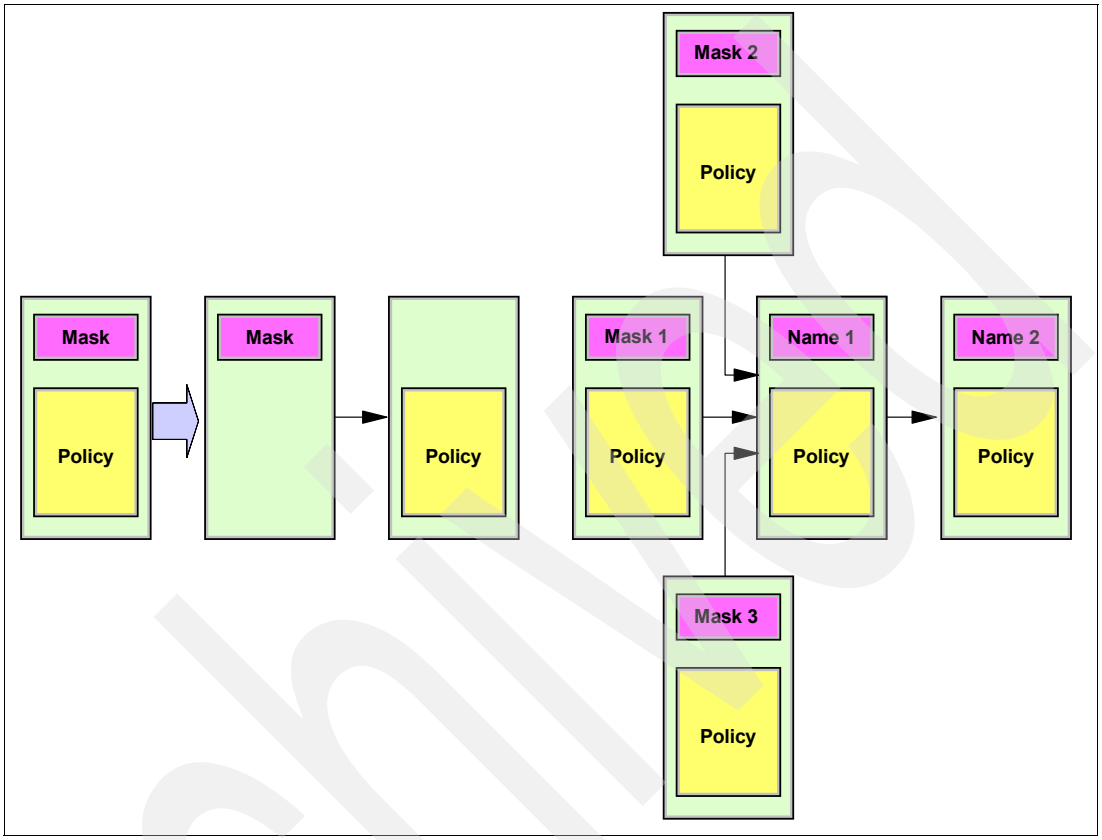


Figure 8-16 Separation of DSNAME mask from the policy

Note: VRSEL(NEW) is required to implement the *separating the data set name mask from the policy* processing.

If you run DFSMSrmm with VRSEL(OLD), for each VRS where COUNT is set to 0, DFSMSrmm issues the message EDG2225I. The processing continues, and DFSMSrmm sets a minimum return code of 4 and ignores the VRS. No data sets can match to or use the VRS.

8.4.2 Release options applied if VRS matched

Release options for matching VRSs can be applied regardless of whether the data set actually is ever VRS retained.

The combination of COUNT(0) and release option changes ensures that a volume can be scratched the same day that a data set on it is created. Figure 8-17 shows you how you can specify COUNT(0) in a DSNAME VRS definition.

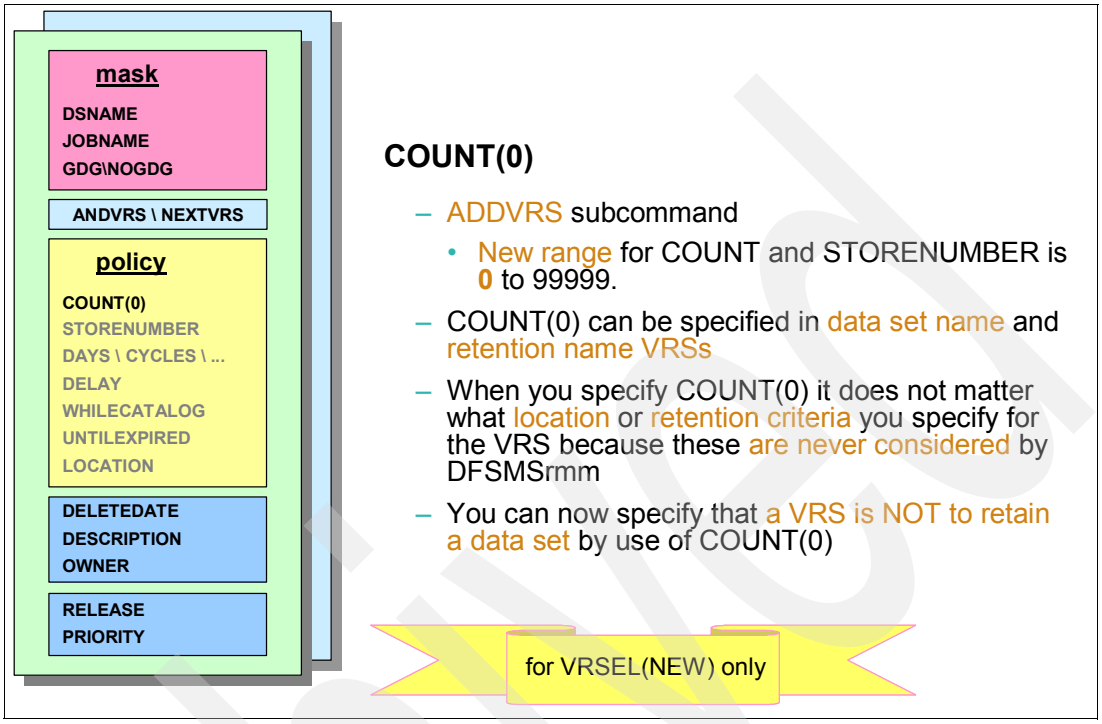


Figure 8-17 VRS count zero processing

When a data set is no longer retained by a vital record specification, DFSMSrmm releases the volume on which the data set resides only if no data set and the volume is retained by a vital record specification. If you use the DFSMSrmm EDGRMMxx PARMLIB OPTION command VRSEL(NEW) option and the RMM ADDVRS RELEASE(EXPIRYDATEIGNORE) operand, DFSMSrmm ignores the volume expiration date and uses information in a vital record specification to control retention. There are two special RELEASE options available, and you can select one or both of these options together:

- EXPIRYDATEIGNORE
- SCRATCHIMMEDIATE

DFSMSrmm does not immediately return a volume to scratch status or to its owner when a volume reaches its expiration date and is not retained by a vital record specification. You must run expiration processing two times to return a volume to scratch status or to its owner. The first run of expiration processing sets the volume status to pending release. The second run of expiration processing completes the return. Running expiration processing two times gives you time to make changes to the volume status before the volume is released.

Note: Sometimes DFSMSrmm cannot make the return in a single run, for example, there may be other release actions required.

If you do not need to run expiration processing in two runs, specify the DFSMSrmm EDGRMMxx PARMLIB OPTION command VRSEL(NEW) option and the RMM ADDVRS RELEASE(SCRATCHIMMEDIATE) operand. This enables you to return volumes to scratch in a single run of expiration processing. For more information about the RELEASE option refer to the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

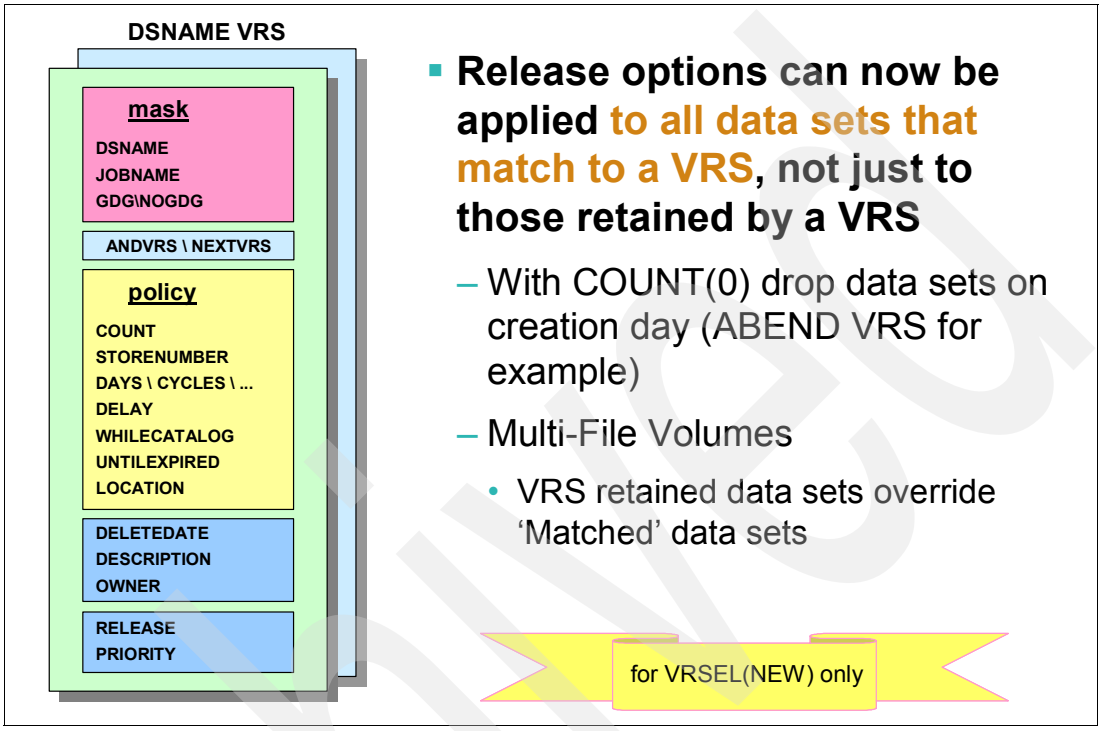


Figure 8-18 VRS release option

8.4.3 Special ABEND and OPEN via DSNAME match

The support for ABEND and OPEN is extended to allow selection of the appropriate policy using the data set name mask. You can use the reserved data set or job names ABEND and OPEN to specify policies for:

- Data sets closed as a result of an abnormal end (ABEND flag in the data set record ON) in a task
- Data sets that are left open (OPEN flag in the volume record ON) or are in use during inventory management.

Figure 8-19 shows you how you can specify the new reserved ABEND and OPEN masks in the ADD VRS subcommand.

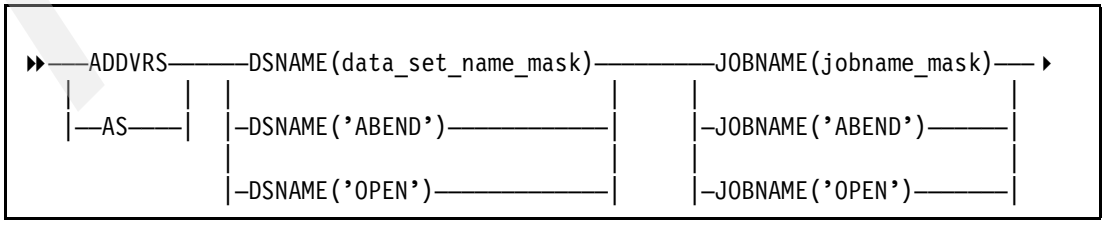


Figure 8-19 Add new ABEND or OPEN VRS

This allows you to use either data set name masks or job name masks to manage open or abend data sets.

The data set name mask or job name can even be used to match to data sets via management class or vital record specification management value as long as the data set name mask specified is not more than a single qualifier.

8.4.4 Find unused VRSs

With this release DFSMSrmm helps you manage your VRSs by VRSEL maintaining the last reference date (DLR) and last reference time (TLR) for each VRS. This new function also includes the following new functions:

- Counts the number of unused VRSs
- Identifies which VRS policy chains are not being used

You can use this information to identify and delete VRSs that are no longer required.

The VRS last reference date and time are now externalized in the report extract file in the output of the LISTVRS subcommand, and the last reference date is also shown in the REPORT file, as shown in Figure 8-20. Also, the REPORT file contains a list of VRS chains not used in this VRSEL run.

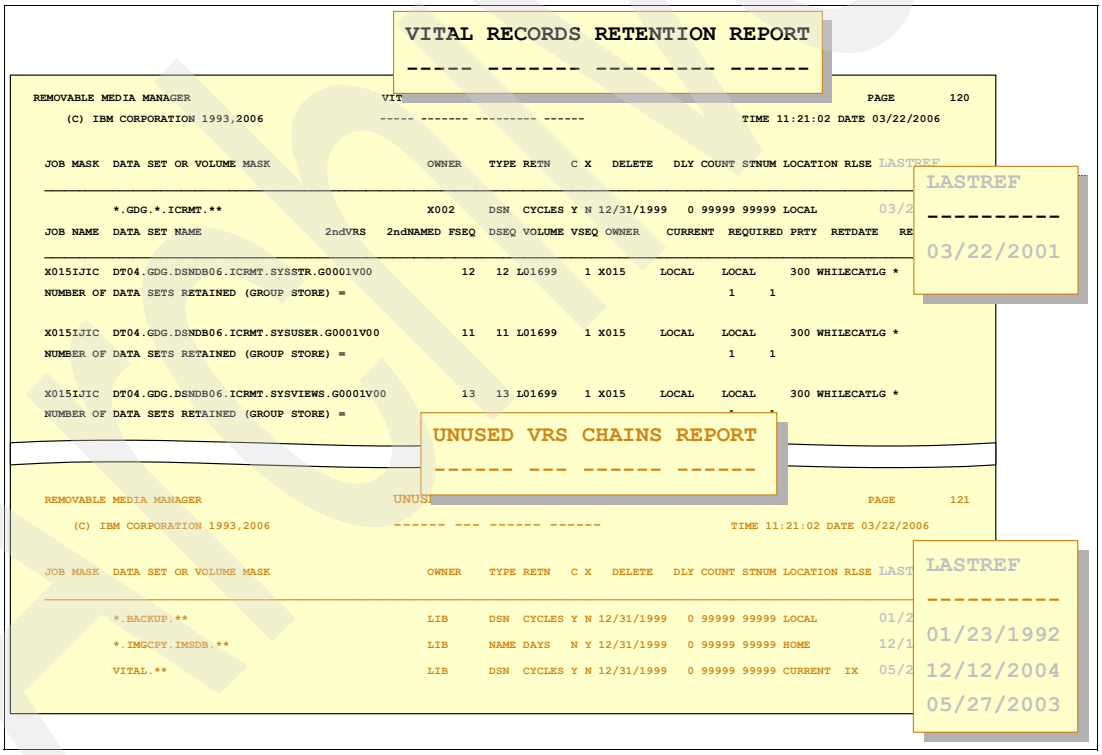


Figure 8-20 Vital records retention report

The MESSAGE file contains an additional message that shows the number of VRSs that are not used by this VRSEL run.

The maintenance of the VRS last reference date and time is supported only if you have specified VRSEL(NEW). In the REPORT file the LASTREF column is listed in any case:

VRSEL(NEW) It will be filled with dates.
VRSEL(OLD) It will be filled with blanks.

8.4.5 Incomplete VRS chains - dummy VRS *broken*

If VRSEL processing finds an incomplete chain it is reported with EDG2230I, but the processing continues. In this case DFSMSrmm sets a minimum return code of 4 and any data set that matches this incomplete chain is retained by a special Name VRS *broken*. Figure 8-21 shows you an example of an incomplete VRS chain.

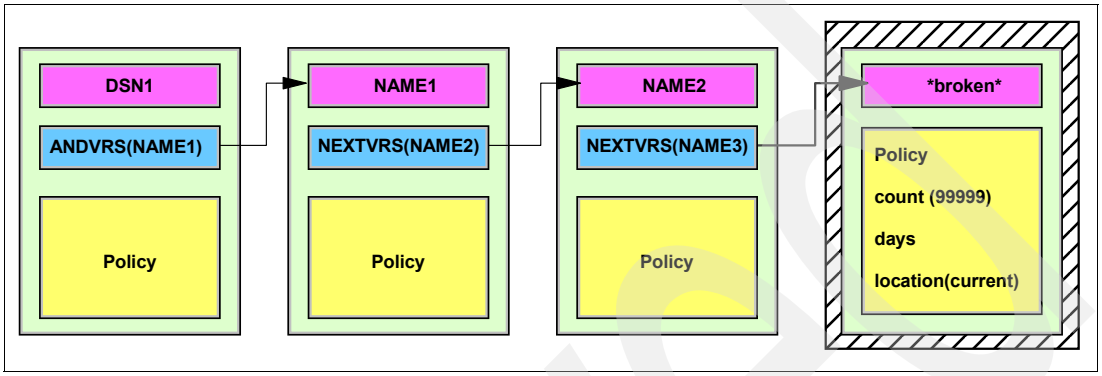


Figure 8-21 Incomplete VRS chain

The two new messages that you can get if the VRSEL processing detects errors are:

EDG2230I `NEXTVRS name_vrs DOES NOT EXIST. CHAINING vrs_type VRS IS vrs_mask.`

Where:

vrs_name The NAME vital record specification defined by NEXTVRS.

vrs_type DSN - DSNAME vital record specification.
VOL - VOLUME vital record specification.
NAME - NAME vital record specification.

vrs_mask The mask that uniquely identifies the vital record specification with the chaining error. For DSNAME vital record specifications the mask includes the data set name and optionally the job name.

Severity Information.

Explanation During inventory management vital record processing, DFSMSrmm checks all vital record specification chains by following the chain using the NEXTVRS values. The vital record specification displayed in the message does not exist in the DFSMSrmm control data set.

Source DFSMSrmm.

Detecting Module EDGVREC0.

System Action Processing continues and DFSMSrmm sets a minimum return code of 4. For VRSEL(OLD) DFSMSrmm retains additional data sets or volumes, in the home location, up to the COUNT value specified in the initial VRS in the chain. For VRSEL(NEW) additional data sets are retained in the current location, permanently.

Operators Response None.

Sysprogr Response ´ Add the missing vital record specification or correct the NEXTVRS value specified on the vital record specification displayed in the message.

Routing Codes N/A.

Descriptor Codes None.

EDG2317E `MIGRATION FROM VRSEL(OLD) TO VRSEL(NEW) IS RECOMMENDED.`

Where:

Severity	Warning.
Explanation	During inventory management vital record processing, DFSMSrmm checks the VRSEL option that you defined in the EDGRMMxx PARMLIB. The VRSEL(OLD) option will be removed in a future release of z/OS. Migrate using VRSEL(NEW).
Source	DFSMSrmm.
Detecting Module	EDGMHKP.
System Action	Processing continues. A minimum return code of 4 is set.
Operators Response	None.
Sysprogr Response´	Plan a migration to VRSEL(NEW). Refer to the migration planning steps documented in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.
Routing Codes	11.
Descriptor Codes	None.

8.4.6 Toleration and removal of old functions

IBM recommends that you perform a migration to VRSEL(NEW). Refer to the migration planning steps documented in the *DFSMSrmm Implementation and Customization Guide*. This prevents use of old VRS operands STARTNUMBER, LOCATION(BOTH), and STORENUMBER(xx,yy).

The toleration APAR OA13355 includes a ++HOLD(ACTION) and requires that VRSs are cleaned up before you are able to run EDGHSKP with VRSEL on a z/OS V1R8 system.

The cleanup actions are documented under message EDG2221E. EDG2221E sets return code 12 instead of 4.

Important: As long as you do not implement COUNT(0) or JOBNAME(ABEND\OPEN) you can run VRSEL processing on either a toleration system or z/OS V1R8.

Error messages

A description of the new message you get if the VRSEL processing detects some errors is:

EDG2222E `type VRS FOR mask SPECIFIES UNSUPPORTED OPTIONS - SOME RETENTION OPTIONS IGNORED.`

Where:

type	*type* Is the type of vital record specification. It can be one of: DSN - DSNAME type vital record specification VOL - VOLUME type vital record specification
mask	This is the vital record specification data set name or volume serial number.
Severity	Information.
Explanation	During vital record processing, DFSMSrmm found a vital record specification that contains unsupported options. The unsupported options are STARTNUMBER and LOCATION(BOTH).
Source	DFSMSrmm.
Detecting Module	EDGVREC0.
System Action	Processing ends. A return code of 12 is set.

Operators Response None.

Sysprogr Response You must replace the vital record specification with other vital record specifications that provide the retention options that you require. For example, if you use LOCATION(BOTH) you can replace it with use of the NEXTVRS operand. This example shows first an unsupported vital record specification and then the equivalent supported vital record specifications that you might use. Unsupported:

```
RMM ADDVRS DSNAME(data_set_name_mask) -
    CYCLES COUNT(5) LOCATION(BOTH) -
    STORENUMBER(2,1)
```

Supported:

```
RMM ADDVRS DSNAME(data_set_name_mask) -
    CYCLES COUNT(5) LOCATION(LOCAL) -
    STORENUMBER(2) NEXTVRS(DIST1C)
RMM ADDVRS NAME(DIST1C) LOCATION(DISTANT) -
    STORENUMBER(1)
```

If you use STARTNUMBER you can replace it with the use of the NEXTVRS operand. This example shows first an unsupported vital record specification and then the equivalent supported vital record specifications that you might use.

Unsupported:

```
RMM ADDVRS DSNAME(data_set_name_mask) -
    CYCLES COUNT(3) LOCATION(VAULT1) -
    STORENUMBER(2) STARTNUMBER(1)
```

Supported:

```
RMM ADDVRS DSNAME(data_set_name_mask) -
    CYCLES COUNT(3) LOCATION(HOME) -
    STORENUMBER(1) NEXTVRS(VLT12C)
RMM ADDVRS NAME(VLT12C) LOCATION(VAULT1) -
    STORENUMBER(2)
```

In addition, IBM suggests that you perform a migration to VRSEL(NEW). See the migration planning steps documented in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

Routing Codes 11.

Descriptor Codes 7.

8.4.7 Conversion to DFSMSrmm from other tape management systems

Before release V1R8, DFSMSrmm converted only DSN, LABEL, and JOB information to UXTABLE entries. Now it also supports MGMTCLAS and ABEND keywords, as shown in Figure 8-22.

```
{                  }{              {keyword       }}
{                  }{              {preferred date}}[,J=create.jobname ]
{D=dsname[-]       }{,LABEL=EXPDT={Julian date   }}[,JOB=create.jobname]
{DSN=dsname[-]     }{,LABEL=RETPD=nnnn             }[,JOBNAME=create.jobname]
 MGMTCLAS=smsclass {,LABEL=WRETPD=nnnn            }[,SELECT=ALL             ]
 M=smsclass         {,ABEND=RETPD=nnnn            }
                    {,ABEND=WRETPD=nnnn           }
                    {,ABEND=EXPDT={keyword      }}
                                  {preferred date}
                                  {Julian date   }
```

Figure 8-22 MGMCLAS and ABEND RDS keywords support

EDGCRFMT and EDGRSRDS conversion programs

If you have a CA-1 retention data set (RDS) entry, then this RDS entry is converted into two steps to create a UXTABLE entry that can be used in the EDGUX100 user exit:

1. EDGCRFMT creates input for EDGCSRDS.
2. EDGCSRDS builds the UXTABLE entries.

Example 8-8 shows you some CA-1 retention data set (RDS) entries.

Example 8-8 CA-1 retention data set (RDS) entries

```
M=SMSCLAS1,LABEL=RETPD=2010/011,JOB=TEST
MGMTCLAS=SMSCLAS2,LABEL=RETPD=2020/011
DSN=RXXXX.XXXX,ABEND=RETPD=2010/001,JOB=TESTTEST
DSN=RXXXX.XXXX,LABEL=RETPD=2010/001,JOB=TESTTEST
DSN=R3650.*,LABEL=RETPD=3650,JOB=COMBIN*
```

These CA-1 retention data set entries are converted to UXTABLE, as shown in Figure 8-23. For more information about a CA-1 to DFSMSrmm conversion refer to the IBM Redbooks publication *Converting to DFSMSrmm from CA-1,* SG24-6241. In this book the conversion is described in detail.

```
* start of RDS entries
         EDGCVRSG DSN=RXXXX.XXXX.G%%%%V%%,  X
               JOB=ABEND,                   X
               RO=001,                      X
               RETPD=2010
         EDGCVRSG DSN=RXXXX.XXXX,           X
               JOB=ABEND,                   X
               RO=001,                      X
               RETPD=2010
         EDGCVRSG DSN=RXXXX.XXXX.G%%%%V%%,  X
               JOB=TESTTEST,                X
               RO=001,                      X
               RETPD=2010
         EDGCVRSG DSN=RXXXX.XXXX,           X
               JOB=TESTTEST,                X
               RO=001,                      X
               RETPD=2010
         EDGCVRSG DSN=R3650.*,              X
               JOB=COMBIN*,                 X
               RO=NO,                       X
               RETPD=3650
         EDGCVRSG DSN=SMSCLAS1,             X
               JOB=TEST,                    X
               RO=011,                      X
               RETPD=2010
         EDGCVRSG DSN=SMSCLAS2,             X
               RO=011,                      X
               RETPD=2020
* default RP value
         EDGCVRSG DSN=*,                    X
               RO=NO,                       X
               RETPD=3
```

Figure 8-23 Sample UXTABLE entries

EDGCSVDS conversion program

EDGCSVDS converts the vaulting policies from the VPDD into K-Records representing the DFSMSrmm storage location movement policies.

Processing is changed to implement data set filter VRSs for all DSN entries found in the VPDD. Instead of storing data set and job names in memory until the vault statements are reached, and then looping through them to create K records, we can now create a K record for the filter as we read the entry from the VPDD and no longer have memory limitations in EDGCSVDS.

EDGCNVT conversion program

EDGCNVT is updated to allow COUNT(0) in the count filed of the EDGCKREC record. This program converts the output of the data extraction programs to DFSMSrmm format (CDS records) or ADDVRS commands, as shown in Figure 8-24.

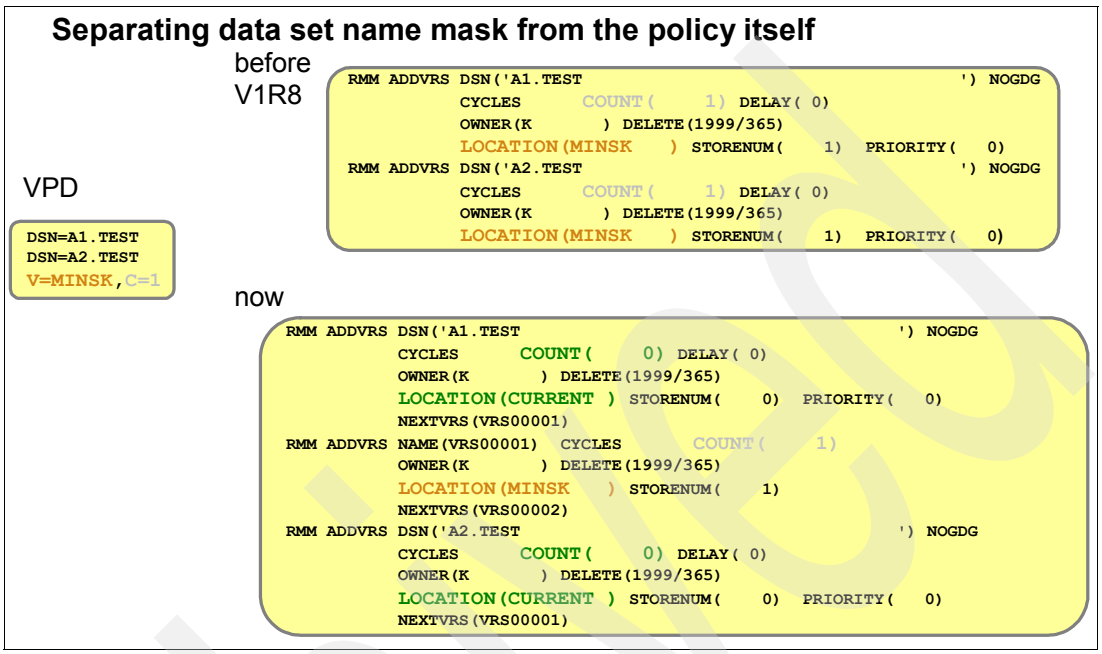


Figure 8-24 EDGCNVTRMM TSO ADDVRS commands

8.5 DFSMSrmm usability items

With this new release there are some new, easy-to-use ISPF dialog and RMM TSO subcommand enhancements. These simplify the tasks performed by the storage administrator and simplify the analysis of data set and volume retention.

8.5.1 Updates to RMM TSO SEARCHVOLUME subcommand

There are enhancements to the TSO RMM SEARCHVOLUME subcommand.

CLIST operand

When you specify the TYPE(LOGICAL) operand and CLIST, DFSMSrmm returns more information in the output file if the obtained logical volume resides on a stacked volume. In such a case, DFSMSrmm returns the first six characters of the container name, the logical volume serial number, and the status value. The status value can be:

SCRATCH If the volume is in scratch status or ready to return to scratch with the SCRATCHIMMEDIATE release option set

INITIALIZE If the volume is in scratch status and contains no valid data

Blank If status is not available

Operand JOBNAME or NOJOBNAME

Use this operand to search for volumes created by the specified job name. A job name is one to eight alphanumeric characters or $, #, or @, and must start SEARCHVOLUME

Subcommand with an alphabetic character, $, #, or @. You can also use a generic job name. Use % in your generic job name mask to match any one character and * to match any character string in the job name. If you do not specify JOBNAME, jobname is not used as a selection. If you specify JOBNAME(*), DFSMSrmm returns all volumes that match the search values specified and that have a job name. Volumes that do not have a job name are not listed.

JOBNAME is mutually exclusive with NOJOBNAME.

Operand LOCATION

```
LOCATION(SHELF | LOCAL | DISTANT | REMOTE | library_name | LOCDEF_location_name |
generic_location_name)
```

Specify to list volumes residing in a specific location. Specify one of the following:

- SHELF

  Volumes stored in shelf locations in a non-system-managed library.

- LOCAL, DISTANT, or REMOTE

  DFSMSrmm built-in storage locations. Use the LOCATION operand together with INTRANSIT to limit the list to only those volumes residing in or moving from the specific location. Use the LOCATION operand together with HOME to limit the list to only those volumes residing in a specific location that has the same home location.

- library_name

  Volumes stored in shelf locations in a specific system-managed library. A library name is one-to-eight alphanumeric characters starting with a non-numeric character. You cannot specify a distributed library name.

- LOCDEF_location_name

  Volumes moving to a storage location that was defined using the LOCDEF command. You can enter any value, as no checking is done against the current list of locations defined to DFSMSrmm.

- generic_location_name

  You can use a generic location name. You can use % in your generic mask to match any one character and * to match any character string.

Operand RELEASEACTION

```
RELEASEACTION(ALL, ERASE, INIT, NOTIFY, REPLACE, RETURN, SCRATCH)
```

Release actions are those that will be set as pending actions when a volume is released. Use this operand to search for volumes that have the specified release action set. Volumes are returned if any of the values you specify are set in the volume. Also see the ACTION operand for how to search for volumes with pending actions.

Specify a value of ALL, or one or more of the following, separated by commas:

ALL	To list all volumes with any pending action
ERASE	To list only volumes that require erasing
INIT	To list only volumes that require initialization
RETURN	To list only volumes that should be returned to their owner
REPLACE	To list only volumes that must be replaced by new volumes and returned to the scratch pool

NOTIFY	To list only volumes for which owners must be notified
SCRATCH	To list all volumes to be returned to scratch status

8.5.2 ISPF lists show retention information

The ISPF dialog has been updated to shown more information about the search data set result list and search volume result list panels.

Volume search result list

In the volume search result list (Figure 8-25) there are now two useful enhancements available:

- The Set Retained column shows whether the volume is retained because the RETAINBY(SET) option is in use and other volumes in the set are either VRS retained or retained by volume expiration date.
- The Status column has been corrected to display volume availability instead of volume status if volume availability is not blank.

```
  Panel  Help  Scroll
 ------------------------------------------------------------------------------
                  DFSMSrmm Volumes (Page 1 of 2)           Row 1 to 11 of 11
Command ===>                                               Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
   Volume          Assigned   Expiration S                Dest-    Tr- Data
S  serial Owner    date       date       R Status  Location ination  ans sets
-- ------ -------- ---------- ---------- - ------- -------- -------- --- -----
   TST008 STC      2007/058   2007/058     VRS     LIB1              N   1
   TST009 HAIMO    2004/296   1999/365   Y MASTER  LIB1              N   1
   TST010 STC      2007/060   1999/365   Y MASTER  LIB1              N   1
   TST011 STC      2007/059   2007/061     VRS     LIB1              N   1
   TST014          2007/060                ENTRY   SHELF    LIB1     Y   0
   TST020          2007/072                SCRATCH LIB1              N   2
   TST021 STC      2007/058   2007/060     VRS     LIB1              N   1
   TST023 STC      2007/059   1999/365     LOAN    LIB1              N   1
   TST026 PAOLOR3  2003/302   2003/365     RELEASE LIB1              N   1
   TST029 MHLRES2  2007/059   2007/071     OPEN    LIB1              N   2
   TS4284 SIEGEL   2007/073   2007/073     MASTER  SHELF             N   99
****************************** Bottom of data ********************************
```

Figure 8-25 DFSMSrmm search volume result panel 1 of 2

Where:

- The Set Retained (SR) column shows whether the volume is:

Y	Yes for set retained
(blank)	Not set retained

- The Volume Status can be one of:

  Availability

LOAN	The volume is on loan.
RELEASE	The volume is pending release.
OPEN	A file on the volume is opened for output.

VRS	The volume is being retained by a vital record specification.
Status	MASTER, SCRATCH, USER, INIT, or ENTRY
MASTER	The volume contains valid user data and cannot be overwritten unless the data set names match.
USER	The volume is assigned to a user and is in use. It contains any type of data and can be overwritten, rewritten, and used repeatedly until the volume's expiration date.
INIT	The volume is awaiting initialization before becoming available for use as a scratch volume.
ENTRY	The volume has been predefined to DFSMSrmm prior to entry into an ATLDS, for use as a scratch volume.
SCRATCH	The volume is free and available for use. It contains no data or the data on the volume has expired or is not valid.

In the right section of the volume search result panel there are no changes, as shown in Figure 8-26

```
  Panel  Help  Scroll
_____________________________________________________________________________
                   DFSMSrmm Volumes (Page 2 of 2)          Row 1 to 11 of 11
Command ===>                                                Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands
Use the LEFT command to view other data columns
   Volume Rack   Media    Home     Media          Recorded Compac-  Attri-
S  serial number name     location type     Label format   tion     butes
-- ------ ------ -------- -------- -------- ----- -------- -------- --------
   TST008 TST008 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST009 TST009 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST010 TST010 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST011 TST011 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST014 TST014 MEDIA3   SHELF    HPCT     SL    128TRACK *        NONE
   TST020 TST021 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST021 TST021 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST023 TST023 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST026 TST026 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST029 TST028 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
   TST284 TST029 MEDIA3   LIB1     HPCT     SL    128TRACK YES      NONE
****************************** Bottom of data ********************************
```

Figure 8-26 DFSMSrmm search volume result panel 2 of 2

Data set search result list

In the right section of the data set search result list shown in Figure 8-27 you now can directly see whether a data set is vital record selected. You must press the right button to show the right panel information of the of the data set search result. The VRS Retained column displays whether data sets are retained by DFSMSrmm VRS retention. It shows whether the data set is:

Y	Yes if data set VRS retained
(blank)	If data set is not VRS retained

```
  Panel  Help  Scroll
_____________________________________________________________________________
EDGPD020        DFSMSrmm Data Sets (Page 1 of 2)            Row 1 to 16 of 16
Command ===>                                                 Scroll ===> PAGE


Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                  Volume          File
S  Data set name                                  serial Owner    seq
-- ---------------------------------------------- ------ -------- -----
   HSM.BACKTAPE.DATASET                           TST010 STC      1
   HSM.BACKTAPE.DATASET                           TST027 STC      1
   HSM.COPY.HMIGTAPE.DATASET                      TST002 STC      1
   HSM.COPY.HMIGTAPE.DATASET                      TST004 STC      1
   HSM.COPY.HMIGTAPE.DATASET                      TST016 STC      1
   HSM.DMP.CRYPCOPY.VMHLOA0.D07072.T141318        TST019 STC      1
   HSM.DMP.CRYPCOPY.VMHLOA1.D07072.T141318        TST019 STC      2
   HSM.DMP.MHLRES.VMHLOA0.D07059.T425215          TST015 STC      1
   HSM.DMP.MHLRES.VMHLOA1.D07059.T425215          TST023 STC      1
   HSM.DMP.MHLRES.VMLDOOB.D06110.T385618          TST013 STC      1
   HSM.DMP.PLUSCOPY.VMHLOA0.D07058.T060317        TST024 STC      1
   HSM.DMP.PLUSCOPY.VMHLOA1.D07058.T060317        TST024 STC      2
   HSM.HMIGTAPE.DATASET                           TST001 STC      1
   HSM.HMIGTAPE.DATASET                           TST012 STC      1
```

Figure 8-27 DFSMSrmm data set display 1 of 2

Press the right button to show the right panel of the search data set result, as shown in Figure 8-28, and to see whether a data set is VRS retained.

```
  Panel  Help  Scroll
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPD030        DFSMSrmm Data Sets (Page 2 of 2)            Row 1 to 24 of 24
Command ===>                                                 Scroll ===> PAGE


Enter HELP or PF1 for the list of available line commands
Use the LEFT command to view other data columns
                                                 Create     Expiration V
S  Data set name                                 date       date       R
-- ---------------------------------------------- ---------- ---------- -
   HSM.BACKTAPE.DATASET                          2007/060   1999/365   Y
   HSM.BACKTAPE.DATASET                          2007/053   1999/365   Y
   HSM.COPY.HMIGTAPE.DATASET                     2007/054   1999/365
   HSM.COPY.HMIGTAPE.DATASET                     2007/060   1999/365
   HSM.COPY.HMIGTAPE.DATASET                     2007/060   1999/365
   HSM.DMP.CRYPCOPY.VMHLOA0.D07072.T141318       2007/072   1999/365
   HSM.DMP.CRYPCOPY.VMHLOA1.D07072.T141318       2007/072   1999/365
   HSM.DMP.MHLRES.VMHLOA0.D07059.T425215         2007/059   1999/365
   HSM.DMP.MHLRES.VMHLOA1.D07059.T425215         2007/059   1999/365
   HSM.DMP.MHLRES.VMLDOOB.D06110.T385618         2006/110   1999/365
   HSM.DMP.PLUSCOPY.VMHLOA0.D07058.T060317       2007/059   1999/365
   HSM.DMP.PLUSCOPY.VMHLOA1.D07058.T060317       2007/059   1999/365
   HSM.HMIGTAPE.DATASET                          2007/054   1999/365
   HSM.HMIGTAPE.DATASET                          2006/087   1999/365
```

Figure 8-28 DFSMSrmm data set display 2 of 2

8.5.3 SELECT primary command in RMM dialog search results

Within the DFSMSrmm ISPF dialog, a new SELECT primary command, as well as changes to the search results lists for data sets, volumes, and vital record specifications, were added.

The SELECT command is supported on all results lists. You can use this primary command to apply the same line command to all selected entries in a search results list table. For example, all volumes in the volume result list are ejected when you issue the command shown in Example 8-9 in a volume search results list.

Example 8-9 Using an asterisk to select all

```
Command ===> SELECT * E
```

You can also specify SEL or S, instead of SELECT. When you do not specify a line command, S is the default.

You can specify a generic value, as shown in Example 8-10, for the first parameter, or a specific value. For example, issue the command shown in Example 8-10 to view volume data for all volumes starting with AB0, or issue a specific volser as shown in Example 8-11 on page 292to release the volume MW0001, if found in the list.

Example 8-10 Using a generic volume mask

```
Command ===> SELECT AB0* V
```

Example 8-11 Specifying a full volume serial number

```
Command ===> SELECT MW0001 R
```

The first parameter is applied on the first key of the table. If you want another column of the table to be taken as the select criteria, issue a SORT command before the select command. For example, first SORT by owner, then issue the command shown in Figure 8-12 on page 274 to eject all volumes of owner SCHLUM.

Example 8-12 Select all information assigned to owner SCHLUM

```
Command ===> SELECT SCHLUM E
```

For example, we use the *Search for data sets* dialog to search for data sets starting with high-level qualifier MHLRES, as shown in Figure 8-29.

```
  Panel  Help
                          DFSMSrmm Data Set Search
Command ===>

Enter fully qualified or partial data set name and job name:

Data set name  . . . . 'mhlres*.**'
Job name . . . . . .          Specific or generic name

Enter optional parameters to qualify search

Owner  . . . . . *            Owner of volumes (Default is your userid)
Volume serial  . . . .      OR Volume serial
List entire set  . . .        YES, For all data sets in the
                              multi-volume set, otherwise NO
Status . . . . . .            PRIVATE, SCRATCH, or blank for all
Retained by VRS  . . .        YES, NO, or blank for all
Create date  . . . . .        Created since YYYY/DDD
Physical file seq  . .        Relative position on the volume
Limit  . . . . . *            Limit search to first nnnn data sets
Clist  . . . . . .            YES to create a data set, or NO, or blank
Program name . . . . .        Specific or generic name

Enter HELP or PF1 for the list of available line commands
```

Figure 8-29 DFSMSrmm search data set dialog

We get a lot of data set information and use the SORT command to sort the data set list by OWNER in descending order to show all data sets have an owner first. Figure 8-30 on page 293 shows you the result and the command that we use, where:

SORT This is the command.

D This is the direction in which it can be A for ascending or D for descending:

ASCENDING (A)
DESCENDING (D)

O This is the identifier in a shortened form of one of the data column names in the data sets list. It can be one of the following:

CREATE (C) - Create date
OWNER (O) - Owner

DATASET (D) - Data set name
VOLUME (VO) - Volume serial
FILESEQ (F) - Physical file sequence number
EXPIRY (E) - Expiration date
VR (VR) - VRS retained

```
  Panel  Help  Scroll
_____________________________________________________________________________
                DFSMSrmm Data Sets (Page 1 of 2)   Row 2,079 to 2,110 of 2,110
Command ===>  SORT D O                                        Scroll ===> CSR

Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                  Volume          File
S  Data set name                                  serial Owner    seq
-- ---------------------------------------------- ------ -------- -----
   MHLRES1.TEST.DATA                              DV2083          1
   MHLRES1.TEST.DATA                              DV2084          1
   MHLRES1.TEST.DATA                              DV2085          1
   MHLRES1.TEST.DATA                              DV2086          1
   MHLRES1.TEST.DATA                              DV2087          1
   MHLRES1.TEST.DATA                              DV2088          1
   MHLRES1.TEST.DATA                              DV2089          1
   MHLRES1.TEST.DATA                              DV2090          1
   MHLRES1.TEST.DATA                              DV2091          1
   MHLRES1.TEST.DATA                              DV2092          1
   MHLRES1.TEST.DATA                              DV2093          1
   MHLRES1.TEST.DATA                              DV2094          1
   MHLRES1.TEST.DATA                              DV2095          1
   MHLRES1.TEST.DATA                              DV2096          1
   MHLRES1.TEST.DATA                              DV2097          1
   MHLRES1.TEST.DATA                              DV2098          1
   MHLRES1.TEST.DATA                              DV2099          1
   MHLRES4.ABARS.OUTPUT.C.C01V0001                TST022 STC      4
   MHLRES4.ABARS.OUTPUT.D.C01V0001                TST022 STC      1
   MHLRES4.ABARS.OUTPUT.I.C01V0001                TST022 STC      3
   MHLRES4.ABARS.OUTPUT.O.C01V0001                TST022 STC      2
   MHLRES5.TEST.FILE1.D130                        MARY01          1
   MHLRES5.TEST.FILE1.D130                        MARY02          1
   MHLRES5.TEST.FILE1.D130                        MARY03          1
   MHLRES5.TEST.FILE1.D130                        MARY04          1
   MHLRES5.TEST.FILE2.D130                        MARY01          2
   MHLRES5.TEST.FILE2.D130                        MARY02          2
   MHLRES5.TEST.FILE2.D130                        MARY03          2
   MHLRES5.TEST.FILE2.D130                        MARY04          2
   MHLRES5.TEST.TAPE                              TST007          1
   MHLRES5.TEST.TAPE                              TST029 MHLRES2  2
   MHLRES6.TEST.TAPE                              TST029 MHLRES2  1
```

Figure 8-30 DFSMSrmm search data set result

Figure 8-31 shows you the results after the panel displays the information in descending order sorted by OWNER and the use of the SELECT command to select all data sets owned by user STC to changing anything.

```
  Panel  Help  Scroll
               DFSMSrmm Data Sets (Page 1 of 2)              4 rows updated
Command ===> SELECT STC C                                  Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                 Volume          File
S  Data set name                                 serial Owner    seq
-- ---------------------------------------------- ------ -------- -----
C  MHLRES4.ABARS.OUTPUT.C.C01V0001                TST022 STC      4
C  MHLRES4.ABARS.OUTPUT.D.C01V0001                TST022 STC      1
C  MHLRES4.ABARS.OUTPUT.I.C01V0001                TST022 STC      3
C  MHLRES4.ABARS.OUTPUT.O.C01V0001                TST022 STC      2
   MHLRES6.TEST.TAPE                             TST029 MHLRES2  1
   MHLRES5.TEST.TAPE                             TST029 MHLRES2  2
   MHLRES1.TEST.DATA                             DV1897          1
   MHLRES1.TEST.DATA                             DV1896          1
   MHLRES1.TEST.DATA                             DV1901          1
   MHLRES1.TEST.DATA                             DV1900          1
   MHLRES1.TEST.DATA                             DV1899          1
   MHLRES1.TEST.DATA                             DV1898          1
   MHLRES1.TEST.DATA                             DV1906          1
   MHLRES1.TEST.DATA                             DV1905          1
   MHLRES1.TEST.DATA                             DV1904          1
   MHLRES1.TEST.DATA                             DV1903          1
   MHLRES1.TEST.DATA                             DV1902          1
   MHLRES1.TEST.DATA                             DV1910          1
   MHLRES1.TEST.DATA                             DV1909          1
   MHLRES1.TEST.DATA                             DV1908          1
   MHLRES1.TEST.DATA                             DV1907          1
   MHLRES1.TEST.DATA                             DV2002          1
   MHLRES1.TEST.DATA                             DV2001          1
   MHLRES1.TEST.DATA                             DV1999          1
   MHLRES1.TEST.DATA                             DV1998          1
   MHLRES1.TEST.DATA                             DV1997          1
   MHLRES1.TEST.DATA                             DV1996          1
   MHLRES1.TEST.DATA                             DV1995          1
   MHLRES1.TEST.DATA                             DV1994          1
   MHLRES1.TEST.DATA                             DV1993          1
   MHLRES1.TEST.DATA                             DV1992          1
   MHLRES1.TEST.DATA                             DV1991          1
```

Figure 8-31 Result after using the SELECT command

Where:

SELECT	DFSMSrmm-supported primary command
STC	Specific owner should be used to set the line command
C	Line command to change the data set information

8.5.4 RMM TSO CHANGEVRS subcommand

There is an RMM TSO CHANGEVRS subcommand available, like all other RMM CHANGE commands, to update previously created VRS definitions defined to DFSMSrmm. The syntax is similar to other change RMM TSO subcommands.

Note: To use the RMM TSO CHANGEVRS subcommand, you need CONTROL access to the STGADMIN.EDG.VRS resource profile to change vital record specifications.

Syntax

Figure 8-32 shows you the syntax of the RMM TSO CHANGEVRS subcommand.

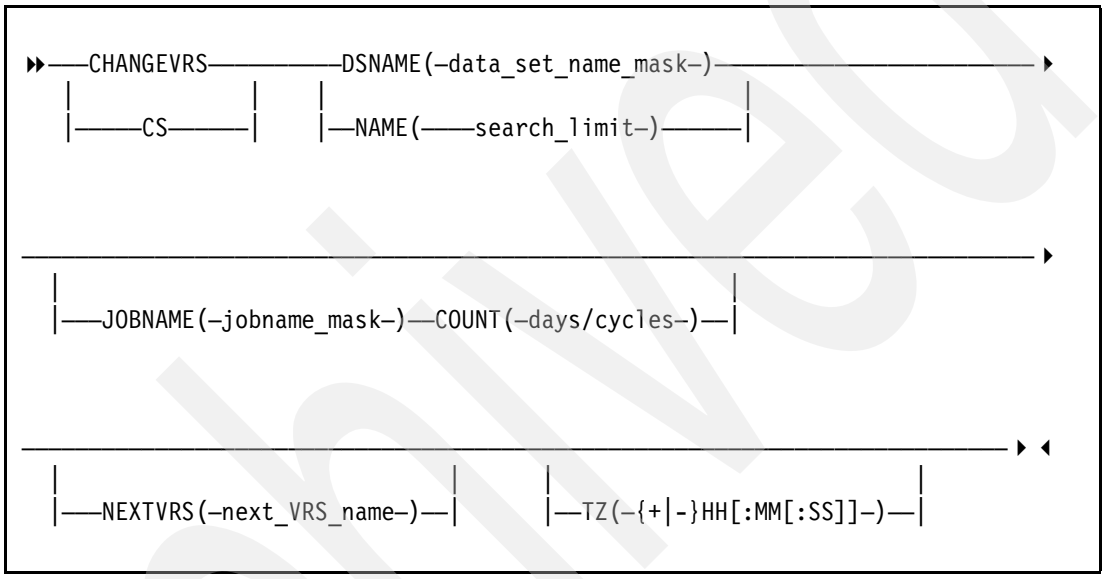


Figure 8-32 CHANGEVRS syntax diagram

Operands

The operands are:

- DSNAME(data_set_name_mask)

  Identifies the type of vital record specification and specifies the mask of a data set name, management class, or management value of an existing vital record specification. The mask can have a fully qualified or a generic name. It can also be one of the reserved words ABEND or OPEN. The data set name mask is 1 to 44 characters, enclosed in quotation marks if any special characters are included. If the data set name mask is not enclosed in quotation marks, PROFILE PREFIX is applied. This operand is required and must immediately follow the CHANGEVRS subcommand. DSNAME is mutually exclusive with the NAME and VOLUME operands.

  Note: DFSMSrmm no longer folds data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

- NAME(VRS_name)

  Identifies the vital record specification type and specifies a name for the vital record specification. A vital record specification name is eight alphanumeric characters chosen by your installation.

- JOBNAME(jobname_mask)

  Identifies the job name for the vital record specification. A job name is one-to-eight alphanumeric characters, $, #, or @. You can specify a specific job name or a job name mask. It can also be one of the reserved words ABEND or OPEN.

  This operand is optional. You must specify the operand, though, if the vital record specification that you want to change has the JOBNAME operand specified.

  If you have data sets with job names that include symbols other than alphanumeric characters, $, #, or @, use a job name mask to cover them.

- COUNT(days/cycles)

  Specifies a retention amount, based on the retention type of the existing vital record specification. Specify COUNT(number_of_days) to request that DFSMSrmm retains all cycles or copies of a data set. Specify COUNT(number_of_cycles) to request that DFSMSrmm retains the number of data set cycles that you specify.

  The value range for data set name vital record specification and retention name vital record specification is 0 to 99999. A value of 99999 indicates that DFSMSrmm retains all cycles of a data set.

  If count() is not specified, the count value in the existing vital record specification record is not changed.

  DFSMSrmm validates the COUNT value as follows:

  - If EXTRADAYS is specified, COUNT must equal STORENUMBER: (STORENUMBER) = (COUNT).
  - Regardless of whether NEXTVRS and ANDVRS are used, COUNT can be (STORENUMBER) <= (COUNT).
  - If DAYS or LASTREFERENCEDAYS retention is used, (STORENUMBER) + (DELAY) <= (COUNT).

- NEXTVRS(next_VRS_name)

  Specifies the name of the next vital record specification in a chain of vital record specifications. If you specify the name of a vital record specification that does not exist, DFSMSrmm uses a dummy vital record specification with the name *broken* instead. This keeps the volume or data set in its current location.

- TZ({+|-}HH[:MM[:SS]])

  Specifies the time zone offset when date and time values are specified. The format is {+|-}HH[:MM[:SS]], where:

  - +|- is the offset direction. Specify + to indicate that the offset is east of the zero median (UT). Specify - to indicate that the offset is west of the zero median (UT). The offset direction is required.
  - HH is hours.
  - MM is minutes.
  - SS is seconds.

  An optional colon (:) separates hours from optional minutes and optional seconds. You can specify a time in the range of 00:00:00–15:00:00 for HH:MM:SS. The MM and SS value range is 00–59.

8.5.5 RMM TSO SEARCHOWNER subcommand

Like other RMM TSO SEARCH subcommands, the SEARCHOWNER command is used to create a list of resources defined to DFSMSrmm.

Syntax is similar to other RMM TSO subcommands, but no dialog support is added yet.

Note: To use the RMM TSO SEARCHOWNER subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the SEARCHOWNER subcommand to create a list of owners defined to DFSMSrmm. You can restrict how many owners DFSMSrmm displays by specifying the LIMIT or END operand. DFSMSrmm searches until it reaches your limit or endpoint, or until it lists all owners that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten.

The information DFSMSrmm returns for each owner in the list is:

- Owner's ID
- Owner's last name
- Owner's first names
- Owner's internal telephone number
- Number of volumes owned

Syntax

Figure 8-33 shows you the syntax of the RMM TSO SEARCHOWNER subcommand.

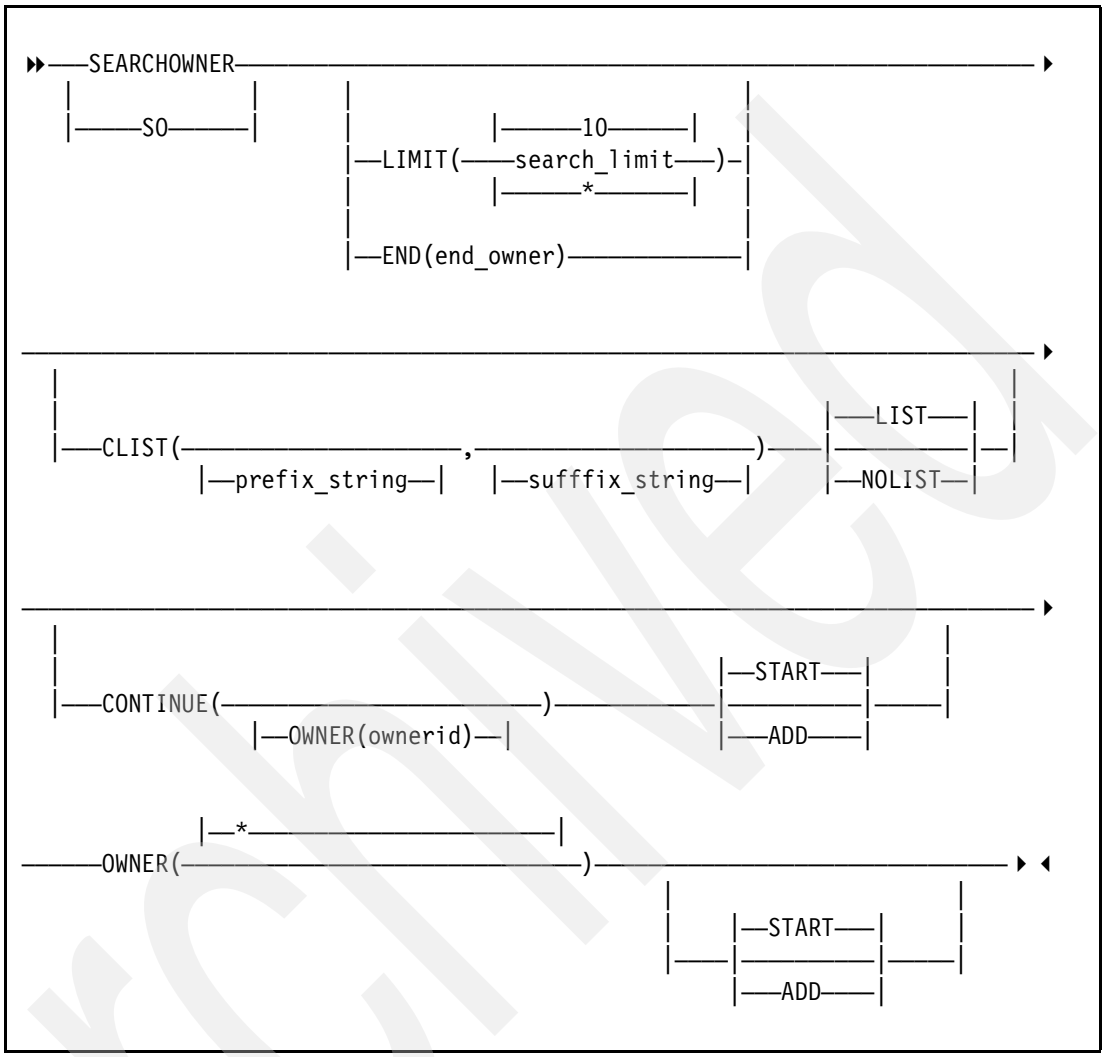


Figure 8-33 SEARCHOWNER syntax diagram

Operands

The operands are:

- ADD

  Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

  ADD is mutually exclusive with START.

  You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

- CLIST(prefix_string,suffix_string)

  Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any owners that you do not want in the list. Then you can run the CLIST at your convenience.

  DFSMSrmm returns the owner serial number for each record if you do not specify (prefix_string and suffix_string). When the owner serial number contains special characters the value is returned within quotation marks.

  You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (prefix_string and suffix_string). These text strings cannot exceed 255 characters. Separate the prefix_string and suffix_string using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns. To enter a null prefix_string, add a pair of separator characters such as ” to the text string (for example, CLIST(”,’ suffix_string’)).

- CONTINUE(OWNER(ownerid))

  Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information may be either a line mode message or an SFI or XML attribute.

  CONTINUE is an optional operand. Use the LIMIT operand to control the maximum number of entries to be returned each time that you start or continue the search.

  To continue a previous search subcommand, the CONTINUE operand value includes the following value to identify the current search position: OWNER(ownerid). ownerid is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

  The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and must be passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

- END(end_owner)

  Specify END as an alternative to the LIMIT operand to enable you to specify both the starting and ending point of the owner search. END is mutually exclusive with LIMIT.

- LIMIT(search_limit | *)

  Specifies the number of entries that DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to request a list of all entries matching your search criteria. LIMIT is mutually exclusive with END. The default value is 10.

- LIST

  Specifies that DFSMSrmm produce a list when the CLIST operand is used. LIST is mutually exclusive with the NOLIST operand. LIST is the default.

- NOLIST

  Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file. NOLIST is mutually exclusive with the LIST operand. LIST is the default.

- OWNER(full_or_generic_owner | *)

  Specifies an owner ID. DFSMSrmm only lists volumes belonging to the owner ID that you specify. Specify a specific owner ID to list volumes belonging to that owner. Specify an asterisk to list all volumes that match the other search criteria regardless of their owner. An owner ID is one-to-eight alphanumeric characters or to six alphanumeric characters, $, #, or @. The first character must not be a number. The default is your TSO user ID.

- START

  Specify this operand to request that records written to the CLIST data set start from the beginning of the data set. START is mutually exclusive with ADD. START is the default value.

Examples

Use the RMM TSO SEARCHOWNER subcommand, as shown in Example 8-13, to create a list of all owners defined to DFSMSrmm.

Example 8-13 List all owners defined to DFSMSrmm

```
   TSO RMM SO OWNER(*) LIMIT(*)
or
   TSO RMM SEARCHOWNER OWNER(*) LIMIT(*)
```

DFSMSrmm displays a list, such as the one shown in Figure 8-34.

```
   Owner    Last Name           First Names         Internal Volumes
   -------- ------------------- ------------------- -------- ----------
   HAIMO    Haimowitz           Bob                 349-5456 1
   HSM      STC DFSMShsm                                     0
   MARY     Lovlace             Mary                475-3231 0
   MHLRES1  Fletcher            Anthony                      0
   MHLRES2  Weisshaar           Gerhard                      1
   MHLRES2D                                                  0
   MHLRES2E                                                  0
   MHLRES3  Perkin              Daniel                       0
   MHLRES4  Coelho              Andre                        1
   MHLRES5  Schlumberger        Norbert             119-3579 1286
   PAOLOR2                                                   0
   PAOLOR3                                                   1
   STC                                                       24
   VAINI                                                     0
   14         ENTRIES LISTED
```

Figure 8-34 DFSMSrmm search owner result

8.5.6 Rexx variable constraint relief

A new operand, VARSTORAGE, was added to the PROFILE command to specify whether variables in the CLIST or authorized REXX variable pools can use storage above the 16 MB line.

There are no changes to RMM for this support.

If you have a large environment and you have a need to list a big number of volumes, normally you get a return code 4 reason code 10 `Insufficient storage for search processing. More records might exist` is avoided.

If your TSO LOGON session parameter size is 30,000, then approximately 2,400 volumes are retrieved with the setting VARSTORAGE(LOW). If you change your settings to VARSTORAGE(HIGH) you can retrieve approximately 20,000 volumes.

VARSTORAGE specifies the storage location to be used for CLIST variables or REXX OUTTRAP variables containing output from authorized commands. A CLIST or REXX exec uses the VARSTORAGE setting of the PROFILE command when the it starts. This setting then remains unchanged for the life of the CLIST or REXX exec, even if the CLIST or REXX exec issues a new PROFILE command with a different VARSTORAGE setting. The new setting only applies when a new CLIST or REXX exec begins.

VARSTORAGE (HIGH) Indicates that CLIST variables and REXX OUTTRAP variables containing output from authorized commands invoked by REXX can be kept in storage above the 16 M line.

VARSTORAGE (LOW) Indicates that CLIST variables and REXX OUTTRAP variables containing output from authorized commands invoked by REXX can only be kept in storage below the 16 M line. If you specify VARSTORAGE with no operands, VARSTORAGE(LOW) is the default. This is the default value when your user profile is created.

To show your current setting use the TSO PROFILE command, as shown in Figure 8-35, without any additional operands.

```
TSO PROF
```

Figure 8-35 Display your current profile settings

Figure 8-36 shows that your current profile settings include the new VARSTORAGE information. If you have not set the VARSTORAGE, you can see that the default setting LOW is set.

```
CHAR(0)  LINE(0)    PROMPT   INTERCOM   NOPAUSE NOMSGID MODE    WTPMSG
NORECOER PREFIX(MHLRES5) PLANGUAGE(ENU) SLANGUAGE(ENU) VARSTORAGE(LOW)
DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
```

Figure 8-36 Your current profile settings

If you have requested to search for all volumes in your DFSMSrmm database, normally you see the message `more volumes exists` on the panel, as shown in Figure 8-37, and the search ended before all volumes were listed.

```
      Panel  Help  Scroll

                    DFSMSrmm Volumes (Page 1 of 2)       More volumes may exist
Command ===>                                                  Scroll ===> CSR
There is not enough storage available to list all the volumes
Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
   Volume          Assigned   Expiration                  Dest-    Tra- Data
S  serial Owner    date       date       Status  Location ination  nsit sets
-- ------ -------- ---------- ---------- ------- -------- -------- ---- -----
   AFS201 VGRMMLIB 2004/303   1999/365   USER    SHELF             N    0
   AFS202 VGRMMLIB 2004/303   1999/365   USER    SHELF             N    0
   BTS001 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS005 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS006 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS007 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS008 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS010 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS011 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS012 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
```

Figure 8-37 Display your incomplete search result

To update this value use the TSO PROFILE command, as shown in Figure 8-38.

```
TSO PROFILE VARSTORAGE(HIGH)
```

Figure 8-38 Update your TSO profile settings

Figure 8-39 shows you your new profile settings include the new VARSTORAGE information.

```
  CHAR(0)  LINE(0)    PROMPT   INTERCOM   NOPAUSE NOMSGID MODE    WTPMSG
  NORECOER PREFIX(MHLRES5) PLANGUAGE(ENU) SLANGUAGE(ENU) VARSTORAGE(HIGH)
  DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
```

Figure 8-39 Your new profile settings

After the update of your profile settings you should no longer get the `more volumes exists` message, except you have a really large environment. Figure 8-40 shows you the search results with 19.713 entries in the list.

```
  Panel  Help  Scroll
 ------------------------------------------------------------------------------
                    DFSMSrmm Volumes (Page 1 of 2)        Row 1 to 33 of 19,713
Command ===>                                                 Scroll ===> CSR

Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
   Volume          Assigned   Expiration                  Dest-    Tra- Data
S  serial Owner    date       date       Status  Location ination  nsit sets
-- ------ -------- ---------- ---------- ------- -------- -------- ---- -----
   AFS201 VGRMMLIB 2004/303   1999/365   USER    SHELF             N    0
   AFS202 VGRMMLIB 2004/303   1999/365   USER    SHELF             N    0
   BTS001 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS005 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS006 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS007 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS008 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS010 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS011 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
   BTS012 VGRMMLIB 2005/005   1999/365   USER    SHELF             N    0
```

Figure 8-40 Display your search result with up to 20.000 entries

8.6 Enabling ISPF Data Set List (DSLIST) support

To enable direct entry into the DFSMSrmm ISPF dialog from the ISPF Data Set List Utility, use the ISPF Configuration Utility to update the ISPF Configuration Table. To enable this function, select the Enable RM/Tape Commands option. For details of how to use the ISPF Configuration Utility, refer to *z/OS ISPF Planning and Customizing,* GC34-4814. Figure 8-41 on page 304 shows that the data set list support is enabled, and also shows the default values for the RM/Tape Command EDGRPD34 and Command APPLID EDG. You do not need to change these values.

8.6.1 Implementation steps

The ISPF Data Set List support can be implemented for a single user or it can be a installation-wide setting. To implement ISPF Data Set List support for a single user:

1. Create a new PDS with VB 255 to store the ISPF default settings.
2. Under ISPF, call 'TSO ISPCCONF' in any command line.
3. Select option **1, Create/Modify Settings and Regenerate Keyword File**, in the ISPF Configuration Utility primary panel, and added the name to the keyword file dsname field that you have allocated before, and added a member name.

   You should get the message `Keyword file loaded.`
4. Select option **3, PDF Exits and Other PDF Settings" in the ISPF Configuration Utility primary panel**.
5. Scroll to the correct place and add a slash (/) to Enable RM/Tape Commands.

6. Exit the dialog by pressing the PF3 key tow times. You should get the message `Keyword file saved.`
7. Select option **4, Build Configuration Table Load Module**.
8. Specify any library in your ISPLLIB concationation as the *Output Configuration Table Load Module Data Set*.
9. Enter to load the module built.
10. Exit ISPF. Ensure that the specified load library is ISPLLIB concatenated.
11. Start ISPF and it works (just for a single user, not for all users).

To implementing the ISPF Data Set List support for all users, follow the steps as described before, but the configuration table load module should be stored in the SISPLPA library. To load the module to the LPA you must IPL your system. For detailed information refer to 8.6.4, “Move the ISPCFIGU module to the SISPLPA library (optional)” on page 313.

8.6.2 Use the ISPF Configuration Utility

Use the ISPF Configuration table to change site-wide defaults and to indicate that installation exit routines are provided for some of the ISPF functions. The ISPF functions that allow installation-written exit routines are data set allocation, print utility, data set compression, data set list utility, member list filter, and data set name change. ISPF checks the configuration table to determine, first, whether exit routines are provided, and second, whether those routines are programs or CLISTs. If you specify both a CLIST and a program, ISPF uses the program.

Enter the command TSO ISPCCONF to start the ISPF Configuration Utility, as shown in Figure 8-41. You can chose the command in the ISPF command shell or in any available command line.

```
  Menu  Utilities  Compilers  Options  Status  Help
──────────────────────────────────────────────────────────────────────────────
                          z/OS Primary Option Menu
Option ===> TSO ISPCCONF

0  Settings      Terminal and user parameters           User ID . : SCHLUM
1  View          Display source data or listings        Time. . . : 18:40
2  Edit          Create or change source data           Terminal. : 3278
3  Utilities     Perform utility functions              Screen. . : 1
4  Foreground    Interactive language processing        Language. : ENGLISH
5  Batch         Submit job for language processing     Appl ID . : PDF
6  Command       Enter TSO or Workstation commands      TSO logon : IKJACCT
7  Dialog Test   Perform dialog testing                 TSO prefix: MHLRES5
9  IBM Products  IBM program development products       System ID : SC70
10 SCLM          SW Configuration Library Manager       MVS acct. : ACCNT#
11 Workplace     ISPF Object/Action Workplace           Release . : ISPF 5.9
12 z/OS System   z/OS system programmer applications
13 z/OS User     z/OS user applications

    Enter X to Terminate using log/list defaults
```

Figure 8-41 Start the ISPF conversion utility

Refer to Figure 8-42 to see how you can select the Create/Modify Settings and Regenerate Keyword File after the ISPF conversion utility is up and running.

```
                        ISPF Configuration Utility                Enter option
______________________________________________________________________________
Option ===> 1

1  Create/Modify Settings and Regenerate Keyword File
2  Edit Keyword File Configuration Table
3  Verify Keyword Table Contents
4  Build Configuration Table Load Module
5  Convert Assembler Configuration Table to Keyword File
6  Build SMP/E USERMOD


Keyword File Data Set
  Data Set . . . 'RMM.ADDONS.CEXECVB'
  Member . . . . SETTING

Configuration Table Assembler Source Data Set
  Data Set . . .
  Member . . . .

  Output File Content for Keyword File
  2  1. Include only non-default values
     2. Include defaults as comments
     3. Include all values


Current Configuration Table
  Keyword File : not available
  Identifier . : ISPCFIGU             Level   . . . : 480R8001
  Compile Date : 2003/04/04           Compile Time :
```

Figure 8-42   Create/Modify Settings and Regenerate Keyword File selection

After the keyword file is loaded select the **PDF Exits and Other PDF Settings** menu, as shown in Figure 8-43.

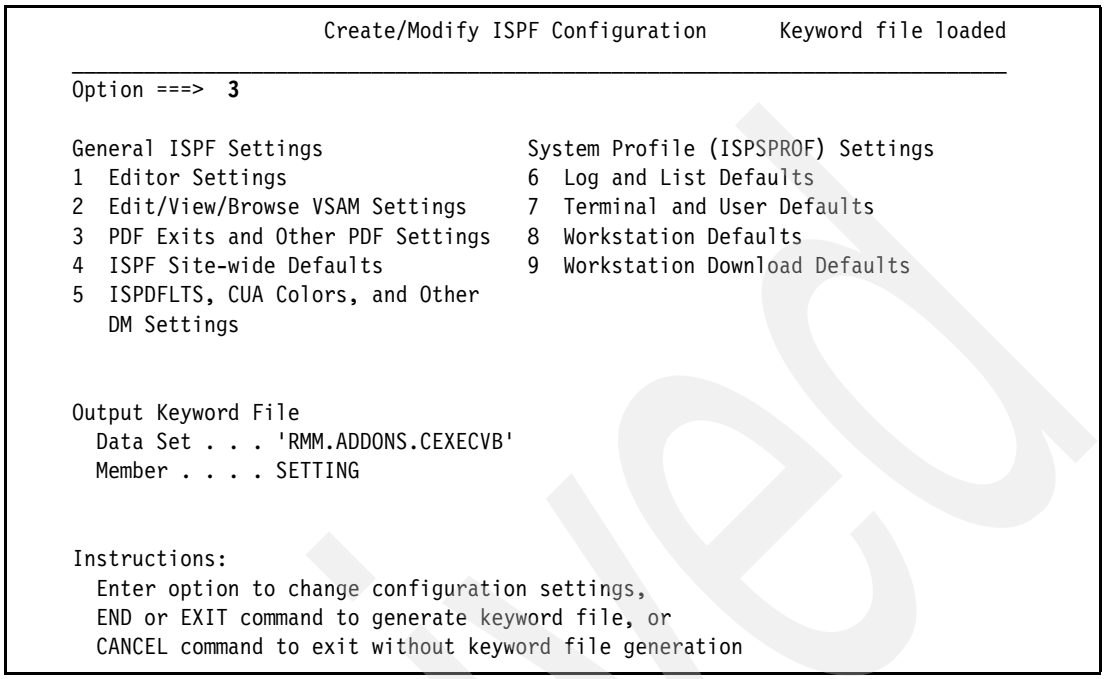


Figure 8-43 PDF Exits and Other PDF Settings selection

The Modify PDF Configuration Settings panel is displayed in a srollable panel and you have scroll down until you can see DSLIST Removable Media Settings, as shown in Figure 8-44.

```
                    Modify PDF Configuration Settings
 ------------------------------------------------------------------------------
Command ===>
                                                                        More:   -
     2. Use IEBCOPY for PDSEs only

  When to use COPY or COPYMOD
  2  1. Use COPY if the target block size is equal to or greater than the
        source block size, COPYMOD otherwise
     2. Use COPY if the target block size is equal to the source block size,
        COPYMOD otherwise
     3. Always use COPYMOD

DSLIST Removable Media Settings
  Enter "/" to select option
  /  Enable RM/Tape Commands

  RM/Tape Command  . . %EDGRPD34
  Command APPLID . . . EDG

Other PDF Settings
  Default PDF Unit . . . . . . . . . . . . SYSALLDA
  Volume for Migrated Data Sets  . . . . . MIGRAT
  Delete Command for Migrated Data Sets    HDELETE
  Allowed Allocation Units . . . . . . . . ANY
  Maximum IEBCOPY Return Code  . . . . . . 0
  Pathname Substitution Character  . . . . Ü

  Enter "/" to select option
     Allocate Before Uncatalog
  /  Verify Expiration Dates
  /  Use SuperC Program Interface
     Monitor Edit Macro Commands via the Activity Monitoring Exit
  /  Allow SUBMIT from Browse
  /  Allow SUBMIT from View
  /  Warn when rename target could be a GDG
  /  Default Edit/Browse/View member list from Option 3.4
  /  Enable View
     Use Panel ISRTSOA in Option 6
     Print using ICF
     Disallow wildcards in the high level qualifier for Data Set List
     Disable all ENQ displays
  /  Fail on LMF lock requests
```

Figure 8-44 Modify PDF Configuration Settings

After you have finished your updates you have to verify your new keyword table using selection 3, in then you have to build a new configuration table using selection 4, as you can see in Figure 8-45.

```
                        ISPF Configuration Utility
 ____________________________________________________________________________
Option ===> 4

1  Create/Modify Settings and Regenerate Keyword File
2  Edit Keyword File Configuration Table
3  Verify Keyword Table Contents
4  Build Configuration Table Load Module
5  Convert Assembler Configuration Table to Keyword File
6  Build SMP/E USERMOD


Keyword File Data Set
  Data Set . . . 'RMM.ADDONS.CEXECVB'
  Member . . . . SETTING

Configuration Table Assembler Source Data Set
  Data Set . . .
  Member . . . .

  Output File Content for Keyword File
  2  1. Include only non-default values
     2. Include defaults as comments
     3. Include all values


Current Configuration Table
  Keyword File : not available
  Identifier . : ISPCFIGU              Level  . . . : 480R8001
  Compile Date : 2003/04/04            Compile Time :
```

Figure 8-45 Build Configuration Table Load Module

A small pop-up window is dispayed to specifiy the input keyword file data set and member and the output configuration table load module data set.

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e                  Build Configuration Table Load Module                    e
e Command ===>                                                              e
e                                                                           e
e Input Keyword File Data set                                               e
e   Data Set . . . 'RMM.ADDONS.CEXECVB'                                     e
e   Member . . . . SETTING                                                  e
e                                                                           e
e Output Configuration Table Load Module Data Set                           e
e   Data Set . . . 'SCHLUM.ISPF.LLIB'                                       e
e                                                                           e
e Optional fields (leave blank for ISPF to use defaults)                    e
e   Object data set  . . .                                                  e
e   Configuration member            (Defaults to ISPCFIGU)                  e
e   VSAM member  . . . . .          (Defaults to ISPCFIGV)                  e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 8-46 Build Configuration Table Load Module pop-up window

Is your target load library is a concatenated library of your LINKLIST you have to refresh the the LINKLIST, as shown in Example 8-14.

Example 8-14 Using the LLA refresh command

```
F LLA,REFRESH
```

Figure 8-47 shows you the successful refresh of your linklist.

```
CSV210I LIBRARY LOOKASIDE REFRESHED
```

Figure 8-47 Successful refresh of linklist

After you have created the new configuration table load module ensure that the specified load library is concatenated to your ISPLLIB, exit ISPF, and start ISPF again.

Important: If the DFSMSrmm REXX exec library, normally called SYS1.SEDGEXE1, is not concatenated in your logon procedure, you must move the two members EDGRMLIB and EDGRPD34 to a concatenated REXX Exec library and update member EDGRPD34 to specify that the RMM ISPF environment is allocated:

```
UseIspfLibdef = true            /* <<   true or false                */
```

8.6.3 Using the ISPF Data Set List Utility support

After you have enabled the ISPF Data Set List (DSLIST) support and restarted your ISPF session, you can use this support. The line commands supported by DFSMSrmm are I, S, M, and D:

I Displays a search results list showing all data sets in the multivolume set for the selected data set.

S Displays the individual data set details. DFSMSrmm determines the first file on the selected volume that matches the selected data set. If other data sets of the same name exist on the volume, the wrong details may be displayed. In that case, use the M line command and then the DFSMSrmm I line command from that results list.

M Displays a search results list showing all data sets defined to DFSMSrmm that match the selected data set name.

D Releases the volume. If the volume is part of a multivolume set, there is the option to release all volumes in the set.

Figure 8-48 shows you the use of the display of individual data set details (S).

```
  Menu  Options  View  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 DSLIST - Data Sets Matching SCHLUM.LI*                          Row 1 of 5
 Command ===>                                                 Scroll ===> PAGE

 Command - Enter "/" to select action              Message          Volume
 ------------------------------------------------------------------------------
          SCHLUM.LIBDUMP                                            SBOX03
          SCHLUM.LIBDUMP.BIN.XMIT                                   SBOXFG
          SCHLUM.LINEITEM.PUNCH                                     SBOX81
  I       SCHLUM.LINEITEM.UNLOAD                   Info - I         TST026
          SCHLUM.LISTDEF.INPUT                                      SBOX09
 ***************************** End of Data Set list ****************************
```

Figure 8-48 Use of the display of individual data set details

The result is shown in Figure 8-49.

```
  Panel  Help
------------------------------------------------------------------------------
                        DFSMSrmm Data Set Details
Command ===>

Data set name . . : 'SCHLUM.LINEITEM.UNLOAD'
Volume serial . . : TST026      Physical file sequence number . . : 1
Owner . . . . . . : SCHLUM      Data set sequence number  . . . . : 1
                                                                More:
Job name  . . . . : SCHLUMR3U
Step name . . . . : STEP010                Record format . . . . : VB
Program name  . . : DSNUTILB               Block size  . . . . . : 32760
DD name . . . . . : SYSREC                 Logical record length : 151
Create date . . . : 2003/300    YYYY/DDD   Block count . . . . . : 36140
Create time . . . : 14:51:16               Total block count . . : 36140
Expiration date . : 2003/300    YYYY/DDD   Percent of volume . . : 2
Original  . . . . :             YYYY/DDD   Device number . . . . : 0B92
System id . . . . : SC63

Last job name . . : PAOLOR3T               Last DD name  . . . . : TAPEVOL
Last step name  . : STEP010                Last device number  . : 0B92
Last program name : IDCAMS
Date last read  . : 2003/302               VRS management value  :
Date last written : 2003/300               Management class  . . : MCDB22
                                           Data class  . . . . . :
Retention date  . :                        Storage class . . . . : SCLIB1
VRS retained  . . : NO                     Storage group . . . . : SGLIB1

Security name . . :
Classification  . :

Primary VRS details:
 VRS name . . . :
 Job name . . . :                            VRS type  . . . . . :
 Subchain name  :                            Subchain start date :

Secondary VRS details:
 Value or class :
 Job name . . . :
 Subchain name  :                            Subchain start date :

Catalog status  . : YES
```

Figure 8-49 Result of display of individual data set details

Figure 8-50 shows all data sets defined to DFSMSrmm that match the selected data set name using function M.

```
  Panel  Help  Scroll
_____________________________________________________________________________
               DFSMSrmm Data Sets (Page 1 of 2)              Row 1 to 1 of 1
Command ===>                                                Scroll ===> PAGE


Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                 Volume          File
S  Data set name                                 serial Owner    seq
-- ---------------------------------------------- ------ -------- -----
   SCHLUM.LINEITEM.UNLOAD                        TST026  SCHLUM  PAOLOR3
****************************** Bottom of data ********************************
```

Figure 8-50 Result if you are selecting function M

Use selection D to releases the volume, as shown in the data set. If you select this function DFSMSrmm shows the confirm volume release panel, as shown in Figure 8-51.

```
  Panel  Help
_____________________________________________________________________________
                      DFSMSrmm Confirm Volume Release
Command ===>

Volume . . . . . : TST026       Location . . . . . : LIB1
  VOL1 volser  . . :            In container . . . . . :
Volume type  . . . : PHYSICAL   Rack number  . . . . . : TST026
Media name . . . . : MEDIA3     Expiration date  . . . . : 2003/365

                                Original expiration date :
Status . . . . . : USER
Description  . . . :
Data set name  . . : 'SCHLUM.LINEITEM.UNLOAD'
                                Release actions:
Media type . . . . : HPCT         Return to SCRATCH pool : NO
Label  . . . . . : SL             Replace volume . . . . : NO
Density  . . . . . : IDRC         Return to owner  . . . : YES
Recording format . : 128TRACK     Initialize volume  . . : NO
Compaction . . . . : YES          Erase volume . . . . . : NO
Attributes . . . . : NONE         Notify owner . . . . . : NO
Availability . : PENDING RELEASE  Expiry date ignore . . : NO
                                  Scratch immediate  . . : NO
Press ENTER to NOFORCE volume, or END command to CANCEL.
```

Figure 8-51 Result if you are selecting function release the volume

Figure 8-52 displays a search result if you have selected function I that includes all volumes residing on the same tape or all data sets residing on the multi-volume set.

```
  Panel  Help  Scroll
_____________________________________________________________________________
                DFSMSrmm Data Sets (Page 1 of 2)            Row 1 to 10 of 10
Command ===>                                                 Scroll ===> PAGE


Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                 Volume          File
S  Data set name                                 serial Owner    seq
-- ---------------------------------------------- ------ -------- -----
   SCHLUM.LINEITEM.UNLOAD                        TST026 MHLRES5  1
   SCHLUM.TESTSTAC.TESTMF02                      TST026 MHLRES5  2
   SCHLUM.TESTSTAC.TESTMF03                      TST026 MHLRES5  3
   SCHLUM.TESTSTAC.TESTMF04                      TST026 MHLRES5  4
   SCHLUM.TESTSTAC.TESTMF05                      TST026 MHLRES5  5
   SCHLUM.TESTSTAC.TESTMF06                      TST026 MHLRES5  6
   SCHLUM.TESTSTAC.TESTMF07                      TST026 MHLRES5  7
   SCHLUM.TESTSTAC.TESTMF08                      TST026 MHLRES5  8
   SCHLUM.TESTSTAC.TESTMF09                      TST026 MHLRES5  9
   SCHLUM.TESTSTAC.TESTMF10                      TST026 MHLRES5  10
****************************** Bottom of data ********************************
```

Figure 8-52 SEARCH volume result

8.6.4 Move the ISPCFIGU module to the SISPLPA library (optional)

After you have installed and verified ISPF, you can enhance its performance by adding the LPA-eligible load modules (in the SISPLPA library) to the LPA list in an LPALSTxx member of PARMLIB. Add those load modules not eligible for LPA (in the SISPLOAD library) to the link list in an LNKLSTxx member of PARMLIB. You can then remove these data sets from the STEPLIB in your TSO LOGON procedure. After adding SISPLPA to LPALST and SISPLOAD to LNKLST, specify CLPA as an initial program load (IPL) parameter to force the SISPLPA modules into the link pack area and to have SISPLOAD added as a system link library.

Finally, after you tested all, you can move the ISPCFIGU member to your ISP. Re-IPL your system to refresh your linklist, as shown in Example 8-15, and then you must re-log in on to your system before you can use this new service.

Example 8-15 Using the LLA refresh command

```
F LLA,REFRESH
```

Figure 8-53 shows you the successful refresh of your linklist.

```
CSV210I LIBRARY LOOKASIDE REFRESHED
```

Figure 8-53 Result of the LLA REFRESH

8.7 Prepare for future releases

In the following sections are suggestions for preparing for future DFSMSrmm releases.

8.7.1 Set a DFSMSrmm control data set ID

Specify one to eight alphanumeric characters that identify the control data set by name, because a CDSID is required in future DFSMSrmm releases. To update your DFSMSrmm control data set control record use the job as shown in Example 8-16.

Example 8-16 Specifying a CDSID

```
//UTIL EXEC PGM=EDGUTIL,PARM='UPDATE'
//SYSPRINT DD SYSOUT=*
//MASTER DD DISP=SHR,DSN=RMM.CONTROL.DSET
//SYSIN DD * D    DDDDD
 CONTROL CDSID(PROD)
/*
```

After you have specified a CDSID in the DFSMSrmm control data set control record you must update your EDGRMMnn member in your corresponding PARMLIB data set, as shown in Example 8-17.

Example 8-17 Sample EDGRMMnn PARMLIB member

```
OPTION    OPMODE(R)                 /* Record-Only Mode         */ -
          ACCOUNTING(J)             /* Accounting from JOB      */ -
          BACKUPPROC(EDGBKUP)       /* Name of BACKUP-proc      */ -
          BLP(RMM)                  /* DFSMSrmm controls BLP    */ -
          CATRETPD(0012)            /* catalog retention        */ -
          CATSYSID(*)               /* all catalogs shared      */ -
          CDSID(PROD)               /* control data set id      */ -
          COMMANDAUTH(OWNER)        /* type of authorization    */ -
          DATEFORM(J)               /* Date format              */ -
          DISPDDNAME(LOANDD)        /* DISP ctrl  DD card       */ -
          DISPMSGID(EDG4054I)       /* DISP message number      */ -
          DSNAME(RMM.PROD.CDS)      /* CDS data set name        */ -
          IPLDATE(N)                /* IPL date checking        */ -
          JRNLNAME(RMM.PROD.JRNL)   /* JRNL data set name       */ -
          JOURNALFULL(75)           /* Percentage JRNL full     */ -
          LINECOUNT(054)            /* Lines per page           */ -
          MASTEROVERWRITE(LAST)     /* Overwriting of a vol     */ -
          MAXHOLD(100)              /* Number of I/O oper.      */ -
          MAXRETPD(NOLIMIT)         /* Maximum retention        */ -
          MEDIANAME(3480)           /* spec. how to move vols   */ -
          MOVEBY(VOLUME)            /* spec. how to move vols   */ -
          MSG(M)                    /* case for message txt     */ -
          NOTIFY(Y)                 /* Notify volume owners     */ -
          PDA(OFF)                  /* PDA is disabled          */ -
          PDABLKCT(255)             /* number of blocks         */ -
          PDABLKSZ(31)              /* blocksize in K           */ -
          PDALOG(OFF)               /* PDA output disabled      */ -
          PREACS(NO)                /* Disable EDGUX100 ACS pr. */ -
          RETAINBY(VOLUME)          /* spec. how to retain vols */ -
          RETPD(0005)               /* Default retention        */ -
          REUSEBIN(STARTMOVE)       /* reuse BIN as soon as pos.*/ -
          SCRATCHPROC(EDGXPROC)     /* ATL/MTL procedure        */ -
          SMFAUD(248)               /* SMF audit records        */ -
          SMFSEC(249)               /* SMF security records     */ -
          SMSACS(YES)               /* enable MV ACS processing */ -
```

```
SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))    ATL*/ -
SYSID(EGZB)                 /* Name of the system       */ -
TPRACF(N)                   /* RACF tape support        */ -
TVEXTPURGE(EXPIRE)          /* set an expiration date   */ -
UNCATALOG(N)                /* Catalog support          */ -
VRSCHANGE(INFO)             /* No additional action     */ -
VRSEL(NEW)                  /* New VRS processing       */ -
VRSJOBNAME(2)               /* DATASETNAME/JOBNAME      */ -
VRSMIN(0000000100,WARN)       /* Warn if < 100 VRSs        */
```

If the CDSID was not previously set you get message EDG0237E MISSING IDENTIFIER FOR THE CONTROL DATA SET. See the description below.

```
EDG0237E MISSING IDENTIFIER FOR THE CONTROL DATA SET.
```

Where:

- Severity

  Information.

- Explanation

  During initialization DFSMSrmm checks to ensure that you have a CDSID specified in the EDGRMMxx PARMLIB member. A CDSID is mandatory.

- Source

  DFSMSrmm.

- Detecting Module

  EDGPARM.

- System Action

  DFSMSrmm initialization stops. This message is followed by message EDG0107A.

- Operators Response

  Notify the system programmer. Reply to message EDG0107A as directed when the system programmer has corrected the error.

- Application Programmer Response

  You must add a CDSID to the OPTION statement in EDGRMMxx before attempting to start DFSMSrmm on a release at z/OS R9 or later. Correct the error in the startup parameters.

- Routing Codes

  3.

- Descriptor Codes

  3.

8.7.2 Re-allocate your DFSMSrmm control data set

Re-allocate your existing DFSMSrmm control data set and use a control interval size (CISZ) of 26624 bytes and BUFFERSPACE of 829440 bytes for the data component of the control data set to help improve inventory management run times through reduced I/O to the control data set. Any suitable control interval size (CISZ) between 10240 and 26624 that meets your needs can be used.

The JCL to allocate the DFSMSrmm control data set in Example 8-18 has been modified to use the recommended values to allocate the VSAM KSDS correctly.

Example 8-18 Sample JCL to allocate the DFSMSrmm CDS

```
//LCLCDSAL JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
// MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ******************************************************************* *
//* STEP 1 Delete old DFSMSrmm control data set *
//* 2 Allocate a new DFSMSrmm control data set *
//* ******************************************************************* *
//STEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DELETE RMM.PROD.CDS CL
  SET MAXCC = 0
/*
//STEP02 EXEC PGM=IDCAMS ALLOCATION MASTER FILE
//SYSPRINT DD SYSOUT=*
//MASTER DD DISP=SHR,UNIT=3390,VOL=SER=DFRMM4
//SYSIN DD *
   DEFINE CLUSTER(NAME(RMM.CONTROL.DSET) -
          FILE(MASTER)                   -
          FREESPACE(15 0)                -
          KEYS(56 0)                     -
          REUSE                          -
          RECSZ(512 9216)                -
          SHR(3 3)                       -
          KILOBYTES(4500 1500)           -
          STORAGECLASS(gspace)           -
          VOLUMES(DFRMM4))               -
   DATA(NAME(RMM.CONTROL.DSET.DATA)      -
          BUFFERSPACE(829440)            -
          CISZ(26624))                   -
   INDEX(NAME(RMM.CONTROL.DSET.INDEX)    -
          CISZ(2048))
/*
```

After you have changed tour VSAM KSDS attributes for the DFSMSrmm control data set you should change the region size parameters in your housekeeping JCL and in the DFRMM procedure as well.

Important: Usa a REGION=0M to use the maximum available region size on your system or specify a minimum of REGION=40MB to speed up your daily housekeeping.

REGION

As you determine the REGION size for the DFSMSrmm started procedure, the amount of virtual storage that DFSMSrmm uses depends on the resources that you have defined. DFSMSrmm virtual storage usage can be affected by any REGION size controls or restrictions that your systems might have in place such as in IEFUSI. The sample DFRMM procedure specifies REGION=40M, which normally provides all the private region below 16 MB and 40 MB above 16 MB. To enable DFSMSrmm to use all available virtual storage, specify REGION=0M. If you want to set a specific region size, consider the following tips along with the current region size of your DFRMM started procedure, to determine whether you need to make any changes to the REGION size.

The VSAM local shared resources (LSR) buffer pool that is built by the DFSMSrmm subsystem for the control data set is obtained above 16 MB. DFSMSrmm builds an LSR buffer pool for the DFRMM started procedure, and also for the EDGUTIL utility batch address space, which has a predetermined size. The LSR buffer pool is 800*data CISZ + 200*index CISZ. Assuming 10 240 for the control data set data CISZ and 2 048 for the control data set index CISZ, the value is 800*10 240+200*2 048=8.4 MB. If you use larger CI sizes, more buffer space is required. For example, if you use a 26K data CISZ, a 21.2 MB buffer size is required.

Recommendation: If the buffer space is larger than 8.6 MB, add the difference to the 40 MB region size that is used by DFSMSrmm and use this value as the REGION size for the DFRMM started procedure.

Here is an example of how you can resolve a S878 abend. As the buffer space increased, an increase of the REGION size resolved the S878 abend. Since the DATA CISZ was increased from 10240 to 26624, the buffer space increased from 8.6 MB to 21.2 MB. That is why HSKPing worked with a REGION size of 40 MB before the CISZ change. The difference of 21.2 and 8.6 requires a REGION size of 52.6 MB, and you have to specify a REGION size of 60 MB.

8.7.3 Update LRECL for REPORT, BACKUP, and JRNLBKUP DD

Update all your JCL to use the new recommended LRECL for the Report, BACKUP, and JRNLBKUP DD statements.

Backing up the control data set

Use the sample JCL in Example 8-19 to get no error until you are backing up the DFSMSrmm control data set. The LRECL of the BACKUP file has be expanded to be increased to 9216 and 9248 bytes.

Example 8-19 Sample JCL to allocate the backup files

```
//EDGHSKP  EXEC PGM=EDGHSKP,PARM='BACKUP(DSS)'
//MESSAGE  DD   DISP=SHR,DSN=RMM.MESSAGE
//SYSPRINT DD   SYSOUT=*
//BACKUP   DD   DISP=(,CATLG),UNIT=TAPE,DSN=BACKUP.CDS(+1),
//         LABEL=(,SL)
//         AVGREC=U,LRECL=9216,BLKSIZE=0,RECFM=U
//JRNLBKUP DD DISP=(,CATLG),UNIT=TAPE,DSN=BACKUP.JRNL(+1),
//         DCB=(RECFM=VB,BLKSIZE=0,LRECL=9248),
//         LABEL=(2,SL),
//         VOL=REF=*.BACKUP
//DSSOPT DD *
   CONCURRENT OPTIMIZE(1) VALIDATE
/*
```

Note: Use this blocksize if you are using DSS or AMS to back up the DFSMSrmm control data set and journal.

Running inventory management

Example 8-20 shows you the expanded logical record length to 255 bytes for the inventory management data set REPORT DD to minimize the risk of having some of the information in a next line.

Example 8-20 Sample JCL to allocate the REPORT file

```
//*        ************************************************************
//STEP04   EXEC PGM=IEFBR14
//DD1      DD   DSN=RMM.HSKP.VRSEL(+1),
//         SPACE=(TRK,(300,30)),
//         DISP=(,CATLG,DELETE),LRECL=255
```

Figure 8-54 shows you an example of how the inventory management report file looks if you have specified a data set with a LRECL with 137 or fewer bytes.

```
REMOVABLE MEDIA MANAGER                              VITAL RECORDS RETENTION REPOR
    (C) IBM CORPORATION 1993,1998                    ----- ------- --------- -----
JOB MASK  DATA SET OR VOLUME MASK                        OWNER    TYPE RETN   C X

          BSYSMF.WEEK.**                                  STCOPC   DSN  CYCLES Y N
JOB NAME  DATA SET NAME                  2ndVRS   2ndNAME  FSEQ  DSEQ VOLUME VSE

SMFWEEK2  BSYSMF.WEEK.G0552V00                                1     1 Q17058
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0551V00                                1     1 Q17191
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0550V00                                1     1 Q17084
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0549V00                                1     1 Q17357
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0548V00                                1     1 Q17287
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0547V00                                1     1 Q17356
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0546V00                                1     1 Q17321
CONT:-
SMFWEEK2  BSYSMF.WEEK.G0545V00                                1     1 Q17068
CONT:-
```

Figure 8-54 REPORT file with a small LRECL

In Figure 8-55 you can see the differences if you are using a larger logical record length to allocate the REPORT file. In this case there are no lines starting with CONT:- included in the report.

```
REMOVABLE MEDIA MANAGER                                VITAL RECORDS RETENTION REPOR
    (C) IBM CORPORATION 1993,1998                      ----- ------- --------- -----
JOB MASK  DATA SET OR VOLUME MASK                           OWNER    TYPE RETN   C X

_____________________________________________________________________________________
          BSYSMF.WEEK.**                                    STCOPC   DSN  CYCLES Y N
JOB NAME  DATA SET NAME                                     FSEQ  DSEQ  VOLUME VSEQ

_____________________________________________________________________________________
SMFWEEK2  BSYSMF.WEEK.G0552V00                                 1     1 Q17058    1
SMFWEEK2  BSYSMF.WEEK.G0551V00                                 1     1 Q17191    1
SMFWEEK2  BSYSMF.WEEK.G0550V00                                 1     1 Q17084    1
SMFWEEK2  BSYSMF.WEEK.G0549V00                                 1     1 Q17357    1
SMFWEEK2  BSYSMF.WEEK.G0548V00                                 1     1 Q17287    1
SMFWEEK2  BSYSMF.WEEK.G0547V00                                 1     1 Q17356    1
SMFWEEK2  BSYSMF.WEEK.G0546V00                                 1     1 Q17321    1
SMFWEEK2  BSYSMF.WEEK.G0545V00                                 1     1 Q17068    1
```

Figure 8-55 REPORT file with a enlarged LRECL

8.7.4 Migrate from VRSEL(OLD) to VRSEL(NEW)

Because the VRSEL(OLD) option is being removed in a future release, and there is a new warning message when you run VRSEL processing on z/OS V1R8, we recommend that you migrate from VRSEL(OLD) to VRSEL(NEW) before moving to z/OS V1R8. If you do not perform this migration you get a warning message (EDG2221E) each time VRSEL is run and EDGHSKP processing ends with job step return code 4.

Any data sets that match to an incomplete chain are retained by a special broken vital record specification. The special broken vital record specification uses the name *broken* and is listed in the REPORT and ACTIVITY files and in the data set matching vital record specification information. With VRSEL(NEW), the broken vital record specification uses a permanent retention date. With VRSEL(OLD), the broken vital record specification ensures that the data set is retained to the maximum of the COUNT value in the first vital record specification.

We recommend using VRSEL(NEW) to:

- Have more flexibility in defining retention and movement policies.
- Override a volume expiration date when a volume is dropped from vital record specification retention and the data set retained on it has the release option EXPIRYDATEIGNORE.
- Return a volume to scratch status in a single inventory management run.

Before starting to use VRSEL(NEW), understand that your existing policies might be applied differently under VRSEL(NEW). You can perform these steps to avoid problems that might occur when the DFSMSrmm performs VRSEL(NEW) processing.

1. Before using VRSEL(NEW), back up your DFSMSrmm control data set using EDGBKUP or EDGHSKP.
2. Perform cleanup on the name vital record specifications by making sure that any retention information in them is correct. DFSMSrmm provides the EDGRVCLN REXX exec described in “EDGRVCLN REXX Procedure to Clean Up Name Vital Record

Specifications" on page 92 to report and clean up problems with name vital record specifications.

3. Run DFSMSrmm inventory management vital record processing so that the DFSMSrmm control data set reflects the cleanup that you have done.
4. Update the DFSMSrmm PARMLIB OPTION VRSEL(NEW) operand.
5. Make sure that all systems sharing DFSMSrmm control data sets have the same PARMLIB options.
6. Run the inventory management VERIFY function against the control data set without introducing any of the new vital record specification functions. When you run VERIFY, changes are not actually made to the DFSMSrmm control data set so that you can look at the results before any changes are made.
7. Inspect the inventory management VERIFY ACTIVITY file by looking at changes in matching vital record specification information, vital record status, and retention date. DFSMSrmm provides a sample job EDGJACTP that you can use with DFSORT to format and print fields in the ACTIVITY file. If you cannot clean up to your satisfaction, you can revert to VRSEL(OLD) at this time.
8. Correct vital record specifications as needed to make sure that the policies that you want are in place.
9. Continue running the VERIFY function and inspect the results until you get the results that you expect using the new functions
10. Begin defining vital record specifications that include the new release options or the use of ANDVRS. We suggest starting slowly until you gain more experience with using the new vital record specification functions. We recommend using the RMM ISPF dialog to add the new vital record specifications or to make changes to existing vital record specifications.
11. Repeat the process from step 7, step 9, and step 10 until you are satisfied that the results are what you expect.
12. Run inventory management production run processing.
13. Check the vital records retention report to make sure that data sets and volumes are retained as you intended.

EDGRVCLN REXX exec

Before you can use VRSEL(NEW), you need to clean up the retention information in your existing vital record specifications. Use the EDGRVCLN REXX procedure to report and update existing vital record specifications that were created before VRSEL(NEW) was introduced. The procedure is intended for use only during the implementation of name vital record specification retention information and for cleanup of vital record specifications even if name vital record specification retention information is not being implemented. EDGRVCLN provides options that you can use to identify retention information that needs to be corrected and options that you can use to correct the information.

The REXX exec has following functions:

LIST(DSNCHAIN) List all vital record specification chains.

LIST(CURRENT) List all vital record specifications that specify LOCATION(CURRENT).

LIST(CYCLES) List all vital record specifications that specify a retention type of CYCLES.

LIST(ERROR) List all name vital record specifications that contain incorrect or incomplete retention information.

LIST(FILTER) List all data set VRSs that are candidates for exploitation of the use of COUNT(0).

FIX(CYCLEBYDAYS) Change all CYCLES vital record specifications to use the CYCLEBYDAYS retention type.

FIX(ERROR) Correct all name vital record specifications that contain incorrect or incomplete retention information as found by LIST(ERROR).

FIX(FILTER) Implement the changes recommend by LIST(FILTER).

EDGRVCLN LIST file

When you use one of the LIST parameters, the procedure produces the DFSMSrmm ADDVRS subcommands that can be issued to define corrected vital record specifications. When you use the LIST(CURRENT), LIST(ERROR), or LIST(CYCLES) parameters, the LIST file includes an RMM DELETEVRS subcommand for each vital record specification in addition to the ADDVRS subcommand. When you use the LIST(FILTER) parameter, the LIST file includes an RMMCHANGEVRS and one or more ADDVRS subcommands for each vital record specification that is a candidate for exploitation of the use of COUNT(0). If you do not want to use the FIX parameters to correct all the identified vital record specifications, you can use the LIST file as input to your own processing. You can edit the file to remove or modify the commands to meet your specific requirements.

EDGRVCLN parameter LIST(DSNCHAIN)

Use this parameter to list all vital record specification chains.

The procedure searches for all data set vital record specifications. The procedure then searches for all name vital record specifications in the chain.

Use JCL as shown in Example 8-21 to list all vital record specification chains.

Example 8-21 Sample JCL to use EDGRVCLN with parameter LIST(DSNCHAIN)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CHAINP    NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CHAIN    NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CHAINP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CHAIN,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
    %EDGRVCLN LIST(DSNCHAIN)
/*
```

Figure 8-56 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(DSNCHAIN)
SEARCHVRS DSNAME 'SCHLUM.TAPE.**' CHAIN  failed with Return Code 4, Reason Code
Error in the CDS
READY
END
```

Figure 8-56 Sample SYSTSPRINT messages using EDGRVCLN with parameter LIST(DSNCHAIN)

The commands generated using parameter LIST(DSNCHAIN) are shown in Figure 8-57.

```
         RMM ADDVRS DSNAME('A.B.C') COUNT(99999) DELETEDATE(1999/365)+
             PRIORITY(0) OWNER(SCHLUM) STORENUMBER(99999) +
             LOCATION(HOME) CYCLES
...
         RMM ADDVRS DSNAME('SCHLUM.RMMDEMO.**') COUNT(1) +
             DELETEDATE(1999/365) +
             DESCRIPTION('MVSSSC SYSTEM BACKUP (PROD)') +
             PRIORITY(0) NEXTVRS(EXTRACAT) OWNER(SCHLUM) STORENUMBER(1) +
             LOCATION(HOME) DAYS +
         RMM ADDVRS NAME(EXTRACAT) DELETEDATE(1999/365) +
             DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
             LOCATION(CURRENT)
         RMM ADDVRS DSNAME('SCHLUM.RMMTEST.MOVE.**') COUNT(99999) +
             DELETEDATE(1999/365) +
             DESCRIPTION('RETAIN AND MOVE DATA SETS') +
             PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) WHILECATALOG +
             LOCATION(REMOTE) CYCLES
```

Figure 8-57 Commands created using EDGRVCLN with parameter LIST(DSNCHAIN)

EDGRVCLN Parameter LIST(CURRENT)

Use this parameter to list all vital record specifications that specify LOCATION(CURRENT).

The procedure searches for all vital record specifications and lists each one that includes the LOCATION(CURRENT).

Use JCL as shown in Example 8-22 to list all vital record specifications that specify LOCATION(CURRENT).

Example 8-22 Sample JCL to use EDGRVCLN with parameter LIST(CURRENT)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CURRENTP NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CURRENT  NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CURRENTP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
```

```
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CURRENT,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(CURRENT)
/*
```

Figure 8-58 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(CURRENT)
4 VRSs with the LOCATION(CURRENT) LISTed.
READY
END
```

Figure 8-58 Sample SYSTSPRINT messages using EDGRVCLN with parameter LIST(CURRENT)

The commands generated using parameter LIST(CURRENT) are shown in Figure 8-59.

```
          RMM DELETEVRS DSNAME('DATA.SET.BACKUP')
          RMM ADDVRS DSNAME('DATA.SET.BACKUP') +
              COUNT(99999) DELETEDATE(1999/365) DESCRIPTION('TEST') +
              PRIORITY(0) OWNER(SCHLUM) STORENUMBER(99999) WHILECATALOG +
              LOCATION(CURRENT) CYCLES
          RMM DELETEVRS DSNAME('D99003')
          RMM ADDVRS DSNAME('D99003') COUNT(3) +
              DELETEDATE(1999/365) DESCRIPTION('TEST') +
              PRIORITY(0) OWNER(SCHLUM) STORENUMBER(3) +
              LOCATION(CURRENT) LASTREFERENCEDAYS
          RMM DELETEVRS NAME(CATALOG)
          RMM ADDVRS NAME(CATALOG) DELETEDATE(1999/365) +
              DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
              LOCATION(CURRENT)
          RMM DELETEVRS NAME(EXTRACAT)
          RMM ADDVRS NAME(EXTRACAT) DELETEDATE(1999/365) +
              DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
              LOCATION(CURRENT)
```

Figure 8-59 Commands created using EDGRVCLN with parameter LIST(CURRENT)

EDGRVCLN parameter LIST(CYCLES)

Use this parameter to list all vital record specifications that specify a retention type of CYCLES. This includes those vital record specifications that include WHILECATALOG, where CYCLES is used as the default retention type.

The procedure searches for all data set vital record specifications and lists each one that includes the CYCLE retention type.

Use JCL as shown in Example 8-23 to list all vital record specification that specify a retention type of CYCLES.

Example 8-23 Sample JCL to use EDGRVCLN with parameter LIST(CYCLES)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
```

```
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CYCLESP  NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CYCLES   NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CYCLESP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CYCLES,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(CYCLES)
/*
```

Figure 8-60 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(CYCLES)
24 DSNAME VRSs with CYCLES retention type LISTed.
READY
END
```

Figure 8-60 Sample SYSTSPRINT messages using EDGRVCLN with parameter LIST(CYCLES)

The commands generated using parameter LIST(CYCLES) are shown in Figure 8-61.

```
          RMM DELETEVRS DSNAME('A.B.C')
          RMM ADDVRS DSNAME('A.B.C') COUNT(99999) +
              DELETEDATE(1999/365) PRIORITY(0) OWNER(SCHLUM) +
              STORENUMBER(99999) +
              LOCATION(HOME) BYDAYSCYCLE
   ....
          RMM DELETEVRS DSNAME('SCHLUM.EDG*.**')
          RMM ADDVRS DSNAME('SCHLUM.EDG*.**') COUNT(6) +
              DELETEDATE(1999/365) DESCRIPTION('TEST') +
              PRIORITY(0) OWNER(SCHLUM) STORENUMBER(6) +
              WHILECATALOG RELEASE(EXPIRYDATEIGNORE SCRATCHIMMEDIATE) +
              LOCATION(HOME) BYDAYSCYCLE
          RMM DELETEVRS DSNAME('SCHLUM.MULTIPLE.DATASET.**')
          RMM ADDVRS DSNAME('SCHLUM.MULTIPLE.DATASET.**') +
              COUNT(1) DELETEDATE(1999/365) DESCRIPTION('TEST')+
              PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) +
              LOCATION(HOME) BYDAYSCYCLE
          RMM DELETEVRS DSNAME('SCHLUM.RMMTEST.MOVE.**')
          RMM ADDVRS DSNAME('SCHLUM.RMMTEST.MOVE.**') COUNT(99999) +
              DELETEDATE(1999/365) +
              DESCRIPTION('RETAIN AND MOVE DATA SETS') +
              PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) WHILECATALOG +
              LOCATION(REMOTE) BYDAYSCYCLE
          RMM DELETEVRS DSNAME('SCHLUM.TAPE.**')
          RMM ADDVRS DSNAME('SCHLUM.TAPE.**') COUNT(1) +
              DELETEDATE(1999/365) DESCRIPTION('TEST NEXT VRS') +
              PRIORITY(0) NEXTVRS(EXTRA) OWNER(SCHLUM) STORENUMBER(1) +
              LOCATION(HOME) BYDAYSCYCLE
          RMM DELETEVRS DSNAME('SCHLUM.TEST1')
          RMM ADDVRS DSNAME('SCHLUM.TEST1') COUNT(1) +
              DELETEDATE(1999/365) PRIORITY(0) OWNER(SCHLUM) +
              STORENUMBER(1) +
              LOCATION(HOME) BYDAYSCYCLE
   ....
```

Figure 8-61 Commands created using EDGRVCLN with parameter LIST(CYCLES)

EDGRVCLN parameter LIST(ERROR)

Use this parameter to list all name vital record specifications that contain incorrect or incomplete retention information. The procedure assumes that any name vital record specification containing a retention type or count value is in error. This is the default value.

The procedure searches for all name vital record specifications and lists each one that includes any retention type or count value.

Use JCL as shown in Example 8-24 to list all vital record specifications that contain incorrect or incomplete retention information.

Example 8-24 Sample JCL to use EDGRVCLN with parameter LIST(ERROR)

```
//EDGRVCLN JOB ,14O.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.ERRORP   NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.ERROR    NONVSAM PURGE
```

```
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.ERRORP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.ERROR,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(ERROR)
/*
```

Figure 8-62 shows you the messages that EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(ERROR)

VRS NAME(CATALOG) is a valid location-only VRS

VRS NAME(D99000) is a valid location-only VRS

VRS NAME(EXTRACAT) is a valid location-only VRS
VRS NAME(TESTAND) had retention values and has been recreated as a
    location-only VRS

VRS NAME(TESTNEXT) is a valid location-only VRS

VRS NAME(TESTX) is a valid location-only VRS
1 NAME VRSs in error are LISTed.
5 NAME VRSs were valid location-only VRSs.
READY
END
```

Figure 8-62 Sample SYSTSPRINT messages using EDGRVCLN with parameter LIST(ERROR)

The commands that are generated using parameter LIST(ERROR) are shown in Figure 8-63.

```
RMM DELETEVRS NAME(TESTAND)
RMM ADDVRS NAME(TESTAND) DELETEDATE(1999/365) +
    DESCRIPTION('TEST LCLVRS1 AND LCLVRS2 FUNCT') +
    ANDVRS(TESTX) OWNER(SCHLUM) STORENUMBER(99999) +
    LOCATION(SHELF)
```

Figure 8-63 Commands created using EDGRVCLN with parameter LIST(ERROR)

EDGRVCLN parameter LIST(FILTER)

Use this parameter to list all data set VRSs that are candidates for exploitation of the use of COUNT(0).

The procedure searches for all data set name VRSs that do not specify COUNT(0) and accumulates those that have common delay, retention, and movement requirements, and NEXT/ANDVRS specified in the first VRS in the chain.

Use JCL as shown in Example 8-25 to list all data set VRSs that are candidates for exploitation of the use of COUNT(0).

Example 8-25 Sample JCL to use EDGRVCLN with parameter LIST(DSNCHAIN)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.FILTERP  NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.FILTER   NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.FILTERP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.FILTER,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(FILTER)
/*
```

Figure 8-64 shows you the messages that EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(FILTER)

Dsname VRSs with retention type CYCLE/BYDAYSCYCLE with the combination of
DELAY() and STORENUMBER() can not be converted.
The following have the same retention criteria and could be converted if
you manually change the VRSs not to have DELAY and STORENUMBER in one VRS:
JOBNAME  DSNAME
TESTYY   TESTYY
TEST*    TESTZZ

RMM commands prepared for the exploitation of the use of COUNT(0) are LISTed.
READY
END
```

Figure 8-64 Sample SYSTSPRINT messages using EDGRVCLN with parameter LIST(FILTER)

The commands that are generated using parameter LIST(FILTER) are shown in Figure 8-65.

```
RMM CHANGEVRS DSNAME('TEST.LCLVRSC.FUNCTION') +
              COUNT(0) +
              NEXTVRS(A0000000)

RMM CHANGEVRS DSNAME('TEST.LCLVRSC.FUNCTION.**') +
              COUNT(0) +
              NEXTVRS(A0000000)

RMM ADDVRS    NAME(A0000000) +
              COUNT(99999) LASTREFERENCEDAYS +
              LOCATION(DISTANT) STORENUMBER(12345) +
              WHILECATALOG UNTILEXPIRED


RMM CHANGEVRS DSNAME('ABEND') +
              JOBNAME(SCHLUM) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM CHANGEVRS DSNAME('ABEND') +
              JOBNAME(SCHLUX) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM CHANGEVRS DSNAME('OPEN') +
              JOBNAME(SCHLUM) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM ADDVRS    NAME(A0000004) +
              COUNT(1) DAYS +
              LOCATION(HOME)


RMM CHANGEVRS DSNAME('SIEGEL.TEST.VRSEL.CATYN') +
              COUNT(0) +
              NEXTVRS(A0000005)

RMM CHANGEVRS DSNAME('SIEGEL.TEST1') +
              COUNT(0) +
              NEXTVRS(A0000005)

RMM ADDVRS    NAME(A0000005) +
              COUNT(3) DAYS +
              LOCATION(HOME)
```

Figure 8-65 Commands created using EDGRVCLN with parameter LIST(FILTER)

FIX(CYCLEBYDAYS)

Use this parameter to change all CYCLES vital record specifications to use the CYCLEBYDAYS retention type.

FIX(ERROR)

Use this parameter to correct all name vital record specifications that contain incorrect or incomplete retention information as found by LIST(ERROR). The corrections are made by

deleting and re-adding the vital record specifications that contain errors. Only use this option when you are ready to correct the errors listed by the LIST(ERROR) option.

FIX(FILTER)

Use this parameter to implement the changes recommend by LIST(FILTER).

The procedure creates one or more retention NAME VRSs for each of the common groups of retentions found, and then uses the CHANGEVRS subcommand to set COUNT(0) and the NEXTVRS to chain to the new NAME VRS(es) created. If DELAY was in use, a retention name VRS is created for the DELAY using DAYS since creation and a NEXTVRS to the retention name VRS(es) for the retention and location pulled from the DSNAME VRS.

9

Tape security

DFSMS V1.8 provides new options for securing tape data sets using System Authorization Facility (SAF). These are designed to allow you to define profiles to protect data sets on tape using the DATASET class without the need to activate the TAPEDSN option or the TAPEVOL class. DFSMS also provides options that you can use to specify that all data sets on a tape volume should have common authorization and that users are authorized to overwrite existing files on a tape volume.

In this chapter we describe the options available in DFSMS V1.8 for tape data set security and how to implement it.

This chapter contains the following sections:

- Tape data set authorization
- Implementation
- Removing TAPEVOL and TAPEDSN processing
- Error messages
- Testing various security settings

9.1 Tape data set authorization

DFSMS V1.8 provides new options for securing tape data sets using the system authorization facility (SAF) to allow you to protect data sets on tape using the RACF DATASET class without the need to activate the TAPEDSN option or the TAPEVOL class. This new tape data set authorization checking allows you to specify that all data sets on a tape volume should have common authorization. You can specify whether users are authorized to overwrite existing files on a tape volume. In addition, you can specify that a user must have access to the first file on a tape volume to add additional files on that tape volume.

9.1.1 Recommendations for tape security

For optimum tape security use the combined capabilities of DFSMSrmm, DFSMSdfp, and RACF. We recommend that you specify the following parameters:

- In DEVSUPxx PARMLIB member:
  - TAPEAUTHDSN=YES
  - TAPEAUTHF1=YES
  - TAPEAUTHRC4=FAIL
  - TAPEAUTHRC8=FAIL
- In EDGRMMxx PARMLIB member:

  OPTION TPRACF(N)
- In RACF:

  SETROPTS NOTAPEDSN NOCLASSACT(TAPEVOL)

The combination of DFSMSrmm, DFSMSdfp, and RACF ensures:

- Full 44 character data set name validation.
- Validation that the correct volume is mounted.
- Control the overwriting of existing tape data sets.
- Management of tape data set retention.
- Control over the creation and destruction of tape volume labels.
- No limitations caused by RACF TAPEVOL profile sizes and TVTOC limitations.
- All tape data sets on a volume have a common authorization.
- Use of generic DATASET profiles, enabling common authorization with DASD data sets.
- Authorization for all tape data sets regardless of the tape label type.
- Authorization for the use of bypass label processing (BLP).
- Exploitation of RACF *erase on scratch* support.
- Use of DFSMSrmm FACILITY class profiles for data sets unprotected by RACF

  Your authorization to use a volume outside of DFSMSrmm control through *ignore* processing also enables authorization to the data sets on that volume.

To aid in the migration to this environment, DFSMSrmm provides the TPRACF(CLEANUP) option, and DEVSUPxx provides TAPEAUTHRC8(WARN) and TAPEAUTHRC4(ALLOW).

The function in DFSMSdfp does not replace all the functional capabilities that the RACF TAPEDSN option, TAPEVOL class, and TVTOC provide. However, together with the functions that DFSMSrmm provides, you do have equivalent capability. The enhanced

DFSMSdfp function addresses the authorization requirements for tape data sets and relies on your use of a tape management system such as DFSMSrmm to perform the following operations:

- Verify full 44 character data set names.
- Control the overwriting of existing tape files.
- Handle tape data set retention.
- Control the creation and destruction of tape labels.

Important: With the DFSMS V1.8 tape security implementation, as opposed to a TAPEVOL and TAPEDSN implementation, you can do the following:

- Write more than 500 data sets to one tape or one tape volume set.
- A tape volume set can expand more than 42 tape volumes.
- Write duplicate data set names to one tape or one tape volume set.

9.1.2 Overview of the TAPEVOL and TAPEDSN processing

The enhancement in authority checking that is available with z/OS DFSMS V1.8 takes into account that today most customers are using a tape management system. A tape management system like DFSMSrmm verifies during reuse of data sets and tape volumes the data set names that were put in place at creation time. This ensures that the same data set resources are always being checked in RACF whenever a particular data set is opened.

How protection of tape data sets works pre-DFSMS V1.8r

Processing in detail differs depending on:

- The setting of SETROPTS TAPEDSN/NOTAPEDSN
- The RACF TAPEVOL class being active or not
- The label type of the tape volume
- The requirement of bypass label processing (BLP)

SETROPTS NOTAPEDSN NOCLASSACT(TAPEVOL)

Tape data sets are not protected if neither RACF option TAPEDSN is being set nor the RACF TAPEVOL class is active. This is true regardless of label type or BLP requirement.

SETROPTS TAPEDSN and SETROPTS CLASSACT(TAPEVOL)

If the RACF TAPEVOL class is active as well as the RACF option TAPEDSN set, tape data are protected at the data set level.

In case of label type SL or AL, during the open of an existing data set a REQUEST=AUTH is performed in the DATASET class with DSNTYPE=T:

1. If there is a matching TAPEVOL class profile, the caller's authority according to the type of open is being checked (READ for input and UPDATE for output).
   - If the caller's authority at the volume level is sufficient, no further checks are performed.
   - If the TAPEVOL profile contains a TVTOC, the data set name is verified at the full length of 44 bytes.
     - If the verification was successful, RACF looks for a matching data set profile and determines whether the caller's authority is sufficient.
     - If verification was not successful the request ends with return code 8 (denied).
   - If the TAPEVOL profile contains no TVTOC see step 2.

2. If there is no matching TAPEVOL profile, RACF looks for a matching DATASET class profile and determines whether the caller is authorized sufficiently (READ for input and UPDATE for output).

Note: Without a TVTOC there is no verification of the full 44 byte length of the data set name.

With a label type of SL or AL, during open of a new data set a REQUEST=AUTH as well as a REQUEST=DEFINE is performed in the DATASET class with DSNTYPE=T:

- If the caller wants to append another data set.
  - If there is a matching TAPEVOL class profile the caller must have at least UPDATE authority, otherwise the request ends with return code 8 (denied).
  - RACF is looking for a matching data set profile and determines that the caller is at least authorized for UPDATE access.
- If the caller wants to overwrite an existing data set.
  - If the new data set name is different from the existing data set, then one of the following is true:
    - The security retention period of that particular existing data set (stored in the protecting DATASET profile) and of all of the following existing data sets must have expired.
    - The caller must have at least UPDATE authority to the volume.
    - RACF is looking for the matching profile of the new data set name and determines that the caller is at least authorized for UPDATE access.
  - If the new data set name is *not* different, then the caller must have UPDATE authority either for the matching DATASET profile or for the matching TAPEVOL profile.
- If the TAPEVOL profile contains a TVTOC, then for the new data set an entry in the TVTOC is maintained.

Note: When a TAPEVOL class profile contains a TVTOC some restrictions apply:

- The maximum number of entries for data sets that a TVTOC can contain is 500.
- The maximum number of volumes that any data set on the tape with an entry in the TVTOC can span is 42.
- From a tape volume set you cannot delete a tape volume if a TVTOC entry indicates that there is a protected data set on the volume.

In case of nonlabeled tapes (NL):

- During open for input the caller must have READ authority to the volume or, if there is a TVTOC for the volume, to the matching data set profile.
- During open for output the caller must have at least UPDATE authority to both the volume and the data set.

In case of nonstandard label tapes (NSL) data management does not do authorization checking.

In case of a request for BLP during open of a data set the caller must be authorized for the ICHBLP resource in the FACILITY class.

SETROPTS TAPEDSN and SETROPTS NOCLASSACT(TAPEVOL)

If only the RACF option TAPEDSN is being set while the TAPEVOL class is not active, tape data is protected at the data set level.

In case of label type SL, AL, or NL during open of a data set for reading, the caller needs at least READ authority for the matching DATASET class profile.

In case of label type SL, AL, or NL during open of a data set for writing, the caller needs at least the following for the matching DATASET class profile:

- UPDATE authority to open an existing data set
- ALTER authority to create a new data set

In the case of nonstandard label tapes (NSL) data management does not do authorization checking.

In case of a request for BLP during open of a data set the caller must be authorized for the ICHBLP resource in the FACILITY class.

SETROPTS NOTAPEDSN and SETROPTS CLASSACT(TAPEVOL)

If the RACF option TAPEDSN is not being set while the TAPEVOL class is active, tape data are protected at the tape volume level.

In the case of label type SL, AL, or NL during open of a data set for reading, the caller needs at least READ authority for the matching TAPEVOL class profile.

In the case of label type SL, AL, or NL during open of a data set for writing, the caller needs at least UPDATE authority for the matching TAPEVOL class profile.

In the case of nonstandard label tapes (NSL) data management does not do authorization checking.

In the case of a request for bypass-label processing during open of a data set the caller must be authorized for the ICHBLP resource in the FACILITY class.

Summary

In order to achieve the highest level of security you should choose an environment that provides discrete TAPEVOL class profiles with TVTOCs as well as the SETROPTS TAPEDSN setting. This environment provides the following advantages:

- Full 44 character data set name validation
- Use of generic DATASET class profiles, enabling common authorization with DASD data sets
- Volume level control in order to guarantee exclusive volume use for applications

9.1.3 How the DFSMS V1.8 tape data set authority checking works

The alternate authority checking is enabled by modifying the contents of member DEVSUPxx in the PARMLIB concatination. Four new keywords are available:

- TAPEAUTHDSN

  YES Enables tape authorization checks in the DATASET class but without DSTYPE=T.

  The system uses the data set name specified in the allocation or JCL to check your authorization to read or write the specified file.

In addition, the system determines the RACF erase-on-scratch setting from the RACF profile and passes it to your tape management system.

When you request bypass label processing (BLP) and the mounted volume uses standard labels, OPEN issues the authorization check that the user is authorized to use BLP. This processing uses the existing ICHBLP resource in the RACF FACILITY class. When you specify TAPEAUTHDSN = YES only, it replaces the check that RACF makes as part of tape volume authorization checking.

NO Indicates OPEN processing to issue RACROUTEs as it did before based on the options set in RACF such as SETROPTS TAPEDSN and SETROPTS CLASSACT(TAPEVOL). This is the default setting.

- TAPEAUTHF1

YES Enables additional tape authorization checks in the DATASET class for existing files on the same tape volume when any other file on the tape volume is opened.

This function depends on the tape management system returning the 44-character data set name and data set sequence number to OPEN/EOV through the IFGTEP during the volume mount exit volume security function. If no data set name is returned by the tape management system, processing is as though this keyword had not been specified.

Although intended to enable an additional authorization check for the first data set when any other data set on the tape volume is opened, the implementation allows your tape management system to request one or more additional authorization checks when any data set on a tape volume is opened. Each additional data set name and data set sequence number returned results in an additional RACROUTE. Do not use this function unless you have a tape management system and it can return a data set name and data set sequence number. A data set sequence number is the label number normally specified in the JCL LABEL keyword and stored in the catalog.

When TAPEAUTHDSN=YES is in use, any additional RACROUTE matches that issued for TAPEAUTHDSN except for the data set name and data set sequence number. Otherwise, TAPEAUTHF1 uses a RACROUTE that matches that used for SETROPTS TAPEDSN. When neither TAPEAUTHDSN nor SETROPTS TAPEDSN is in use, TAPEAUTHF1 support is not provided.

NO Disables additional tape authorization checks in the DATASET class for existing files on the same tape volume when any other file on the tape volume is opened. This is the default setting.

- TAPEAUTHRC4

ALLOW This applies to authorization checks in the DATASET class, and applies only to the results of TAPEAUTHDSN = YES and TAPEAUTHF1 = YES processing.

This allows accessing of data sets that are not protected by a security profile. RC4 refers to the return code value of 4 returned from SAF as a result of the RACROUTE issued by

OPEN/CLOSE/EOV. A return code of 4 in general means that the resource is not protected.

FAIL Denies accessing of data sets that are not protected by a security profile. Use this setting in a PROTECTALL(FAIL) environment. This is the default setting.

- TAPEAUTHRC8

WARN Enables warning mode for all tape authorization checks in the DATASET class as a result of TAPEAUTHDSN = YES and TAPEAUTHF1 = YES processing. Allows accessing of data sets that typically cannot be accessed. RACF issues an ICH408I message to indicate why access is not allowed. However, OPEN/EOV allows access.

FAIL Denies accessing of data sets according to the result of the check in DATASET class. This is the default setting.

In Table 9-1 you can see the different PROTECTALL settings in RACF and the equivalent settings of the new TAPEAUTHRC4 and TAPEAUTHRC8 options for securing tape data sets using the System Authorization Facility (SAF).

Table 9-1 Compare RACF and DEFSUP protection

RACF settings	DEVSUP settings	Action
PROTECTALL(NONE)	N/A	None
PROTECTALL(WARN)	TAPEAUTHRC4(ALLOW) TAPEAUTHRC8(WARN)	Enables warning mode
PROTECTALL(FAIL)	TAPEAUTHRC4(FAIL) TAPEAUTHRC8(FAIL)	Denies accessing of data sets: ► That are not protected ► According to the result of the check

Figure 9-1shows the RACF checking sequence when you have set the new tape security options TAPEAUTHDSN and TAPEAUTHF1 and activated it. If you have not specified to bypass DFSMSrmm processing by using the JCL parameter EXPDT=98000 or ACCODE=XCANORES, then RACF checking is made. The data sets must be RACF protected and the user must have access to the data set profile protecting the data sets.

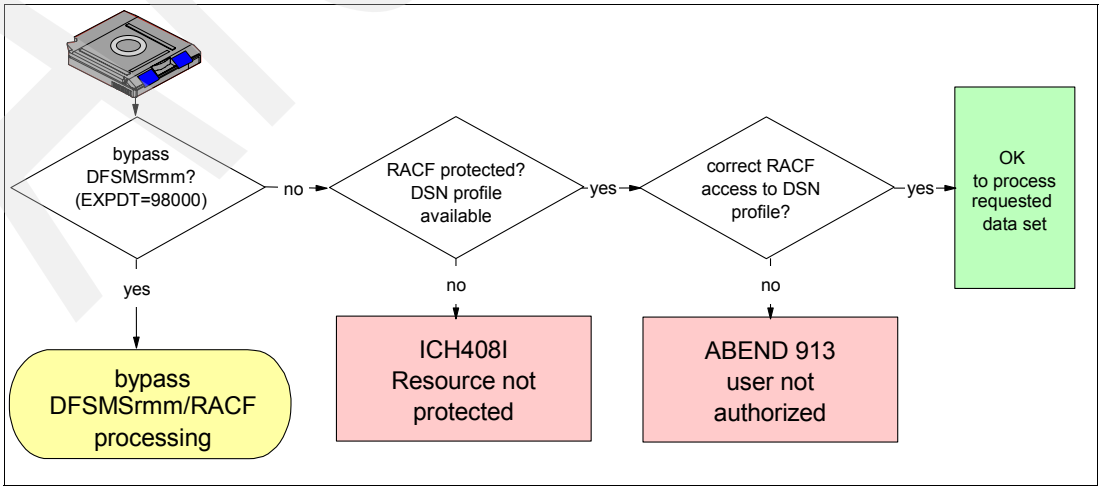


Figure 9-1 New tape security flowchart

If you have specified to bypass DFSMSrmm processing by using the JCL parameter, EXPDT=98000 or ACCODE=XCANORES DFSMSrmm makes additional security checks, as shown in Figure 9-2. Instead of having different RACF resources to check the bypass processing for volumes defined or not defined in the DFSMSrmm control data set, you can specify the resource STGADMIN.EDG.IGNORE.TAPE.* in class FACILITY.

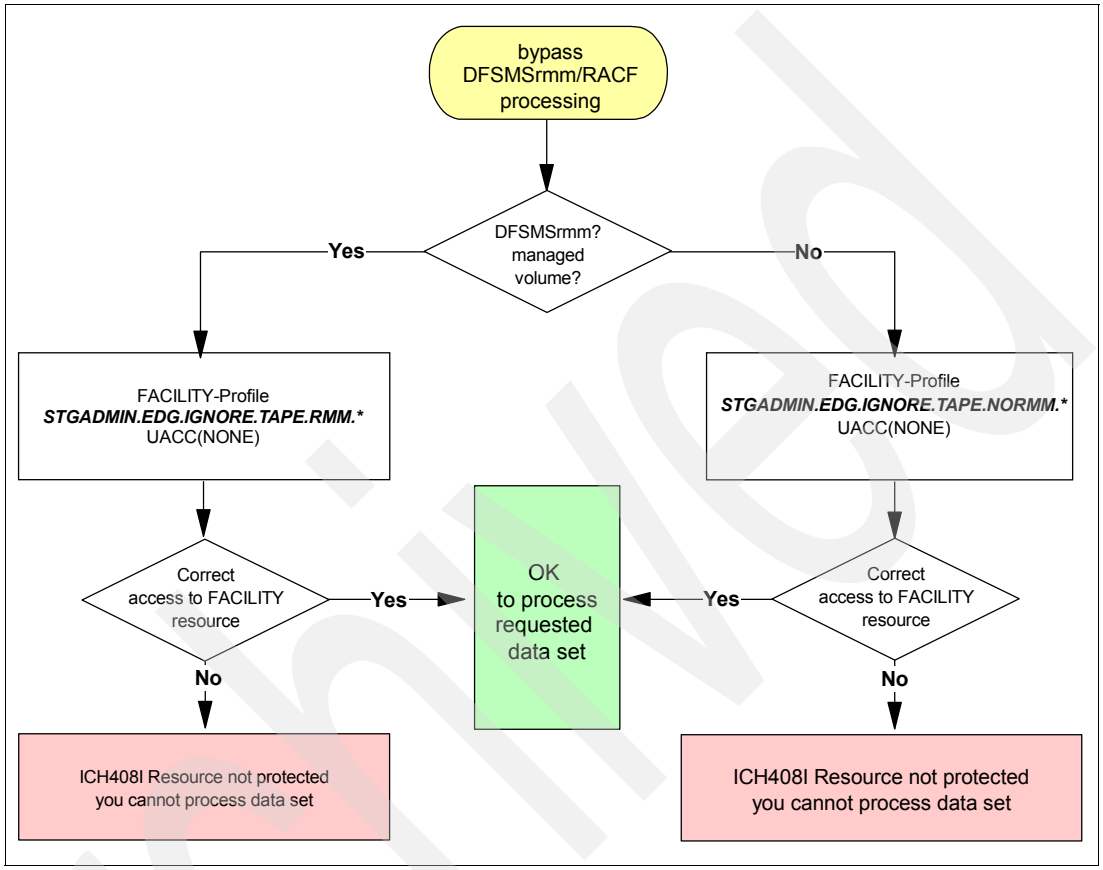


Figure 9-2 Bypass DFSMSrmm processing flowchart

Note: You can specify a profile in RACF class FACILITY for each single volume by using the full volume serial number at the end of the resource name like:

`STGADMIN.EDG.IGNORE.TAPE.RMM.V12345`

or for a range of volumes using the asterisk as part of the volume serial number like the following for all three different kinds of resources:

`STGADMIN.EDG.IGNORE.TAPE.NORMM.ABC*`

9.2 Implementation

In this section we describe how you can implement the new tape data set authorization. This section describes the new implementation of tape data set autorization using the new DEVSUP settings to protect data sets on tape using the DATASET class.

Note: Before you start this new tape data set authorization implementation you should search your DFSMSrmm control data set and check that for all data sets residing on tape, at a minimum, the high-level qualifier is defined to RACF.

You should protect the use of the set MVS DEVSUP command so that only a small number of people have access to this resource and are able to modify your tape data set authorization settings. Example 9-1 shows you how you can protect the use of the MVS DEVSUP command.

Example 9-1 Protect MVS SET DEVSUP command

```
//MHLRES5C JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*
//* ****************************************************************
//* * TESTING Tape Data Set Authorization                           *
//* ****************************************************************
//DELETE   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RDEF OPERCMDS MVS.SET.DEVSUP UACC(NONE)
 PE MVS.SET.DEVSUP CL(OPERCMDS) ID(MHLRES5 MHLRES2) ACC(NONE)
 SETR REFRESH RACLIST(OPERCMDS)
/*
```

Note: Refresh the RACF OPERCMDS because the OPERCMDS are in the RACLIST class.

Use the MVS SET DEVSUP command in Example 9-2 to test your RACF protection of the MVS SET DEVSUP command.

Example 9-2 Unauthorized use of set DEVSUP

```
/T DEVSUP=18
```

This MVS SET COMMAND is not working because the user MHLRES5 is not permitted to use this command. Figure 9-3 shows you the RACF error messages you receive.

```
 RESPONSE=SC64
   ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
     MVS.SET.DEVSUP CL(OPERCMDS)
     INSUFFICIENT ACCESS AUTHORITY
     ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

Figure 9-3 RACF error messages

9.2.1 Check all high-level qualifiers on tape

You should check that each high-level qualifier on volumes in DFSMSrmm status MASTER or USER are defined to RACF as a user or group. You can create different reports to check the high-level qualifiers currently used:

- Only on volumes in status SCRATCH
- Only on volumes that have a status other than SCRATCH
- Using all volumes in any volume status

9.2.2 Update DEVSUPnn PARMLIB member

Update the DEVSUPxx member in your PARMLIB to specify your installation default for device support options. DEVSUPxx is processed during the NIP phase of IPL. After IPL, you can use system command SET DEVSUP=XX to activate the DEVSUP changes.

To enable the new tape data set protection add the following parameters to the DEVSUPxx PARMLIB member:

TAPEAUTHDSN To enable tape authorization checks in the DATASET class

TAPEAUTHF1 Enables additional tape authorization checks in the DATASET class for existing files on the same tape volume when any other file on the tape volume is opened.

TAPEAUTHRC4 Use this keyword to control PROTECTALL processing for tape data sets. This applies to the results of RACROUTE processing when both TAPEAUTHDSN=YES and TAPEAUTHF1=YES are specified.

TAPEAUTHRC8 Use this keyword as an aid to the implementation of TAPEAUTHDSN and TAPEAUTHF1.

Provides a managed and controlled implementation of tape authorization checks in the DATASET class, and applies only to the results of TAPEAUTHDSN=YES and TAPEAUTHF1=YES processing.

Note: While TAPEAUTHRC4=FAIL and TAPEAUTHRC8=FAIL are specified to implement the DFSMS V1.8 tape data set security enhancement, the TAPEAUTHRC4=ALLOW and TAPEAUTHRC8=WARN options are provided to ease the migration to the new tape security implementation. We recommend that you initially specify APEAUTHRC4=ALLOW and TAPEAUTHRC8=WARN.

Figure 9-4 shows a sample DEVSUPxx PARMLIB member including the default for compaction for new data sets written to 3480, 3490, or 3590 tape subsystems. The category used for this system for MEDIA1, MEDIA2, MEDIA3, and MEDIA5 tapes is the status scratch. The PRIVATE category for tapes is the private status, and the ERROR category is for all volumes in which OAM detects any error. This member also includes the final settings for the new tape data set authorization implementation after you have successfully tested this new function.

```
COMPACT = YES,                    /* INSTALLATION DEFAULT FOR IDRC */
MEDIA1 = 0021,
MEDIA2 = 0022,
MEDIA3 = 0023,
MEDIA5 = 0025,
ERROR = 002E,
PRIVATE = 002F,
TAPEAUTHDSN = YES,
TAPEAUTHF1 = YES,
TAPEAUTHRC4 = FAIL,
TAPEAUTHRC8 = FAIL
```

Figure 9-4 Sample DEVSUPxx PARMLIB member

Where:

- COMPACT

  Specifies the default for compaction for new data sets written to 3480, 3490, or 3590 tape subsystems.

- MEDIAx

  Used to specify category codes for library partitioning.

- ERROR

  Used to specify category codes for library partitioning.

- PRIVATE

  Used to specify category codes for library partitioning.

- TAPEAUTHDSN=YES

  Enables tape authorization checks in the DATASET class but without DSTYPE=T.

  DSTYPE=T indicates to RACF that the check is for the data set on a tape volume and that special RACF tape data set and a tape volume processing is to be performed. Without DSTYPE=T RACF authorization checking considers only profiles in the DATASET class.

  The system uses the data set name specified in the allocation or JCL to check your authorization to read or write the specified file.

  In addition, the system determines the RACF erase-on-scratch setting from the RACF profile and passes it to your tape management system.

  Use this option only when you have a tape management system, such as DFSMSrmm, installed and actively checking that the 44-character data set name specified by the user matches the data set name on tape. Without a tape management system, tape data set open processing can only validate the last 17 characters of the data set name against the tape volume labels.

  When you request bypass label processing (BLP) and the mounted volume uses standard labels, OPEN issues the authorization check that the user is authorized to use BLP. This processing uses the existing ICHBLP resource in the RACF FACILITY class. When you specify TAPEAUTHDSN=YES only, it replaces the check that RACF makes as part of tape volume authorization checking.

- TAPEAUTHF1=YES

  Enables additional tape authorization checks in the DATASET class for existing files on the same tape volume when any other file on the tape volume is opened.

  This function depends on the tape management system returning the 44-character data set name and data set sequence number to OPEN/EOV through the IFGTEP during the volume mount exit volume security function. If no data set name is returned by the tape management system, processing is as though this keyword had not been specified.

  Although intended to enable an additional authorization check for the first data set when any other data set on the tape volume is opened, the implementation allows your tape management system to request one or more additional authorization checks when any data set on a tape volume is opened. Each additional data set name and data set sequence number returned results in an additional RACROUTE. Do not use this function unless you have a tape management system and it can return a data set name and data set sequence number. A data set sequence number is the label number normally specified in the JCL LABEL keyword and stored in the catalog.

  When TAPEAUTHDSN=YES is in use, any additional RACROUTE matches that issued for TAPEAUTHDSN except for the data set name and data set sequence number. Otherwise, TAPEAUTHF1 uses a RACROUTE that matches that used for SETROPTS

TAPEDSN. When neither TAPEAUTHDSN nor SETROPTS TAPEDSN is in use, TAPEAUTHF1 support is not provided.

- TAPEAUTHRC4

  Denies accessing of data sets that are not protected by a security profile.

- TAPEAUTHRC8

  Denies accessing of data sets that typically cannot be accessed.

Use the MVS SET DEVSUP command to implement the new tape data set security settings. Figure 9-5 shows the successful result of the command.

```
T DEVSUP=18
IEE252I MEMBER DEVSUP18 FOUND IN SYS1.PARMLIB
IEE536I DEVSUP   VALUE 18 NOW IN EFFECT
IEA253I DEVSUP  348OX RECORDING MODE DEFAULT IS COMPACTION.
IEA253I DEVSUP  ISO/ANSI TAPE LABEL VERSION DEFAULT IS V3
IEA253I DEVSUP  TAPE OUTPUT DEFAULT BLOCK SIZE LIMIT IS 32760
IEA253I DEVSUP  COPYSDB DEFAULT IS INPUT
IEA253I DEVSUP  TAPEAUTHDSN: YES
IEA253I DEVSUP  TAPEAUTHF1: YES
IEA253I DEVSUP  TAPEAUTHRC4: FAIL
IEA253I DEVSUP  TAPEAUTHRC8: FAIL
```

Figure 9-5 Result of the set DEVSUP command

Restriction: Only the new tape security parameters are updated using the SET DEVSUP command. All existing parameters are not updated and cannot be changed without an IPL.

9.3 Removing TAPEVOL and TAPEDSN processing

In this section we provide instructions for changing from using RACF TAPEVOL and TAPEDSN to the new tape data set authorization checking.

9.3.1 Check and modify your RACF settings

First check your RACF settings using the RACF SETR LIST command in a TSO session, as shown in Example 9-3.

Example 9-3 RACF SETR LIST command used in TSO

```
Command ===> SETR LIST
```

Example 9-4 shows you the same command used in a batch job.

Example 9-4 RACF SETR LIST command used in a batch job

```
//SCHLUM   JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*
//* ********************************************************************
//* * TESTING Tape Data Set Authorization                               *
//* ********************************************************************
//SETRLIST EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
```

```
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR LIST
/*
```

The following figures show the output with the current RACF settings. Figure 9-6 shows you the RACF ATTRIBUTES, STATISTICS, and ACTIVE CLASSES.

```
ATTRIBUTES = INITSTATS WHEN(PROGRAM -- BASIC)
STATISTICS = DATASET DASDVOL GDASDVOL GTERMINL O2OMPE TAPEVOL TERMINAL
ACTIVE CLASSES = DATASET USER GROUP ACCTNUM ACICSPCT APPCLU APPCPORT APPCSERV
                 APPCTP APPL BCICSPCT CBIND CCICSCMD CDT CONSOLE CPSMOBJ
                 CPSMXMP CSFKEYS CSFSERV DASDVOL DCICSDCT DIGTCERT DIGTRING
                 DSNR ECICSDCT EJBROLE FACILITY FCICSFCT FIELD FSSEC GCICSTRN
                 GCPSMOBJ GCSFKEYS GDASDVOL GEJBROLE GMQADMIN GSDSF GXFACILI
                 HCICSFCT IBMOPC ILMADMIN JCICSJCT JESJOBS JESSPOOL KCICSJCT
                 KEYSMSTR LOGSTRM MCICSPPT MQADMIN NCICSPPT NETCMDS NETSPAN
                 NODES NODMBR OPERCMDS O2OMPE PCICSPSB PMBR PRINTSRV PROGRAM
                 PTKTDATA PTKTVAL QCICSPSB RACFVARS RACGLIST RRSFDATA
                 RVARSMBR SCICSTST SDSF SERVAUTH SERVER STARTED STORCLAS
                 SURROGAT SYSMVIEW TAPEVOL TCICSTRN TMEADMIN TSOAUTH TSOPROC
                 UCICSTST UNIXPRIV VCICSCMD VTAMAPPL WRITER XFACILIT
```

Figure 9-6 RACF ATTRIBUTES, STATISTICS, and ACTIVE CLASSES

In Figure 9-7 you can see the active RACF GENERIC PROFILE CLASSES.

```
GENERIC PROFILE CLASSES =  DATASET ACCTNUM ACICSPCT AIMS ALCSAUTH APPCLU
                           APPCPORT APPCSERV APPCSI APPCTP APPL CACHECLS
                           CBIND CCICSCMD CIMS CONSOLE CPSMOBJ CPSMXMP
                           CSFKEYS CSFSERV DASDVOL DBNFORM DCEUUIDS DCICSDCT
                           DEVICES DIGTCERT DIGTCRIT DIGTNMAP DIGTRING
                           DIRACC DIRAUTH DIRECTRY DIRSRCH DLFCLASS DSNADM
                           DSNR EJBROLE FACILITY FCICSFCT FIELD FILE FIMS
                           FSOBJ FSSEC GMBR IBMOPC ILMADMIN INFOMAN IPCOBJ
                           JAVA JCICSJCT JESINPUT JESJOBS JESSPOOL KEYSMSTR
                           LDAPBIND LFSCLASS LOGSTRM MCICSPPT MDSNBP MDSNCL
                           MDSNDB MDSNJR MDSNPK MDSNPN MDSNSC MDSNSG MDSNSM
                           MDSNSP MDSNTB MDSNTS MDSNUF MDSNUT MGMTCLAS
                           MQADMIN MQCHAN MQCMDS MQCONN MQNLIST MQPROC
                           MQQUEUE NDSLINK NETCMDS NETSPAN NODES NODMBR
                           NOTELINK NVASAPDT OIMS OPERCMDS O2OMPE PCICSPSB
                           PERFGRP PIMS PMBR PRINTSRV PROCACT PROCESS
                           PROPCNTL PSFMPL PTKTDATA PTKTVAL RACFVARS RACGLIST
                           RMTOPS RODMMGR ROLE RRSFDATA RVARSMBR SCDMBR
                           SCICSTST SDSF SECLMBR SERVAUTH SERVER SFSCMD
                           SIMS SMESSAGE SOMDOBJS STARTED STORCLAS SUBSYSNM
                           SURROGAT SYSMVIEW TAPEVOL TCICSTRN TEMPDSN
                           TERMINAL TIMS TMEADMIN TSOAUTH TSOPROC UNIXMAP
                           UNIXPRIV VMBATCH VMBR VMCMD VMMAC VMMDISK VMNODE
                           VMPOSIX VMRDR VMSEGMT VTAMAPPL VXMBR WRITER
                           XFACILIT
```

Figure 9-7 RACF GENERIC PROFILE CLASSES

Figure 9-8 shows you the current active RACF GENERIC COMMAND CLASSES.

```
GENERIC COMMAND CLASSES =  DATASET ACCTNUM ACICSPCT AIMS ALCSAUTH APPCLU
                           APPCPORT APPCSERV APPCSI APPCTP APPL CACHECLS
                           CBIND CCICSCMD CIMS CONSOLE CPSMOBJ CPSMXMP
                           CSFKEYS CSFSERV DASDVOL DBNFORM DCEUUIDS DCICSDCT
                           DEVICES DIGTCERT DIGTCRIT DIGTNMAP DIGTRING
                           DIRACC DIRAUTH DIRECTRY DIRSRCH DLFCLASS DSNADM
                           DSNR EJBROLE FACILITY FCICSFCT FIELD FILE FIMS
                           FSOBJ FSSEC GMBR IBMOPC ILMADMIN INFOMAN IPCOBJ
                           JAVA JCICSJCT JESINPUT JESJOBS JESSPOOL KEYSMSTR
                           LDAPBIND LFSCLASS LOGSTRM MCICSPPT MDSNBP MDSNCL
                           MDSNDB MDSNJR MDSNPK MDSNPN MDSNSC MDSNSG MDSNSM
                           MDSNSP MDSNTB MDSNTS MDSNUF MDSNUT MGMTCLAS
                           MQADMIN MQCHAN MQCMDS MQCONN MQNLIST MQPROC
                           MQQUEUE NDSLINK NETCMDS NETSPAN NODES NODMBR
                           NOTELINK NVASAPDT OIMS OPERCMDS O2OMPE PCICSPSB
                           PERFGRP PIMS PMBR PRINTSRV PROCACT PROCESS
                           PROPCNTL PSFMPL PTKTDATA PTKTVAL RACFVARS RACGLIST
                           RMTOPS RODMMGR ROLE RRSFDATA RVARSMBR SCDMBR
                           SCICSTST SDSF SECLMBR SERVAUTH SERVER SFSCMD
                           SIMS SMESSAGE SOMDOBJS STARTED STORCLAS SUBSYSNM
                           SURROGAT SYSMVIEW TAPEVOL TCICSTRN TEMPDSN
                           TERMINAL TIMS TMEADMIN TSOAUTH TSOPROC UNIXMAP
                           UNIXPRIV VMBATCH VMBR VMCMD VMMAC VMMDISK VMNODE
                           VMPOSIX VMRDR VMSEGMT VTAMAPPL VXMBR WRITER
                           XFACILIT
```

Figure 9-8 RACF GENERIC COMMAND CLASSES

Figure 9-8 shows you the RACF GENLIST, GLOBAL CHECKING and the settings of the RACLIST classes.

```
GENLIST CLASSES =  NONE
GLOBAL CHECKING CLASSES =  O2OMPE
SETR RACLIST CLASSES =  ACCTNUM APPCPORT APPCSERV APPCTP APPL CBIND CDT
                        CSFKEYS CSFSERV DIGTCERT DIGTRING FACILITY FIELD
                        ILMADMIN JESSPOOL NETCMDS NODES OPERCMDS O2OMPE
                        PRINTSRV PTKTDATA PTKTVAL RACFVARS RRSFDATA SDSF
                        SERVAUTH SERVER STARTED SURROGAT SYSMVIEW TSOAUTH
                        TSOPROC UNIXPRIV VTAMAPPL WRITER XFACILIT
```

Figure 9-9 RACF RACLIST CLASSES

In Figure 9-10 you can see that the tape data set protection is active (TAPEDSN) and that the high-level qualifier PASSWORD will be added to each single-level qualifier data set that is used on this system.

```
GLOBAL=YES RACLIST ONLY =  NONE
AUTOMATIC DATASET PROTECTION IS NOT IN EFFECT
ENHANCED GENERIC NAMING IS IN EFFECT
REAL DATA SET NAMES OPTION IS INACTIVE
JES-BATCHALLRACF OPTION IS INACTIVE
JES-XBMALLRACF OPTION IS INACTIVE
JES-EARLYVERIFY OPTION IS ACTIVE
PROTECT-ALL OPTION IS NOT IN EFFECT
TAPE DATA SET PROTECTION IS ACTIVE
SECURITY RETENTION PERIOD IN EFFECT IS  9999 DAYS.
ERASE-ON-SCRATCH IS INACTIVE
SINGLE LEVEL NAME PREFIX IS PASSWORD
LIST OF GROUPS ACCESS CHECKING IS ACTIVE.
INACTIVE USERIDS ARE NOT BEING AUTOMATICALLY REVOKED.
NO DATA SET MODELLING BEING DONE.
PASSWORD PROCESSING OPTIONS:
  PASSWORD CHANGE INTERVAL IS 180 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS   0 DAYS.
  MIXED CASE PASSWORD SUPPORT IS NOT IN EFFECT
  NO PASSWORD HISTORY BEING MAINTAINED.
  USERIDS NOT BEING AUTOMATICALLY REVOKED.
  NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
  NO INSTALLATION PASSWORD SYNTAX RULES ARE PRESENT.
DEFAULT RVARY PASSWORD IS IN EFFECT FOR THE SWITCH FUNCTION.
DEFAULT RVARY PASSWORD IS IN EFFECT FOR THE STATUS FUNCTION.
SECLABEL CONTROL IS NOT IN EFFECT
GENERIC OWNER ONLY IS IN EFFECT
COMPATIBILITY MODE IS NOT IN EFFECT
MULTI-LEVEL QUIET IS NOT IN EFFECT
MULTI-LEVEL STABLE IS NOT IN EFFECT
NO WRITE-DOWN IS NOT IN EFFECT
MULTI-LEVEL ACTIVE IS NOT IN EFFECT
CATALOGUED DATA SETS ONLY, IS NOT IN EFFECT
USER-ID FOR JES NJEUSERID IS : ????????
USER-ID FOR JES UNDEFINEDUSER IS : ++++++++
PARTNER LU-VERIFICATION SESSIONKEY INTERVAL DEFAULT IS "NEVER EXPIRES".
ADDCREATOR IS NOT IN EFFECT
KERBLVL =     1
MULTI-LEVEL FILE SYSTEM IS NOT IN EFFECT
MULTI-LEVEL INTERPROCESS COMMUNICATIONS IS NOT IN EFFECT
MULTI-LEVEL NAME HIDING IS IN EFFECT
SECURITY LABEL BY SYSTEM IS NOT IN EFFECT
PRIMARY LANGUAGE DEFAULT : ENU
SECONDARY LANGUAGE DEFAULT : ENU
```

Figure 9-10 Other RACF-related information

In our example above you can see that the RACF TAPEVOL class is active and TAPE DATA SET PROTECTION is set so we deactivate both functions using the RACF commands, as shown in Example 9-5.

Example 9-5 Deactivate the RACF TAPEVOL class and TAPE DATA SET PROTECTION

```
//MHLRES5C JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*
//* ****************************************************************
//* * TESTING Tape Data Set Authorization                          *
//* ****************************************************************
//DELETE   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETROPTS NOCLASSACT(TAPEVOL)
 SETROPTS NOTAPEDSN
 SETR REFRESH GENERIC(TAPEVOL)
/*
```

9.3.2 Check and modify your DFSMShsm settings

Check your DFSMShsm settings and switch off the TAPESECURITY if this option is set. Figure 9-11 shows you a sample ARCCMDxx PARMLIB member where you can see that the SETSYS TAPESECURITY(RACF) option is used.

```
...
/*****************************************************************/
/*              DFSMSHSM RACF SPECIFICATIONS                     */
/*****************************************************************/
/*                                                               */

SETSYS                    /* DO NOT PUT RACF-INDICATION        */ -
  NORACFIND               /* ON BACKUP AND MIGRATION           */
                          /* COPIES OF DATA SETS.              */

SETSYS                    /* USE RACF TO PROVIDE TAPE SECURITY */ -
  TAPESECURITY(RACF EXPIRATIONINCLUDE)        /* USE 99365 EXPDT */

SETSYS                    /* DO NOT ALLOW ERASE-ON-SCRATCH     */ -
  NOERASEONSCRATCH        /* ON ANY DFSMSHSM BACKUP            */
                          /* VERSIONS AND MIGRATION COPIES     */

SETSYS                    /* BACKUP DISCRETE RACF PROFILES     */ -
  PROFILEBACKUP

/*                                                               */
...
```

Figure 9-11 Sample DFSMShsm PARMLIB member

The use of the TAPESECURITY(RACF) means that DFSMShsm protects each backup, migration, and dump tape with RACF. DFSMShsm also protects alternate backup and migration tapes generated as a result of TAPECOPY processing. The RACF subparameter does not support backup or migration of password-protected data sets.

Note: The RACF option of the SETSYS TAPESECURITY command directs DFSMShsm to automatically add RACF protection to scratch tapes.

If the TAPESECURITY is set to RACF, change your ARCCMDxx PARMLIB member, as shown in Figure 9-12, to remove the RACF setting. Restart your DFSMShsm so that the new setting is used.

```
...
/*****************************************************************/
/*               DFSMSHSM RACF SPECIFICATIONS                    */
/*****************************************************************/
/*                                                               */

SETSYS                      /* DO NOT PUT RACF-INDICATION        */ -
  NORACFIND                 /* ON BACKUP AND MIGRATION           */
                            /* COPIES OF DATA SETS.              */

SETSYS                      /* USE 99365 TO WIRTE  AN EXPDT      */ -
  TAPESECURITY(EXPIRATIONINCLUDE)      /*  TO EACH TAPE         */

SETSYS                      /* DO NOT ALLOW ERASE-ON-SCRATCH     */ -
  NOERASEONSCRATCH          /* ON ANY DFSMSHSM BACKUP            */
                            /* VERSIONS AND MIGRATION COPIES     */

SETSYS                      /* BACKUP DISCRETE RACF PROFILES     */ -
  PROFILEBACKUP

/*                                                               */
...
```

Figure 9-12 Sample DFSMShsm PARMLIB member with RACF setting

Note: You must re-start you DFSMShsm, because there is no SETSYS command available to switch off the RACF setting.

9.3.3 Clean up your TAPEVOL profiles using DFSMSrmm settings

Modify your DFSMSrmm settings in EDGRMMxx if you have specified that DFSMSrmm maintains the security profiles that protect tape volumes specified in the TPRACF OPTION operand.

TPRACF

Check the type of RACF tape support that you have selected. If you have specified TPRACF(AUTOMATIC) or TPRACF(PREDEFINED) you should now change this setting to one of the following:

TPRACF(CLEANUP) DFSMSrmm ensures that TAPEVOL profiles and discrete tape DATASET profiles are deleted during recycling of scratch volumes and existing TAPEVOL profiles are deleted when volumes are deleted from the DFSMSrmm CDS. When you use this option, DFSMSrmm never creates any RACF profiles for you. This processing is only provided for VLPOOLs with RACF(Y). TPRACF(CLEANUP) is intended to be used when you are changing how tape data sets are protected. For example, if you no longer wish

to use TAPEVOL profiles and are enabling the use of DATASET profiles, TPRACF(CLEANUP) can be used for this occasion. When you specify TPRACF(CLEANUP), DFSMSrmm deletes RACF tape profiles for any volumes in your installation based on the following VLPOOL values:

- VLPOOL RACF(N), DFSMSrmm does no processing of tape profiles for volumes in the pool at any time.
- VLPOOL RACF(Y), RACF tape profiles are deleted when RMM CHANGEVOLUME or DELETEVOLUME subcommands are issued. DFSMSrmm deletes TAPEVOL and discrete tape data set profiles during recycling of scratch tapes if the profiles exist.

TPRACF(NONE) DFSMSrmm do not manipulate an TAPEVOL profile in any circumstances.

Example 9-6 shows out the update to your EDGRMMxx PARMLIB member OPTION command if you would like DFSMSrmm to clean up your security TAPEVOL profiles.

Example 9-6 Update the OPTION command

```
  OPTION    OPMODE(R)                 /* Record-Only Mode         */ -
            ACCOUNTING(J)             /* Accounting from JOB      */ -
            BACKUPPROC(EDGBKUP)       /* Name of BACKUP-proc      */ -
            BLP(RMM)                  /* DFSMSrmm controls BLP    */ -
            CATRETPD(0012)            /* catalog retention        */ -
            CATSYSID(*)               /* all catalogs shared      */ -
            CDSID(PROD)               /* control data set id      */ -
            COMMANDAUTH(OWNER)        /* type of authorization    */ -
            DATEFORM(J)               /* Date format              */ -
            DISPDDNAME(LOANDD)        /* DISP ctrl  DD card       */ -
            DISPMSGID(EDG4054I)       /* DISP message number      */ -
            DSNAME(RMM.PROD.CDS)      /* CDS data set name        */ -
            IPLDATE(N)                /* IPL date checking        */ -
            JRNLNAME(RMM.PROD.JRNL)   /* JRNL data set name       */ -
            JOURNALFULL(75)           /* Percentage JRNL full     */ -
            LINECOUNT(054)            /* Lines per page           */ -
            MASTEROVERWRITE(LAST)     /* Overwriting of a vol     */ -
            MAXHOLD(100)              /* Number of I/O oper.      */ -
            MAXRETPD(NOLIMIT)         /* Maximum retention        */ -
            MEDIANAME(3480)           /* spec. how to move vols   */ -
            MOVEBY(VOLUME)            /* spec. how to move vols   */ -
            MSG(M)                    /* case for message txt     */ -
            NOTIFY(Y)                 /* Notify volume owners     */ -
            PDA(OFF)                  /* PDA is disabled          */ -
            PDABLKCT(255)             /* number of blocks         */ -
            PDABLKSZ(31)              /* blocksize in K           */ -
            PDALOG(OFF)               /* PDA output disabled      */ -
            PREACS(NO)                /* Disable EDGUX100 ACS pr. */ -
            RETAINBY(VOLUME)          /* spec. how to retain vols */ -
            RETPD(0005)               /* Default retention        */ -
            REUSEBIN(STARTMOVE)       /* reuse BIN as soon as pos.*/ -
            SCRATCHPROC(EDGXPROC)     /* ATL/MTL procedure        */ -
            SMFAUD(248)               /* SMF audit records        */ -
            SMFSEC(249)               /* SMF security records     */ -
            SMSACS(YES)               /* enable MV ACS processing */ -
```

```
SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))    ATL*/ -
SYSID(EGZB)                  /* Name of the system       */ -
TPRACF(CLEANUP)              /* RACF tape support        */ -
TVEXTPURGE(EXPIRE)           /* set an expiration date   */ -
UNCATALOG(N)                 /* Catalog support          */ -
VRSCHANGE(INFO)              /* No additional action     */ -
VRSEL(NEW)                   /* New VRS processing       */ -
VRSJOBNAME(2)                /* DATASETNAME/JOBNAME      */ -
VRSMIN(0000000100,WARN)      /* Warn if < 100 VRSs       */
```

Example 9-7 shows you the correct setting of the of the VLPOOL command in the EDGRMMnn PARMLIB member.

Example 9-7 Update the VLPOOL command

```
VLPOOL        PREFIX(TST*)                                    -
              TYPE(S)                                         -
              DESCRIPTION('LOGISCHE VTS VOLUMES          ')   -
              MEDIANAME(3490)                                 -
              RACF(Y)                                         -
              EXPDTCHECK(N)           /* NOT TO CHECK OR VALIDATE */
VLPOOL        PREFIX(E*)                                      -
              TYPE(S)                                         -
              DESCRIPTION('Enhanced 3590 cartridges      ')   -
              MEDIANAME(3490)                                 -
              RACF(Y)                                         -
              EXPDTCHECK(N)           /* NOT TO CHECK OR VALIDATE */
VLPOOL        PREFIX(M*)                                      -
              TYPE(S)                                         -
              DESCRIPTION('MEDIA5 cartridges             ')   -
              MEDIANAME(3490)                                 -
              RACF(Y)                                         -
              EXPDTCHECK(N)           /* NOT TO CHECK OR VALIDATE */
```

Where RACF(Y) specifies that you want DFSMSrmm to delete RACF tape profiles for the volumes in the pool.

9.3.4 Clean up your TAPEVOL profiles using commands

If you would like to clean up your security server directly you can use the TSO RMM SEACHVOLUME subcommand with the CLIST operand to create RACF delete TAPEVOL commands. After you have checked the commands you can execute them and delete all your existing TAPEVOL profiles directly. Example 9-8 shows you sample JCL to create the RDELETE statements.

Example 9-8 Create RACF RDELETE commands

```
//LCLRDEL  JOB ,RMM,NOTIFY=&SYSUID,
//         MSGCLASS=H,CLASS=O,MSGLEVEL=(1,1),REGION=0M
//CLEANUP  DD   PGM=IDCAMS
//SYSPRINT DD   DUMMY
//SYSIN    DD   *
  DELETE RMM.DELETE.RACF.TAPEVOL.TST NONVSAM PURGE
/*
//LCLSVSM  EXEC PGM=IKJEFT01,DYNAMNBR=99
//SYSTSPRT DD   SYSOUT=*
//RMMCLIST DD   DISP=(,CATLG),DSN=RMM.DELETE.RACF.TAPEVOL.TST,
//         SPACE=(TRK,(45,45),RLSE),LRECL=80,RECFM=FB,
```

```
//         UNIT=SYSDA
//SYSTSIN  DD   *
 RMM SV VOLUME(TST*) LIMIT(*) OWNER(*) -
     CLIST('RDELETE TAPEVOL ','')
/*
//
```

Where:

- RMMCLIST DD

  The RMM TSO SEARCH subcommands with the CLIST operand is writing to this data set to store the commands.

- CLIST(prefix_string,suffix_string)

  Specifies a CLIST to create a data set of executable commands or to prepare an import list. You can edit the data set to remove any volumes that you do not want in the list. Then you can run the CLIST at your convenience.

  DFSMSrmm returns the volume serial number for each record if you do not specify (prefix_string and suffix_string). When the volume serial number contains special characters the value is returned within quotation marks.

  You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (prefix_string and suffix_string). These text strings cannot exceed 255 characters. Separate the prefix_string and suffix_string using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns. To enter a null prefix_string, add a pair of separator characters such as ” to the text string (for example, CLIST(”,’ suffix_string’)).

Figure 9-13 shows you how the RACF commands are created using the job above.

```
RDELETE TAPEVOL TST000
RDELETE TAPEVOL TST001
RDELETE TAPEVOL TST002
RDELETE TAPEVOL TST003
RDELETE TAPEVOL TST004
RDELETE TAPEVOL TST005
RDELETE TAPEVOL TST006
RDELETE TAPEVOL TST007
RDELETE TAPEVOL TST008
RDELETE TAPEVOL TST009
RDELETE TAPEVOL TST010
RDELETE TAPEVOL TST011
RDELETE TAPEVOL TST012
RDELETE TAPEVOL TST013
RDELETE TAPEVOL TST014
RDELETE TAPEVOL TST015
RDELETE TAPEVOL TST016
RDELETE TAPEVOL TST017
RDELETE TAPEVOL TST018
RDELETE TAPEVOL TST019
RDELETE TAPEVOL TST020
RDELETE TAPEVOL TST021
RDELETE TAPEVOL TST022
```

Figure 9-13 Sample RACF RDELETE commands

After you have checked the commands you can execute them using the JCL shown in Example 9-9.

Example 9-9 Execute RACF RDELETE commands

```
//LCLRDEL  JOB ,RMM,NOTIFY=&SYSUID,
//         MSGCLASS=H,CLASS=O,MSGLEVEL=(1,1),REGION=OM
//EXECUTE  EXEC PGM=IKJEFTO1,DYNAMNBR=99
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 EX 'RMM.DELETE.RACF.TAPEVOL.TST'
/*
//
```

9.4 Error messages

Until the TAPEVOL profiles are deleted you may see some security violations and you should check your system log from time to time for messages.

Figure 9-14 shows message IEC150I, which is generated if you do not have the correct access to create a given data set.

```
JOB22272  IEC150I 913-60,IFG0196T,GENER99,STEP01,SYSUT2,0B91,TST026,DTAUS
JOB22272  IEA995I SYMPTOM DUMP OUTPUT  928
          SYSTEM COMPLETION CODE=913  REASON CODE=00000060
           TIME=20.17.17  SEQ=02226  CPU=0000  ASID=002A
           PSW AT TIME OF ERROR  075C1000   80C49CBE  ILC 2  INTC 0D
             NO ACTIVE MODULE FOUND
             NAME=UNKNOWN
             DATA AT PSW  00C49CB8 - 41003B7A  0A0D41F0  38BE56F0
             GR 0: 00000000_00C49F84   1: 00000000_A4913000
                2: 00000000_000097F4   3: 00000000_00C4940A
                4: 00000000_007DD1F8   5: 00000000_007DD58C
                6: 00000000_007DD534   7: 00000000_007DD58C
                8: 00000000_007DD554   9: 00000000_007C71A8
                A: 00000000_007DBCE0   B: 00000000_00C4CB04
                C: 00000000_80C4CBE4   D: 00000000_007DD4B8
                E: 00000000_80C49542   F: 00000010_00000060
           END OF SYMPTOM DUMP
JOB22272  IEF450I GENER99 STEP01 - ABEND=S913 U0000 REASON=00000060  929
```

Figure 9-14 IEC150I message

You get the message:

IEC150I 913-*rc,mod,jjj,sss, ddname[-#],dev,ser,dsname(member)*

Where:

rc	Associates this message with system completion code 913 and with the return code.
jjj	The job name.
sss	The step name.
ddname[-#]	DDname (followed by a concatenation number if it is part of a concatenation and not the first DD statement in the concatenation).

dev	The device number.
ser	The volume serial number.
mod	The name of the module in which the error occurred.
dsname(member)	The data set name. Member name if specified. The explanation for the hex return code is as follows: For RACF errors, see message IDC3009I for the return code.
Severity	Information.
Explanation	The error occurred during 1) the processing of an OPEN macro instruction or during end-of-volume for a password-protected data set after the operator attempted to enter a password in response to message IEC301A, or 2) the processing of an OPEN macro instruction involving a checkpoint data set. A VSAM data set is being opened with a DCB instead of an ACB.

The explanation for the hex return code is as follows: For RACF errors, see message IDC3009I for the return code.

Return code	Explanation
60	One of the following occurred: • The user is not authorized to define a data set with the specified name. The specified data set name and file sequence indicator do not match the corresponding names in the Tape Volume Table of Contents (TVTOC). • The user is not authorized to access this data set.

Attention: In this case the volumes gets the volume error status of SECURITY CONFLICT in the VOLCAT and is protected for future use until you have reset this situation, if the volume is SMS managed:

```
 LINE       VOLUME USE      VOLUME              CHECKPT  LIBRARY   STORAGE
 OPERATOR   SERIAL ATTR     ERROR STATUS        VOLUME   NAME      GRP NAME
---(1)----  -(2)-- --(3)--  -------(4)--------  --(5)--  --(6)---  --(7)---
            TST020 SCRATCH  SECURITY CONFLICT   NO       LIB1      *SCRTCH*
----------  ------ ----------  BOTTOM  OF  DATA  ----------  ------  ----
```

Figure 9-15 shows you an ICH408I message you get now for tapes. You currently get this message for data sets residing on DASD volumes if you do not have the correct access to the specified security resource in RACF facility class.

```
JOB22265  IEF233A M 0B91,PRIVAT,SL,GENER99,STEP01,MHLRES1.TESTSTAC.TESTYF01
JOB22265  ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)  781
            MHLRES1.TESTSTAC.TESTYF01 CL(DATASET ) VOL(TST02O)
            INSUFFICIENT ACCESS AUTHORITY
            FROM MHLRES1.** (G)
            ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
JOB22265  IEC518I SOFTWARE ERRSTAT: RACFPROT 0B91,TST02O,SL,GENER99,STEP01
JOB22265  IEC502E RK 0B91,TST02O,SL,GENER99,STEP01
JOB22265  IEC150I 913-38,IFG0194F,GENER99,STEP01,SYSUT2,0B91,,MHLRES1.TESTSTAC.T
JOB22265  IEA995I SYMPTOM DUMP OUTPUT  785
          SYSTEM COMPLETION CODE=913  REASON CODE=00000038
           TIME=19.54.16  SEQ=02221  CPU=0000  ASID=002A
           PSW AT TIME OF ERROR  075C1000   80C49CBE  ILC 2  INTC 0D
             NO ACTIVE MODULE FOUND
             NAME=UNKNOWN
             DATA AT PSW  00C49CB8 - 41003B7A  0A0D41F0  38BE56F0
             GR 0: 00000000_00C49F84   1: 00000000_A4913000
                2: 00000000_000097F4   3: 00000000_00C4940A
                4: 00000000_007C51F8   5: 00000000_007C558C
                6: 00000000_007C5534   7: 00000000_007C558C
                8: 00000000_007C5554   9: 00000000_007C3040
                A: 00000000_007DBCE0   B: 00000000_00C4CB04
                C: 00000000_80C4CBE4   D: 00000000_007C54B8
                E: 00000000_80C49542   F: 00000010_00000038
           END OF SYMPTOM DUMP
JOB22265  IEF450I GENER99 STEP01 - ABEND=S913 U0000 REASON=00000038  786
                 TIME=19.54.16
JOB22265  -                                      --TIMINGS (MINS.)--
JOB22265  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
JOB22265  -GENER99           STEP01   *S913     51    .00    .00    .56    362
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Figure 9-15 Normal ICH408I security violation

Note: If the volume is an SMS managed volume the status of the volume is not changed in the VOLCAT and the volume can be continually accessed.

9.5 Testing various security settings

The following jobs give you an overview of how the different tape security settings work and the error messages that you get if your job abends. Table 9-2 shows you the different settings that we tested.

Table 9-2 Different RACF, DEVSUP, and DFSMSrmm settings

Test case/ success	RACF settings					TAPEAUTH settings in the DEVSUPnn				DFSMSrmm [2]	
	TAPEVOL	TAPEDSN	THM00n[4]	MHLRES1.**	MHLRES7.**	DSN	F1	RC4	RC8	TPRACF	RACF
All test cases between 1 and 19 are without the use of the new TAPEAUTH settings in the DEVSUPnn member. Create two new data sets on a tape volume, but at this time there are no DEVSUP settings.											

Test case/ success	RACF settings					TAPEAUTH settings in the DEVSUPnn				DFSMSrmm [2]	
	TAPEVOL	TAPEDSN	THM00n[4]	MHLRES1.**	MHLRES7.**	DSN	F1	RC4	RC8	TPRACF	RACF
1 ☑	ACTIVE	ACTIVE	N/A [1]	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
Read the previously created data sets with different RACF settings and different access to the DATASET and TAPEVOL profile.											
2 ☑	ACTIVE	ACTIVE	ALTER	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
3 ☑	ACTIVE	ACTIVE	ALTER	NONE	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
4 ☑	ACTIVE	ACTIVE	NONE	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
5 ☒	ACTIVE	ACTIVE	NONE	NONE	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
6 ☑	N/A [1]	ACTIVE	N/A [1]	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
7 ☒	N/A [1]	ACTIVE	N/A [1]	NONE	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
8 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
9 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	NONE	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
Create an additional file to the previously created two tape data sets with different RACF settings.											
10 ☑	ACTIVE	ACTIVE	ALTER	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
11 ☒	ACTIVE	ACTIVE	ALTER	ALTER	NONE	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
12 ☑	ACTIVE	N/A [1]	ALTER	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
13 ☑	ACTIVE	N/A [1]	ALTER	ALTER	NONE	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
14 ☑	N/A [1]	ACTIVE	ALTER	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
15 ☒	N/A [1]	ACTIVE	ALTER	ALTER	NONE	N/A [1]	N/A [1]	N/A [1]	N/A [1]	A [3]	Y
Read the previously created data sets, but at this time the DFSMSrmm RACF support is switched off.											
16 ☑	ACTIVE	ACTIVE	ALTER	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	**N**	**N**
17 ☑	ACTIVE	ACTIVE	ALTER	ALTER	NONE	N/A [1]	N/A [1]	N/A [1]	N/A [1]	**N**	**N**
18 ☑	N/A [1]	ACTIVE	N/A [1]	ALTER	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	**N**	**N**
19 ☒	N/A [1]	ACTIVE	N/A [1]	NONE	ALTER	N/A [1]	N/A [1]	N/A [1]	N/A [1]	**N**	**N**
All test cases below are now tested with the new TAPEAUTH settings in the DEVSUPnn member, the RACF CLASS TAPEVOL is not active, and option TAPEDSN is inactive. Create two new data sets on a tape volume, but at this time RACF TAPEVOL and TAPEDSN are inactive and the DFSMSrmm RACF support is switched off.											
20 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	YES	YES	FAIL	Fail	N	N
Read the previously created data sets with different RACF access to MHLRES1.** and MHLRES7.**.											
21 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	YES	YES	FAIL	FAIL	N	N
22 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	ALTER	YES	YES	FAIL	FAIL	N	N
23 ☒	N/A [1]	N/A [1]	N/A [1]	ALTER	NONE	YES	YES	FAIL	FAIL	N	N
24 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	NONE	YES	YES	FAIL	FAIL	N	N
Create an additional file, MHLRES7.RACF.TEST2.FILEn, with different RACF access to MHLRES1.** and MHLRES5.**.											
25 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	YES	YES	FAIL	FAIL	N	N
26 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	ALTER	YES	YES	FAIL	FAIL	N	N
27 ☒	N/A [1]	N/A [1]	N/A [1]	ALTER	NONE	YES	YES	FAIL	FAIL	N	N
28 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	NONE	YES	YES	FAIL	FAIL	N	N
Read the previously created data sets with different RACF access to MHLRES1.** and MHLRES7.**.											

Test case/ success	RACF settings					TAPEAUTH settings in the DEVSUPnn				DFSMSrmm [2]	
	TAPEVOL	TAPEDSN	THM00n[4]	MHLRES1.**	MHLRES7.**	DSN	F1	RC4	RC8	TPRACF	RACF
29 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	YES	**NO**	FAIL	FAIL	N	N
30 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	ALTER	YES	**NO**	FAIL	FAIL	N	N
31 ☒	N/A [1]	N/A [1]	N/A [1]	ALTER	NONE	YES	**NO**	FAIL	FAIL	N	N
32 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	NONE	YES	**NO**	FAIL	FAIL	N	N
Create an additional file, MHLRES1.RACF.TEST2.FILEn, with different RACF access to MHLRES1.** and MHLRES5.*.											
33 ☑	N/A [1]	N/A [1]	N/A [1]	ALTER	ALTER	YES	**NO**	FAIL	FAIL	N	N
34 ☑	N/A [1]	N/A [1]	N/A [1]	NONE	ALTER	YES	**NO**	FAIL	FAIL	N	N
35 ☒	N/A [1]	N/A [1]	N/A [1]	ALTER	NONE	YES	**NO**	FAIL	FAIL	N	N
36 ☒	N/A [1]	N/A [1]	N/A [1]	NONE	NONE	YES	**NO**	FAIL	FAIL	N	N

[1] N/A means that this function is not activated at this time or that no TAPEVOL profile exists.
[2] The DFSMSrmm settings are stored in the current active DFSMSrmm EDGRMMnn PARMLIB member.
In this table we show only the tape security related to DFSMSrmm operands that we modified for our test cases.
TPRACF is an operand of the OPTION command and RACF is an option of the VLPOOL command.
[3] "A" is the abbreviation of the TPRACF(AUTOMATIC) processing.
[4] THM001 is the TAPEVOL profile protecting the volume that we used to create the two data sets for test cases 1 to 19.

9.5.1 Test case 1

In test case 1:

Function Create two new tape sets on a volume in status scratch. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors and DFSMSrmm has created a new RACF TAPEVOL profile for volume THM001, and the user MHLRES5 has ALTER access to it.

Example 9-10 shows you the JCL and the settings that we used to create the two data sets.

Example 9-10 Sample JCL used for test case 1

```
//RACFTS01 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*    ***************************************************************
//*    * /F DFRMM,M=R2                                               *
//*    ***************************************************************
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
```

```
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES1.RACF.TEST1.FILE1 NONVSAM NOSCRATCH
 DELETE MHLRES7.RACF.TEST1.FILE2 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD   DISP=(,CATLG),DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,99),
//         UNIT=ATL3
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(02,SL),DSN=MHLRES7.RACF.TEST1.FILE2,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=(,RETAIN,,REF=*.STEP01.SYSUT2)
//SYSIN    DD   DUMMY
```

Example 9-11 shows you the most important results of the RACF commands.

Example 9-11 RACF command results of test case 1

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED
```

```
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES5.**') ALL
INFORMATION FOR DATASET MHLRES5.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES5         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.2 Test case 2

For test case 2:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors because the user has ALTER access to the TAPEVOL profile and to the two profiles in class DATASET.

Example 9-12 shows you the JCL and the settings that we used to read the two data sets.

Example 9-12 Sample JCL used for test 2

```
//RACFTS02 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**          ACC(ALTER)
//*        MHLRES5.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        THM001              ACC(ALTER)
//*        TAPEVOL             ACTIVE
//*        TAPEDSN             ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACC(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
```

```
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF03)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-13 shows you the most important results of the RACF commands.

Example 9-13 RACF commands result of test case 2

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
MHLRES5   ALTER       000000
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM
```

```
NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.3 Test case 3

For test case 3:

Function Read the two previously created tape sets, but in this case the user has no access to the first data set. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors because the user has ALTER access to the TAPEVOL profile and to the second data set. This is only possible because after RACF has checked the access to the data set profile it checks the access to the TAPEVOL profile. If one of two checks ended with a return code zero, the access is allowed.

Example 9-14 shows you the JCL and the settings that we used to read the two data sets.

Example 9-14 Sample JCL used for test 3

```
//RACFTS03 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
```

```
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACC(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF04)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-15 shows the most important results of the RACF commands.

Example 9-15 RACF commands result of case 3

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
MHLRES5   ALTER       000000
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM
```

```
NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.4 Test case 4

For test case 4:

Function Read the two previously created tape sets, but in this case the user has no access to the TAPEVOL profile. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors because the user has ALTER access to the data set profiles and there is no check for the access to the TAPEVOL profile.

Example 9-16 shows you the JCL and the settings that we used to read the two data sets.

Example 9-16 Sample JCL used for test 4

```
//RACFTS04 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**          ACC(ALTER)
//*        MHLRES5.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        THM001              ACC(NONE)
//*        TAPEVOL             ACTIVE
//*        TAPEDSN             ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(NONE)
```

```
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF05)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-17 shows you the most important results of the RACF commands.

Example 9-17 RACF commands of test case 4

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              NONE     NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
MHLRES5   NONE        000000
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
```

```
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

9.5.5 Test case 5

For test case 5:

Function Read the two previously created tape sets, but in this case the user has no access to the TAPEVOL profile and no access to the first file. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result In this case we get a security violation for the first data set because we have no access to the data set profile nor to the TAPEVOL profile. The second data set can be read without any errors.

Example 9-16 on page 363 shows you the JCL and the settings that we used to read the two data sets.

Example 9-18 Sample JCL used for test 5

```
//RACFTS05 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
```

```
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(NONE)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(NONE)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF06)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-19 shows the most important results of the RACF commands.

Example 9-19 RACF commands of test case 5

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE               NONE    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
MHLRES5   NONE        000000
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM
```

```
NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-20 you can see that we get a security violation for file one.

Example 9-20 Test case 5 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  THM001 CL(TAPEVOL )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST1.FILE1 CL(DATASET ) VOL(THM001)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-60,IFG0194F,RACFTS05,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST1.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000060
 TIME=18.11.17  SEQ=00982  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000060
 END OF SYMPTOM DUMP
IEF472I RACFTS05 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000060
....
```

9.5.6 Test case 6

For test case 6:

Function	Read the two previously created tape sets, but in this case the RACF CLASS TAPEVOL is inactive. Program IEBGENER is used to read the files.
Data set names	MHLRES1.RACF.TEST1.FILE1 MHLRES7.RACF.TEST1.FILE1
Result	The job ended without any errors because the user has ALTER access to both data set profiles. The RACF TAPEVOL profile is not checked.

Example 9-21 shows you the JCL and the settings that we used to read the two data sets.

Example 9-21 Sample JCL used for test 6

```
//RACFTS06 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            NO ACCESS
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*           MHLRES1.RACF.TEST1.FILE1
//*           MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) DELETE
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
```

```
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF07)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-22 shows the most important results of the RACF commands.

Example 9-22 RACF commands of test case 6

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR TAPEDSN
WARNING: TAPEDSN OPTION ACTIVE, TAPEVOL CLASS IS NOT ACTIVE
READY

 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5          NONE             ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

            SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO
```

```
AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.7 Test case 7

For test case 7:

Function Read the two previously created tape sets, but in this case the RACF CLASS TAPEVOL is inactive and the user has no access to the first data set. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result In this case we get a security violation for the first data set because we have no access to the data set profile and the TAPEVOL profile is not checked because RACF CLASS TAPEVOL is inactive. The second data set can be read without any errors.

Example 9-23 shows you the JCL and the settings that we used to read the two data sets.

Example 9-23 Sample JCL used for test 7

```
//RACFTS07 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            NO ACCESS
//*        TAPEVOL           INACTIVE
```

```
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF08)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-24 shows the most important results of the RACF commands.

Example 9-24 RACF commands of test case 7

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR TAPEDSN
WARNING: TAPEDSN OPTION ACTIVE, TAPEVOL CLASS IS NOT ACTIVE
READY

 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5          NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
```

```
----      ------   ------ -----
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-25 you can see the security violation that we get for the first file, but in this case only for the data set profile.

Example 9-25 Test case 7 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST1.FILE1 CL(DATASET ) VOL(THM001)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-60,IFG0194F,RACFTS07,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST1.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000060
 TIME=18.13.50  SEQ=00983  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000060
 END OF SYMPTOM DUMP
IEF472I RACFTS07 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000060
....
```

9.5.8 Test case 8

For test case 8:

Function Read the two previously created tape sets, but in this case both RACF CLASSES TAPEVOL and TAPEDSN are inactive. Program IEBGENER is used to read the files. In this case you have all needed RACF access, and the job ended without any errors, as expected.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors because there are no security checks.

Example 9-26 shows you the JCL and the settings that we used to read the two data sets.

Example 9-26 Sample JCL used for test 8

```
//RACFTS08 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
```

```
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           INACTIVE
//*        THM001            NO ACCESS
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF09)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-27 shows the most important results of the RACF commands.

Example 9-27 RACF commands of test case 8

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR NOTAPEDSN

 RL TAPEVOL THM001 ALL
```

```
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-27 on page 375 you can see that there was no security violation.

9.5.9 Test case 9

For test case 9:

Function Read the two previously created tape sets. Both RACF CLASSES TAPEVOL and TAPEDSN are inactive and the user has no access to the data set profile that protects the second file. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE1

Result The job ended without any errors because there are no security checks.

Example 9-28 shows the JCL and the settings that we used to read the two data sets.

Example 9-28 Sample JCL used for test 9

```
//RACFTS09 JOB (999,POK),MSGLEVEL=1,
//         NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(A)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**          ACC(NONE)
//*        MHLRES5.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        THM001              NO ACCESS
//*        TAPEVOL             INACTIVE
//*        TAPEDSN             INACTIVE
//*
//*    FUNCTION
//*        READ THE PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 RL TAPEVOL THM001 ALL
```

```
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF10)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-29 shows the most important results of the RACF commands.

Example 9-29 RACF commands of test case 9

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR NOTAPEDSN

 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000001

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
```

```
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

9.5.10 Test case 10

For test case 10:

Function Create an additional third new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE3

Result The job ended without any errors because we have ALTER access to the RACF TAPEVOL profile for volume THM001 and the user MHLRES5 has ALTER access to the data set profile.

Example 9-30 shows you the JCL and the settings that we used to create the third data set.

Example 9-30 Sample JCL used for test 10

```
//RACFTS10 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
```

```
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST1.FILE3
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE3 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(03,SL),DSN=MHLRES7.RACF.TEST1.FILE3,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF11)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-31 shows the most important results of the RACF commands.

Example 9-31 RACF commands of test case 10

```
 RL TAPEVOL THM001
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.11 Test case 11

For test case 11:

Function Create an additional fourth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE4

Result In this case we get a security violation for the new fourth data set we would like to create because we have no access to the fourth data set profile.

Example 9-32 shows you the JCL and the settings that we used to create the third data set.

Example 9-32 Sample JCL used for test 11

```
//RACFTS11 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SETS
//*            MHLRES7.RACF.TEST1.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
```

```
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST1.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF12)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-33 shows the most important results of the RACF commands.

Example 9-33 RACF commands of test case 11

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-34 you can see that there was no security violation.

Example 9-34 Test case 11 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST1.FILE4 CL(DATASET ) VOL(THM001)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-60,IFG0194F,RACFTS11,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST1.FILE4
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000060
 TIME=08.00.54  SEQ=01017  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000060
```

```
 END OF SYMPTOM DUMP
IEF472I RACFTS11 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000060
....
```

9.5.12 Test case 12

For test case 12:

Function Repeat the creation of the additional fourth new tape sets on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE4

Result The job ended without any errors because we have access to the RACF TAPEVOL profile for volume THM00, although the user MHLRES5 has ALTER access to the data set profile. This access is not checked because RACF class TAPEDSN is inactive.

Example 9-35 shows you the JCL and the settings that we have to create the third data set.

Example 9-35 Sample JCL used for test 12

```
//RACFTS12 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**          ACC(ALTER)
//*        MHLRES5.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        THM001              ACC(ALTER)
//*        TAPEVOL             ACTIVE
//*        TAPEDSN             INACTIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST1.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE THM001CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
```

```
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST1.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF13)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-36 shows the most important results of the RACF commands.

Example 9-36 RACF commands of test case 12

```
 RL TAPEVOL THM001ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM
```

```
NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.13 Test case 13

For test case 13:

Function Create an additional fifth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE4

Result The job ended without any errors because the RACF option NOTAPEDSN class is being set and the user MHLRES5 has ALTER access to the TAPEVOL.

Example 9-37 shows the JCL and the settings that we have to create the third data set.

Example 9-37 Sample JCL used for test 13

```
//RACFTS13 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
```

```
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           INACTIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SETS
//*            MHLRES7.RACF.TEST1.FILE5
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE THM001CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE5 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*--------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(05,SL),DSN=MHLRES7.RACF.TEST1.FILE5,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF14)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-38 shows you the most important results of the RACF commands.

Example 9-38 RACF commands of test case 13

```
 RL TAPEVOL THM001ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
```

```
 OO    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 OO    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 OO    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA
```

SECU Example 9-39 on page 390 RITY LEVEL

```
----------------------------------------
NO SECURITY LEVEL
```

9.5.14 Test case 14

For test case 14:

Function Create an additional sixth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE6

Result The job ended without any errors because the user MHLRES5 has ALTER access to the data set profile and RACF class TAPEVOL is inactive.

Example 9-39 shows you the JCL we used.

Example 9-39 Sample JCL used for test 14

```
//RACFTS14 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           ACTIVE
//*        THM001           ACC(ALTER)
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SETS
//*            MHLRES7.RACF.TEST1.FILE6
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
```

```
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE6 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(06,SL),DSN=MHLRES7.RACF.TEST1.FILE6,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF15)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-40 shows the most important results of the RACF commands.

Example 9-40 RACF commands

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR TAPEDSN
WARNING: TAPEDSN OPTION ACTIVE, TAPEVOL CLASS IS NOT ACTIVE

 RL TAPEVOL THM001ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE            ALTER     NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
```

```
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.15 Test case 15

For test case 15:

Function Create an additional seventh new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library.

Data set names MHLRES7.RACF.TEST1.FILE7

Result In this case we get a security violation for the new seventh data set that we would like to create because we have no access to the data set profile and the RACF CLASS TAPEVOL is inactive.

Example 9-41 shows the JCL and the settings that we used to create the third data set.

Example 9-41 Sample JCL used for test 15

```
//RACFTS15 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(Y)
//*        RACF(Y)
//*
//*
//*    DEVSUP
```

```
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SETS
//*            MHLRES7.RACF.TEST1.FILE7
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST1.FILE6 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*--------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(07,SL),DSN=MHLRES7.RACF.TEST1.FILE7,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM001
//SYSIN    DD   DUMMY
```

Example 9-42 shows you the most important results of the RACF commands.

Example 9-42 RACF commands of test case 14

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR TAPEDSN
WARNING: TAPEDSN OPTION ACTIVE, TAPEVOL CLASS IS NOT ACTIVE
READY
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
READY
 SETR REFRESH RACLIST(TAPEVOL)
RACLIST REFRESH of class TAPEVOL ignored. The class is not active yet.
```

```
SETROPTS command complete.

 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM
```

```
NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-43 you can see that there was no security violation.

Example 9-43 Test case 15 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST1.FILE7 CL(DATASET ) VOL(THM001)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-60,IFG0194F,RACFTS15,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST1.FILE7
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000060
 TIME=08.49.02  SEQ=01037  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000060
 END OF SYMPTOM DUMP
IEF472I RACFTS15 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000060
....
```

9.5.16 Test case 16

For test case 16:

Function Read the six previously created tape sets, but in this case both RACF CLASSES TAPEVOL and TAPEDSN are inactive. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE2
MHLRES7.RACF.TEST1.FILE3
MHLRES7.RACF.TEST1.FILE4
MHLRES7.RACF.TEST1.FILE5
MHLRES7.RACF.TEST1.FILE6

Result The job ended without any errors because no security checks are being performed.

Example 9-28 on page 377 shows the JCL and the settings that we used to read the two data sets.

Example 9-44 Sample JCL used for test case 16

```
//RACFTS16 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*    ***************************************************************
//*    * /F DFRMM,M=02                                               *
//*    ***************************************************************
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           ACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ ALL PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*            MHLRES7.RACF.TEST1.FILE3
//*            MHLRES7.RACF.TEST1.FILE4
//*            MHLRES7.RACF.TEST1.FILE5
//*            MHLRES7.RACF.TEST1.FILE6
//*            MHLRES7.RACF.TEST1.FILE7
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//*--------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
```

```
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE3,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(3,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP04   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE4,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(4,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP05   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE5,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(5,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP06   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE6,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(6,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP07   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE7,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(7,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF17)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-45 shows the most important results of the RACF commands.

Example 9-45 RACF commands of test case 16

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
```

```
-----  --------   ----------------  -----------  -------
 OO    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 OO    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 OO    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

9.5.17 Test case 17

For test case 17:

Function Read the six previously created tape sets, but in this case RACF CLASS TAPEVOL is active and option TAPEDSN is being set. Program IEBGENER is used to read the files.

Data set names
MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE2
MHLRES7.RACF.TEST1.FILE3
MHLRES7.RACF.TEST1.FILE4
MHLRES7.RACF.TEST1.FILE5
MHLRES7.RACF.TEST1.FILE6

Result The job ended without any errors, though the user has no access to the data set profile MHLRES1.**. In this situation RACF checks that the user has UPDATE access to the TAPEVOL profile.

Example 9-46 shows the JCL and the settings that we used to read the two data sets.

Example 9-46 Sample JCL used for test 17

```
//RACFTS17 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**          ACC(NONE)
//*        MHLRES5.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        THM001              ACC(ALTER)
//*        TAPEVOL             ACTIVE
//*        TAPEDSN             ACTIVE
//*
//*    FUNCTION
//*        READ ALL PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*            MHLRES7.RACF.TEST1.FILE3
//*            MHLRES7.RACF.TEST1.FILE4
//*            MHLRES7.RACF.TEST1.FILE5
//*            MHLRES7.RACF.TEST1.FILE6
//*            MHLRES7.RACF.TEST1.FILE7
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR CLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
```

```
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE3,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(3,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP04   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE4,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(4,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP05   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE5,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(5,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP06   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE6,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(6,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP07   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE7,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(7,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
```

```
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF18)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-47 shows the most important results of the RACF commands.

Example 9-47 RACF commands of test case 17

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5          NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1          NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7          NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
```

```
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

9.5.18 Test case 18

For test case 18:

Function Read the six previously created tape sets, but in this case RACF CLASS TAPEVOL is inactive and option TAPEDSN is being set. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE2
MHLRES7.RACF.TEST1.FILE3
MHLRES7.RACF.TEST1.FILE4
MHLRES7.RACF.TEST1.FILE5
MHLRES7.RACF.TEST1.FILE6

Result The job ended without any errors because the user has access to the data set profile and the RACF TAPEVOL class is inactive.

Example 9-48 shows the JCL and the settings that we used to read the two data sets.

Example 9-48 Sample JCL used for test 18

```
//RACFTS18 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ ALL PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
```

```
//*            MHLRES7.RACF.TEST1.FILE2
//*            MHLRES7.RACF.TEST1.FILE3
//*            MHLRES7.RACF.TEST1.FILE4
//*            MHLRES7.RACF.TEST1.FILE5
//*            MHLRES7.RACF.TEST1.FILE6
//*            MHLRES7.RACF.TEST1.FILE7
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE3,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(3,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP04   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE4,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(4,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP05   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE5,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(5,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP06   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
```

```
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE6,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(6,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP07   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE7,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(7,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF19)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-49 shows you the most important results of the RACF commands.

Example 9-49 RACF commands of test case 18

```
 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE             ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

```
 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.19 Test case 19

For test case 19:

Function Read the six previously created tape sets, but in this case RACF CLASS TAPEVOL is inactive and option TAPEDSN is being set. Program IEBGENER is used to read the files.

Data set names MHLRES1.RACF.TEST1.FILE1
MHLRES7.RACF.TEST1.FILE2
MHLRES7.RACF.TEST1.FILE3
MHLRES7.RACF.TEST1.FILE4
MHLRES7.RACF.TEST1.FILE5
MHLRES7.RACF.TEST1.FILE6

Result We get a security violation because we have no access to the data set profile MHLRES1.** and the RACFCLASS TAPEVOL is inactive.

Example 9-50 shows the JCL and the settings that we used to read the two data sets.

Example 9-50 Sample JCL used for test 19

```
//RACFTS19 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=N/A
//*        TAPEAUTHF1=N/A
//*        TAPEAUTHRC4=N/A
```

```
//*        TAPEAUTHRC8=N/A
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        THM001            ACC(ALTER)
//*        TAPEVOL           INACTIVE
//*        TAPEDSN           ACTIVE
//*
//*    FUNCTION
//*        READ ALL PREVIOUSLY CREATED DATA SETS
//*            MHLRES1.RACF.TEST1.FILE1
//*            MHLRES7.RACF.TEST1.FILE2
//*            MHLRES7.RACF.TEST1.FILE3
//*            MHLRES7.RACF.TEST1.FILE4
//*            MHLRES7.RACF.TEST1.FILE5
//*            MHLRES7.RACF.TEST1.FILE6
//*            MHLRES7.RACF.TEST1.FILE7
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR TAPEDSN
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
 SETR REFRESH RACLIST(TAPEVOL)
 RL TAPEVOL THM001 ALL
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR REFRESH GENERIC(DATASET)
 SETROPTS LIST
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST1.FILE1,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE2,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE3,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(3,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP04   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE4,
```

```
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(4,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP05   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE5,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(5,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP06   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE6,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(6,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP07   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST1.FILE7,
//         VOL=(,RETAIN,,,SER=(THM001)),LABEL=(7,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
```

Example 9-51 shows you the most important results of the RACF commands.

Example 9-51 RACF commands

```
 SETR NOCLASSACT(TAPEVOL)
READY
 SETR TAPEDSN
WARNING: TAPEDSN OPTION ACTIVE, TAPEVOL CLASS IS NOT ACTIVE
READY
 PE THM001 CLASS(TAPEVOL) ID(MHLRES5) ACCESS(ALTER)
READY
 SETR REFRESH RACLIST(TAPEVOL)
RACLIST REFRESH of class TAPEVOL ignored. The class is not active yet.
SETROPTS command complete.

 RL TAPEVOL THM001 ALL
CLASS      NAME
-----      ----
TAPEVOL    THM001

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MHLRES5         NONE              ALTER    NO

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
STC       ALTER       000005
MHLRES5   ALTER       000000

 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO
```

```
AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-52 you can see that there was no security violation.

Example 9-52 Test case 19 job output

```
ICH70001I MHLRES5  LAST ACCESS AT 18:37:04 ON SUNDAY, AUGUST 5, 2007
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST1.FILE1 CL(DATASET ) VOL(THM001)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-60,IFG0194F,RACFTS19,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST1.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000060
 TIME=18.39.36  SEQ=00996  CPU=0000  ASID=0039
```

```
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000060
 END OF SYMPTOM DUMP
IEF472I RACFTS19 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000060
....
```

9.5.20 Test case 20

This test case and all following test cases are under the condition that we specified different settings of the new TAPEAUTH settings in the DEFSUPnn member and both RACF CLASSES TAPEVOL and TAPEDSN are inactive.

Function Create two new tape sets on a tape volume in status scratch. Program IEBGENER is used to write the files. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result The job ended without any errors because the user has access to the data set profiles.

Example 9-53 shows the JCL and the settings that we used to create the two data sets.

Example 9-53 Sample JCL used for test 20

```
//RACFTS20 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES5.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
```

```
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES5.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES5.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES1.RACF.TEST2.FILE1 NONVSAM NOSCRATCH
 DELETE MHLRES7.RACF.TEST2.FILE2 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD   DISP=(,CATLG),DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,99),
//         UNIT=ATL3
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(02,SL),DSN=MHLRES7.RACF.TEST2.FILE2,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=(,RETAIN,,REF=*.STEP01.SYSUT2)
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
```

9.5.21 Test case 21

For test case 21:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN and TAPEAUTHF1 are set and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result The job ended without any errors because the user has access to the data set profiles.

Example 9-54 shows you the JCL that we used for test case 21.

Example 9-54 Sample JCL used for test 21

```
//RACFTS21 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**          ACC(ALTER)
//*        MHLRES7.**          ACC(ALTER)
//*        TAPEVOL             INCATIVE
//*        TAPEDSN             INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
```

```
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF22)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-55 shows you the most important results of the RACF commands.

Example 9-55 Test case 21 job output

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
```

```
NO SECURITY LEVEL

CATEGORIES
```

9.5.22 Test case 22

For test case 22:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN and TAPEAUTHF1 are set and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result We got two security violations because the user has no access to the data set profile MHLRES1.**, which protects the first file on the volume. So we could not access any data set on the volume.

Example 9-56 shows the JCL that we used for test case 22.

Example 9-56 Sample JCL used for test 22

```
//RACFTS22 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**          ACC(NONE)
//*        MHLRES7.**          ACC(ALTER)
//*        TAPEVOL             INCATIVE
//*        TAPEDSN             INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
```

```
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF23)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-57 shows you the most important results of the RACF commands.

Example 9-57 List of the RACF profile in class TAPEVOL

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   NONE         SYS1           NON-VSAM


 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)
```

```
NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-28 on page 377 you can see the security violation for the first file we got.

Example 9-58 Test case 9 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS22,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST2.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.34.36  SEQ=00892  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS22 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS22,STEP02,SYSUT1,0B23,,MHLRES7.RACF.TEST2.FILE2
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.35.22  SEQ=00893  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
```

```
  AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
        2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
        4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
        6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
        8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
        A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
        C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
        E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS22 STEP02 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.23 Test case 23

For test case 23:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result Although we have access to the first file on the volume, we got a security violation because the user has no access to the data set profile MHLRES7.** to read the second file.

Example 9-59 shows the JCL that we used for test case 22.

Example 9-59 Sample JCL used for test 23

```
//RACFTS23 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
```

```
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*--------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF24)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-60 shows the most important results of the RACF commands.

Example 9-60 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
```

```
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-61 you can see the security violation we got for the second file.

Example 9-61 Test case 9 job output

```
...
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE2 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS23,STEP02,SYSUT1,0B23,,MHLRES7.RACF.TEST2.FILE2
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.36.09  SEQ=00894  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS23 STEP02 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.24 Test case 24

For test case 24:

Function	Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.
Data set names	MHLRES1.RACF.TEST2.FILE1 MHLRES7.RACF.TEST2.FILE2
Result	We got two security violations because the user has no access to the data set profiles MHLRES1.** and MHLRES7.**.

Example 9-62 shows the JCL that we used for test case 22.

Example 9-62 Sample JCL used for test 24

```
//RACFTS24 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*           MHLRES1.RACF.TEST2.FILE1
//*           MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
```

```
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF25)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-63 shows the most important results of the RACF commands.

Example 9-63 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1          NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
```

```
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-64 you can see that we got two security violations, one for file 1 and one for file 2.

Example 9-64 Test case 24 job output

```
...
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS24,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST2.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=10.40.55  SEQ=01052  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS24 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE2 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS24,STEP02,SYSUT1,0B23,,MHLRES7.RACF.TEST2.FILE2
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=10.41.42  SEQ=01053  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
```

```
          4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
          6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
          8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
          A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
          C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
          E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
....
```

9.5.25 Test case 25

For test case 25:

Function Create an additional third new tape set on the previously used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE3

Result The job ended without any errors because we have access to both data set profiles protecting the first file on the volume and the new file we are creating.

Example 9-65 shows you the JCL and the settings that we used to create the third data set.

Example 9-65 Sample JCL used for test 25

```
//RACFTS25 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE3
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES5.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
```

```
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE3 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(03,SL),DSN=MHLRES7.RACF.TEST2.FILE3,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF26)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-66 shows the most important results of the RACF commands.

Example 9-66 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

            SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
```

```
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.26 Test case 26

For test case 26:

Function Create an additional fourth new tape set on the previously used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE4

Result We got a security violation and the file is not created because we have no access to the RACF data set profile MHLRES1.** that protects the first file on the volume.

Example 9-67 shows the JCL that we used for test case 26.

Example 9-67 Sample JCL used for test 9

```
//RACFTS26 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
```

```
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST2.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF27)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-68 shows the most important results of the RACF commands.

Example 9-68 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED
```

```
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job shown in Example 9-69 you can see that we received a security violation to the data set profile MHLRES1.** that protects the first file on the volume.

Example 9-69 Test case 26 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS26,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST2.FILE4
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.39.27  SEQ=00897  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
```

```
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS26 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.27 Test case 27

For test case 27:

Function Recreate an additional fourth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE4

Result We got a security violation and the file is not created because we have no access to the RACF data set profile MHLRES7.** that protects the data set we would like to create.

Example 9-70 shows the JCL that we used for test case 26.

Example 9-70 Sample JCL used for test 9

```
//RACFTS27 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
```

```
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST2.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF28)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-71 shows you the most important results of the RACF commands.

Example 9-71 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
  ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)
```

```
LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-72 you can see that there was no security violation reading the first file, but we were not able to read the second file because we have no access to the profile that protects this file.

Example 9-72 Test case 27 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE4 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS27,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST2.FILE4
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.40.27  SEQ=00898  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS27 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.28 Test case 28

For test case 28:

Function Recreate an additional fourth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN and TAPEAUTHF1 are set, and the option for TAPEAUTHRC4 and TAPEAUTHRC8 is set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE4

Result We got a security violation because we have no access to the RACF data set profile MHLRES1.** that protects the first file on the volume nor to the profile MHLRES7.** that protects the data set we would like to create.

Example 9-73 shows the JCL that we used for test case 28.

Example 9-73 Sample JCL used for test 28

```
//RACFTS28 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*           MHLRES7.RACF.TEST2.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
```

```
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST2.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF29)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-74 shows the most important results of the RACF commands.

Example 9-74 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   NONE         SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)
```

```
NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-75 you can see that the new data set was not created.

Example 9-75 Test case 28 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE4 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS28,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST2.FILE4
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.41.27  SEQ=00899  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS28 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.29 Test case 29

For test case 29:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result The job ended without any errors because the user has access to both data set profiles protecting this files.

Example 9-76 shows the JCL that we used for test case 29.

Example 9-76 Sample JCL used for test 29

```
//RACFTS29 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**          ACC(ALTER)
//*        MHLRES7.***         ACC(ALTER)
//*        TAPEVOL             INCATIVE
//*        TAPEDSN             INCATIVE
//*
//*    FUNCTION
//*        Read the two DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
```

```
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF30)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-77 shows the most important results of the RACF commands.

Example 9-77 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.30 Test case 30

For test case 30:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result We got a security violation because the user has no access to the data set profile MHLRES1.**. The second file can be read without any errors.

Example 9-78 shows the JCL that we used for test case 30.

Example 9-78 Sample JCL used for test 30

```
//RACFTS30 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*            MHLRES1.RACF.TEST2.FILE1
//*            MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
```

```
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF31)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-79 shows the most important results of the RACF commands.

Example 9-79 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ---------------   -------   -----
 00    MHLRES1         NONE         NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
```

```
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-80 you can see that there was a security violation for the first file.

Example 9-80 Test case 30 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS30,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST2.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.47.20  SEQ=00901  CPU=0000  ASID=0039
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS30 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.31 Test case 31

For test case 31:

Function Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES1.RACF.TEST2.FILE1
MHLRES7.RACF.TEST2.FILE2

Result We got a security violation because the user has no access to the data set profile MHLRES7.**. The first file can be read without any errors.

Example 9-81 shows the JCL that we used for test case 31.

Example 9-81 Sample JCL used for test 31

```
//RACFTS31 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*           MHLRES1.RACF.TEST2.FILE1
//*           MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
```

```
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF32)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-82 shows the most important results of the RACF commands.

Example 9-82 Results of the RACF commands

```
  LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1          NON-VSAM

NO INSTALLATION DATA

            SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
```

```
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-83 you can see that we got a security violation for data set MHLRES7.RACF.TEST2.FILE2.

Example 9-83 Test case 31 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE2 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS31,STEP02,SYSUT1,0B23,,MHLRES7.RACF.TEST2.FILE2
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=16.49.21  SEQ=00902  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS31 STEP02 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.32 Test case 32

Test case 32:

Function	Read the two previously created tape sets. Program IEBGENER is used to read the files. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.
Data set names	MHLRES1.RACF.TEST2.FILE1 MHLRES7.RACF.TEST2.FILE2
Result	We got a security violation because the user has no access to both data set profiles MHLRES1.** and MHLRES7.**. No file can be read.

Example 9-84 shows the JCL that we used for test case 32.

Example 9-84 Sample JCL used for test 32

```
//RACFTS32 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE TWO NEW DATA SETS
//*           MHLRES1.RACF.TEST2.FILE1
//*           MHLRES7.RACF.TEST2.FILE2
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES1.RACF.TEST2.FILE1,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(1,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//STEP02   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RACF.TEST2.FILE2,
//         VOL=(,RETAIN,,,SER=(THM002)),LABEL=(2,SL)
//SYSUT2   DD   DUMMY
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
```

Example 9-85 shows you the most important results of the RACF commands.

Example 9-85 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-86 you can see that we got a security violation for both data set profiles protecting this files.

Example 9-86 Test case 32 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES1.RACF.TEST2.FILE1 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES1.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS32,STEP01,SYSUT1,0B23,,MHLRES1.RACF.TEST2.FILE1
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=11.52.12  SEQ=01054  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS32 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE2 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS32,STEP02,SYSUT1,0B23,,MHLRES7.RACF.TEST2.FILE2
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=11.53.03  SEQ=01055  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C380   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B4410   5: 00000000/00000000_007B47A4
         6: 00000000/00000000_007B474C   7: 00000000/00000000_007B47A4
         8: 00000000/00000000_007B476C   9: 00000000/00000000_007B56B8
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B46D0
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS32 STEP02 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.33 Test case 33

For test case 33:

Function Create an additional fourth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE5

Result The job ended without any errors because we have access to both data set profiles protecting the first file on the volume and the new file that we are creating.

Example 9-87 shows the JCL that we used for test case 33.

Example 9-87 Sample JCL used for test 33

```
//RACFTS33 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*           MHLRES7.RACF.TEST2.FILE4
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES5.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
```

```
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE4 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(04,SL),DSN=MHLRES7.RACF.TEST2.FILE4,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF34)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-88 shows you the most important results of the RACF commands.

Example 9-88 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE         NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

             SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)
```

```
NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.34 Test case 34

For test case 34:

Function Create an additional fifth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE5

Result The job ended without any errors, though we have no access to the first file on the volume. This is not checked because the option TAPEAUTHF1 is not set to yes.

Example 9-89 shows the JCL that we used for test case 34.

Example 9-89 Sample JCL used for test 34

```
//RACFTS34 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(ALTER)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE5
//*
//RACFCMDS EXEC PGM=IKJEFT01
```

```
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(ALTER) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE5 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(05,SL),DSN=MHLRES7.RACF.TEST2.FILE5,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF35)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-90 shows you the most important results of the RACF commands.

Example 9-90 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1           NON-VSAM

NO INSTALLATION DATA
```

```
 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

9.5.35 Test case 35

For test case 35:

Function Create an additional sixth new tape sets on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE5

Result We got a security violation and the file is not created because we have no access to the RACF data set profile MHLRES7.** that protects the data set that we would like to create.

Example 9-91 shows the JCL that we used for test case 35.

Example 9-91 Sample JCL used for test 35

```
//RACFTS35 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
```

```
//*        MHLRES1.**        ACC(ALTER)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE6
//*
//RACFCMDS EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(ALTER) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE6 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(06,SL),DSN=MHLRES7.RACF.TEST2.FILE6,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
//SUBMIT   EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(RACF36)
//SYSUT2   DD   SYSOUT=(A,INTRDR)
//SYSIN    DD   DUMMY
```

Example 9-92 shows you the most important results of the RACF commands.

Example 9-92 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)
```

```
NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
-----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-93 you can see that we got a security violation to data set profile MHLRES7.**.

Example 9-93 Test case 35 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE6 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS35,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST2.FILE6
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=17.20.50  SEQ=00905  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000   80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
```

```
  AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
        2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
        4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
        6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
        8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
        A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
        C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
        E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
IEF472I RACFTS35 STEP01 - COMPLETION CODE - SYSTEM=913 USER=0000 REASON=00000038
....
```

9.5.36 Test case 36

For test case 36:

Function Recreate the additional sixth new tape set on the previous used tape volume. Program IEBGENER is used to copy a member of a library. TAPEAUTHDSN is set to YES but TAPEAUTHF1 is set to NO. The options for TAPEAUTHRC4 and TAPEAUTHRC8 are set to FAIL.

Data set names MHLRES7.RACF.TEST2.FILE5

Result We got a security violation, and the file is not created because we have no access to the RACF data set profile MHLRES7.** that protects the data set that we would like to create. The first file protection is not checked because the TAPEAUTHF1 checking is not set.

Example 9-94 shows the JCL that we used for test case 36.

Example 9-94 Sample JCL used for test 36

```
//RACFTS36 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC70
//*
//*    RMM OPTIONS:
//*        TPRACF(N)
//*        RACF(N)
//*
//*
//*    DEVSUP
//*        TAPEAUTHDSN=YES
//*        TAPEAUTHF1=YES
//*        TAPEAUTHRC4=FAIL
//*        TAPEAUTHRC8=FAIL
//*
//*    RACF
//*        MHLRES1.**        ACC(NONE)
//*        MHLRES7.**        ACC(NONE)
//*        TAPEVOL           INCATIVE
//*        TAPEDSN           INCATIVE
//*
//*    FUNCTION
//*        CREATE AN ADDITIONAL NEW DATA SET
//*            MHLRES7.RACF.TEST2.FILE6
//*
//RACFCMDS EXEC PGM=IKJEFT01
```

```
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 SETR NOCLASSACT(TAPEVOL)
 SETR NOTAPEDSN
 PE 'MHLRES1.**' ACC(NONE) ID(MHLRES5)
 PE 'MHLRES7.**' ACC(NONE) ID(MHLRES5)
 SETR REFRESH GENERIC(DATASET)
 LD DATASET('MHLRES1.**') ALL
 LD DATASET('MHLRES7.**') ALL
 SETR LIST
/*
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE MHLRES7.RACF.TEST2.FILE6 NONVSAM NOSCRATCH
 SET MAXCC=0
/*
//*-------------------------------------------------------------*
//STEP01   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=MHLRES5.RACF.CNTL(TEXT)
//SYSUT2   DD LABEL=(06,SL),DSN=MHLRES7.RACF.TEST2.FILE6,
//         DISP=(,CATLG,DELETE),UNIT=ATL3,RETPD=02,
//         VOLUME=SER=THM002
//SYSIN    DD   DUMMY
//*-------------------------------------------------------------*
```

Example 9-95 shows the most important results of the RACF commands.

Example 9-95 Results of the RACF commands

```
 LD DATASET('MHLRES1.**') ALL
INFORMATION FOR DATASET MHLRES1.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES1         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE         SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL

 LD DATASET('MHLRES7.**') ALL
INFORMATION FOR DATASET MHLRES7.** (G)
```

```
LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    MHLRES7         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
    NONE        SYS1           NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
----------------------------------------
NO SECURITY LEVEL
```

In the output of the job in Example 9-96 you can see that we got a security violation for the data set profile MHLRES7.** only.

Example 9-96 Test case 36 job output

```
....
IEF237I DMY  ALLOCATED TO SYSIN
ICH408I USER(MHLRES5 ) GROUP(SYS1    ) NAME(MARY LOVELACE - RESI)
  MHLRES7.RACF.TEST2.FILE6 CL(DATASET ) VOL(THM002)
  INSUFFICIENT ACCESS AUTHORITY
  FROM MHLRES7.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194F,RACFTS36,STEP01,SYSUT2,0B23,,MHLRES7.RACF.TEST2.FILE6
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=913  REASON CODE=00000038
 TIME=17.21.52  SEQ=00906  CPU=0000  ASID=002C
 PSW AT TIME OF ERROR  075C1000  80C50EDE  ILC 2  INTC 0D
   NO ACTIVE MODULE FOUND
   NAME=UNKNOWN
   DATA AT PSW  00C50ED8 - 41003B7A  0A0D41F0  38BE56F0
   AR/GR 0: A3280FB6/00000000_00C511A4   1: 00000000/00000000_A4913000
         2: 00000000/00000000_0000C3E4   3: 00000000/00000000_00C5062A
         4: 00000000/00000000_007B51F8   5: 00000000/00000000_007B558C
         6: 00000000/00000000_007B5534   7: 00000000/00000000_007B558C
         8: 00000000/00000000_007B5554   9: 00000000/00000000_007B3328
         A: 00000000/00000000_007CE390   B: 00000000/00000000_00C53B2C
         C: 00000000/00000000_80C53C0C   D: 00000000/00000000_007B54B8
         E: 00000000/00000000_80C50762   F: 00000002/00000010_00000038
 END OF SYMPTOM DUMP
....
```

A

APAR text

This appendix contains the text of APARs that must be reviewed to successfully implement DFSMS V1.8.

APARs referenced in the book

OA17704

```
SMF record type 42 subtype 6 for above 2 GB have incorrect content
     ****************************************************************
* USERS AFFECTED: All VSAM RLS users who use RMF to view       *
*                 statistics.                                  *
****************************************************************
* PROBLEM DESCRIPTION: In RMF V1R8 VSAM LRU Overview SYSPLEX   *
*                      panel:                                  *
*                      (1)The following fields contain          *
*                      incorrect values:                       *
*                      Buffer Size Goal Below 2GB              *
*                      Buffer Size Goal Above 2GB              *
*                      Buffer Size High Below 2GB              *
*                      Fixed Storage Below 2GB                 *
*                      Fixed Storage Above 2GB                 *
*                      Real Storage % Below 2GB                *
*                      Real Storage % Above 2GB                *
*                      Fixed Pages Low Below 2GB               *
*                      Fixed Pages High Below 2GB              *
*                      Fixed Pages Avg Below 2GB               *
*                      Fixed Pages Low Above 2GB               *
*                      Fixed Pages High Above 2GB              *
*                      Fixed Pages Avg Above 2GB               *
*                      Buffer Counts by Pool Low/High/Avg      *
*                      2K to 32K Above 2GB                     *
*                                                              *
*                      (2)After restarting RMF address space   *
*                      or adding data masks via                *
*                      F RMF,F III,VSAMRLS(ADD(DSName)),       *
*                      customers stopped seeing any RMF data.  *
*                                                              *
*                      (3)When RlsAboveThebarMaxPoolSize is 0, *
*                      customers stop seeing changes to the    *
*                      following fields:                       *
*                      Buffer Size Goal Above 2GB              *
*                      Fixed Storage Above 2GB                 *
*                      Real Storage % Above 2GB                *
*                      Fixed Pages Low Above 2GB               *
*                      Fixed Pages High Above 2GB              *
*                      Fixed Pages Avg Above 2GB               *
*                      Buffer Counts by Pool Low/High/Avg      *
*                      2K to 32K Above 2GB                     *
*                                                              *
****************************************************************
* RECOMMENDATION:                                              *
****************************************************************
The SMF fields for these RMF fields were not correct, causing
RMF to report incorrect values.
```

OA11708

Support for the SMS,SMSVSAM,TERMINATESERVER command.

Note: The fix for OA13332 must be installed before OA11708, and the fix for OA14666 must be installed after OA11708.

```
************************************************************
* USERS AFFECTED: All HDZ11GO users and above issuing the  *
*                 V SMS,SMSVSAM,TERMINATESERVER command    *
*                 while sharing systems are issuing lock   *
*                 requests for lock structure IGWLOCK00.   *
************************************************************
* PROBLEM DESCRIPTION: When the V SMS,SMSVSAM,TERMINATESERVER *
*                      command is issued, sharing systems may *
*                      see delays for lock requests issued *
*                      against the RLS lock structure      *
*                      IGWLOCK00.  The delays occur during *
*                      the XCF lock table cleaup phase.    *
************************************************************
* RECOMMENDATION:                                          *
************************************************************
When the V SMS,SMSVSAM,TERMINATESERVER command is issued, the
SMSVSAM address space terminates without releasing active
system locks in IGWLOCK00.  When a lock structure connector
disconnects and active locks exist, XCF will perform a lock
table scan by the active connectors to cleanup the failing
connectors locks.  During the lock table scan, the lock
structure is quiesced to all sharing systems, resulting in
delays and lengthed transaction times (increased FCVRWAIT
times for CICS users).

Problem
Conclusion:


The V SMS,SMSVSAM,TERMINATESERVER command has been modifed to
release all system related locks prior to explicitly issuing
an IXLDISC for IGWLOCK00.  If the installation has also closed
all RLS data sets and terminated all registered RLS client
address spaces, then the XCF lock table scan will be skipped.
If however, any active locks exist when the TERMINATESERVER
command is issued, new message MSGIGW413I will be issued:

IGW413I  SMSVSAM DISCONNECTED FROM LOCK STRUCTURE IGWLOCK00
         A NON-ZERO RETURN/REASON CODE WAS RETURNED FROM
         IXLDISC:
         RETURN CODE (in hex): returncode
         REASON CODE (in hex): reasoncode

         Explanation: A V SMS,SMSVSAM,TEMRINATESERVER
         command was issued to terminate the SMSVSAM address
         space. SMSVSAM issued an IXLDISC to disconnect
         from lock structure IGWLOCK00. A non-zero return/
```

```
reason code was returned from IXLDISC.

Source: DFSMSdfp

Detecting Module:
IDAVSTAI

System Action: SMSVSAM termination continues.
Refer to the z/OS MVS Sysplex Services Refrence
manual for a description of the return/reason
code returned by IXLDISC.

Operator Response: RC=4 RSN=xxxx0401
(IXLRSNCODEOWNINGRESOURCES), is expected if any
data sets are opened for RLS access or there are
active RLS client spaces registered with SMSVSAM.
If all data sets are closed and all RLS client
spaces are ended, and RSN=xxxx0401 is
received, collect LOGREC data and report the
problem to IBM. For any other non-zero return/
reason code, report the problem to IBM.
```

OA13332

```
     ****************************************************************
* USERS AFFECTED: All HDZ11G0 users and above installing       *
*                 RLS APAR OA11708.                            *
****************************************************************
* PROBLEM DESCRIPTION: Add sysplex lock contention support     *
*                      for APAR OA11708.                       *
****************************************************************
* RECOMMENDATION:                                              *
****************************************************************
RLS APAR OA11708 provides new support to the V SMS,SMSVSAM,
TERMINATESERVER command.  The new support will modify the
TERMINATESERVER command to release all "system" type locks
in IGWLOCK00.  If a rebuild of IGWLOCK00 is started at the
same time as a TERMINATESERVER command is issued, the current
RLS lock contention exits will deny the lock release requests
before the rebuild path serializes with the TERMINATESERVER
path.  The "system" lock contention exit must be modified
to allow the lock releases to continue until the rebuild
quiesce phase is entered and properly serialzes with the
TERMINATESERVER.

Problem
Conclusion:


RLS special lock contention exit has been modified to allow
contention processing to continue until the RLS rebuild quiesce
phase completes.  This APAR must be installed on all systems
in the plex before OA11708 is installed on any one system.
```

OA14666

```
*****************************************************************
* USERS AFFECTED: All HDZ11G0 users and above with OA11708     *
*                 installed and attempting to issue MVS        *
*                 command V SMS,SMSVSAM,TERMINATESERVER for    *
*                 multiple images at the same time while       *
*                 contention for record locks exist.           *
*****************************************************************
* PROBLEM DESCRIPTION: The V SMS,SMSVSAM,TERMINATESERVER       *
*                      command did not complete on system1     *
*                      and system1 did not reply to the        *
*                      disconnect event for IGWLOCK00          *
*                      from system2.  MSGIXL041E is displayed. *
*****************************************************************
* RECOMMENDATION:                                              *
*****************************************************************
APAR OA11708 added logic to the TERMINATESERVER path to
serialize all lock activity prior to releasing system related
locks and explicitly disconnecting from IGWLOCK00.  If two or
more systems issue the V SMS,SMSVSAM,TERMINATESERVER command
on systems with OA11708 installed and record lock contention
exists, then one SMSVSAM may hang in the terminateserver path
after obtaining record lock latches and waiting on contention
to complete for lock releases of system related locks.  The
contention is not completing because XCF has not completed
lock table cleanup activities.  XCF has not completed lock
table cleanup because the hung SMSVSAM has not replied to
the disconnect event for IGWLOCK00 from the other terminating
server(s).  Therefore the terminateserver path is deadlocked.
To recover from the hung SMSVSAM in the TERMINATESERVER path,
issue the FORCE SMSVSAM,ARM command.

Problem
Conclusion:


The RLS locking code has been modified to not obtain record lock
latches in the disconnect cleanup path when this SMSVSAM is
also terminating.  The terminateserver path already holds the
necessary record lock serialization.
```

OA16191

```
USERS AFFECTED:
All users of DFSMShsm V1R3, V1R5, V1R6, V1R7 and V1R8 who use the DFSMShsm
TAPECOPY and BACKVOL CDS commands. This also affects installations that backup
their control data sets during AUTOBACKUP processing.
PROBLEM DESCRIPTION:
Installations that perform TAPECOPY or CDS backup processing can encounter an
ABEND878 during these processes.
RECOMMENDATION:
Potential ABEND878 abends can occur during TAPECOPY and CDS backup processing if
below the line storage is exhausted in the DFSMShsm address space.
```

Problem Conclusion:
Temporary Fix:
Comments:
DFSMShsm is enhancing TAPECOPY to allow both the tape input and tape output buffers to optionally be above the 16MB line. DFSMShsm is also enhancing CDS BACKUP to allow the output buffers to be above the 16MB line. To request the TAPECOPY and CDS backup buffers be above the line issue the following DFSMShsm PATCH command:
PATCH .MCVT.+196 BITS(...1....).
To request that DFSMShsm return to using buffers below the 16MB line for TAPECOPY and CDS Backup, issue the following DFSMShsm PATCH command:
PATCH .MCVT.+196 BITS(...0....).
Modules/Macros:
ARCCATBS ARCCPCTS ARCCVT
SRLS:
NONE

OA16192

REDUCE VIRTUAL STORAGE CONTRAINT RESULTING IN ABEND878 BELOW THE LINE BY MOVING DFSMSDSS TO OWN ADDRESS SPACE FOR MOST FUNCTIONS.
Error Description:
New Function
Local Fix:
Responder Page
Problem Summary:
Problem Conclusion:
Temporary Fix:
Apply AA16192
Comments:
This support allows installations to request that DFSMShsm invoke DFSMSdss via its cross-memory application interface.
To request that DFSMShsm invoke DFSMSdss via its cross-memory application interface installations must issue the following DFSMShsm patch command:
PATCH .MCVT.+433 X'FF'
Turning the patch ON (X'FF') to request DFSMSdss cross-memory processing, the patch can only be issued from the DFSMShsm startup command member (ARCCCMDnn). Attempts to patch this field to a value of X'FF' outside of DFSMShsm startup initialization will fail with message MSGARC0235I.
Installations which later decide they do not wish to invoke DFSMSdss via its cross-memory application interface, must issue the DFSMShsm patch command to patch the MCVT+433 to any value other than X'FF'.
For example, PATCH .MCVT.+433 X'00'.
Patching this field to any value other than X'FF' may be done at any time. Issuing the patch outside of DFSMShsm startup initialization is acceptable if the value being patched is something other than X'FF'. Otherwise, the command will fail with message MSGARC0235I.
This enhancement introduces 1 new DFSMShsm message MSGARC1017I and changes 1 existing message, ARC0235I.
The following changes should be made to the MVS System Messages Vol 2 - SA22-7632-10
ARC1017I- LOAD OF ADRXMAIA FAILED. THE CROSS-MEMORY INVOCATION
OF DFSMSDSS DISABLED FOR MIGRATION, BACKUP, RECOVER,
DUMP, RESTORE AND CDS BACKUP

Explanation: The load of ADRXMAIA failed. Cross-memory invocation of DFSMSdss in disabled for migration, backup, recover, dump, restore and CDS backup
Source: DFSMShsm
System Action: DFSMShsm processing continues. DFSMSdss will not be invoked using its cross-memory application interface.
Application Programmer Response: Determine why the load failed.
The most likely cause is that ADRXMAIA cannot be found in the load library concatentation.
Programmer Response: None
Operator Response: None
The changed message:
ARC0235I CANNOT PATCH PROTECTED DATA
Explanation: One of the following has occurred:
An attempt was made to patch data that is located in protected fields.
An attempt was made to patch a field in the DFSMShsm MCVT record to request that DFSMShsm invoke DFSMSdss in cross-memory mode for migration, backup, recovery, dump, restore and CDS backup processing. Requests to patch a value of X'FF' in this field are only permitted during DFSMShsm startup initialization processing. Requests to patch this field to any other value, preventing further DFSMSdss cross-memory invocations, is permitted outside of DFSMShsm startup initialization.
System Action: The command ends with no data modified. DFSMShsm processing continues.
Application Programmer Response: If the installations intent is to request DFSMShsm to invoke DFSMSdss in cross-memory mode for migration, backup, recovery, dump, restore and CDS backup processing, the patch command will complete successfully if placed in the DFSMShsm startup member of the SYS1.PARMLIB(ARCCMDxx).
The following chapter should be added to the DFSMShsm Implementation and Customization Guide
DFSMSdss Address Space IDs used by DFSMShsm
When DFSMShsm invokes DFSMSdss via the DFSMSdss cross-memory application interface, DFSMShsm will request that DFSMSdss use a unique address space identifier for each unique DFSMShsm function and host id. The address space identifier for each function will be in the format of "DFHnXXXX", where 'n' will represent the unique DFSMShsm host id and 'XXXX' will represent the DFSMShsm function. The following functional abbreviations will be used for 'XXXX'
Dump - "DUMP"
Restore = "REST"
Migration = "MIGR"
Backup = "BACK"
Recover = "RCVR"
CDS Backup = "CDSB"
For instance, migration for HSM hostid=1 would result in a generated address space identifier of DFH1MIGR.
Fast Replication Backup and Restore will also generate unique DFSMSdss address space identifiers but the format of those names will be in the format of "DSSFRftt" where "f'= a functin identifier of "B" for backup or "R" for recover. The "tt"=task number, which will be a character value between 1 and 64.
Modules/Macros:
SRLS:
Return Codes:
Circumvention:
Message To Submitter:

OA18319

This is a problem with DFDSS dropping SMS RLS attributes from data sets under certain circumstances if the volumes that the data set is on is specified.

Description:

A VSAM KSDS with data and index components on different volumes may lose its RLSDATA during DSS COPY. The problem reported came about because input volumes were specified on a LOGINDYNAM parameter with COPY statemets similar to the following:

```
COPY DATASET(INCLUDE(DATASET.KSDS)) -
 LOGINDYNAM( -
 (VOLUM1) -
 (VOLUM2) -
) -
 DELETE CATALOG -
 SPHERE
```

When the data component resides on VOLUM2 and the index component on VOLUM1, the the RLS data is not propogated on COPY. If the order of the volumes in the LOGINDYNAM is swapped then the target data set will retain all of its RLS information.
Local Fix:
Do not specify input volumes.

OA18465

PTF List:
Release 180 : UA29904 available 06/10/13 (F610)
Parent APAR:
Child APAR List:

Submitter

Page
Error Description:
Provide relief for DFSMShsm v1r8 recycle data loss APAR OA18372.

Local Fix:
Hold RECYCLE and mark any partial ML2 and backup tapes full using the DELVOL volser MIGRATION(MARKFULL) or DELVOL volser BACKUP(MARKFULL) command, as appropriate.
A TOOL can be downloaded to determine if tapes have been impacted with the error reported in this APAR. See the README member contained in the tool for details.
FTP server: testcase.boulder.ibm.com
directory : /mvs/fromibm
filename : OA18372.ANALYSIS.TOOL.TR <== case-sensitive
mode : BINARY
Note: This tool is a tersed PDS data set which contains DFSORT JCL. The data set must be downloaded and untersed to a PDS output data set using TRSMAIN.
Downloading the tool:
1) Allocate a local ds with attributes of a TERSED data set:
DCB=(RECFM=FB,LRECL=1024,BLKSIZE=6144,DSORG=PS)

```
2) Log on to FTP server (user: ANONYMOUS, password: email)
   Then issue the following sequence of commands:
     cd mvs/fromibm
     bin
     get OA18372.ANALYSIS.TOOL.TR 'local.ds.name' (replace
     quit
3) Unterse the local file using job below:
    //STEP1 EXEC PGM=TRSMAIN,PARM=UNPACK
    //SYSPRINT DD SYSOUT=*
    //INFILE DD DISP=SHR,
    //         DSNAME=%indsname
    //OUTFILE DD SPACE=(TRK,(10,10,20),RLSE),
    //         VOL=(,,,1),DISP=(NEW,CATLG),UNIT=SYSDA,
    //         DSNAME=%outdsname
               Responder Page
                                                        Problem Summary:
USERS AFFECTED:
All V1R8 DFSMShsm users of RECYCLE.
PROBLEM DESCRIPTION:
Relief for OA18372.
RECOMMENDATION:
Relief for OA18372.
Problem Conclusion:
DFSMShsm RECYCLE processing will call error-processing module ARCERP to abnormally
terminate under the conditions that would otherwise result in the loss of data
described in APAR OA18372.
Although the recycle function may terminate abnormally under certain error
conditions, DFSMShsm V1R8 customers can resume normal recycle processing without
fear of data loss.
OA18372 will provide a complete resolution to the problem that will eliminate the
deliberate abends added by OA18465.  Should any of the following symptoms occur
after OA18465 is installed, the customer is requested to notify IBM Software
Support so that the error conditions leading to the abend can be analyzed.
The following symptoms will be observed when the abend occurs:
For recycle of a Migration Tape:
ARC0200I TRAP IN MODULE ARCRCYPQ, CODE=0007, ABEND ONCE ADDED
ARC0900I DFSMSHSM ERROR CODE 0007 IN MODULE ARCRCYPQ TYPE ABEND
ARC0003I ARCRCYVS TASK ABENDED, CODE 800003EF IN MODULE ARCERP
         AT OFFSET 04B0, STORAGE LOCATION=xxxxxxxx
ARC0834I RECYCLE TASK xxxx ENDING. RC=0023, REAS=000
The resulting dump will have the following title:
ARCRCYVS TASK ABENDED CODE=800003EF IN MODULE ARCERP AT
   AT OFFSET=04B0, STORAGE LOCATION=xxxxxxxx
For recycle of a Backup Tape:
ARC0200I TRAP IN MODULE ARCRCYPQ, CODE=0005, ABEND ONCE ADDED
ARC0900I DFSMSHSM ERROR CODE 0005 IN MODULE ARCRCYPQ TYPE ABEND
ARC0003I ARCRCYVS TASK ABENDED, CODE 800003ED IN MODULE ARCERP
         AT OFFSET 04B0, STORAGE LOCATION=xxxxxxxx
ARC0834I RECYCLE TASK xxxx ENDING. RC=0023, REAS=000
The resulting dump will have the following title:
ARCRCYVS TASK ABENDED CODE=800003ED IN MODULE ARCERP AT
   AT OFFSET=04B0, STORAGE LOCATION=xxxxxxxx
New return codes 5 and 7 will be added to Table 1 in the DFSMShsm Diagnosis manual
(GC521083) for ARCRCYPQ.
Additional keywords: MSGARC0200I MSGARC0208I MSGARC0900I RC5 RC7
```

```
                    MSGARC0003I MSGARC0834I RC23 RC0 REAS0 RSN0
                    ABENDU3EF ABENDU3ED ABEND3EF ABEND3ED
                    ABENDU1005 ABENDU1007 ABEND1005 ABEND1007
Temporary Fix:
*********
 HIPER
*********
Comments:
Modules/Macros:
ARCRCYPQ ARCTEOV
SRLS:
GC52108301
Return Codes:
Circumvention:
Message To Submitter:
```

OA19493

There is a problem under circumstances with use of the DFSMShsm migration fast reconnect. Certain options, which happen to be used by a well known data manipulation product can result in a recalled data set that is updated not being recognised as needing a new migration, and being reconnected to the prior version of the data.

```
Description:
   DS1IND08 is not updated when a DSORG=PS data set is recalled,
opened INOUT, updated (by appending records to the end), and
successfully closed.   DS1RECAL and DS1DSCHA are ON.
*****************************************************************
* USERS AFFECTED: All using DFHSM Fast Subsequent Migrate      *
*                 function available in z/OS1.7.               *
*****************************************************************
* PROBLEM DESCRIPTION: HSM Fast Subsequent Migrate function    *
*                      fails to detect data set was modified   *
*                      on recall. This could potentially       *
*                      cause a loss of data.                   *
*****************************************************************
* RECOMMENDATION:                                              *
*****************************************************************
OPEN processing for BSAM INOUT did not set output indicator
DS1IND08 in the F1DSCB. After an OPEN for INOUT followed
by a CLOSE TYPE=T, the data set is written to. The result
is the HSM FSM function incorrectly reconnects to the
original migrated copy.
```

OA20170

There is a problem with the CBRSMF macro which does not describe the offset to the ST38FLGs field correctly due to some missing fields in the definition.

```
Description:
   Field ST38RCLD at +x'74' is missing and some comments are
incomplete in the mapping macro for OAM type 85 subtype 38
in SYS1.MACLIB(CBRSMF). Subtype 38 is for OSMC SINGLE OBJECT
RECALL UTILITY.
```

```
Local Fix:
   none
```

OA20242

```
Message ADR146I does not get issued when using INDDNAME/INDYNAM for a logical
COPY.

Error.
Description:

In DFSMSdss release 1.8 the INDDNAME and INDYNAM keywords are
obsolete for data set COPY and DUMP functions. ADR146I should be
issued when these kewords are specified. This message IS
issued for data set DUMP with INDDNAME, but is NOT issued for
data set DUMP with INDYNAM or data set COPY with either INDDNAME
or INDYNAM.

This is the message text for a data set  DUMP with INDDNAME:
ADR146I (R/I)-RIO3 (13), OBSOLETE KEYWORD 'INDDNAME ' SPECIFIED.
'PHYSINDDNAME ' WILL BE USED.
```

OA09928

```
Error Description:
New function to enhance migration duplex performance with D/T3592-J tape drives.
Local Fix:
Problem Summary:
USERS AFFECTED:
Users of DFSMShsm's migration function when duplexing to 3592-J tape drives.  The
affected z/OS levels are V1R3-V1R6.
PROBLEM DESCRIPTION:
Performance suffers when DFSMShsm migration duplexes to the 3592-J tape drive.
RECOMMENDATION:
Migration elapsed times rise dramatically when SYNCDEVs are done on two 3592-J
tape drives simultaneously.  The effect is most noticeable when migrating small
data sets.
Problem Conclusion:
Temporary Fix:
 HIPER

Comments:
Duplex migration to D/T3592 -J tape drives is significantly enhanced when SYNCDEVs
are not done for the alternate tape.
Customers can now suppress syncs on the alternate tape during duplex migration.
Syncs for the migration alternate tape can be turned off via:
PATCH .MCVT.+196 BITS(..1.....)
and back on via:
PATCH .MCVT.+196 BITS(..0.....)
The default is to do syncs on the alternate tapes.
This patch is not recommended for earlier technology tape drives such as the 3490
and 3590.
```

Prior APAR OW45264 describes a similar patch for turning off alternate tape syncs for Recycle. This can also dramatically improve performance with the 3592 -J tape drive.
Search keywords D/T3592 3592J 3592J1A 3592-J1A .

OW45264

Error Description:
AE fix continuation OW43224 UW71243 UW71244 UW71245   Recycle takes too long for D/T3590 drives.
Local Fix:
Problem Summary:
USERS AFFECTED:
All DFHSM Recycle users of Duplex tape.
PROBLEM DESCRIPTION:
Recycle takes too long for D/T3590 drives.
RECOMMENDATION:
Recycle processing for Duplex tapes performs a SYNCDEV for
Problem Conclusion:
DFSMShsm issues a SYNCDEV after each successful data set is processed for both the primary and alternate output tapes. This increases the elapsed time Recycle will take to process in a duplex tape environment.
Bypassing the SYNCDEV for the duplex alternate tape during Recycle is now supported. This reduces the time Recycle takes by bypassing the SYNCDEV of the duplex alternate tape after each data set. To bypass the alternate tapes SYNCDEV operation, enter the following PATCH command for your release:
For releases HDZ11D0 AND HDZ11E0:
PATCH .YGCB.+88 BITS(......1.)
For releases HDZ11F0, HDZ11G0, HDZ11H0, HDZ11J0 & HDZ11K0:
PATCH .YGCB.+C8 BITS(......1.)
To preserve the PATCH when HSM is restarted, place the PATCH command in the HSM startup parmlib member ARCCMDxx.
To remove the patch and allow the SYNCDEV to be again occur on the alternate volume, enter the following PATCH command for your release and remove the previous PATCH from ARCCMDxx:
For releases HDZ11D0 AND HDZ11E0:
PATCH .YGCB.+88 BITS(......0.)
For releases HDZ11F0, HDZ11G0, HDZ11H0, HDZ11J0 & HDZ11K0:
PATCH .YGCB.+C8 BITS(......0.)
If you had the following PATCH installed from APAR OW43224:
PATCH .MCVT.+295 BITS(....1...)
it should be removed at this time.

OW43224

Error Description:
Recycle takes too long for D/T3590 drives.
Local Fix:

Problem Summary:
USERS AFFECTED:
All DFHSM Recycle users of Duplex tape.

PROBLEM DESCRIPTION:
Recycle takes too long for D/T3590 drives.

RECOMMENDATION:
Recycle processing for Duplex tapes performs a SYNCDEV for both the primary and alternate tape after each dataset.

Problem Conclusion:
DFSMShsm issues a SYNCDEV after each successful dataset is processed for both the primary and alternate output tapes.
This increases the elapsed time Recycle will take to process in a duplex tape environment. Bypassing the SYNCDEV for the duplex alternate tape during Recycle is now supported. This reduces the time Recycle takes by bypassing the SYNCDEV of the duplex alternate tape after each dataset. To bypass the alternate tapes SYNCDEV operation, a new HSM PATCH may be issued.
See APAR OW45264 for the latest PATCH information.

OA17734

Problem Summary:
   USERS AFFECTED:
Users of VSAM RLS.
PROBLEM DESCRIPTION:
This Health Check is the interface to VSAM RLS Diagnostic Contention console command:
                              D SMS,SMSVSAM,DIAG(CONTENTION)
RECOMMENDATION:
This APAR implements VSAM RLS HealthCheck IBMVSAMRLS - VSAMRLS_DIAG_CONTENTION.
Problem Conclusion:
Publication
"z/OS V1R8.0 MVS System Messages, Vol 9 (IGF-IWM)"
(SA22-7639-12) Chapter 6.0 has been supplemented
with msgs IGWR101I IGWR102E IGWR103I IGWR105I.
Publication
"IBM Health Checker for z/OS User's Guide"
(SA22-7994-03) has been suplemented to include a section "VSAM Record Level Sharing (VSAMRLS) checks (IBMVSAMRLS) VSAMRLS_DIAG_CONTENTION"
Register this check by issuing the following console command:
setprog exit,add,exitname=hzsaddcheck,modname=igwrdadd

This registration must be repeated after every IPL for this check to be loaded and available to the HealthChecker.
Temporary Fix:
Comments:
The Health Check is developed to address the centralization of VSAM RLS diagnostic services. Mainly the Diag Contention console command:
D SMS,SMSVSAM,DIAG(CONTENTION)
**** AE06/11/20 FIX IN ERROR. SEE APAR OA19054 FOR DESCRIPTION
Modules/Macros:
HDZ1180J IDAVSCSR IDAVSSPC IDAVSSSR IGWRDADD IGWRDDSP IGWRDM01
IGWRDPCS IGWRDRDS IGWSCMD IGWSROUT IGWSRTE2
SRLS:
SA22799403 SA22763912

OA16372

```
APAR Identifier ...... OA16372      Last Changed ........ 07/05/17
I/O ERROR ON READ. MM ERROR CODES=X'00140408' DB2 REASON CODE=
X'00C200A4'. REPRO RETURNS RPLFDBWD = X'21080018'

Symptom ...... MS 21080018          Status ........... CLOSED  PER
Severity ................... 2      Date Closed ......... 07/05/17
Component .......... 5695DF106      Duplicate of ........
Reported Release ......... 1K0      Fixed Release ............ 999
Component Name DFSMS VSAM MEDA      Special Notice       ATTENTION
Current Target Date ..07/05/31      Flags
SCP ...................
Platform ............               PERVASIVE


Status Detail: APARCLOSURE - APAR is being closed.

PE PTF List:

PTF List:
Release 1K0   : PTF not available yet
Release 180   : PTF not available yet


Parent APAR:
Child APAR list: OA17415


ERROR DESCRIPTION:
Volume serials out of order.  This results in I/O errors on
read.
MM ERROR CODES=X'00140408'
DB2 REASON CODE=X'00C200A4'.
REPRO returns
IDC3351I ** VSAM I/O RETURN CODE IS 24 - RPLFDBWD = X'21080018'


LOCAL FIX:
DELETE NOSCRATCH and DEFINE RECATALOG with the volumes
specified in the correct order.  The LISTCAT for the data set
is needed before doing the DELETE so the correct order for the
volumes can be determined.


PROBLEM SUMMARY:
****************************************************************
* USERS AFFECTED: All HDZ11K0 users and above of VSAM reading  *
*                 or writing to a multi-volume                 *
*                 VSAM data set where the volumes have been    *
*                 recataloged in a different order             *
*                 than originally extended to.                 *
****************************************************************
* PROBLEM DESCRIPTION: VSAM GET/PUT requests fail with         *
*                      RC=8RPLFDBK X'21080018' (media manager  *
```

```
*                     error x'00140408') when accessinga VSAM *
*                     data set and the data set has           *
*                     more than 3 volumes which have          *
*                     been recataloged out of original        *
*                     extend order.                           *
*************************************************************
* RECOMMENDATION:                                           *
*************************************************************
VSAM Open is modified to detect Volumes out of order
condition.  IEC161I 071(SFI)-026 informational msesage
will be issued.  User should put volumes back in the
original order before proceeding.



PROBLEM CONCLUSION:
For out of sequence volumes, the following 2 errors will
no longer be given:
Media Manager ERROR CODES=X'00140408'
Record Management RPLFDBWD = X'21080018'
However, the user should take measures to correct the out
of sequence volumes as indicated by message IEC161I 'action
required'.



TEMPORARY FIX:


COMMENTS:


MODULES/MACROS:   IDAAMB   IDA0192Z


SRLS:      SA22763712


RTN CODES:


CIRCUMVENTION:


MESSAGE TO SUBMITTER:
```

OA17415

```
APAR Identifier ...... OA17415      Last Changed ........ 06/11/01
I/O ERROR ON READ. MM ERROR CODES=X'00140408' DB2 REASON CODE=
X'00C200A4'. *COMPANION FIX FOR OA16372*

Symptom ...... MS 21080018          Status ............ CLOSED  PER
Severity ................... 2      Date Closed ......... 06/09/19
```

```
Component .......... 5695DF122      Duplicate of ........
Reported Release ..... 1KO      Fixed Release ............ 999
Component Name VSAM REC LEV SH      Special Notice          HIPER
Current Target Date ..06/09/18      Flags
SCP ...................               FUNCTIONLOSS
Platform ............

Status Detail: SHIPMENT - Packaged solution is available for
                          shipment.

PE PTF List:

PTF List:
Release 1KO   : UA29376 available 06/10/31 (F610 )
Release 180   : UA29377 available 06/10/31 (F610 )


Parent APAR:    OA16372
Child APAR list:


ERROR DESCRIPTION:
Volume serials out of order.  This results in I/O errors on
read.
MM ERROR CODES=X'00140408'
DB2 REASON CODE=X'00C200A4'.
REPRO returns
IDC3351I ** VSAM I/O RETURN CODE IS 24 - RPLFDBWD = X'21080018'


LOCAL FIX:
DELETE NOSCRATCH and DEFINE RECATALOG with the volumes
specified in the correct order.  The LISTCAT for the data set
is needed before doing the DELETE so the correct order for the
volumes can be determined.


PROBLEM SUMMARY:
****************************************************************
* USERS AFFECTED: All HDZ11KO users and above of VSAM RLS      *
*                 reading or writing to a multi-volume         *
*                 VSAM data set where the volumes have been    *
*                 recataloged in a different order than        *
*                 originally extended to.                      *
****************************************************************
* PROBLEM DESCRIPTION: VSAM GET/PUT requests fail with RC=8    *
*                      RPLFDBK X'21080018' (media manager      *
*                      error x'00140408') when accessing       *
*                      a VSAM data set for RLS  and the        *
*                      data set has more than three volumes    *
*                      which have been recatalog out of        *
*                      the original extend order.              *
****************************************************************
* RECOMMENDATION:                                              *
****************************************************************
```

```
New code provided in HDZ11K0 and above for greater than 255
extent relief modified the VSAM RLS I/O path to not search all
internal extent control blocks for the referenced CI.  If the
secondary volumes are recatalog in a different order than
the order they were originally extended on, and the I/O
request is for a CI on a volume now "out of sequence", the
request will fail with the RC=8 RSN21080018.


PROBLEM CONCLUSION:
The VSAM RLS OPEN path was modified to allow the I/O path
to search all extent control blocks for the requested CI.


TEMPORARY FIX:
*********
* HIPER *
*********
++APAR AVAILABLE ON REQUEST.


COMMENTS:


MODULES/MACROS:   IDAVRSE0 IDAV192Z


SRLS:      NONE


RTN CODES:


CIRCUMVENTION:
```

OA20293

```
OA20293      Last Changed ........ 07/06/13
 AFTER OA17011 IGD17364I IS ISSUED WHEN MANAGEMENT CLASS
 RETENTION LIMIT IS ZERO AND TAPE DATA SETS ARE DELETED

 Symptom ...... MS IGD17364I         Status ........... CLOSED  PER
 Severity ..................... 2     Date Closed ......... 07/05/24
 Component .......... 5695DF101      Duplicate of ........
 Reported Release ......... 180      Fixed Release ............ 999
 Component Name STORAGE MGMT SU      Special Notice    PE     HIPER
 Current Target Date ..07/06/08      Flags
 SCP ...................
 Platform ............                                  DATALOSS

 Status Detail: SHIPMENT - Packaged solution is available for
                           shipment.

 PE PTF List:    UA28408 UA28410 UA28407 UA28409 UA28411 UA90305
```

```
UA90352 UA90351 UA90307 UA90306

PTF List:
Release 1J0   : UA34457 available 07/06/13 (1000 )
Release 1K0   : UA34458 available 07/06/13 (1000 )
Release 180   : UA34459 available 07/06/13 (1000 )
Release 190   : UA34460 available 07/06/13 (1000 )

Tentative Affected Releases and Current Relief Available:
Release 180   : Relief is available in the form of: LOCALFIX


Parent APAR:
Child APAR list:


ERROR DESCRIPTION:
After applying OA17011 the follow message may appear in the
joblog:

IGD17364I DATA SET my.tape.data.set.name
NOT AUTHORIZED TO EXPIRATION DATE SPECIFIED
EXPIRATION DATE RESET TO MAXIMUM ALLOWED 0.0

This occurs when customers specify a Retention Limit of zero
in the data set's Management Class.

The problem with this PTF is that tape data sets are
also getting deleted.  The documentation states "Management
class is not used for tape data sets." in the DFSMSdfp Storage
Administration Reference.

Request to development:  Customer's have also expressed that
IGD17364I should be issued in the SYSLOG.  Please see marketing
request MRO205072635
=====
Additional Symptoms:
  Tape volumes are prematurely released during RMM Housekeeping
    because the special expiration date explicitly specified
    in the JCL by the user is overridden when the management
    class assigned to the tape data set specifies Retention
    Limit = 0.  Management class attributes should not apply
    to tape data sets.
  For newly-created tape data sets, at open time the RMM
    EDGUX100 installation exit checks the JCL expiration
    date to see if a special date (e.g., 99000, 99365, 98010,
    etc.) is specified.  If so, an appropriate VRS Management
    Value will be assigned to the data set, that will match
    during housekeeping with an RMM vital record specification
    ( VRS ) that defines the retention policy to be applied
    to the data set.
  When OA17011 is applied, if Retention Limit is set to zero,
    the tape data set's special expiration date in the JCL
    is overridden causing the 'Original Expiration Date' field
    and the 'VRS Management Value' field in the RMM CDS data
```

```
    set record to be blank, and thus, the tape data set/volume
    is retained improperly.  Specify Retention Limit = NOLIMIT
    to circumvent the problem.
  OEM tape management system users (e.g., CA-1) may also be
    affected.
  Additional Keywords:
    DFSMSrmm DFRMM RMM 5695DF186 RMMCJH EXPDT OEXPDT UXTABLE
      EDGCVRSX VRSMV MV DATALOSS SCRATCH SCRATCHED CA1 TMS


LOCAL FIX:
Specify NOLIMIT for Retention Limit in the Management Class.


PROBLEM SUMMARY:
****************************************************************
* USERS AFFECTED: USERS AT Z/OS DFSMSDFP RELEASE 1GO AND       *
*                 HIGHER.                                      *
****************************************************************
* PROBLEM DESCRIPTION: OA17011 supports MGMTCLAS maximum       *
*                      retention period limit for library tape *
*                      datasets which caused problems where    *
*                      the assigned MGMTCLAS has maximum       *
*                      retention limit set to 0.               *
****************************************************************
* RECOMMENDATION:                                              *
****************************************************************
Customers who have been assigning MGMTCLAS with maximum
retention limit set to 0 find library tape volumes being
returned to scratch status unexpectedly.


PROBLEM CONCLUSION:
With this APAR , the code is now changed back to ignore the
MGMTCLAS retention limit processing.  For those customers who
have made MGMTCLAS adjustment by specifying a meaningful
retention limit will not be honored after the installation of
this APAR.   Therefore, MSG IGD17364I will no longer be issued.


TEMPORARY FIX:


COMMENTS:


MODULES/MACROS:   IGDVTSTP


SRLS:      NONE


RTN CODES:
```

CIRCUMVENTION:

MESSAGE TO SUBMITTER:

OA17011

```
APAR Identifier ...... OA17011      Last Changed ........ 07/05/24
  RETENTION PERIOD LIMIT IN MANAGEMENT CLASS EXPIRATION DATE EXPDT
  RETPD

  Symptom ...... IN INCORROUT         Status ........... CLOSED  PER
  Severity ..................... 4    Date Closed ......... 06/06/23
  Component .......... 5695DF101      Duplicate of ........
  Reported Release ......... 1J0      Fixed Release ............ 999
  Component Name STORAGE MGMT SU      Special Notice      ATTENTION
  Current Target Date ..06/08/20      Flags
  SCP ...................                              NEW FUNCTION
  Platform ............


  Status Detail: SHIPMENT - Packaged solution is available for
                            shipment.

  PE PTF List:

  PTF List:
  Release 1G0   : UA28407 available 06/09/16 (F609 )
  Release 1H0   : UA28408 available 06/09/16 (F609 )
  Release 1J0   : UA28409 available 06/09/16 (F609 )
  Release 1K0   : UA28410 available 06/09/16 (F609 )
  Release 180   : UA28411 available 06/09/16 (F609 )


  Parent APAR:
  Child APAR list:


  ERROR DESCRIPTION:
  This PMR was created as a result of field test for PMR
  74060,499,000. During field testing it was discovered
  that the Retention limit in the management class did not
  limit the expiration date specified in the Data Class or
  the JCL.


  LOCAL FIX:
  specify the expiration date which is not above your desired
  retention limit.


  PROBLEM SUMMARY:
  ****************************************************************
  * USERS AFFECTED: Users at DFSMSdfp 1.G.0 and higher releases. *
  ****************************************************************
```

```
* PROBLEM DESCRIPTION: Management class retention limit not    *
*                       enforced.                               *
****************************************************************
* RECOMMENDATION:                                              *
****************************************************************
When JCL specified RETPD exceeds the maximum retention specified
in management class, the management class value should override
the JCL value.


PROBLEM CONCLUSION:
IGDVTSTP is changed to check the correct field for management
class.


TEMPORARY FIX:


COMMENTS:
 **** PE07/03/08 FIX IN ERROR. SEE APAR OA20293  FOR DESCRIPTION


MODULES/MACROS:   IGDVTSTP


SRLS:      NONE


RTN CODES:


CIRCUMVENTION:


MESSAGE TO SUBMITTER:
```

B

Code samples

This appendix contains sample code that may be useful when implementing DFSMS V1.8.

Each sample program is documented separately so that it can be used individually as required.

In Table B-1 we list the programs provided.

Table B-1 SMF extract sample programs

Program	Function
SMF42T1 (see “SMF record type 42 sub type 1 data” on page 478)	PDSE SMF TYPE42 records subtype 1
SMF85TA (see “SMF record type 85 subtype 1-7 data display program” on page 513)	OAM SMF TYPE85 records subtypes 1-7
SMF85TO (see “SMF record type 85 subtype 38 data display program” on page 484)	OAM SMF TYPE85 records subtype 38
SMF85TI (see “SMF record type 85 subtype 39 data display program” on page 490)	OAM SMF TYPE85 records subtype 39
SMF85TJ (see “SMF record type 85 subtype 40 data display program” on page 496)	OAM SMF TYPE85 records subtype 40
SMF85TH (see “SMF record type 85 subtype 32-35 data display program” on page 502)	OAM SMF TYPE85 records subtypes 32-35
SMF42TG (see “SMF record type 42 subtype 16 data display program” on page 522)	VSAM RLS SMF TYPE42 records subtype 16
SMF42TI (see “SMF Record type 42 subtype 18 data display program” on page 534)	VSAM RLS SMF TYPE42 records subtype 18
SMF42TJ (see “SMF record type 42 subtype 19 data display program” on page 547)	VSAM RLS SMF TYPE42 records subtype 19

SMF record type 42 sub type 1 data

SMF record type 42 subtype 1 captures the PDSE buffer management function statistics.

We provide sample source code to print out the contents of the records as captured by SMF.

Sample JCL to run this program is provided at 3.9.1, “BMF data capture preparation” on page 66. The program is provided in assembler source form so that it can be assembled using the current system macros, but otherwise should not require any user modification for immediate use.

Figure B-1 shows an example of the JCL required to assemble and store the formatting program in a load module data set. The assembler source is expected to be in data set PDSERES.SMF42T1.JCL(SMF42T1A), and the result is in data set PDSERES.SMF42T1J.PDS.

```
//MHLRES1L JOB (1234567,COMMENT),MHLRES1,TIME=10,NOTIFY=MHLRES1
//ASMHCL PROC
//ASM      EXEC PGM=ASMA90,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=PDSERES.SMF42T1.JCL(SMF42T1A)
/*
//LKED.SYSLMOD  DD DSN=PDSERES.SMF42T1J.PDS,DISP=SHR
//LKED.SYSIN   DD *
 NAME SMF42T1(R)
```

Figure B-1 Example of JCL to assemble and link SMF type 42 formatting program

The source code contained in Example B-1 should be cut and pasted into a data set, then assembled. The source lines are provided to check that all lines are cut and pasted, but we do not recommend attempting to capture the line numbers. The line numbers run continuously from start to end.

Details on the PDSE buffer management function (BMF) and use of the SMF42T1 program can be found in *Partitioned Data Set Extended Usage Guide*, SG24-6106, and a summary at 3.9.3, “SMF statistics interpretation” on page 69.

Example: B-1 Source of program to print out contents of SMF TYPE 42 subtype 1 records

```
         MACRO                                                          00010011
&NAME    SEGSTART                                                       00020011
&NAME    STM   14,12,12(13)           SAVE HIS REGS IN HIS SAVE AREA   00030011
R0       EQU   0                                                        00040011
```

```
R1       EQU   1                                                      00050011
R2       EQU   2                                                      00060011
R3       EQU   3                                                      00070011
R4       EQU   4                                                      00080011
R5       EQU   5                                                      00090011
R6       EQU   6                                                      00100011
R7       EQU   7                                                      00110011
R8       EQU   8                                                      00120011
R9       EQU   9                                                      00130011
R10      EQU   10                                                     00140011
R11      EQU   11                                                     00150011
RB       EQU   12                                                     00160011
R13      EQU   13                                                     00170011
R14      EQU   14                                                     00180011
R15      EQU   15                                                     00190011
         BALR  12,0                  SET UP ADDRESSABILITY            00200011
         USING *,12                  USE REG 12 AS BASE REG           00210011
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE   00220011
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3  00230011
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS    00240011
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13 00250011
         MEND                                                         00260011
         MACRO                                                        00270011
&NAME2   SEGEND                                                       00280011
&NAME2   L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE    00290011
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA   00300011
         XR    R15,R15                                                00310011
         BR    14                    RETURN TO CALLING RTN VIA REG 14 00320011
SAVEREGS DC    18F'0'                SET UP SAVE AREA                 00330011
         MEND                                                         00340011
SMFR42T1  SEGSTART                                                    00350011
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF     00360011
*  THE SMF TYPE 42 SUBTYPE 1 RECORDS, WHICH ARE THE SMS BMF           00370011
*  RECORDS                                                            00380011
*  THE IFASMFDP PROGRAM MUST HAVE BEEN USED                           00390011
*  TO SELECT RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR     00400011
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.             00410011
*                                                                     00420011
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.                   00430011
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 42 RECORD                00440011
*  R3    START OF WHOLE RECORD                                        00450011
*  R4    START OF SMF42S1 SECTION                                     00460011
*  R5    START OF SMF4201A BMF TOTALS SECTION                         00470011
*  R6    START OF SMF4201B BMF BUFFER MANAGEMENT STATISTICS SECTION   00480011
*  OTHER REGISTER USES                                                00490011
*  R12   OVERALL BASE REGISTER                                        00500011
*  R7    RECORD TYPE/SUBTYPE CHECKING/WORKING                         00510011
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING                         00520011
*  R9    USED FOR OFFSET LENGTH ON TRIPLETS                           00530011
*  R10   USED FOR BCT                                                 00540011
*  R11   USED FOR TIME CONVERSION                                     00550011
*  R14   USED FOR TIME CONVERSION                                     00560011
*  R15   USED FOR TIME CONVERSION                                     00570011
*                                                                     00580011
         OPEN SMFIN               QSAM GET LOCATE PROCESSING IS USED 00590011
         OPEN (PRINTDCB,(OUTPUT)) QSAM PUT MOVE   PROCESSING IS USED 00600011
         PUT  PRINTDCB,PRINTHDR                                       00610011
READ     GET SMFIN                                                    00620011
         LR    R3,R1              COPY PARAMETER POINTER             00630011
         USING SMF42,R3          -> SMF RECORD                       00640011
```

```
*   CHECK IF TYPE 02                                                    00650011
         CLI   SMF42RTY,X'02' (SAME DISPLAEMENT SMF ADMIN RECORD 02)    00660011
         BE    IGNORE                                                   00670011
         CLI   SMF42RTY,X'2A'      *   CHECK IF TYPE 42                00680011
         BNE   IGNORE                                                   00690011
CHKSTYP1 DS    0H                                                       00700011
         CLI   SMF42STY+1,X'01'    *   CHECK IF SUBTYPE 1              00710011
         BNE   IGNORE                                                   00720011
* IS TYPE 42 SUBTYPE 1, SO EXTRACT DATA                                 00730011
* FIRST EXTRACT THE RECORD TIME AND DATE AND CONVERT TO HUMAN           00740011
* THEN  ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.               00750011
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION        00760011
* ADD R8 TO R3                                                          00770011
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS, HOWEVER MANY THERE ARE   00780011
         MVC   TIMEF,SMF42TME     SAVE THE SMF TIME IN 100THS OF SECS   00790011
         XR    R14,R14              CLEAR HIGH END OF PAIR             00800011
         L     R15,TIMEF           COMPLETE EVEN/ODD PAIR CONTENTS     00810011
         LA    R11,100            START BY DIVIDING BY 100 TO GET SECS 00820011
         DR    R14,R11             DIVIDE                             00830011
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS            00840011
* REMAINDER NOW IN R14 WHICH WE IGNORE                                  00850011
* QUOTIENT IN R15 - IE SECONDS WHICH WE CONVERT TO MINS + SECS          00860011
         XR    R14,R14              CLEAR HIGH END OF PAIR             00870011
         LA    R11,60             DIVIDE BY 60 TO GET MINS            00880011
         DR    R14,R11          DIVIDE TO GET MINUTES & SECONDS AS REM. 00890011
* REMAINDER NOW IN R14 WHICH IS SECONDS WHICH WE MUST SAVE              00900011
* QUOTIENT IN R15 - IE MINUTES FOR MORE PROCESSING                      00910011
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL           00920011
         OI    TIMET+7,X'0F'      FIX SIGN                            00930011
         UNPK  TIMEX+6(2),TIMET+6(2)  UNPACK SECONDS                   00940011
         MVI   TIMEX+5,C':'                                            00950011
         XR    R14,R14              CLEAR HIGH END OF PAIR             00960011
         LA    R11,60             DIVIDE BY 60 TO GET HOURS           00970011
         DR    R14,R11          DIVIDE TO GET HOURS & MINUTES AS REM.  00980011
* REMAINDER NOW IN R14 WHICH IS MINUTES WHICH WE MUST SAVE              00990011
* QUOTIENT IN R15 - IE HOURS WHICH WE MUST SAVE                         01000011
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL           01010011
         OI    TIMET+7,X'0F'      FIX SIGN                            01020011
         UNPK  TIMEX+3(2),TIMET+6(2)  UNPACK MINUTES                   01030011
         MVI   TIMEX+2,C':'                                            01040011
         CVD   R15,TIMET           CONVERT TO PACKED DECIMAL           01050011
         OI    TIMET+7,X'0F'      FIX SIGN                            01060011
         UNPK  TIMEX+0(2),TIMET+6(2)  UNPACK HOURS                     01070011
         MVC   P42TME1,TIMEX                                           01080011
         MVC   P42TME,TIMEX                                            01090011
         UNPK  P42DTE(7),SMF42DTE(4)                                   01100011
         OI    P42DTE+3,X'F0'                                          01110011
         CLC   P42DTE(2),=CL2'01'                                      01120011
         BNE   *+10                                                    01130011
         MVC   P42DTE(2),=C'20'                                        01140011
         MVC   P42DTE(2),=C'20'                                        01150011
         MVC   P42DTE1,P42DTE                                          01160011
*                                                                      01170011
         LA    R4,SMF42END    END OF HEADER, START OF DATA             01180011
         USING SMF42S1,R4                                              01190011
         L     R8,SMF42BMO    OFFSET TO BMF TOTALS SECTIONS            01200011
         LH    R9,SMF42BML    LENGTH OF BMF TOTALS SECTIONS            01210011
         LH    R10,SMF42BMN   NUMBER OF BMF TOTALS SECTIONS            01220011
BMFTRIP  DS    0H                                                      01230011
         CVD   R10,DWORD                                               01240011
```

```
         OI    DWORD+7,X'0F'                                      01250011
         UNPK  P42T#(7),DWORD+4(4)                                01260011
         PUT   PRINTDCB,PRBMSL1                                   01270011
         LA    R5,0(R3,R8)                                        01280011
         USING SMF4201A,R5                                        01290011
         L     R7,SMF42TNA                                        01300011
         CVD   R7,DWORD                                           01310011
         OI    DWORD+7,X'0F'                                      01320011
         UNPK  P42TNA(7),DWORD+4(4)                               01330011
         L     R7,SMF42TMT                                        01340011
         CVD   R7,DWORD                                           01350011
         OI    DWORD+7,X'0F'                                      01360011
         UNPK  P42TMT(7),DWORD+4(4)                               01370011
         L     R7,SMF42TRT                                        01380011
         CVD   R7,DWORD                                           01390011
         OI    DWORD+7,X'0F'                                      01400011
         UNPK  P42TRT(7),DWORD+4(4)                               01410011
         L     R7,SMF42TRH                                        01420011
         CVD   R7,DWORD                                           01430011
         OI    DWORD+7,X'0F'                                      01440011
         UNPK  P42TRH(7),DWORD+4(4)                               01450011
         L     R7,SMF42TDT                                        01460011
         CVD   R7,DWORD                                           01470011
         OI    DWORD+7,X'0F'                                      01480011
         UNPK  P42TDT(7),DWORD+4(4)                               01490011
         L     R7,SMF42TDH                                        01500011
         CVD   R7,DWORD                                           01510011
         OI    DWORD+7,X'0F'                                      01520011
         UNPK  P42TDH(7),DWORD+4(4)                               01530011
         L     R7,SMF42BUF                                        01540011
         CVD   R7,DWORD                                           01550011
         OI    DWORD+7,X'0F'                                      01560011
         UNPK  P42BUF(7),DWORD+4(4)                               01570011
         L     R7,SMF42BMX                                        01580011
         CVD   R7,DWORD                                           01590011
         OI    DWORD+7,X'0F'                                      01600011
         UNPK  P42BMX(7),DWORD+4(4)                               01610011
         LH    R7,SMF42LRU                                        01620011
         CVD   R7,DWORD                                           01630011
         OI    DWORD+7,X'0F'                                      01640011
         UNPK  P42LRU(7),DWORD+4(4)                               01650011
         LH    R7,SMF42UIC                                        01660011
         CVD   R7,DWORD                                           01670011
         OI    DWORD+7,X'0F'                                      01680011
         UNPK  P42UIC(7),DWORD+4(4)                               01690011
         DROP  R5                                                 01700011
         PUT   PRINTDCB,PRBMSL2                                   01710011
         PUT   PRINTDCB,PRBMSL3                                   01720011
         PUT   PRINTDCB,PRINTBLK                                  01730011
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS         01740011
         LA    R8,0(R8,R9)                                        01750011
         BCT   R10,BMFTRIP                                        01760011
*  WHEN BCT REACHES ZERO GO ON WITH THE SCLASS ENTRIES            01770011
*  PROCESS THE SC ENTRIES TRIPLET.                                01780011
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START           01790011
*  SECOND HW IS THE LENGTH OF EACH TRIPLET                        01800011
*  THIRD HW IS THE NUMBER OF TRIPLETS                             01810011
         L     R8,SMF42SCO    OFFSET TO THE SCLASS SECTION        01820011
         LH    R9,SMF42SCL    LENGTH OF THE SCLASS SECTIONS       01830011
         LH    R10,SMF42SCN   NUMBER OF SCLASS SECTIONS           01840011
```

```
SCOTRIP  DS    0H                                                       01850011
         LA    R6,0(R3,R8)                                              01860011
         USING SMF42O1B,R6                                              01870011
         MVC   P42PNN,SMF42PNN                                          01880011
         PUT   PRINTDCB,PRINTL1                                         01890011
         L     R7,SMF42SRT                                              01900011
         CVD   R7,DWORD                                                 01910011
         OI    DWORD+7,X'0F'                                            01920011
         UNPK  P42SRT(7),DWORD+4(4)                                     01930011
         L     R7,SMF42SRH                                              01940011
         CVD   R7,DWORD                                                 01950011
         OI    DWORD+7,X'0F'                                            01960011
         UNPK  P42SRH(7),DWORD+4(4)                                     01970011
         L     R7,SMF42SDT                                              01980011
         CVD   R7,DWORD                                                 01990011
         OI    DWORD+7,X'0F'                                            02000011
         UNPK  P42SDT(7),DWORD+4(4)                                     02010011
         L     R7,SMF42SDH                                              02020011
         CVD   R7,DWORD                                                 02030011
         OI    DWORD+7,X'0F'                                            02040011
         UNPK  P42SDH(7),DWORD+4(4)                                     02050011
WRITEIT  DS    0H                                                       02060011
         PUT   PRINTDCB,PRINTL2                                         02070011
         PUT   PRINTDCB,PRINTL3                                         02080011
         PUT   PRINTDCB,PRINTBLK                                        02090011
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               02100011
*                                                                       02110011
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD                          02120011
         LA    R8,0(R8,R9)                                              02130011
         BCT   R10,SCOTRIP                                              02140011
         B     READ                                                     02150011
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS        02160011
         B     READ                                                     02170011
FINI     DS    0H                                                       02180011
         SEGEND                                                         02190011
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,EODAD=FINI    02200011
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133               02210011
DWORD    DS    D                                                        02220011
         ORG   DWORD                                                    02230011
         DC    C'12345678'                                              02240011
TIMET    DS    D                WORKAREA FOR TIME CONVERSION            02250011
TIMEX    DS    D                WORKAREA FOR TIME CONVERSION            02260011
TIMEF    DS    F                WORKAREA FOR TIME CONVERSION            02270011
PRINTBLK DC    CL133' '                                                 02280011
PRINTHDR DC    CL133'1SMF TYPE 42 S/TYPE 1 RECORDS. COLS USE SMF NAMES' 02290011
PRINTL1  DC    CL133' SMF42PNN (SCLASS):'                               02300011
         ORG   PRINTL1+20                                               02310011
P42PNN   DC    CL30' '                                                  02320011
         ORG                                                            02330011
PRBMSL1  DC    CL133' BMF TOTALS SET #:'                                02340011
         ORG   PRBMSL1+20                                               02350011
P42T#    DC    CL8' '                                                   02360011
         ORG                                                            02370011
*                                                                       02380011
PRBMSL2  DC    CL133' '                                                 02390011
         ORG   PRBMSL2+1                                                02400011
P42TMEH1 DC    CL8'HH:MM:SS '                                           02410011
         DC    CL1' '                                                   02420011
P42DTEH1 DC    CL8'YYYYDDD '                                            02430011
P42TNAH  DC    CL9'SMF42TNA'                                            02440011
```

```
P42TMTH  DC    CL9'SMF42TMT'                                              02450011
P42TRTH  DC    CL9'SMF42TRT'                                              02460011
P42TRHH  DC    CL9'SMF42TRH'                                              02470011
P42TDTH  DC    CL9'SMF42TDT'                                              02480011
P42TDHH  DC    CL9'SMF42TDH'                                              02490011
P42BUFH  DC    CL9'SMF42BUF'                                              02500011
P42BMXH  DC    CL9'SMF42BMX'                                              02510011
P42LRUH  DC    CL9'SMF42LRU'                                              02520011
P42UCIH  DC    CL9'SMF42UIC'                                              02530011
         ORG                                                              02540011
*                                                                         02550011
PRBMSL3  DC    CL133' '                                                   02560011
         ORG   PRBMSL3+1                                                  02570011
P42TME1  DC    CL8' '                                                     02580011
         DC    CL1' '                                                     02590011
P42DTE1  DC    CL8' '                                                     02600011
P42TNA   DC    CL9' '   SMF42TNA                                          02610011
P42TMT   DC    CL9' '   SMF42TMT                                          02620011
P42TRT   DC    CL9' '   SMF42TRT                                          02630011
P42TRH   DC    CL9' '   SMF42TRH                                          02640011
P42TDT   DC    CL9' '   SMF42TDT                                          02650011
P42TDH   DC    CL9' '   SMF42TDH                                          02660011
P42BUF   DC    CL9' '   SMF42BUF                                          02670011
P42BMX   DC    CL9' '   SMF42BMX                                          02680011
P42LRU   DC    CL9' '   SMF42LRU                                          02690011
P42UIC   DC    CL9' '   SMF42UIC                                          02700011
         ORG                                                              02710011
*                                                                         02720011
PRINTL2  DC    CL133' '                                                   02730011
         ORG   PRINTL2+1                                                  02740011
P42TMEH  DC    CL8'HH:MM:SS '                                             02750011
         DC    CL1' '                                                     02760011
P42DTEH  DC    CL8'YYYYDDD '                                              02770011
P42SRTH  DC    CL9'SMF42SRT'                                              02780011
P42SRHH  DC    CL9'SMF42SRH'                                              02790011
P42SDTH  DC    CL9'SMF42SDT'                                              02800011
P42SDHH  DC    CL9'SMF42SDH'                                              02810011
         ORG                                                              02820011
*                                                                         02830011
PRINTL3  DC    CL133' '                                                   02840011
         ORG   PRINTL3+1                                                  02850011
P42TME   DC    CL8' '                                                     02860011
         DC    CL1' '                                                     02870011
P42DTE   DC    CL8' '                                                     02880011
P42SRT   DC    CL9' '     SMF42SRT                                        02890011
P42SRH   DC    CL9' '     SMF42SRH                                        02900011
P42SDT   DC    CL9' '     SMF42SDT                                        02910011
P42SDH   DC    CL9' '     SMF42SDH                                        02920011
         ORG                                                              02930011
SMFDSECT DSECT                                                            02940011
*        IFASMFR  (42) THIS DOES NOT EXPAND THE SUBTYPES AS IT SHOULD     02950011
         IGWSMF  SMF42_01=YES                                             02960011
         END                                                              02970011
```

SMF record type 85 subtype 38 data display program

Program SMG85TO displays the contents of selected fields of SMF record Type 85 subtype 38 data. It is not intended to provide a comprehensive report on OAM activity but rather to verify that retrieval from tape to DASD is occurring when an object is recalled.

Note: The source code refers to an error in the CBRSMF macro. This is described by APAR OA20170 (see “OA20170” on page 464).

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF85TO.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-2 on page 485 and all subsequent contents through to Figure B-5 on page 488 one after the other into member SMFT85OA. The result should contain 201 lines.

Note: In Figure B-5 on page 488 the line labelled SMFIN has a continuation character. This must be in column 72.

```
 MACRO
&NAME    SEGSTART
&NAME    STM   14,12,12(13)            SAVE HIS REGS IN HIS SAVE AREA
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
RB       EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         BALR  12,0                  SET UP ADDRESSABILITY
         USING *,12                  USE REG 12 AS BASE REG
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13
         MEND
         MACRO
         SEGEND
         L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA
         XR    R15,R15
         BR    14                    RETURN TO CALLING RTN VIA REG 14
SAVEREGS DC    18F'0'                SET UP SAVE AREA
         MEND
SMFR85TO  SEGSTART
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF
*  THE SMF TYPE 85 SUBTYPE 38 RECORDS, WHICH ARE THE OAM DATA SET
*  RECALL SUMMARY RECORDS
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED
*  TO SELECT TYPE 85 SUBTYPE 38
*  RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.
*
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 85 RECORD
*  R3    START OF WHOLE RECORD
*  THERE IS 1 DSECTS TO BE MAPPED
*  R4    START OF ST38 SINGLE OBJECT RECALL SECTION
*  R5    SPARE
*  R6    SPARE
*  R7    SPARE
```

Figure B-2 SMF85TO assembler source (part 1 of 4)

```
*  OTHER REGISTER USES
*  R12   OVERALL BASE REGISTER
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING
*  R9    LENGTH OF PARTICULAR DSECT
*  R10   NUMBER OF ENTRIES IN THE TRIPLET
*
*  QSAM GET LOCATE PROCESSING IS USED
*
         OPEN SMFIN
         OPEN (PRINTDCB,(OUTPUT))
         PUT  PRINTDCB,PRINTHDR
READ     GET SMFIN
* COPY PARAMETER POINTER
         LR    R3,R1
* R3 -> SMF RECORD
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION
         USING CBRSMF85,R3
*   CHECK IF TYPE 85
         CLI   SMF85RTY,X'55'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
CHKSTYP1 DS    0H
*   CHECK IF SUBTYPE 38
         CLI   SMF85STY+1,X'26'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
* IS TYPE 85 SUBTYPE 38, SO EXTRACT DATA
* R3 IS THE START OF THE WHOLE RECORD
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION
* ADD R8 TO R3
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS
         LA    R4,SMF85END
         USING ST38,R4
         L     R8,SMF85OSO
         LH    R9,SMF85OSL
         LH    R10,SMF85OSN
*  PROCESS THE SUMMARY ENTRIES TRIPLET.
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START
*  SECOND HW IS THE LENGTH OF EACH TRIPLET
*  THIRD HW IS THE NUMBER OF TRIPLETS
*  FIELDS USED IN THE REPORT CORRESPOND TO THE RECORDS TAKEN FROM
*  THE SMF RECORD TYPE 85 SUBTYPE 38 RECORDS.
*  COLN COMES FROM ST38COLN
*  CNID COMES FROM ST38CNID
*  ETC
*  ST38FLGS IS INTERPRETED AS FLG0 ON OR OFF
*
```

Figure B-3 SMF85TO assembler source (part 2 of 4)

```
*************************************************************
* NOTE THAT THERE IS A PROBLEM WITH THE CBRSMF MACRO WHICH DOES
* NOT DESCRIBE THE OFFSET TO ST38FLGS CORRECTLY. THE CODE
* ACCOMODATES THIS, BUT IT WILL HAVE TO BE ADJUSTED WHEN THE
* CBRSMF MACRO IS FIXED.
*************************************************************
*
SCOTRIP  DS    0H
         LA    R4,0(R3,R8)
         MVC   COLN,ST38COLN
*  CONVERT CNID
         L     R1,ST38CNID
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  CNID(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL1
         MVC   OBJN,ST38OBJN
         MVC   SGN,ST38SGN
*  CONVERT OLEN
         L     R1,ST38OLEN
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  OLEN(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL2
*
         MVC   VSN,ST38VSN
         MVC   MT,ST38MT
*  CONVERT TKN
         L     R1,ST38TKN
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  TKN(11),DWORD+2(6)
         MVC   VT,ST38VT
         MVC   BT,ST38BT
*  CONVERT FLAGS
         LA    R1,ST38FLGS+4  **** TEMPORARY ADJUSTMENT DUE TO
*                              ERROR IN CBRSMF RECORD. REMOVE
*                              THE +4 WHEN ST38FLGS ASSEMBLES AT
*                              OFFSET X'7C'
*        CVD   R1,DWORD
*        OI    DWORD+7,X'0F'
*        UNPK  FLGS(11),DWORD+2(6)
         MVC   FLGS,=CL20'FLG0 OFF'
         TM    0(R1),ST38FLG0   IS THE FLAG ON?
         BNO   FLG0OFF
         MVC   FLGS(08),=C'FLG0 ON '
FLG0OFF  EQU *
         PUT   PRINTDCB,PRINTL3
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
WRITEIT  DS    0H
         PUT   PRINTDCB,PRINTBLK
```

Figure B-4 SMF85TO assembler source (part 3 of 4)

```
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS
*
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD
         LA    R8,0(R8,R9)
         BCT   R10,SCOTRIP
         B     READ
IGNORE   DS    OH EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS
         B     READ
FINISH   DS    OH
         SEGEND
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,            C
               EODAD=FINISH
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133
DWORD    DS    D
         ORG   DWORD
         DC    C'12345678'
PRINTBLK DC    CL133' '
PRINTHDR DC    CL133'1SMF TYPE 85 SUBTYPE 38 RECORDS'
PRINTL1  DC    CL133' COLN/CNID:'
         ORG   PRINTL1+23
COLN     DC    CL44' '
         DC    C' '
CNID     DC    CL20' ' CONVERTED FROM BL4
         ORG
*
PRINTL2  DC    CL133' OBJN/SGN/OLEN:'
         ORG   PRINTL2+23
OBJN     DC    CL44' '
         DC    CL1' '
SGN      DC    CL8' '
         DC    CL1' '
OLEN     DC    CL20' ' CONVERTED FROM BL4
         ORG
*
PRINTL3  DC    CL133' VSN/MT/TKN/VT/BT/FLGS:'
         ORG   PRINTL3+23
VSN      DC    CL6' '
         DC    CL1' '
MT       DC    CL2' '
         DC    CL1' '
TKN      DC    CL20' '      CONVERTED FROM BL4
         DC    CL1' '
VT       DC    CL2' '
         DC    CL1' '
BT       DC    CL2' '
         DC    CL1' '
FLGS     DC    CL20' '      INTERPRETED
         ORG
SMFDSECT DSECT
         IFASMFR  (85) THIS INCLUDES CBRSMF MACRO
         END
```

Figure B-5 SMF85TO assembler source (part 4 of 4)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-6 into your PDS MHLRES1.SMF85TO.SOURCE as member SMFT85OJ. The result should contain 33 lines.

```
//
/*PRIORITY  15
//MHLRES1O JOB (1234567,COMMENT),MHLRES1,TIME=1O,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
//ASMHCL PROC
//ASM      EXEC PGM=ASMA9O,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(1O,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,2O,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF85TO.SOURCE(SMF85TOA)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TO.LOAD,DISP=(,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF85TO(R)
```

Figure B-6 SMF85TO JCL to assemble and link the program

This creates data set MHLRES1.SMF85TO.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TO.LOAD,DISP=(,CATLG,DELETE),

//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TO.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

SMF record type 85 subtype 39 data display program

Program SMG85TI displays the contents of selected fields of SMF record type 85 subtype 39 data. It is not intended to provide a comprehensive report on OAM activity but rather to verify that immediate backup is occurring.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF85TI.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-7 on page 491 and all subsequent contents through to Figure B-10 on page 494 one after the other into member SMFT85IA. The result should contain 178 lines.

Note: In Figure B-9 on page 493 the line labelled SMFIN has a continuation character. This must be in column 72.

```
 MACRO
&NAME    SEGSTART
&NAME    STM   14,12,12(13)            SAVE HIS REGS IN HIS SAVE AREA
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
RB       EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         BALR  12,0                  SET UP ADDRESSABILITY
         USING *,12                  USE REG 12 AS BASE REG
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13
         MEND
         MACRO
&NAME    SEGEND
&NAME    L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA
         XR    R15,R15
         BR    14                    RETURN TO CALLING RTN VIA REG 14
SAVEREGS DC    18F'0'                SET UP SAVE AREA
         MEND
SMFR85TI  SEGSTART
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF
*  THE SMF TYPE 85 SUBTYPE 39 RECORDS, WHICH ARE THE OAM DATA SET
*  IMMEDIATE BACKUP RECORDS.
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED
*  TO SELECT TYPE 85 SUBTYPE 39
*  RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.
*
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 85 RECORD
*  R3    START OF WHOLE RECORD
*  THERE IS 1 DSECTS TO BE MAPPED
*  R4    START OF ST39 OSMC IMMEDIATE BACKUP COPY
*  R5    SPARE
*  R6    SPARE
*  R7    SPARE
```

Figure B-7 SMF85TI assembler source (part 1 of 4)

```
*  OTHER REGISTER USES
*  R12   OVERALL BASE REGISTER
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING
*  R9    LENGTH OF PARTICULAR DSECT
*  R10   NUMBER OF ENTRIES IN THE TRIPLET
*
*  QSAM GET LOCATE PROCESSING IS USED
*
         OPEN SMFIN
         OPEN (PRINTDCB,(OUTPUT))
         PUT  PRINTDCB,PRINTHDR
READ     GET SMFIN
* COPY PARAMETER POINTER
         LR    R3,R1
* R3 -> SMF RECORD
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION
         USING CBRSMF85,R3
*   CHECK IF TYPE 85
         CLI   SMF85RTY,X'55'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
CHKSTYP1 DS    0H
*   CHECK IF SUBTYPE 39
         CLI   SMF85STY+1,X'27'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
* IS TYPE 85 SUBTYPE 39, SO EXTRACT DATA
* R3 IS THE START OF THE WHOLE RECORD
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION
* ADD R8 TO R3
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS
         LA    R4,SMF85END
         USING ST39,R4
         L     R8,SMF85OSO
         LH    R9,SMF85OSL
         LH    R10,SMF85OSN
*  PROCESS THE SUMMARY ENTRIES TRIPLET.
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START
*  SECOND HW IS THE LENGTH OF EACH TRIPLET
*  THIRD HW IS THE NUMBER OF TRIPLETS
*  FIELDS USED IN THE REPORT CORRESPOND TO THE RECORDS TAKEN FROM
*  THE SMF RECORD TYPE 85 SUBTYPE 39 RECORDS.
*  COLN COMES FROM ST39COLN
*  CNID COMES FROM ST39CNID
*  ETC
*  ST39FLGS IS NOT INTERPRETED - EACH BIT JUST SHOWN AS 1 OR 0
*
SCOTRIP  DS    0H
         LA    R4,0(R3,R8)
         MVC   COLN,ST39COLN
```

Figure B-8 SMF85TI assembler source (part 2 of 4)

```
*  CONVERT CNID
         L     R1,ST39CNID
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  CNID(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL1
         MVC   OBJN,ST39OBJN
         MVC   SGN,ST39SGN
*  CONVERT OLEN
         L     R1,ST39OLEN
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  OLEN(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL2
*
         MVC   VSN,ST39TVSN
         MVC   MT,ST39SMT
*  CONVERT TKN
         L     R1,ST39BTKN
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  TKN(11),DWORD+2(6)
*  PRINT   FLAGS
         UNPK  FLGS(09),ST39FLGS(5)  UNPK 1 MORE THAN NEEDED
         MVI   FLGS+8,C' ' BLANK OUTTHE EXTRA BYTE
         PUT   PRINTDCB,PRINTL3
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
WRITEIT  DS    0H
         PUT   PRINTDCB,PRINTBLK
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS
*
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD
         LA    R8,0(R8,R9)
         BCT   R10,SCOTRIP
         B     READ
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS
         B     READ
FINISH   DS    0H
          SEGEND
 SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,            C
                EODAD=FINISH
 PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133
 DWORD    DS    D
          ORG   DWORD
          DC    C'12345678'
 PRINTBLK DC    CL133' '
 PRINTHDR DC    CL133'1SMF TYPE 85 SUBTYPE 39 RECORDS'
```

Figure B-9 SMF85TI assembler source (part 3 of 4)

```
PRINTL1  DC    CL133' COLN/CNID:'
         ORG   PRINTL1+23
COLN     DC    CL44' '
         DC    C' '
CNID     DC    CL20' ' CONVERTED FROM BL4
         ORG
*
PRINTL2  DC    CL133' OBJN/SGN/OLEN:'
         ORG   PRINTL2+23
OBJN     DC    CL44' '
         DC    CL1' '
SGN      DC    CL8' '
         DC    CL1' '
OLEN     DC    CL20' ' CONVERTED FROM BL4
         ORG
*
PRINTL3  DC    CL133' VSN/MT/TKN/FLGS:'
         ORG   PRINTL3+23
VSN      DC    CL6' '
         DC    CL1' '
MT       DC    CL2' '
         DC    CL1' '
TKN      DC    CL20' '      CONVERTED FROM BL4
         DC    CL1' '
FLGS     DC    CL20' '      AS-IS
         ORG
SMFDSECT DSECT
         IFASMFR  (85) THIS INCLUDES CBRSMF MACRO
         END
```

Figure B-10 SMF85TI assembler source (part 4 of 4)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-6 on page 489 into your PDS MHLRES1.SMF85TI.SOURCE as member SMFT85IJ. The result should contain 32 lines.

Run the job when the member has been created.

```
//MHLRES1O JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
/*JOBPARM S=*
//ASMHCL PROC
//ASM      EXEC PGM=ASMA9O,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF85TI.SOURCE(SMF85TI)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TI.LOAD,DISP=(,CATLG,DELETE),
//     UNIT=SYSALLDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF85TI(R)
```

Figure 9-16 SMF85TI JCL to assemble and link the program

This creates data set MHLRES1.SMF85TI.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TI.LOAD,DISP=(,CATLG,DELETE),

//           UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TI.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

SMF record type 85 subtype 40 data display program

Program SMG85TJ displays the contents of selected fields of SMF record type 85 subtype 40 data. It is not intended to provide a comprehensive report on OAM activity but rather to verify that immediate backup is occurring.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF85TJ.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-11 on page 497 and all subsequent contents through to Figure B-14 on page 500, one after the other, into member SMFT85JA. The result should contain 178 lines.

```
         MACRO
&NAME    SEGSTART
&NAME    STM   14,12,12(13)            SAVE HIS REGS IN HIS SAVE AREA
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
RB       EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         BALR  12,0                  SET UP ADDRESSABILITY
         USING *,12                  USE REG 12 AS BASE REG
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13
         MEND
         MACRO
&NAME    SEGEND
&NAME    L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA
         XR    R15,R15
         BR    14                    RETURN TO CALLING RTN VIA REG 14
SAVEREGS DC    18F'0'                SET UP SAVE AREA
         MEND
SMFR85TJ  SEGSTART
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF
*  THE SMF TYPE 85 SUBTYPE 40 RECORDS, WHICH ARE THE
*  OAM COMMAND RECYCLE RECORDS.
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED
*  TO SELECT TYPE 85 SUBTYPE 40
*  RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.
*
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 85 RECORD
*  R3    START OF WHOLE RECORD
*  THERE IS 1 DSECTS TO BE MAPPED
*  R4    START OF ST40 OSMC COMMAND RECYCLE
*  R5    START OF VOLUME ARRAY - ASSUMED TO START AT THE END OF BASE
*  R6    NUMBER OF VOLUMES
*  R7    SPARE
```

Figure B-11 SMF85TJ assembler source (part 1 of 4)

```
*  OTHER REGISTER USES
*  R12   OVERALL BASE REGISTER
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING
*  R9    LENGTH OF PARTICULAR DSECT
*  R10   NUMBER OF ENTRIES IN THE TRIPLET
*
*  QSAM GET LOCATE PROCESSING IS USED
*
         OPEN SMFIN
         OPEN (PRINTDCB,(OUTPUT))
         PUT  PRINTDCB,PRINTHDR
READ     GET SMFIN
* COPY PARAMETER POINTER
         LR    R3,R1
* R3 -> SMF RECORD
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION
         USING CBRSMF85,R3
*   CHECK IF TYPE 85
         CLI   SMF85RTY,X'55'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
CHKSTYP1 DS    0H
*   CHECK IF SUBTYPE 40
         CLI   SMF85STY+1,X'28'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
* IS TYPE 85 SUBTYPE 40, SO EXTRACT DATA
* R3 IS THE START OF THE WHOLE RECORD
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION
* ADD R8 TO R3
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS
         LA    R4,SMF85END
         USING ST40,R4
         L     R8,SMF85OSO
         LH    R9,SMF85OSL
         LH    R10,SMF85OSN
*  PROCESS THE SUMMARY ENTRIES TRIPLET.
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START
*  SECOND HW IS THE LENGTH OF EACH TRIPLET
*  THIRD HW IS THE NUMBER OF TRIPLETS
*  FIELDS USED IN THE REPORT CORRESPOND TO THE RECORDS TAKEN FROM
*  THE SMF RECORD TYPE 85 SUBTYPE 40 RECORDS.
*  STRD COMES FROM ST40STRD
*  ENDD COMES FROM ST40ENDD
*  ETC
*
```

Figure B-12 SMF85TJ assembler source (part 2 of 4)

```
SCOTRIP  DS    0H
         LA    R4,0(R3,R8)
         LA    R5,ST40END     POINT TO THE VOLUME ARRAY
         USING ST40VOLD,R5
*   MOVE STRD
         MVC   STRD,ST40STRD
*   MOVE ENDD
         MVC   ENDD,ST40ENDD
* CONVERT VOLN
         LH    R1,ST40VOLN
         LR    R6,R1        SAVE FOR LATER  WHEN PRINTING THE VOLS
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  VOLN(11),DWORD+2(6)
* CONVERT PCTV
         LH    R1,ST40PCTV
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  PCTV(11),DWORD+2(6)
* CONVERT LIM
         LH    R1,ST40LIM
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  LIM(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL1
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
VLOOP    MVC   VSN,ST40VSN
         PUT   PRINTDCB,PRINTL2
         LA    R5,6(R5)
         BCT   R6,VLOOP
WRITEIT  DS    0H
         PUT   PRINTDCB,PRINTBLK
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS
*
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD
         LA    R8,0(R8,R9)
         BCT   R10,SCOTRIP
         B     READ
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS
         B     READ
FINISH   DS    0H
         SEGEND
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,             C
               EODAD=FINISH
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133
DWORD    DS    D
         ORG   DWORD
         DC    C'12345678'
PRINTBLK DC    CL133' '
```

Figure B-13 SMF85TJ assembler source (part 3 of 4)

```
PRINTHDR DC    CL133'1SMF TYPE 85 SUBTYPE 40 RECORDS'
PRINTL1  DC    CL133' STRD/ENDD/VOLN/PCTV/LIM:'
         ORG   PRINTL1+26
STRD     DC    CL10' '
         DC    C' '
ENDD     DC    CL10' '
         DC    C' '
VOLN     DC    CL12' ' CONVERTED FROM BINARY
         DC    C' '
PCTV     DC    CL12' ' CONVERTED FROM BINARY
         DC    C' '
LIM      DC    CL12' ' CONVERTED FROM BINARY
         ORG
*
PRINTL2  DC    CL133' VSN:'
         ORG   PRINTL2+5
VSN      DC    CL6' '
         ORG
*
SMFDSECT DSECT
         IFASMFR  (85) THIS INCLUDES CBRSMF MACRO
         END
```

Figure B-14 SMF85TJ assembler source (part 4 of 4)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-15 into your PDS MHLRES1.SMF85TJ.SOURCE as member SMFT85JJ. The result should contain 32 lines.

Run the job when the member has been created.

```
//MHLRES1O JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
/*JOBPARM S=*
//ASMHCL PROC
//ASM      EXEC PGM=ASMA9O,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF85TJ.SOURCE(SMF85TJ)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TJ.LOAD,DISP=SHR TLG,DELETE),
//*    UNIT=SYSALLDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF85TJ(R)
```

Figure B-15 SMF85TJ JCL to assemble and link the program

This creates data set MHLRES1.SMF85TJ.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TJ.LOAD,DISP=(,CATLG,DELETE),

//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TJ.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

SMF record type 85 subtype 32-35 data display program

Program SMG85TH displays the contents of selected fields of SMF record type 85 subtypes 32/33/34/25 data. It is not intended to provide a comprehensive report on OAM activity but rather to verify that immediate backup is occurring.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF85TH.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-16 on page 503 and all subsequent contents through to Figure B-24 on page 511, one after the other, into member SMF85TH. The result should contain 451 lines.

```
         MACRO                                                          00010099
&NAME    SEGSTART                                                       00020099
&NAME    STM   14,12,12(13)           SAVE HIS REGS IN HIS SAVE AREA   00030099
R0       EQU   0                                                        00040099
R1       EQU   1                                                        00050099
R2       EQU   2                                                        00060099
R3       EQU   3                                                        00070099
R4       EQU   4                                                        00080099
R5       EQU   5                                                        00090099
R6       EQU   6                                                        00100099
R7       EQU   7                                                        00110099
R8       EQU   8                                                        00120099
R9       EQU   9                                                        00130099
R10      EQU   10                                                       00140099
R11      EQU   11                                                       00150099
RB       EQU   12                                                       00160099
R13      EQU   13                                                       00170099
R14      EQU   14                                                       00180099
R15      EQU   15                                                       00190099
         BALR  12,0                  SET UP ADDRESSABILITY            00200099
         USING *,12                  USE REG 12 AS BASE REG           00210099
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE   00220099
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3  00230099
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS    00240099
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13 00250099
         MEND                                                           00260099
         MACRO                                                          00270099
&NAME    SEGEND                                                         00280099
&NAME    L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE    00290099
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA   00300099
         XR    R15,R15                                                  00310099
         BR    14                    RETURN TO CALLING RTN VIA REG 14 00320099
SAVEREGS DC    18F'0'                SET UP SAVE AREA                 00330099
         MEND                                                           00340099
        MACRO                                                           00350099
        BINDEC   &KEY                                                   00360099
        L     R7,ST32&KEY                                               00370099
        CVD   R7,DWORD                                                  00380099
        OI    DWORD+7,X'0F'                                             00390099
        UNPK  &KEY.(7),DWORD+4(4)                                       00400099
        MEND                                                            00410099
SMFR85TH  SEGSTART                                                      00420099
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS PARTS OF 00430099
*  THE SMF TYPE 85 SUBTYPE 32-35 RECORDS.                               00440099
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED        00450099
*  TO SELECT ANY OR ALL OF TYPE 85 SUBTYPES 32-35                       00460099
*  RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR                 00470099
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.               00480099
*                                                                       00490099
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.                     00500099
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 85 RECORD                  00510099
*  R3    START OF WHOLE RECORD                                          00520099
*  THERE IS 1 DSECTS TO BE MAPPED                                       00530099
*  R4    START OF SUBTYPE RECORDS                                       00540099
*  R5    FOR DIVIDING (DIVISOR)                                         00550099
*  R6    FOR DIVIDING - EVEN-ODD PAIR WITH R7  (DIVIDEND)               00560099
```

Figure B-16 SMF85TH assembler source (part 1 of 9)

```
*  R7    FOR DIVIDING                                                00570099
*  OTHER REGISTER USES                                               00580099
*  R12   OVERALL BASE REGISTER                                       00590099
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING                        00600099
*  R9    LENGTH OF PARTICULAR DSECT                                  00610099
*  R10   NUMBER OF ENTRIES IN THE TRIPLET                            00620099
*                                                                    00630099
*  QSAM GET LOCATE PROCESSING IS USED                                00640099
*                                                                    00650099
         OPEN SMFIN                                                 00660099
         OPEN (PRINTDCB,(OUTPUT))                                   00670099
         PUT  PRINTDCB,PRINTHDR                                     00680099
READ     GET SMFIN                                                  00690099
* COPY PARAMETER POINTER                                            00700099
         LR    R3,R1                                                00710099
* R3 -> SMF RECORD                                                  00720099
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION                     00730099
         USING CBRSMF85,R3                                          00740099
*   CHECK IF TYPE 85                                                00750099
         CLI   SMF85RTY,X'55'                                       00760099
         BNE   IGNORE                                               00770099
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS         00780099
CHKSTYP1 DS    0H                                                   00790099
*   CHECK IF ANY OF SUBTYPE 32-35                                   00800099
         CLI   SMF85STY+1,X'20'                                     00810099
         BNE   *+18                                                 00820099
         MVI   STYPE,C'2'                                           00830099
         MVC   FUNC,=CL34'(STORAGE GROUP PROCESSING)'               00840099
         B     STOK                                                 00850099
         CLI   SMF85STY+1,X'21'                                     00860099
         BNE   *+18                                                 00870099
         MVI   STYPE,C'3'                                           00880099
         MVC   FUNC,=CL34'(DASD SPACE MANAGEMENT PROCESSING)'       00890099
         B     STOK                                                 00900099
         CLI   SMF85STY+1,X'22'                                     00910099
         BNE   *+18                                                 00920099
         MVI   STYPE,C'4'                                           00930099
         MVC   FUNC,=CL34'(OPTICAL DISK RECOVERY UTILITY)'          00940099
         B     STOK                                                 00950099
         CLI   SMF85STY+1,X'23'                                     00960099
         BNE   *+18                                                 00970099
         MVI   STYPE,C'5'                                           00980099
         MVC   FUNC,=CL34'(MOVE VOLUME (MOVEVOL) UTILITY)'          00990099
         B     STOK                                                 01000099
*        OTHERWISE IGNORE                                           01010099
         B     IGNORE                                               01020099
STOK     EQU   *                                                    01030099
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS         01040099
* IS ONE OF TYPE 85 SUBTYPE 32-35 SO EXTRACT DATA                   01050099
* R3 IS THE START OF THE WHOLE RECORD                               01060099
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.           01070099
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION    01080099
* ADD R8 TO R3                                                      01090099
```

Figure B-17 SMF85TH assembler source (part 2 of 9)

```
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS                          01100099
         LA    R4,SMF85END                                            01110099
         USING ST32,R4                                                01120099
         L     R8,SMF85OSO                                            01130099
         LH    R9,SMF85OSL                                            01140099
         LH    R10,SMF85OSN                                           01150099
*  PROCESS THE SUMMARY ENTRIES TRIPLET.                               01160099
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START               01170099
*  SECOND HW IS THE LENGTH OF EACH TRIPLET                            01180099
*  THIRD HW IS THE NUMBER OF TRIPLETS                                 01190099
*  FIELDS USED IN THE REPORT CORRESPOND TO THE RECORDS TAKEN FROM      01200099
*  THE SMF RECORD TYPE 85 SUBTYPE 32-35 RECORDS.                      01210099
*  SGN  COMES FROM ST32SGN                                            01220099
*  VSN1 COMES FROM ST32VSN1                                           01230099
*  MT   COMES FROM ST32OMT                                            01240099
*  ETC                                                                01250099
*  STXXFLGS IS NOT INTERPRETED - EACH BIT JUST SHOWN AS 1 OR 0        01260099
*                                                                     01270099
SCOTRIP  DS    0H                                                     01280099
         LA    R4,0(R3,R8)                                            01290099
         UNPK  YYDDD(7),SMF85DTE                                      01300099
         CLI   YYDDD+1,C'0'                                           01310099
         BE   SETD0                                                   01320099
         CLI   YYDDD+1,C'1'                                           01330099
         BE   SETD1                                                   01340099
*        OTHERWISE ABEND AS SOMETHING HAS GONE WRONG                  01350099
         DC    F'0'                                                   01360099
SETD0    MVC   YYDDD(2),=C'19'                                        01370099
         B     SETDZ                                                  01380099
SETD1    MVC   YYDDD(2),=C'20'                                        01390099
*                                                                     01400099
SETDZ    EQU   *                                                      01410099
*    CONVERT THE TIME FROM HUNDREDTHS OF SEC SINCE MIDNIGHT           01420099
         LA    R5,100 PREPARE TO DIVIDE BY 100                        01430099
         LA    R6,0                                                   01440099
         L     R7,SMF85TME  GET THE TIME                              01450099
         DR    R6,R5   -> SECS IN R7, HUNS IN R6                      01460099
         CVD   R6,DWORD                                               01470099
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING               01480099
         UNPK  HUS,DWORD+6(2)                                         01490099
*        DC    F'0'                                                   01500099
* NOW GET THE SECS                                                    01510099
         LA    R5,60 PREPARE TO DIVIDE BY 60                          01520099
         LA    R6,0                                                   01530099
         DR    R6,R5   -> MINS IN R7, SECS REMAINDER IN R6            01540099
         CVD   R6,DWORD                                               01550099
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING               01560099
         UNPK  SS,DWORD+6(2)                                          01570099
* NOW GET THE MINS                                                    01580099
         LA    R6,0                                                   01590099
         DR    R6,R5   -> HRS IN R7, MINS REMAINDER IN R6             01600099
         CVD   R6,DWORD                                               01610099
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING               01620099
         UNPK  MM,DWORD+6(2)                                          01630099
         CVD   R7,DWORD     DO HOURS                                  01640099
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING               01650099
         UNPK  HH,DWORD+6(2)                                          01660099
```

Figure B-18 SMF85TH assembler source (part 3 of 9)

```
*                                                                       01670099
         PUT   PRINTDCB,PRINTL0                                         01680099
*  COPY    SGN                                                         01690099
         MVC   SGN,ST32SGN                                              01700099
*  COPY    VSM0                                                        01710099
         MVC   VSN0,ST32VSN0                                            01720099
*  COPY    VSM1                                                        01730099
         MVC   VSN1,ST32VSN1                                            01740099
*  COPY    MT                                                          01750099
         MVC   MT,ST32OMT                                               01760099
         PUT   PRINTDCB,PRINTL1                                         01770099
*                                                                       01780099
         BINDEC PDWO   CONVERT                                          01790099
         BINDEC PDWK   CONVERT                                          01800099
         BINDEC PDRO   CONVERT                                          01810099
         BINDEC PDRK   CONVERT                                          01820099
         BINDEC PDDO   CONVERT                                          01830099
         BINDEC PDDK   CONVERT                                          01840099
         PUT   PRINTDCB,PRINTL2                                         01850099
         BINDEC PDWO   CONVERT                                          01860099
         BINDEC PDWK   CONVERT                                          01870099
         BINDEC PORO   CONVERT                                          01880099
         BINDEC PORK   CONVERT                                          01890099
         BINDEC PODO   CONVERT                                          01900099
         BINDEC PODK   CONVERT                                          01910099
         PUT   PRINTDCB,PRINTL3                                         01920099
         BINDEC PTWO   CONVERT                                          01930099
         BINDEC PTWK   CONVERT                                          01940099
         BINDEC PTRO   CONVERT                                          01950099
         BINDEC PTRK   CONVERT                                          01960099
         BINDEC PTDO   CONVERT                                          01970099
         BINDEC PTDK   CONVERT                                          01980099
         PUT   PRINTDCB,PRINTL4                                         01990099
         BINDEC BOWO   CONVERT                                          02000099
         BINDEC BOWK   CONVERT                                          02010099
         BINDEC BORO   CONVERT                                          02020099
         BINDEC BORK   CONVERT                                          02030099
         BINDEC BODO   CONVERT                                          02040099
         BINDEC BODK   CONVERT                                          02050099
         PUT   PRINTDCB,PRINTL5                                         02060099
         BINDEC BTWO   CONVERT                                          02070099
         BINDEC BTWK   CONVERT                                          02080099
         BINDEC BTRO   CONVERT                                          02090099
         BINDEC BTRK   CONVERT                                          02100099
         BINDEC BTDO   CONVERT                                          02110099
         BINDEC BTDK   CONVERT                                          02120099
         PUT   PRINTDCB,PRINTL6                                         02130099
         BINDEC B2OWO CONVERT                                           02140099
         BINDEC B2OWK CONVERT                                           02150099
         BINDEC B2ORO CONVERT                                           02160099
         BINDEC B2ORK CONVERT                                           02170099
         BINDEC B2ODO CONVERT                                           02180099
         BINDEC B2ODK CONVERT                                           02190099
```

Figure B-19 SMF85TH assembler source (part 4 of 9)

```
 PUT   PRINTDCB,PRINTL7                                          02200099
         BINDEC B2TWO  CONVERT                                          02210099
         BINDEC B2TWK  CONVERT                                          02220099
         BINDEC B2TRO  CONVERT                                          02230099
         BINDEC B2TRK  CONVERT                                          02240099
         BINDEC B2TDO  CONVERT                                          02250099
         BINDEC B2TDK  CONVERT                                          02260099
         PUT   PRINTDCB,PRINTL8                                         02270099
         BINDEC DTUP   CONVERT                                          02280099
         BINDEC DTDE   CONVERT                                          02290099
*  CONVERT 4KIN                                                        02300099
         L     R1,ST324KIN                                              02310099
         CVD   R1,DWORD                                                 02320099
         OI    DWORD+7,X'0F'                                            02330099
         UNPK  N4KIN(11),DWORD+2(6)                                     02340099
*  CONVERT 4KDE                                                        02350099
         L     R1,ST324KDE                                              02360099
         CVD   R1,DWORD                                                 02370099
         OI    DWORD+7,X'0F'                                            02380099
         UNPK  N4KDE(11),DWORD+2(6)                                     02390099
*  CONVERT 32KI                                                        02400099
         L     R1,ST3232KI                                              02410099
         CVD   R1,DWORD                                                 02420099
         OI    DWORD+7,X'0F'                                            02430099
         UNPK  N32KI(11),DWORD+2(6)                                     02440099
*  CONVERT 32KD                                                        02450099
         L     R1,ST3232KD                                              02460099
         CVD   R1,DWORD                                                 02470099
         OI    DWORD+7,X'0F'                                            02480099
         UNPK  N32KD(11),DWORD+2(6)                                     02490099
         BINDEC NCE    CONVERT                                          02500099
         PUT   PRINTDCB,PRINTL9                                         02510099
*  INTERPRET THE FLAGS                                                 02520099
         UNPK  FLGS(09),ST32FLGS(5)  UNPK 1 MORE THAN NEEDED            02530099
         MVI   FLGS+8,C' ' BLANK OUTTHE EXTRA BYTE                      02540099
         NC    FLGS(08),=8X'0F'                                         02550099
         TR    FLGS(8),TRTAB                                            02560099
         BINDEC NTE    CONVERT                                          02570099
         BINDEC RCLD   CONVERT                                          02580099
         BINDEC RCLK   CONVERT                                          02590099
         BINDEC LOBI   CONVERT                                          02600099
         BINDEC LOBD   CONVERT                                          02610099
         PUT   PRINTDCB,PRINTL10                                        02620099
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS             02630099
         PUT   PRINTDCB,PRINTBLK                                        02640099
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               02650099
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD                          02660099
         LA    R8,0(R8,R9)                                              02670099
         BCT   R10,SCOTRIP                                              02680099
         B     READ                                                     02690099
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS        02700099
         B     READ                                                     02710099
```

Figure B-20 SMF85TH assembler source (part 5 of 9)

```
FINISH   DS    OH                                                      02720099
         SEGEND                                                        02730099
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,            C02740099
               EODAD=FINISH                                          02750099
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133            02760099
DWORD    DS    D                                                     02770099
         ORG   DWORD                                                 02780099
         DC    C'12345678'                                           02790099
TRWORK   DS    CL33                                                  02800099
TRTAB    DC    C'0123456789ABCDEF'                                   02810099
PRINTBLK DC    CL133' '                                              02820099
PRINTHDR DC    CL133'1SMF TYPE 85 SUBTYPE 32-35 RECORDS'             02830099
PRINTLO  DC    CL133' SMFDTE/TME:'                                   02840099
         ORG   PRINTLO+38                                            02850099
YYDDD    DC    CL7' '                                                02860099
         DC    CL1' '                                                02870099
HH       DC    CL2' '                                                02880099
         DC    C':'                                                  02890099
MM       DC    CL2' '                                                02900099
         DC    C':'                                                  02910099
SS       DC    CL2' '                                                02920099
         DC    C'.'                                                  02930099
HUS      DC    CL3' '                                                02940099
         DC    CL1' '                                                02950099
         ORG                                                         02960099
PRINTL1  DC    CL133' STYPE/SGN/VSNO/VSN1/MT:'                       02970099
         ORG   PRINTL1+38                                            02980099
STYPE3   DC    CL1'3'  PREFIX TO SUBTYPES 32-35                      02990099
STYPE    DC    CL1' '  CONVERTED                                     03000099
         DC    CL1' '                                                03010099
SGN      DC    CL8' '                                                03020099
         DC    C' '                                                  03030099
VSNO     DC    CL6' '                                                03040099
         DC    C' '                                                  03050099
VSN1     DC    CL6' '                                                03060099
         DC    C' '                                                  03070099
MT       DC    CL2' '                                                03080099
         DC    C' '                                                  03090099
FUNC     DC    CL34' '                                               03100099
         ORG                                                         03110099
*                                                                    03120099
PRINTL2  DC    CL133' PDWO/PDWK/PDRO/PDRK/PDDO/PDDK:'                03130099
         ORG   PRINTL2+38                                            03140099
PDWO     DC    CL12' '  CONVERTED                                    03150099
         DC    CL1' '                                                03160099
PDWK     DC    CL12' '  CONVERTED                                    03170099
         DC    CL1' '                                                03180099
PDRO     DC    CL12' '  CONVERTED                                    03190099
         DC    CL1' '                                                03200099
PDRK     DC    CL12' '  CONVERTED                                    03210099
         DC    CL1' '                                                03220099
PDDO     DC    CL12' '  CONVERTED                                    03230099
         DC    CL1' '                                                03240099
PDDK     DC    CL12' '  CONVERTED                                    03250099
         DC    CL1' '                                                03260099
         ORG                                                         03270099
*                                                                    03280099
```

Figure B-21 SMF85TH assembler source (part 6 of 9)

```
PRINTL3  DC    CL133' POWO/POWK/PORO/PORD/PODO/PODK:'                    03290099
         ORG   PRINTL3+38                                               03300099
POWO     DC    CL12' '  CONVERTED                                       03310099
         DC    CL1' '                                                   03320099
POWK     DC    CL12' '  CONVERTED                                       03330099
         DC    CL1' '                                                   03340099
PORO     DC    CL12' '  CONVERTED                                       03350099
         DC    CL1' '                                                   03360099
PORK     DC    CL12' '  CONVERTED                                       03370099
         DC    CL1' '                                                   03380099
PODO     DC    CL12' '  CONVERTED                                       03390099
         DC    CL1' '                                                   03400099
PODK     DC    CL12' '  CONVERTED                                       03410099
         DC    CL1' '                                                   03420099
         ORG                                                            03430099
PRINTL4  DC    CL133' PTWO/PTWK/PTRO/PTRD/PTDO/PTDK:'                   03440099
         ORG   PRINTL4+38                                               03450099
PTWO     DC    CL12' '  CONVERTED                                       03460099
         DC    CL1' '                                                   03470099
PTWK     DC    CL12' '  CONVERTED                                       03480099
         DC    CL1' '                                                   03490099
PTRO     DC    CL12' '  CONVERTED                                       03500099
         DC    CL1' '                                                   03510099
PTRK     DC    CL12' '  CONVERTED                                       03520099
         DC    CL1' '                                                   03530099
PTDO     DC    CL12' '  CONVERTED                                       03540099
         DC    CL1' '                                                   03550099
PTDK     DC    CL12' '  CONVERTED                                       03560099
         DC    CL1' '                                                   03570099
         ORG                                                            03580099
PRINTL5  DC    CL133' BOWO/BOWK/BORO/BORK/BODO/BODK:'                   03590099
         ORG   PRINTL5+38                                               03600099
BOWO     DC    CL12' '  CONVERTED                                       03610099
         DC    CL1' '                                                   03620099
BOWK     DC    CL12' '  CONVERTED                                       03630099
         DC    CL1' '                                                   03640099
BORO     DC    CL12' '  CONVERTED                                       03650099
         DC    CL1' '                                                   03660099
BORK     DC    CL12' '  CONVERTED                                       03670099
         DC    CL1' '                                                   03680099
BODO     DC    CL12' '  CONVERTED                                       03690099
         DC    CL1' '                                                   03700099
BODK     DC    CL12' '  CONVERTED                                       03710099
         DC    CL1' '                                                   03720099
         ORG                                                            03730099
PRINTL6  DC    CL133' BTWO/BTWK/BTRO/BTRK/BTDO/BTDK:'                   03740099
         ORG   PRINTL6+38                                               03750099
BTWO     DC    CL12' '  CONVERTED                                       03760099
         DC    CL1' '                                                   03770099
BTWK     DC    CL12' '  CONVERTED                                       03780099
         DC    CL1' '                                                   03790099
BTRO     DC    CL12' '  CONVERTED                                       03800099
         DC    CL1' '                                                   03810099
BTRK     DC    CL12' '  CONVERTED                                       03820099
         DC    CL1' '                                                   03830099
```

Figure B-22 SMF85TH assembler source (part 7 of 9)

```
BTDO     DC    CL12' '  CONVERTED                                    03840099
         DC    CL1' '                                                03850099
BTDK     DC    CL12' '  CONVERTED                                   03860099
         DC    CL1' '                                                03870099
         ORG                                                         03880099
PRINTL7  DC    CL133' B2OWO/B2OWK/B2ORO/B2ORK/B2ODO/B2ODK:'          03890099
         ORG   PRINTL7+38                                            03900099
B2OWO     DC    CL12' '  CONVERTED                                  03910099
         DC    CL1' '                                                03920099
B2OWK     DC    CL12' '  CONVERTED                                  03930099
         DC    CL1' '                                                03940099
B2ORO     DC    CL12' '  CONVERTED                                  03950099
         DC    CL1' '                                                03960099
B2ORK     DC    CL12' '  CONVERTED                                  03970099
         DC    CL1' '                                                03980099
B2ODO     DC    CL12' '  CONVERTED                                  03990099
         DC    CL1' '                                                04000099
B2ODK     DC    CL12' '  CONVERTED                                  04010099
         DC    CL1' '                                                04020099
         ORG                                                         04030099
PRINTL8  DC    CL133' B2TWO/B2TWK/B2TRO/B2TRK/B2TDO/B2TDK:'          04040099
         ORG   PRINTL8+38                                            04050099
B2TWO     DC    CL12' '  CONVERTED                                  04060099
         DC    CL1' '                                                04070099
B2TWK     DC    CL12' '  CONVERTED                                  04080099
         DC    CL1' '                                                04090099
B2TRO     DC    CL12' '  CONVERTED                                  04100099
         DC    CL1' '                                                04110099
B2TRK     DC    CL12' '  CONVERTED                                  04120099
         DC    CL1' '                                                04130099
B2TDO     DC    CL12' '  CONVERTED                                  04140099
         DC    CL1' '                                                04150099
B2TDK     DC    CL12' '  CONVERTED                                  04160099
         DC    CL1' '                                                04170099
         ORG                                                         04180099
PRINTL9  DC    CL133' DTUP/DTDE/4KIN/4KDE/32KI/32KD/NCE:'            04190099
         ORG   PRINTL9+38                                            04200099
DTUP      DC    CL12' '  CONVERTED                                  04210099
         DC    CL1' '                                                04220099
DTDE      DC    CL12' '  CONVERTED                                  04230099
         DC    CL1' '                                                04240099
N4KIN     DC    CL12' '  CONVERTED                                  04250099
         DC    CL1' '                                                04260099
N4KDE     DC    CL12' '  CONVERTED                                  04270099
         DC    CL1' '                                                04280099
N32KI     DC    CL12' '  CONVERTED                                  04290099
         DC    CL1' '                                                04300099
N32KD     DC    CL12' '  CONVERTED                                  04310099
         DC    CL1' '                                                04320099
NCE       DC    CL12' '  CONVERTED                                  04330099
         ORG                                                         04340099
```

Figure B-23 SMF85TH assembler source (part 8 of 9)

```
PRINTL10 DC    CL133' FLGS/NTE/RCLD/RCLK/LOBI/LOBD:'                    04350099
         ORG   PRINTL10+38                                             04360099
FLGS     DC    CL9' '   INTERPRETED AS 0 OR 1                         04370099
         DC    CL4' '                                                  04380099
NTE       DC    CL12' '  CONVERTED                                    04390099
         DC    CL1' '                                                  04400099
RCLD      DC    CL12' '  CONVERTED                                    04410099
         DC    CL1' '                                                  04420099
RCLK      DC    CL12' '  CONVERTED                                    04430099
         DC    CL1' '                                                  04440099
LOBI      DC    CL12' '  CONVERTED                                    04450099
         DC    CL1' '                                                  04460099
LOBD      DC    CL12' '  CONVERTED                                    04470099
         ORG                                                           04480099
SMFDSECT DSECT                                                         04490099
         IFASMFR  (85) THIS INCLUDES CBRSMF MACRO                      04500099
         END                                                           04510099
```

Figure B-24 SMF85TH assembler source (part 9 of 9)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-25 into your PDS MHLRES1.SMF85TH.SOURCE as member SMF85THJ. The result should contain 32 lines.

Run the job when the member has been created.

```
//MHLRES1O JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
/*JOBPARM S=*
//ASMHCL PROC
//ASM      EXEC PGM=ASMA90,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF85TH.SOURCE(SMF85TH)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TH.LOAD,DISP=(,CATLG,DELETE),
//     UNIT=SYSALLDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF85TH(R)
```

Figure B-25 SMF85TH JCL to assemble and link the program

This creates data set MHLRES1.SMF85TH.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TH.LOAD,DISP=(,CATLG,DELETE),

//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TH.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

SMF record type 85 subtype 1-7 data display program

Program SMG85TA displays the contents of selected fields of SMF record type 85 subtypes 1/2/3/4/5/6/7 data. It is not intended to provide a comprehensive report on OAM activity but rather to verify that immediate backup is occurring.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF85TA.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-26 on page 514 and all subsequent contents through to Figure B-32 on page 520, one after the other, into member SMFT85AA. The result should contain 333 lines.

```
         MACRO
&NAME    SEGSTART
&NAME    STM   14,12,12(13)            SAVE HIS REGS IN HIS SAVE AREA
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
RB       EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         BALR  12,0                  SET UP ADDRESSABILITY
         USING *,12                  USE REG 12 AS BASE REG
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13
         MEND
         MACRO
&NAME    SEGEND
&NAME    L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA
         XR    R15,R15
         BR    14                    RETURN TO CALLING RTN VIA REG 14
SAVEREGS DC    18F'0'                SET UP SAVE AREA
         MEND
SMFR85TA  SEGSTART
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS PARTS OF
*  THE SMF TYPE 85 SUBTYPE 1-7 RECORDS.
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED
*  TO SELECT ANY OR ALL OF TYPE 85 SUBTYPES 1-7
*  RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.
*
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 85 RECORD
*  R3    START OF WHOLE RECORD
*  THERE IS 1 DSECTS TO BE MAPPED
*  R4    START OF SUBTYPE RECORDS
*  R5    SPARE
*  R6    SPARE
*  R7    SPARE
*  OTHER REGISTER USES
*  R12   OVERALL BASE REGISTER
```

Figure B-26 SMF85TA assembler source (part 1 of 7)

```
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING
*  R9    LENGTH OF PARTICULAR DSECT
*  R10   NUMBER OF ENTRIES IN THE TRIPLET
*
*  QSAM GET LOCATE PROCESSING IS USED
*
         OPEN SMFIN
         OPEN (PRINTDCB,(OUTPUT))
         PUT  PRINTDCB,PRINTHDR
READ     GET SMFIN
* COPY PARAMETER POINTER
         LR    R3,R1
* R3 -> SMF RECORD
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION
         USING CBRSMF85,R3
*   CHECK IF TYPE 85
         CLI   SMF85RTY,X'55'
         BNE   IGNORE
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
CHKSTYP1 DS    0H
*   CHECK IF ANY OF SUBTYPE 1-7
         CLI   SMF85STY+1,X'01'
         BNE   *+18
         MVI   STYPE,C'1'
         MVC   FUNC,=CL15'OSREQ ACCESS'
         B     STOK
         CLI   SMF85STY+1,X'02'
         BNE   *+18
         MVI   STYPE,C'2'
         MVC   FUNC,=CL15'OSREQ STORE'
         B     STOK
         CLI   SMF85STY+1,X'03'
         BNE   *+18
         MVI   STYPE,C'3'
         MVC   FUNC,=CL15'OSREQ RETRIEVE'
         B     STOK
         CLI   SMF85STY+1,X'04'
         BNE   *+18
         MVI   STYPE,C'4'
         MVC   FUNC,=CL15'OSREQ QUERY'
         B     STOK
         CLI   SMF85STY+1,X'05'
         BNE   *+18
         MVI   STYPE,C'5'
         MVC   FUNC,=CL15'OSREQ CHANGE'
         B     STOK
         CLI   SMF85STY+1,X'06'
         BNE   *+18
         MVI   STYPE,C'6'
         MVC   FUNC,=CL15'OSREQ DELETE'
         B     STOK
```

Figure B-27 SMF85TA assembler source (part 2 of 7

```
*
         CLI   SMF85STY+1,X'07'
         BNE   *+18
         MVI   STYPE,C'7'
         MVC   FUNC,=CL15'OSREQ UNACCESS'
         B     STOK
*        OTHERWISE IGNORE
         B     IGNORE
STOK     EQU   *
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS
* IS ONE OF TYPE 85 SUBTYPE 1-7 SO EXTRACT DATA
* R3 IS THE START OF THE WHOLE RECORD
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION
* ADD R8 TO R3
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS
         LA    R4,SMF85END
         USING ST1,R4
         L     R8,SMF85OSO
         LH    R9,SMF85OSL
         LH    R10,SMF85OSN
*  PROCESS THE SUMMARY ENTRIES TRIPLET.
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START
*  SECOND HW IS THE LENGTH OF EACH TRIPLET
*  THIRD HW IS THE NUMBER OF TRIPLETS
*  FIELDS USED IN THE REPORT CORRESPOND TO THE RECORDS TAKEN FROM
*  THE SMF RECORD TYPE 85 SUBTYPE 1-7 RECORDS.
*  COLN COMES FROM ST1COLN
*  OBJN COMES FROM ST1OBJN
*  ETC
*  ST1FLGS IS NOT INTERPRETED - EACH BIT JUST SHOWN AS 1 OR 0
*
SCOTRIP  DS    0H
         LA    R4,0(R3,R8)
         LA    R4,0(R3,R8)
         UNPK  YYDDD(7),SMF85DTE
         CLI   YYDDD+1,C'0'
         BE   SETD0
         CLI   YYDDD+1,C'1'
         BE   SETD1
*        OTHERWISE ABEND AS SOMETHING HAS GONE WRONG
         DC    F'0'
SETD0    MVC   YYDDD(2),=C'19'
         B     SETDZ
SETD1    MVC   YYDDD(2),=C'20'
*
```

Figure B-28 SMF85TA assembler source (part 3 of 7)

```
SETDZ    EQU   *
*    CONVERT THE TIME FROM HUNDREDTHS OF SEC SINCE MIDNIGHT
         LA    R5,100 PREPARE TO DIVIDE BY 100
         LA    R6,0
         L     R7,SMF85TME  GET THE TIME
         DR    R6,R5   -> SECS IN R7, HUNS IN R6
         CVD   R6,DWORD
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING
         UNPK  HUS,DWORD+6(2)
*        DC    F'0'
* NOW GET THE SECS
         LA    R5,60 PREPARE TO DIVIDE BY 60
         LA    R6,0
         DR    R6,R5   -> MINS IN R7, SECS REMAINDER IN R6
         CVD   R6,DWORD
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING
         UNPK  SS,DWORD+6(2)
* NOW GET THE MINS
         LA    R6,0
         DR    R6,R5   -> HRS IN R7, MINS REMAINDER IN R6
         CVD   R6,DWORD
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING
         UNPK  MM,DWORD+6(2)
         CVD   R7,DWORD     DO HOURS
         OI    DWORD+7,X'0F'  FIX THE SIGN FOR PRINTING
         UNPK  HH,DWORD+6(2)
*
         PUT   PRINTDCB,PRINTL0
*
         PUT   PRINTDCB,PRINTL1
*  COPY    COLN
         MVC   COLN,ST1COLN
*  COPY    OBJN
         MVC   OBJN,ST1OBJN
         PUT   PRINTDCB,PRINTL2
*
*  COPY    SGN
         MVC   SGN,ST1SGN
*  COPY    SCN
         MVC   SCN,ST1SCN
*  COPY    MCN
         MVC   MCN,ST1MCN
*
*  CONVERT LEN
         L     R1,ST1LEN
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  LEN(11),DWORD+2(6)
```

Figure B-29 SMF85TA assembler source (part 4 of 7)

```
*  CONVERT      TTOK  & TOK
* DO TOK FIRST
*        TOK IS 8 BYTES BINARY -> 16 BYTES PRINTABLE
*        MVC   TOK,ST1TOK
         UNPK  TRWORK(15),ST1TOK+1(8)  15 BYTES (ONE REDUNDANT BYTE)
         NC    TRWORK(15),=15X'0F'
         TR    TRWORK(15),TRTAB
         MVC   TOK+2(14),TRWORK
         UNPK  TRWORK(3),ST1TOK(2)  LAST BYTE + ONE REDUNDANT BYTE
         NC    TRWORK(3),=3X'0F'
         TR    TRWORK(3),TRTAB
         MVC   TOK(2),TRWORK
*
*        TTOK IS 16 BYTES BINARY -> 32 BYTES PRINTABLE
*        HAVE TO UNPACK THIS AS TWO SETS OF 16 AS PER TOK
*        FIRST DO 16 BYTES, THE REPEAT FOR THE NEXT TWO
         UNPK  TRWORK(15),ST1TTOK+1(8)  15 BYTES  (ONE REDUNDANT BYTE)
         NC    TRWORK(15),=15X'0F'
         TR    TRWORK(15),TRTAB
         MVC   TTOK+2(14),TRWORK
         UNPK  TRWORK(3),ST1TTOK(2)  LAST BYTES + ONE REDUNDANT BYTE
         NC    TRWORK(3),=3X'0F'
         TR    TRWORK(3),TRTAB
         MVC   TTOK(2),TRWORK
* NOW DO IT ALL AGAIN WITH OFFSET OF 8 ON ST1TTOK AND OFSET OF 16 ON
*        TTOK
         UNPK  TRWORK(15),ST1TTOK+1+8(8) 15 BYTES (ONE REDUNDANT BYTE)
         NC    TRWORK(15),=15X'0F'
         TR    TRWORK(15),TRTAB
         MVC   TTOK+2+16(14),TRWORK
         UNPK  TRWORK(3),ST1TTOK+8(2)  LAST BYTES + ONE REDUNDANT BYTE
         NC    TRWORK(3),=3X'0F'
         TR    TRWORK(3),TRTAB
         MVC   TTOK+16(2),TRWORK
*
*  TRANSLATE TO PRINTABLE
         PUT   PRINTDCB,PRINTL3
*
* COPY VSN & VMT
         MVC   VSN,ST1VSN
         MVC   MT,ST1VMT
*  CONVERT RC & RS
*  CONVERT RC
         L     R1,ST1RC
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  RC(08),DWORD+4(4)
*  CONVERT RS
         L     R1,ST1RS
         CVD   R1,DWORD
         OI    DWORD+7,X'0F'
         UNPK  RS(08),DWORD+4(4)
```

Figure B-30 SMF85TA assembler source (part 5 of 7)

```
*  PRINT   FLAGS
         UNPK  FLGS(09),ST1FLGS(5)  UNPK 1 MORE THAN NEEDED
         MVI   FLGS+8,C' ' BLANK OUT THE EXTRA BYTE
         NC    FLGS(08),=8X'0F'
         TR    FLGS(8),TRTAB
         PUT   PRINTDCB,PRINTL4
WRITEIT  DS    0H
         PUT   PRINTDCB,PRINTBLK
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS
*
*  WHEN BCT REACHES ZERO GO GET ANOTHER RECORD
         LA    R8,0(R8,R9)
         BCT   R10,SCOTRIP
         B     READ
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS
         B     READ
FINISH   DS    0H
         SEGEND
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,
               EODAD=FINISH
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133
DWORD    DS    D
         ORG   DWORD
         DC    C'12345678'
TRWORK   DS    CL33
TRTAB    DC    C'0123456789ABCDEF'
PRINTBLK DC    CL133' '
PRINTHDR DC    CL133'1SMF TYPE 85 SUBTYPE 1-7 RECORDS'
PRINTL0  DC    CL133' SMFDTE/TME:'
         ORG   PRINTL0+27
YYDDD    DC    CL7' '
         DC    CL1' '
HH       DC    CL2' '
         DC    C':'
MM       DC    CL2' '
         DC    C':'
SS       DC    CL2' '
         DC    C'.'
HUS      DC    CL3' '
         DC    CL1' '
         ORG
PRINTL1  DC    CL133' STYPE:'
         ORG   PRINTL1+27
STYPE    DC    CL1' '  CONVERTED
         DC    CL1' '
FUNC     DC    CL15' '
         ORG
```

Figure B-31 SMF85TA assembler source (part 6 of 7)

```
PRINTL2  DC    CL133' COLN/OBJN:'
         ORG   PRINTL2+27
COLN     DC    CL44' '
         DC    C' '
OBJN     DC    CL20' ' CONVERTED FROM BL4
         ORG
*
PRINTL3  DC    CL133' SGN/SCN/MCN/LEN/TTOK/TOK:'
         ORG   PRINTL3+27
SGN      DC    CL8' '
         DC    CL1' '
SCN      DC    CL8' '
         DC    CL1' '
MCN      DC    CL8' '
         DC    CL1' '
LEN      DC    CL12' '
         DC    CL1' '
TTOK     DC    CL32' '      CONVERTED
         DC    CL1' '
TOK      DC    CL16' '      CONVERTED
         ORG
*
PRINTL4  DC    CL133' VSN/MT/RC/RS/FLGS:'
         ORG   PRINTL4+27
VSN      DC    CL6' '      COPIED
         DC    CL1' '
MT       DC    CL2' '      COPIED
         DC    CL1' '
RC       DC    CL8' '      CONVERTED
         DC    CL1' '
RS       DC    CL8' '      CONVERTED
         DC    CL1' '
FLGS     DC    CL20' '      AS-IS
         ORG
SMFDSECT DSECT
         IFASMFR  (85) THIS INCLUDES CBRSMF MACRO
         END
```

Figure B-32 SMF85TA assembler source (part 7 of 7)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-33 into your PDS MHLRES1.SMF85TA.SOURCE as member SMFT85AJ. The result should contain 32 lines.

Run the job when the member has been created.

```
//MHLRES1O JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
//ASMHCL PROC
//ASM      EXEC PGM=ASMA9O,REGION=OM,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF85TA.SOURCE(SMF85TAA)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TA.LOAD,DISP=(,CATLG,DELETE),
//*    UNIT=SYSALLDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF85TA(R)
```

Figure B-33 SMF85TA JCL to assemble and link the program

This creates data set MHLRES1.SMF85TA.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TA.LOAD,DISP=(,CATLG,DELETE),

//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TA.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

SMF record type 42 subtype 16 data display program

Program SMG42TG displays the contents of selected fields of SMF record type 42 subtypes 16 data. It is not intended to provide a comprehensive report on RLS activity but rather to verify that data sets are being allocated below or above the 2 GB bar as required.

There are three steps to build the program which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF42TG.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-34 on page 523 and all subsequent contents through to Figure B-43 on page 532, one after the other, into member SMF42TG. The result should contain 708 lines.

```
         MACRO                                                          00010043
&NAME    SEGSTART                                                       00020043
&NAME    STM   14,12,12(13)           SAVE HIS REGS IN HIS SAVE AREA    00030043
R0       EQU   0                                                        00040043
R1       EQU   1                                                        00050043
R2       EQU   2                                                        00060043
R3       EQU   3                                                        00070043
R4       EQU   4                                                        00080043
R5       EQU   5                                                        00090043
R6       EQU   6                                                        00100043
R7       EQU   7                                                        00110043
R8       EQU   8                                                        00120043
R9       EQU   9                                                        00130043
R10      EQU   10                                                       00140043
R11      EQU   11                                                       00150043
RB       EQU   12                                                       00160043
R13      EQU   13                                                       00170043
R14      EQU   14                                                       00180043
R15      EQU   15                                                       00190064
         BALR  12,0                  SET UP ADDRESSABILITY              00200043
         USING *,12                  USE REG 12 AS BASE REG             00210043
         USING *+4096,6                                                 00220074
         LA    6,4095(12)                                               00230074
         LA    6,1(6)                                                   00240074
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE     00250043
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3    00260043
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS      00270043
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13   00280043
         MEND                                                           00290043
         MACRO                                                          00300043
&NAME    SEGEND                                                         00310043
&NAME    L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE      00320043
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA     00330043
         XR    R15,R15                                                  00340043
         BR    14                    RETURN TO CALLING RTN VIA REG 14   00350043
SAVEREGS DC    18F'0'                SET UP SAVE AREA                   00360043
         MEND                                                           00370043
         MACRO                                                          00380053
         BINDEC   &KEY                                                  00390053
         L     R7,SMF42&KEY                                             00400053
         CVD   R7,DWORD                                                 00410053
         OI    DWORD+7,X'0F'                                            00420053
         UNPK  P42&KEY.(7),DWORD+4(4)                                   00430053
         MEND                                                           00440053
         MACRO                                                          00450073
         BINDECA  &KEY                                                  00460073
         L     R7,SMF2A&KEY                                             00470073
         CVD   R7,DWORD                                                 00480073
         OI    DWORD+7,X'0F'                                            00490073
         UNPK  P2A&KEY.(7),DWORD+4(4)                                   00500073
         MEND                                                           00510073
SMF42TGA  SEGSTART                                                      00520099
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF       00530043
*  THE SMF TYPE 42 SUBTYPE 16 RECORDS, WHICH ARE THE SMS DATA SET       00540043
*  SUMMARY RECORDS                                                      00550043
*  IT IS ASSUMED THAT THE IFASMFDP PROGRAM HAS ALREADY BEEN USED        00560043
*  TO SELECT RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR       00570043
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.               00580043
*                                                                       00590043
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.                     00600043
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 42 RECORD                  00610043
*  R3    START OF WHOLE RECORD                                          00620043
*  THERE ARE 4 DSECTS TO BE MAPPED                                      00630043
*  2 FOR THE BELOW 64-BIT DATA                                          00640043
*  2 FOR THE ABOVE 64-BIT DATA                                          00650043
*  R4    START OF TRIPLETS                                              00660099
*  R5    START OF DATA                                                  00670099
*  R7                                                                   00680099
*  OTHER REGISTER USES                                                  00690043
*  R12   OVERALL BASE REGISTER                                          00700043
*  R6    2ND     BASE REGISTER                                          00710099
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING                           00720043
*  R9    LENGTH OF PARTICULAR DSECT                                     00730043
*  R10   NUMBER OF ENTRIES IN THE TRIPLET                               00740043
*                                                                       00750043
```

Figure B-34 SMF42TG assembler source (part 1 of 10)

```
*  QSAM GET LOCATE PROCESSING IS USED                                   00760043
*                                                                       00770043
         OPEN SMFIN                                                     00780043
         OPEN (PRINTDCB,(OUTPUT))                                       00790043
*        LA   R1,PGA142L1                                               00800060
         PUT  PRINTDCB,PRINTHDR                                         00810043
READ     GET SMFIN                                                      00820043
* COPY PARAMETER POINTER                                               00830043
         LR    R3,R1                                                    00840043
* R3 -> SMF RECORD                                                     00850043
* USE SMF R3 RECORD MAPPING FOR INITIAL VERSION                        00860043
         USING SMF42,R3                                                 00870043
*   CHECK IF TYPE 02                                                   00880043
         CLI   SMF42RTY,X'02' (SAME DISPLAEMENT SMF ADMIN RECORD 02)    00890043
         BE    IGNORE                                                   00900043
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS             00910048
*   CHECK IF TYPE 42                                                   00920043
         CLI   SMF42RTY,X'2A'                                           00930043
         BNE   IGNORE                                                   00940043
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS             00950048
CHKSTYP1 DS    0H                                                       00960043
*   CHECK IF SUBTYPE 16                                                00970043
         CLI   SMF42STY+1,X'10'                                         00980043
         BNE   IGNORE                                                   00990049
*        DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS             01000052
* IS TYPE 42 SUBTYPE 16, SO EXTRACT DATA                               01010043
* R3 IS THE START OF THE WHOLE RECORD                                  01020043
* FIRST ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.              01030043
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION       01040043
* ADD R8 TO R3                                                         01050043
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS                          01060043
         LA    R4,SMF42END  (WHERE THE 1ST DATA SECTION STARTS)         01070043
         USING SMF42SG,R4                                               01080043
         MVC   TIMEF,SMF42TME     SAVE THE SMF TIME IN 100THS OF SECS   01090069
         XR    R14,R14              CLEAR HIGH END OF PAIR              01100069
         L     R15,TIMEF           COMPLETE EVEN/ODD PAIR CONTENTS      01110069
         LA    R11,100            START BY DIVIDING BY 100 TO GET SECS  01120069
         DR    R14,R11             DIVIDE                              01130069
* REMAINDER NOW IN R14 WHICH WE IGNORE                                 01140069
* QUOTIENT IN R15 - IE SECONDS WHICH WE CONVERT TO MINS + SECS         01150069
         XR    R14,R14              CLEAR HIGH END OF PAIR              01160069
         LA    R11,60              DIVIDE BY 60 TO GET MINS            01170069
         DR    R14,R11           DIVIDE TO GET MINUTES & SECONDS AS REM. 01180069
* REMAINDER NOW IN R14 WHICH IS SECONDS WHICH WE MUST SAVE             01190069
* QUOTIENT IN R15 - IE MINUTES FOR MORE PROCESSING                     01200069
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL          01210069
         OI    TIMET+7,X'0F'      FIX SIGN                            01220069
         UNPK  TIMEX+6(2),TIMET+6(2)  UNPACK SECONDS                   01230069
         MVI   TIMEX+5,C':'                                            01240069
         XR    R14,R14              CLEAR HIGH END OF PAIR             01250069
         LA    R11,60              DIVIDE BY 60 TO GET HOURS           01260069
         DR    R14,R11           DIVIDE TO GET HOURS & MINUTES AS REM.  01270069
* REMAINDER NOW IN R14 WHICH IS MINUTES WHICH WE MUST SAVE             01280069
* QUOTIENT IN R15 - IE HOURS WHICH WE MUST SAVE                        01290069
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL          01300069
         OI    TIMET+7,X'0F'      FIX SIGN                            01310069
         UNPK  TIMEX+3(2),TIMET+6(2)  UNPACK MINUTES                   01320069
         MVI   TIMEX+2,C':'                                            01330069
         CVD   R15,TIMET           CONVERT TO PACKED DECIMAL          01340069
         OI    TIMET+7,X'0F'      FIX SIGN                            01350069
         UNPK  TIMEX+0(2),TIMET+6(2)  UNPACK HOURS                     01360069
         MVC   P42TME,TIMEX                                            01370069
         MVC   P42TME2,TIMEX                                           01380069
         MVC   P2ATME,TIMEX                                            01390075
         MVC   P2ATME2,TIMEX                                           01400075
         UNPK  P42DTE3(7),SMF42DTE(4)                                  01410069
         OI    P42DTE3+3,X'F0'                                         01420069
         CLC   P42DTE3(2),=CL2'01'                                     01430069
         BNE   *+10                                                    01440069
         MVC   P42DTE3(2),=C'20'                                       01450069
         MVC   P42DTE5,P42DTE3                                         01460069
         MVC   P2ADTE3,P42DTE3                                         01470075
         MVC   P2ADTE5,P42DTE3                                         01480075
```

Figure B-35 SMF42TG assembler source (part 2 of 10)

```
*  PROCESS THE DS SUMMARY ENTRIES TRIPLET.                                01490069
*  FIRST FULLWORD IS OFFSET TO WHERE THE TRIPLETS START                   01500069
*  SECOND HW IS THE LENGTH OF ALL  TRIPLETS                               01510069
*  THIRD HW IS THE NUMBER OF TRIPLETS                                     01520069
GA142PRP EQU   *                                                         01530057
         L     R8,SMF42GD1  OFFSET  TO START OF CF DATASET SUMMARY       01540099
*                           BELOW THE 64-BIT LINE                       01550043
         LH    R9,SMF42GD2    LENGTH OF THE SCLASS SECTIONS              01560057
         LH    R10,SMF42GD3   NUMBER OF SCLASS SECTIONS                  01570057
         LTR  R10,R10                                                    01580057
         BZ    G1A42PRP                                                  01590060
         XR    R14,R14        CLEAR HIGH ORDER                           01600057
         STH   R14,P42TGA1C   START THE COUNT AT 0                       01610057
         LR    R15,R9         LOAD LEGTH IN R9                           01620057
         DR    R14,R10         GET LENGTH OF SMF420JA DSECTS             01630057
         LR    R9,R15                                                    01640057
         LA    R5,0(R3,R8)  POINT R5 TO THE RECORD START                 01650043
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                    01660057
GA142TRP DS    0H                                                        01670057
        LH    R14,P42TGA1C  GET CURRENT COUNT                           01680057
        LA    R14,1(R14)    ADD ONE                                     01690057
        STH   R14,P42TGA1C  PUT CURRENT COUNT                           01700057
        CVD   R14,DWORD                                                 01710057
        OI    DWORD+7,X'0F'                                             01720057
        UNPK  P42TGA1#(7),DWORD+4(4)                                    01730057
        PUT   PRINTDCB,PGA142L1                                         01740057
        LA    R5,0(R5,R9)   ADVANCE TO DSECT                            01750057
         USING SMF420GA,R5                                              01760057
         MVC   P42GAE,SMF42GAE                                          01770057
         BINDEC GAA                                                     01780053
         MVC   P42GAB(44),SMF42GAB                                      01790099
         MVC   P42GAC(44),SMF42GAC                                      01800054
         PUT   PRINTDCB,PGA142L2                                        01810060
         PUT   PRINTDCB,PGA142L3                                        01820060
         MVC   P42GAF(30),SMF42GAF                                      01830062
         MVC   P42GAH(30),SMF42GAH                                      01840065
         MVC   P42GAI(9),=CL9'UNKNOWN'   SET UNKNOWN IN CASE NEITHER ON 01850099
*        TM    SMF42GAI,SMF42GAIO      IS BIT 0 ON?                     01860099
         TM    SMF42GAI,X'80'          IS BIT 0 ON?                     01870099
         BO    P42GAIO                 NO SEE IF                        01880099
         B     P42GAI1T                                                 01890099
P42GAIO  MVC   P42GAI(8),=C'DATA'      YES SET DATA                     01900099
         B     P42GAIOK                                                 01910099
P42GAI1T EQU   *                                                        01920099
*        TM    SMF42GAI,SMF42GAI1      IS BIT 1 ON?                     01930099
         TM    SMF42GAI,X'40'          IS BIT 1 ON?                     01940099
         BO    P42GAI1                 NO DON'T SET                     01950099
         B     P42GAIOK                                                 01960099
P42GAI1  MVC   P42GAI(8),=C'INDEX'     YES SET INDEX                    01970099
P42GAIOK EQU   *                                                        01980099
         MVC   P42GAJ(12),SMF42GAJ                                      01990066
         PUT   PRINTDCB,PGA142L4                                        02000099
         PUT   PRINTDCB,PGA142L5                                        02010099
         MVC   P42GAP(16),SMF42GAP                                      02020067
         BINDEC GCA                                                     02030099
         BINDEC GCB                                                     02040099
         BINDEC GCC                                                     02050099
         BINDEC GCD                                                     02060099
         BINDEC GCE                                                     02070099
         BINDEC GCF                                                     02080099
         BINDEC GCI                                                     02090099
         BINDEC GCL                                                     02100099
         BINDEC GCM                                                     02110099
         BINDEC GCN                                                     02120099
         PUT   PRINTDCB,PGA142L6                                        02130099
         PUT   PRINTDCB,PGA142L7                                        02140099
         PUT   PRINTDCB,PRINTBLK                                        02150043
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS                02160043
*                                                                        02170043
*  WHEN BCT REACHES ZERO GO ON TO THE NEXT SET                           02180056
         BCT   R10,GA142TRP                                             02190057
```

Figure B-36 SMF42TG assembler source (part 3 of 10)

```
G1A42PRP EQU   *                                                       02200060
         L     R8,SMF42GD4  OFFSET TO START OF DATASET CF/SYS SUMMARY   02210099
*                           BELOW THE 64-BIT LINE                      02220060
         LH    R9,SMF42GD5    LENGTH OF THE SCLASS SECTIONS             02230060
         LH    R10,SMF42GD6   NUMBER OF SCLASS SECTIONS                 02240060
         LTR   R10,R10                                                  02250060
         BZ    GA12APRP                                                 02260083
         XR    R14,R14        CLEAR HIGH ORDER                          02270060
         STH   R14,P42TG1AC   START THE COUNT AT 0                      02280060
         LR    R15,R9         LOAD LEGTH IN R9                          02290060
         DR    R14,R10         GET LENGTH OF SMF42OJA DSECTS            02300060
         LR    R9,R15                                                   02310060
         LA    R5,0(R3,R8)  POINT R5 TO THE RECORD START                02320060
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   02330060
G1A42TRP DS    0H                                                       02340060
        LH    R14,P42TG1AC  GET CURRENT COUNT                          02350060
        LA    R14,1(R14)    ADD ONE                                    02360060
        STH   R14,P42TG1AC  PUT CURRENT COUNT                          02370060
        CVD   R14,DWORD                                                02380060
        OI    DWORD+7,X'0F'                                            02390060
        UNPK  P42TG1A#(7),DWORD+4(4)                                   02400061
        PUT   PRINTDCB,PG1A42L1                                        02410060
        LA    R5,0(R5,R9)   ADVANCE TO DSECT                           02420060
         USING SMF42OGB,R5                                             02430060
         BINDEC GBA                                                    02440082
         CH     R7,=H'0'                                               02450099
         BE     G1A42SKP           IF THE COUNT IS ZERO SKIP REPORT    02460099
         MVC   P42GBB(44),SMF42GBB                                     02470060
         MVC   P42GBC(44),SMF42GBC                                     02480093
         PUT   PRINTDCB,PG1A42L2                                       02490099
         PUT   PRINTDCB,PG1A42L3                                       02500099
         MVC   P42GBE,SMF42GBE                                         02510060
         MVC   P42GBF(30),SMF42GBF                                     02520066
         MVC   P42GBH(8),SMF42GBH                                      02530066
         MVC   P42GBI(9),=CL9'UNKNOWN'   SET UNKNOWN IN CASE NEITHER ON 02540099
*        TM    SMF42GBI,SMF42GBIO      IS BIT 0 ON?                    02550099
         TM    SMF42GBI,X'80'          IS BIT 0 ON?                    02560099
         BO    P42GBIO                 NO SEE IF                       02570099
         B     P42GBI1T                                                02580099
P42GBIO  MVC   P42GBI(8),=C'DATA'      YES SET DATA                    02590099
         B     P42GBIOK                                                02600099
P42GBI1T EQU   *                                                       02610099
*        TM    SMF42GBI,SMF42GBI1      IS BIT 1 ON?                    02620099
         TM    SMF42GBI,X'40'          IS BIT 1 ON?                    02630099
         BO    P42GBI1                 NO DON'T SET                    02640099
         B     P42GBIOK                                                02650099
P42GBI1  MVC   P42GBI(8),=C'INDEX'     YES SET INDEX                   02660099
P42GBIOK EQU   *                                                       02670099
         MVC   P42AO9(12),SMF42AO9                                     02680066
         PUT   PRINTDCB,PG1A42L4                                       02690099
         PUT   PRINTDCB,PG1A42L5                                       02700099
         MVC   P42GBP(16),SMF42GBP                                     02710093
         BINDEC GIA                                                    02720099
         BINDEC GIB                                                    02730099
         BINDEC GIC                                                    02740099
         BINDEC GID                                                    02750099
         BINDEC GIE                                                    02760099
         BINDEC GIF                                                    02770099
         BINDEC GIL                                                    02780099
         BINDEC GIR                                                    02790099
         BINDEC GIS                                                    02800099
         PUT   PRINTDCB,PG1A42L6                                       02810099
         PUT   PRINTDCB,PG1A42L7                                       02820099
         PUT   PRINTDCB,PRINTBLK                                       02830065
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               02840065
*                                                                       02850065
*  WHEN BCT REACHES ZERO GO ON TO THE NEXT SET                          02860065
G1A42SKP DS    0H                                                      02870099
         BCT   R10,G1A42TRP                                            02880065
GA12APRP EQU   *                                                       02890073
         L     R8,SMF2AGD1  OFFSET  TO START OF CF DATASET SUMMARY     02900099
*                           ABOVE THE 64-BIT LINE                      02910090
         LH    R9,SMF2AGD2    LENGTH OF THE SCLASS SECTIONS             02920073
         LH    R10,SMF2AGD3   NUMBER OF SCLASS SECTIONS                 02930073
         LTR   R10,R10                                                  02940073
         BZ    G1A2APRP                                                 02950099
         XR    R14,R14        CLEAR HIGH ORDER                          02960073
         STH   R14,P2ATGA1C   START THE COUNT AT 0                      02970073
```

Figure B-37 SMF42TG assembler source (part 4 of 10)

```
         LR    R15,R9         LOAD LEGTH IN R9                          02980073
         DR    R14,R10         GET LENGTH OF SMF2AOJA DSECTS            02990073
         LR    R9,R15                                                   03000073
         LA    R5,0(R3,R8)  POINT R5 TO THE RECORD START               03010073
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   03020073
GA12ATRP DS    0H                                                       03030073
        LH    R14,P2ATGA1C  GET CURRENT COUNT                          03040073
        LA    R14,1(R14)    ADD ONE                                    03050073
        STH   R14,P2ATGA1C  PUT CURRENT COUNT                          03060073
        CVD   R14,DWORD                                                03070073
        OI    DWORD+7,X'0F'                                            03080073
        UNPK  P2ATGA1#(7),DWORD+4(4)                                   03090073
        PUT   PRINTDCB,PGA12AL1                                        03100073
        LA    R5,0(R5,R9)   ADVANCE TO DSECT                           03110073
         USING SMF2AOGA,R5                                             03120073
         MVC   P2AGAE,SMF2AGAE                                         03130073
         BINDECA GAA                                                   03140073
         MVC   P2AGAB(44),SMF2AGAB                                     03150099
         MVC   P2AGAC(44),SMF2AGAC                                     03160073
         PUT   PRINTDCB,PGA12AL2                                       03170073
         PUT   PRINTDCB,PGA12AL3                                       03180073
         MVC   P2AGAF(30),SMF2AGAF                                     03190073
         MVC   P2AGAH(30),SMF2AGAH                                     03200073
         MVC   P2AGAI(9),=CL9'UNKNOWN'   SET UNKNOWN IN CASE NEITHER ON 03210099
*        TM    SMF2AGAI,SMF2AGAIO      IS BIT 0 ON?                    03220099
         TM    SMF2AGAI,X'80'          IS BIT 0 ON?                    03230099
         BO    P2AGAI0                 NO SEE IF                       03240099
         B     P2AGAI1T                                                03250099
P2AGAI0  MVC   P2AGAI(8),=C'DATA'      YES SET DATA                    03260099
         B     P2AGAIOK                                                03270099
P2AGAI1T EQU   *                                                       03280099
*        TM    SMF2AGAI,SMF2AGAI1      IS BIT 1 ON?                    03290099
         TM    SMF2AGAI,X'40'          IS BIT 1 ON?                    03300099
         BO    P2AGAI1                 NO DON'T SET                    03310099
         B     P2AGAIOK                                                03320099
P2AGAI1  MVC   P2AGAI(8),=C'INDEX'     YES SET INDEX                   03330099
P2AGAIOK EQU   *                                                       03340099
         MVC   P2AGAJ(12),SMF2AGAJ                                     03350073
         PUT   PRINTDCB,PGA12AL4                                       03360099
         PUT   PRINTDCB,PGA12AL5                                       03370099
         MVC   P2AGAP(16),SMF2AGAP                                     03380073
         BINDECA GCA                                                   03390099
         BINDECA GCB                                                   03400099
         BINDECA GCC                                                   03410099
         BINDECA GCD                                                   03420099
         BINDECA GCE                                                   03430099
         BINDECA GCF                                                   03440099
         BINDECA GCI                                                   03450099
         BINDECA GCL                                                   03460099
         BINDECA GCM                                                   03470099
         BINDECA GCN                                                   03480099
         PUT   PRINTDCB,PGA12AL6                                       03490099
         PUT   PRINTDCB,PGA12AL7                                       03500099
         PUT   PRINTDCB,PRINTBLK                                       03510073
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               03520073
*                                                                       03530073
*  WHEN BCT REACHES ZERO GO ON TO THE NEXT SET                          03540073
         BCT   R10,GA12ATRP                                            03550073
G1A2APRP EQU   *                                                       03560073
         L     R8,SMF2AGD4  OFFSET TO START OF DATASET CF/SYS SUMMARY   03570099
*                           ABOVE THE 64-BIT LINE                      03580099
         LH    R9,SMF2AGD5    LENGTH OF THE SCLASS SECTIONS             03590073
         LH    R10,SMF2AGD6   NUMBER OF SCLASS SECTIONS                 03600073
         LTR  R10,R10                                                  03610073
         BZ    READ         (LAST ONE)                                 03620099
         XR    R14,R14        CLEAR HIGH ORDER                        03630073
         STH   R14,P2ATG1AC   START THE COUNT AT 0                      03640073
         LR    R15,R9         LOAD LEGTH IN R9                        03650073
         DR    R14,R10         GET LENGTH OF SMF2AOJA DSECTS          03660073
         LR    R9,R15                                                  03670073
         LA    R5,0(R3,R8)  POINT R5 TO THE RECORD START               03680073
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                  03690073
```

Figure B-38 SMF42TG assembler source (part 5 of 10)

```
G1A2ATRP DS    0H                                                       03700073
         LH    R14,P2ATG1AC  GET CURRENT COUNT                          03710099
         LA    R14,1(R14)    ADD ONE                                    03720099
         STH   R14,P2ATG1AC  PUT CURRENT COUNT                          03730099
         CVD   R14,DWORD                                                03740099
         OI    DWORD+7,X'0F'                                            03750099
         UNPK  P2ATG1A#(7),DWORD+4(4)                                   03760099
         PUT   PRINTDCB,PG1A2AL1                                        03770099
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                           03780099
         USING SMF2A0GB,R5                                              03790073
         BINDECA  GBA                                                   03800082
         MVC   P2AGBB(44),SMF2AGBB                                      03810073
         MVC   P2AGBC(44),SMF2AGBC                                      03820093
         PUT   PRINTDCB,PG1A2AL2                                        03830099
         PUT   PRINTDCB,PG1A2AL3                                        03840099
         MVC   P2AGBE,SMF2AGBE                                          03850073
         MVC   P2AGBF(30),SMF2AGBF                                      03860073
         MVC   P2AGBH(8),SMF2AGBH                                       03870073
         MVC   P2AGBI(9),=CL9'UNKNOWN'   SET UNKNOWN IN CASE NEITHER ON 03880099
*        TM    SMF2AGBI,SMF2AGBIO      IS BIT 0 ON?                     03890099
         TM    SMF2AGBI,X'80'          IS BIT 0 ON?                     03900099
         BO    P2AGBIO                 NO SEE IF                        03910099
         B     P2AGBI1T                                                 03920099
P2AGBIO  MVC   P2AGBI(8),=C'DATA'      YES SET DATA                     03930099
         B     P2AGBIOK                                                 03940099
P2AGBI1T EQU   *                                                        03950099
*        TM    SMF2AGBI,SMF2AGBI1      IS BIT 1 ON?                     03960099
         TM    SMF2AGBI,X'40'          IS BIT 1 ON?                     03970099
         BO    P2AGBI1                 NO DON'T SET                     03980099
         B     P2AGBIOK                                                 03990099
P2AGBI1  MVC   P2AGBI(8),=C'INDEX'     YES SET INDEX                    04000099
P2AGBIOK EQU   *                                                        04010099
         MVC   P2AA09(12),SMF2AA09                                      04020073
         PUT   PRINTDCB,PG1A2AL4                                        04030099
         PUT   PRINTDCB,PG1A2AL5                                        04040099
         MVC   P2AGBP(16),SMF2AGBP                                      04050093
         BINDECA GIA                                                    04060099
         BINDECA GIB                                                    04070099
         BINDECA GIC                                                    04080099
         BINDECA GID                                                    04090099
         BINDECA GIE                                                    04100099
         BINDECA GIF                                                    04110099
         BINDECA GIL                                                    04120099
         BINDECA GIR                                                    04130099
         BINDECA GIS                                                    04140099
         PUT   PRINTDCB,PG1A2AL6                                        04150099
         PUT   PRINTDCB,PG1A2AL7                                        04160099
         PUT   PRINTDCB,PRINTBLK                                        04170073
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               04180073
*  WHEN BCT REACHES ZERO GO ON TO THE NEXT SET                          04190073
         BCT   R10,G1A2ATRP                                             04200073
         B     READ                                                     04210043
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS        04220043
         B     READ                                                     04230043
FINISH   DS    0H                                                       04240043
         SEGEND                                                         04250044
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,             C04260043
               EODAD=FINISH                                             04270043
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133               04280043
DWORD    DS    D                                                        04290043
         ORG   DWORD                                                    04300043
         DC    C'12345678'                                              04310043
TIMET    DS    D                 WORKAREA FOR TIME CONVERSION           04320069
TIMEX    DS    D                 WORKAREA FOR TIME CONVERSION           04330069
TIMEF    DS    F                 WORKAREA FOR TIME CONVERSION           04340069
PRINTBLK DC    CL133' '                                                 04350043
PRINTHDR DC    CL133'1SMF TYPE 42 SUBTYPE 16 RECORDS'                   04360043
************ SMF42GA1 ************                                      04370060
PGA142L1 DC    CL133' SMF42GA1 SYSPLEX D/S RESPONSE SUMMARY SET #:'     04380099
         ORG   PGA142L1+51                                              04390099
P42TGA1# DC    CL8' '                                                   04400057
         ORG                                                            04410057
P42TGA1C DC    H'0'  TO COUNT THE NUMBER OF SETS                        04420057
```

Figure B-39 SMF42TG assembler source (part 6 of 10)

```
*                                                                       04430057
*                                                                       04440043
PGA142L2 DC    CL133' '                                                 04450060
         ORG   PGA142L2+1                                               04460060
P42DTE3H DC    CL8'YYYYDDD '                                            04470062
P42TMEH  DC    CL9'HH:MM:SS '                                           04480069
P42GAAH  DC    CL9'SMF42GAA'                                            04490054
P42GABH  DC    CL44'SMF42GAB'                                           04500099
P42GACH  DC    CL45'SMF42GAC'                                           04510054
         ORG                                                            04520043
*                                                                       04530043
PGA142L3 DC    CL133' '                                                 04540060
         ORG   PGA142L3+1                                               04550060
P42DTE3  DC    CL8' '                                                   04560062
P42TME   DC    CL9'HH:MM:SS '                                           04570069
P42GAA   DC    CL9' '                                                   04580062
P42GAB   DC    CL44' '                                                  04590099
P42GAC   DC    CL45' '                                                  04600099
         ORG                                                            04610043
PGA142L4 DC    CL133' '                                                 04620062
         ORG   PGA142L4+1+8                                             04630063
P42GAEH  DC    CL26'SMF42GAE'                                           04640099
P42GAFH  DC    CL31'SMF42GAF'                                           04650062
P42GAHH  DC    CL32'SMF42GAH'                                           04660073
P42GAIH  DC    CL9'SMF42GAI'  INTERPRETED                               04670099
P42GAJH  DC    CL13'SMF42GAJ'                                           04680099
         ORG                                                            04690062
*                                                                       04700062
PGA142L5 DC    CL133' '                                                 04710062
         ORG   PGA142L5+1+8                                             04720064
P42GAE   DC    CL26' '                                                  04730099
P42GAF   DC    CL31' '                                                  04740065
P42GAH   DC    CL32' '                                                  04750073
P42GAI   DC    CL9' '                                                   04760099
P42GAJ   DC    CL13' '                                                  04770066
         ORG                                                            04780062
PGA142L6 DC    CL133' '                                                 04790099
         ORG   PGA142L6+1+8                                             04800099
P42GAPH  DC    CL17'SMF42GAP'                                           04810099
P42GCAH  DC    CL9'SMF42GCA'                                            04820099
P42GCBH  DC    CL9'SMF42GCB'                                            04830099
P42GCCH  DC    CL9'SMF42GCC'                                            04840099
P42GCDH  DC    CL9'SMF42GCD'                                            04850099
P42GCEH  DC    CL9'SMF42GCE'                                            04860099
P42GCFH  DC    CL9'SMF42GCF'                                            04870099
P42GCIH  DC    CL9'SMF42GCI'                                            04880099
P42GCLH  DC    CL9'SMF42GCL'                                            04890099
P42GCMH  DC    CL9'SMF42GCM'                                            04900099
P42GCNH  DC    CL9'SMF42GCN'                                            04910099
         ORG                                                            04920099
*                                                                       04930099
PGA142L7 DC    CL133' '                                                 04940099
         ORG   PGA142L7+1+8                                             04950099
P42GAP   DC    CL17' '                                                  04960099
P42GCA   DC    CL9' '                                                   04970099
P42GCB   DC    CL9' '                                                   04980099
P42GCC   DC    CL9' '                                                   04990099
P42GCD   DC    CL9' '                                                   05000099
P42GCE   DC    CL9' '                                                   05010099
P42GCF   DC    CL9' '                                                   05020099
P42GCI   DC    CL9' '                                                   05030099
P42GCL   DC    CL9' '                                                   05040099
P42GCM   DC    CL9' '                                                   05050099
P42GCN   DC    CL9' '                                                   05060099
         ORG                                                            05070099
************ SMF42G1A ************                                      05080099
PG1A42L1 DC    CL133' SMF42G1A DATA SET/MVS SYSTEM SUMMARY SET #:'      05090099
         ORG   PG1A42L1+45                                              05100099
P42TG1A# DC    CL8' '                                                   05110060
         ORG                                                            05120060
P42TG1AC DC    H'0'  TO COUNT THE NUMBER OF SETS                        05130060
PG1A42L2 DC    CL133' '                                                 05140060
         ORG   PG1A42L2+1                                               05150060
P42DTE5H DC    CL8'YYYYDDD '                                            05160062
P42TME2H DC    CL9'HH:MM:SS '                                           05170070
```

Figure B-40 SMF42TG assembler source (part 7 of 10)

```
P42TME2H DC    CL9'HH:MM:SS '                                          05170070
P42GBAH  DC    CL9'SMF42GBA'                                           05180060
P42GBBH  DC    CL45'SMF42GBB'                                          05190060
P42GBCH  DC    CL45'SMF42GBC'                                          05200060
         DC    CL1' '                                                  05210060
         ORG                                                           05220060
PG1A42L3 DC    CL133' '                                                05230060
         ORG   PG1A42L3+1                                              05240060
P42DTE5  DC    CL8' '                                                  05250062
P42TME2  DC    CL9'HH:MM:SS '                                          05260070
P42GBA   DC    CL9' '                                                  05270060
P42GBB   DC    CL45' '                                                 05280060
P42GBC   DC    CL45' '                                                 05290060
         DC    CL1' '                                                  05300060
         ORG                                                           05310060
                                                                       05320063
PG1A42L4 DC    CL133' '                                                05330062
         ORG   PG1A42L4+1+8                                            05340063
P42GBEH  DC    CL30'SMF42GBE'                                          05350099
P42GBFH  DC    CL31'SMF42GBF'                                          05360099
P42GBHH  DC    CL32'SMF42GBH'                                          05370099
P42GBIH  DC    CL9'SMF42GBI' INTERPRETED                               05380099
P42AO9H  DC    CL13'SMF42AO9'                                          05390099
         ORG                                                           05400062
PG1A42L5 DC    CL133' '                                                05410062
         ORG   PG1A42L5+1+8                                            05420063
P42GBE   DC    CL30' '                                                 05430099
P42GBF   DC    CL31' '                                                 05440099
P42GBH   DC    CL32' '                                                 05450099
P42GBI   DC    CL9' '                                                  05460099
P42AO9   DC    CL13' '                                                 05470099
         ORG                                                           05480062
PG1A42L6 DC    CL133' '                                                05490099
         ORG   PG1A42L6+1+8                                            05500099
P42GBPH  DC    CL17'SMF42GBP'                                          05510099
P42GIAH  DC    CL9'SMF42GIA'                                           05520099
P42GIBH  DC    CL9'SMF42GIB'                                           05530099
P42GICH  DC    CL9'SMF42GIC'                                           05540099
P42GIDH  DC    CL9'SMF42GID'                                           05550099
P42GIEH  DC    CL9'SMF42GIE'                                           05560099
P42GIFH  DC    CL9'SMF42GIF'                                           05570099
P42GILH  DC    CL9'SMF42GIL'                                           05580099
P42GIRH  DC    CL9'SMF42GIR'                                           05590099
P42GISH  DC    CL9'SMF42GIS'                                           05600099
         ORG                                                           05610099
PG1A42L7 DC    CL133' '                                                05620099
         ORG   PG1A42L7+1+8                                            05630099
P42GBP   DC    CL17' '                                                 05640099
P42GIA   DC    CL9' '                                                  05650099
P42GIB   DC    CL9' '                                                  05660099
P42GIC   DC    CL9' '                                                  05670099
P42GID   DC    CL9' '                                                  05680099
P42GIE   DC    CL9' '                                                  05690099
P42GIF   DC    CL9' '                                                  05700099
P42GIL   DC    CL9' '                                                  05710099
P42GIR   DC    CL9' '                                                  05720099
P42GIS   DC    CL9' '                                                  05730099
         ORG                                                           05740099
************ SMF2AGA1 ************                                     05750073
PGA12AL1 DC    CL133' SMF2AGA1 SYSPLEX D/S RESPONSE SUMMARY SET #:'    05760099
         ORG   PGA12AL1+51                                             05770099
P2ATGA1# DC    CL8' '                                                  05780073
         ORG                                                           05790073
P2ATGA1C DC    H'0'  TO COUNT THE NUMBER OF SETS                       05800073
PGA12AL2 DC    CL133' '                                                05810073
         ORG   PGA12AL2+1                                              05820073
P2ADTE3H DC    CL8'YYYYDDD '                                           05830073
P2ATMEH  DC    CL9'HH:MM:SS '                                          05840073
P2AGAAH  DC    CL9'SMF2AGAA'                                           05850073
P2AGABH  DC    CL44'SMF2AGAB'                                          05860099
P2AGACH  DC    CL45'SMF2AGAC'                                          05870073
         ORG                                                           05880073
```

Figure B-41 SMF42TG assembler source (part 8 of 10)

```
PGA12AL3 DC    CL133' '                                                 05890073
         ORG   PGA12AL3+1                                              05900073
P2ADTE3  DC    CL8' '                                                   05910073
P2ATME   DC    CL9'HH:MM:SS '                                           05920073
P2AGAA   DC    CL9' '                                                   05930073
P2AGAB   DC    CL44'SMF2AGAB'                                           05940099
P2AGAC   DC    CL45'SMF2AGAC'                                           05950073
         ORG                                                            05960073
PGA12AL4 DC    CL133' '                                                 05970073
         ORG   PGA12AL4+1+8                                            05980073
P2AGAEH  DC    CL27'SMF2AGAE'                                           05990099
P2AGAFH  DC    CL31'SMF2AGAF'                                           06000073
P2AGAHH  DC    CL32'SMF2AGAH'                                           06010073
P2AGAIH  DC    CL9'SMF2AGAI'  INTERPRETED                               06020099
P2AGAJH  DC    CL13'SMF2AGAJ'                                           06030099
         ORG                                                            06040073
PGA12AL5 DC    CL133' '                                                 06050073
         ORG   PGA12AL5+1+8                                            06060073
P2AGAE   DC    CL27' '                                                  06070099
P2AGAF   DC    CL31' '                                                  06080073
P2AGAH   DC    CL32' '                                                  06090073
P2AGAI   DC    CL9' '                                                   06100099
P2AGAJ   DC    CL13' '                                                  06110099
         ORG                                                            06120073
PGA12AL6 DC    CL133' '                                                 06130099
         ORG   PGA12AL6+1+8                                            06140099
P2AGAPH  DC    CL17'SMF2AGAP'                                           06150099
P2AGCAH  DC    CL9'SMF2AGCA'                                            06160099
P2AGCBH  DC    CL9'SMF2AGCB'                                            06170099
P2AGCCH  DC    CL9'SMF2AGCC'                                            06180099
P2AGCDH  DC    CL9'SMF2AGCD'                                            06190099
P2AGCEH  DC    CL9'SMF2AGCE'                                            06200099
P2AGCFH  DC    CL9'SMF2AGCF'                                            06210099
P2AGCIH  DC    CL9'SMF2AGCI'                                            06220099
P2AGCLH  DC    CL9'SMF2AGCL'                                            06230099
P2AGCMH  DC    CL9'SMF2AGCM'                                            06240099
P2AGCNH  DC    CL9'SMF2AGCN'                                            06250099
         ORG                                                            06260099
PGA12AL7 DC    CL133' '                                                 06270099
         ORG   PGA12AL7+1+8                                            06280099
P2AGAP   DC    CL17' '                                                  06290099
P2AGCA   DC    CL9' '                                                   06300099
P2AGCB   DC    CL9' '                                                   06310099
P2AGCC   DC    CL9' '                                                   06320099
P2AGCD   DC    CL9' '                                                   06330099
P2AGCE   DC    CL9' '                                                   06340099
P2AGCF   DC    CL9' '                                                   06350099
P2AGCI   DC    CL9' '                                                   06360099
P2AGCL   DC    CL9' '                                                   06370099
P2AGCM   DC    CL9' '                                                   06380099
P2AGCN   DC    CL9' '                                                   06390099
         ORG                                                            06400099
************ SMF2AG1A ************                                      06410073
PG1A2AL1 DC    CL133' SMF2AG1A DATA SET/MVS SYSTEM SUMMARY SET #:'      06420099
         ORG   PG1A2AL1+44                                             06430099
P2ATG1A# DC    CL8' '                                                   06440073
         ORG                                                            06450073
P2ATG1AC DC    H'0'  TO COUNT THE NUMBER OF SETS                        06460073
PG1A2AL2 DC    CL133' '                                                 06470073
         ORG   PG1A2AL2+1                                              06480073
P2ADTE5H DC    CL8'YYYYDDD '                                            06490073
P2ATME2H DC    CL9'HH:MM:SS '                                           06500073
P2AGBAH  DC    CL9'SMF2AGBA'                                            06510073
P2AGBBH  DC    CL45'SMF2AGBB'                                           06520073
P2AGBCH  DC    CL45'SMF2AGBC'                                           06530073
         ORG                                                            06540073
PG1A2AL3 DC    CL133' '                                                 06550073
         ORG   PG1A2AL3+1                                              06560073
P2ADTE5  DC    CL8' '                                                   06570073
P2ATME2  DC    CL9'HH:MM:SS '                                           06580073
P2AGBA   DC    CL9' '                                                   06590073
P2AGBB   DC    CL45' '                                                  06600073
P2AGBC   DC    CL45' '                                                  06610073
         ORG                                                            06620073
```

Figure B-42 SMF42TG assembler source (part 9 of 10)

```
PG1A2AL4 DC    CL133' '                                                 06630073
         ORG   PG1A2AL4+1+8                                            06640073
P2AGBEH  DC    CL30'SMF2AGBE'                                           06650099
P2AGBFH  DC    CL31'SMF2AGBF'                                           06660099
P2AGBHH  DC    CL32'SMF2AGBH'                                           06670099
P2AGBIH  DC    CL9'SMF2AGBI' INTERPRETED                                06680099
P2AAO9H  DC    CL13'SMF2AAO9'                                           06690099
         ORG                                                           06700073
PG1A2AL5 DC    CL133' '                                                 06710073
         ORG   PG1A2AL5+1+8                                            06720073
P2AGBE   DC    CL30' '                                                  06730099
P2AGBF   DC    CL31' '                                                  06740099
P2AGBH   DC    CL32' '                                                  06750099
P2AGBI   DC    CL9' '                                                   06760099
P2AAO9   DC    CL13' '                                                  06770099
         ORG                                                           06780073
PG1A2AL6 DC    CL133' '                                                 06790099
         ORG   PG1A2AL6+1+8                                            06800099
P2AGBPH  DC    CL17'SMF2AGBP'                                           06810099
P2AGIAH  DC    CL9'SMF2AGIA'                                            06820099
P2AGIBH  DC    CL9'SMF2AGIB'                                            06830099
P2AGICH  DC    CL9'SMF2AGIC'                                            06840099
P2AGIDH  DC    CL9'SMF2AGID'                                            06850099
P2AGIEH  DC    CL9'SMF2AGIE'                                            06860099
P2AGIFH  DC    CL9'SMF2AGIF'                                            06870099
P2AGILH  DC    CL9'SMF2AGIL'                                            06880099
P2AGIRH  DC    CL9'SMF2AGIR'                                            06890099
P2AGISH  DC    CL9'SMF2AGIS'                                            06900099
         ORG                                                           06910099
PG1A2AL7 DC    CL133' '                                                 06920099
         ORG   PG1A2AL7+1+8                                            06930099
P2AGBP   DC    CL17' '                                                  06940099
P2AGIA   DC    CL9' '                                                   06950099
P2AGIB   DC    CL9' '                                                   06960099
P2AGIC   DC    CL9' '                                                   06970099
P2AGID   DC    CL9' '                                                   06980099
P2AGIE   DC    CL9' '                                                   06990099
P2AGIF   DC    CL9' '                                                   07000099
P2AGIL   DC    CL9' '                                                   07010099
P2AGIR   DC    CL9' '                                                   07020099
P2AGIS   DC    CL9' '                                                   07030099
         ORG                                                           07040099
SMFDSECT DSECT                                                         07050073
*        IFASMFR  (42) THIS DOES NOT EXPAND THE SUBTYPES               07060043
         IGWSMF  SMF42_OG=YES                                          07070043
         END                                                           07080043
```

Figure B-43 SMF42TG assembler source (part 10 of 10)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-44 into your PDS MHLRES1.SMF42TGA.SOURCE as member SMF42TGJ. The result should contain 31 lines.

Run the job when the member has been created.

```
//MHLRES1L JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
/*JOBPARM S=*
//ASMHCL PROC
//ASM      EXEC PGM=ASMA90,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFES0),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFES0),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF42TG.SOURCE(SMF42TG)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TG.LOAD,DISP=(,CATLG,DELETE),
//      UNIT=SYSDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF42TG(R)
```

Figure B-44 SMF42TGJ JCL to assemble and link the program SMF42TG

This creates data set MHLRES1.SMF42TG.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TG.LOAD,DISP=(,CATLG,DELETE),

//           UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TG.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

Program SMF42TG alteration

The SMF records are generally TEXT or binary values, and when binary values, are usually fixed bin 32.

The program refers to each field four times:

- The SMF source field as mapped by the IGWSMF macro
- The location in the code where the field is copied or converted
- The storage location where the output line header is generated
- The storage location where the copied or converted data is stored

If text fields are changed, you need to take account of the length of the field. If the replacement is shorter than the field it replaces, the length does not have to be adjusted, but the output is longer than it need be. If the replacement field is longer than the field it replaces, the lengths must be adjusted to avoid overlay or truncation of storage. For example, the following simple changes can be made:

- Character changes

  If field SMF42A00 (which is currently reserved) was changed so that it contained useful information, and field SMF42GAJ was no longer needed, all references to GAJ could be changed to A00 and the program would change the headings and the source of the data when assembled and display the new information.

- Binary changes

  Not all currently defined fields in the IGWSMF macro for SMF record type 42 subtype 16 are defined in the program, as there are many. The program can be altered to swap one binary field for another easily. For example, if field SMF42GCA was no longer needed, but SMF42GDX was required, if all references to SMF42GCA were changed to SMF42GDX, the program would change the headings and the source of the data when assembled and display the new information.

The above the 2 GB bar part of the program fields defined with prefix SMF2A, and in the output definitions as P2A, work in the same manner as those defined SMF42 and P42 for the below 2 GB bar as described above.

SMF Record type 42 subtype 18 data display program

Program SMG42TI displays the contents of selected fields of SMF record Type 42 subtypes 18 data. It is not intended to provide a comprehensive report on RLS activity, but rather to verify that data is being allocated below or above the 2 GB bar as required.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF42TI.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-45 on page 535 and all subsequent contents through to Figure B-55 on page 545, one after the other, into member SMF42TI. Set the EDIT session up in NUMBER OFF mode. The result should contain 531 lines.

```
         MACRO                                                          00010091
&NAME    SEGSTART                                                       00020091
&NAME    STM   14,12,12(13)           SAVE HIS REGS IN HIS SAVE AREA   00030091
R0       EQU   0                                                        00040091
R1       EQU   1                                                        00050091
R2       EQU   2                                                        00060091
R3       EQU   3                                                        00070091
R4       EQU   4                                                        00080091
R5       EQU   5                                                        00090091
R6       EQU   6                                                        00100091
R7       EQU   7                                                        00110091
R8       EQU   8                                                        00120091
R9       EQU   9                                                        00130091
R10      EQU   10                                                       00140091
R11      EQU   11                                                       00150091
RB       EQU   12                                                       00160091
R13      EQU   13                                                       00170091
R14      EQU   14                                                       00180091
R15      EQU   15                                                       00190091
         BALR  12,0                  SET UP ADDRESSABILITY            00200091
         USING *,12                  USE REG 12 AS BASE REG           00210091
         USING *+4096,6                                               00220091
         LA    6,4095(12)                                             00230091
         LA    6,1(6)                                                 00240091
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE   00250091
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3  00260091
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS    00270091
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13 00280091
         MEND                                                         00290091
         MACRO                                                        00300091
&NAME2   SEGEND                                                       00310091
&NAME2   L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE    00320091
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA   00330091
         XR    R15,R15                                                00340091
         BR    14                    RETURN TO CALLING RTN VIA REG 14 00350091
SAVEREGS DC    18F'0'                SET UP SAVE AREA                 00360091
         MEND                                                         00370091
         MACRO                                                        00380091
         BINDEC   &KEY                                                00390091
         L     R7,SMF42&KEY                                           00400091
         CVD   R7,DWORD                                               00410091
         OI    DWORD+7,X'0F'                                          00420091
         UNPK  P42&KEY.(7),DWORD+4(4)                                 00430091
         MEND                                                         00440091
SMF42TI   SEGSTART                                                    00450091
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF      00460091
*  THE SMF TYPE 42 SUBTYPE 18 RECORDS, WHICH ARE THE SMS              00470091
*  COUPLING FACILITY CACHE PARTITION SUMMARY                          00480091
*  THE IFASMFDP PROGRAM MUST HAVE BEEN USED                           00490091
*  TO SELECT RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR     00500091
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.             00510091
*                                                                     00520091
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.                   00530091
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 42 RECORD                00540091
*  R3    START OF WHOLE RECORD                                        00550091
*  R4    START OF SMF42SI SECTION                                     00560091
```

Figure B-45 SMF42TI assembler source (part 1 of 11)

```
*  R5    START OF EACH SECTION IN BM LRU ACTIVITY GROUP                  00570091
*  OTHER REGISTER USES                                                   00580091
*  R12   OVERALL BASE REGISTER                                           00590091
*  R6    2ND BASE REGISTER                                               00600091
*  R7    RECORD TYPE/SUBTYPE CHECKING/WORKING                            00610091
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING                            00620091
*  R9    USED FOR OFFSET LENGTH ON TRIPLETS                              00630091
*  R10   USED FOR BCT                                                    00640091
*  R11   USED FOR TIME CONVERSION                                        00650091
*  R14   USED FOR TIME CONVERSION                                        00660091
*  R15   USED FOR TIME CONVERSION                                        00670091
*                                                                        00680091
         OPEN SMFIN               QSAM GET LOCATE PROCESSING IS USED   00690091
         OPEN (PRINTDCB,(OUTPUT))  QSAM PUT MOVE   PROCESSING IS USED   00700091
         PUT  PRINTDCB,PRINTHDR                                         00710091
READ     GET SMFIN                                                      00720091
         LR    R3,R1              COPY PARAMETER POINTER               00730091
         USING SMF42,R3          -> SMF RECORD                         00740091
*   CHECK IF TYPE 02 WHICH CAN ALWAYS BE SELECTED                       00750091
         CLI   SMF42RTY,X'02' (SAME DISPLAEMENT SMF ADMIN RECORD 02)    00760091
         BE    IGNORE                                                   00770091
         CLI   SMF42RTY,X'2A'      *   CHECK IF TYPE 42                00780091
         BNE   IGNORE                                                   00790091
CHKSTYP1 DS    0H                                                       00800091
         CLI   SMF42STY+1,X'12'    *   CHECK IF SUBTYPE 18              00810091
          BNE   IGNORE                                                  00820091
* IS TYPE 42 SUBTYPE 18, SO EXTRACT DATA                                00830091
* FIRST EXTRACT THE RECORD TIME AND DATE AND CONVERT TO HUMAN           00840091
* THEN  ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.               00850091
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION        00860091
* ADD R8 TO R3                                                          00870091
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS, HOWEVER MANY THERE ARE   00880091
* DO THE TIME & DATE EXTRACTS NOW SO THE REGS CAN BE REUSED IF NEEDED   00890091
         MVC   TIMEF,SMF42TME     SAVE THE SMF TIME IN 100THS OF SECS   00900091
*                                                                       00910091
         XR    R14,R14             CLEAR HIGH END OF PAIR              00920091
         L     R15,TIMEF           COMPLETE EVEN/ODD PAIR CONTENTS     00930091
         LA    R11,100            START BY DIVIDING BY 100 TO GET SECS 00940091
         DR    R14,R11             DIVIDE                              00950091
*         DC    F'0'  CREATE AN ABEND TO LOOK AT THE RECORDS            00960091
* REMAINDER NOW IN R14 WHICH WE IGNORE                                  00970091
* QUOTIENT IN R15 - IE SECONDS WHICH WE CONVERT TO MINS + SECS          00980091
         XR    R14,R14              CLEAR HIGH END OF PAIR             00990091
         LA    R11,60             DIVIDE BY 60 TO GET MINS             01000091
         DR    R14,R11          DIVIDE TO GET MINUTES & SECONDS AS REM. 01010091
* REMAINDER NOW IN R14 WHICH IS SECONDS WHICH WE MUST SAVE              01020091
* QUOTIENT IN R15 - IE MINUTES FOR MORE PROCESSING                      01030091
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL           01040091
         OI    TIMET+7,X'0F'      FIX SIGN                             01050091
         UNPK  TIMEX+6(2),TIMET+6(2)  UNPACK SECONDS                    01060091
         MVI   TIMEX+5,C':'                                            01070091
         XR    R14,R14              CLEAR HIGH END OF PAIR             01080091
         LA    R11,60             DIVIDE BY 60 TO GET HOURS            01090091
         DR    R14,R11          DIVIDE TO GET HOURS & MINUTES AS REM.  01100091
```

Figure B-46 SMF42TI assembler source (part 2 of 11

```
* REMAINDER NOW IN R14 WHICH IS MINUTES WHICH WE MUST SAVE              01110091
* QUOTIENT IN R15 - IE HOURS WHICH WE MUST SAVE                         01120091
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL           01130091
         OI    TIMET+7,X'0F'      FIX SIGN                             01140091
         UNPK  TIMEX+3(2),TIMET+6(2)  UNPACK MINUTES                   01150091
         MVI   TIMEX+2,C':'                                           01160091
         CVD   R15,TIMET           CONVERT TO PACKED DECIMAL           01170091
         OI    TIMET+7,X'0F'      FIX SIGN                             01180091
         UNPK  TIMEX+0(2),TIMET+6(2)  UNPACK HOURS                     01190091
         MVC   P42TME,TIMEX                                           01200091
         MVC   P42TME2,TIMEX                                          01210091
         MVC   P42TME3,TIMEX                                          01220091
         UNPK  P42DTE(7),SMF42DTE(4)                                  01230091
         OI    P42DTE+3,X'F0'                                         01240091
         CLC   P42DTE(2),=CL2'01'                                     01250091
         BNE   *+10                                                   01260091
         MVC   P42DTE(2),=C'20'                                       01270091
         MVC   P42DTE(2),=C'20'                                       01280091
         MVC   P42DTE2,P42DTE                                         01290091
         MVC   P42DTE3,P42DTE                                         01300091
         LA    R4,SMF42END  -> PRODUCT SECTION                        01310091
         USING SMF42SI,R4                                             01320091
IA142PRP EQU   *                                                      01330091
         L     R8,SMF42IM1  OFFSET                                    01340091
*                    TO SYSPLEX CACHE PARTITION TOTAKLS                01350091
         LH    R10,SMF42IM3   NUMBER OF COPIES OF THE DSECT           01360091
         LH    R9,SMF42IM2    LENGTH OF ALL SMF420JA DSECTS           01370091
         LTR  R10,R10                                                 01380091
         BZ    IC142PRP GO ON TO THE SINGLE CACHE                     01390091
         XR    R14,R14        CLEAR HIGH ORDER                        01400091
         STH   R14,P42TIACC   START THE COUNT AT 0                    01410091
         LR    R15,R9         LOAD LEGTH IN R9                        01420091
         DR    R14,R10         GET LENGTH OF SMF420JA DSECTS          01430091
         LR    R9,R15                                                 01440091
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                 01450091
*                           THE START OF THE RECORD                   01460091
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                 01470091
IA142TRP DS    0H                                                     01480091
         LH    R14,P42TIACC  GET CURRENT COUNT                        01490091
         LA    R14,1(R14)    ADD ONE                                  01500091
         STH   R14,P42TIACC  PUT CURRENT COUNT                        01510091
         CVD   R14,DWORD                                              01520091
         OI    DWORD+7,X'0F'                                          01530091
         UNPK  P42TIAC#(7),DWORD+4(4)                                 01540091
         PUT   PRINTDCB,PIA142L1                                      01550091
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                         01560091
         USING SMF420IA,R5                                            01570091
         BINDEC IAA                                                   01580091
         MVC   P42IBG(16),SMF42IBG                                    01590091
         BINDEC IBH                                                   01600091
         MVC   P42IO1(12),SMF42IO1                                    01610091
         BINDEC IAD                                                   01620091
         BINDEC IAF                                                   01630091
         BINDEC IAG                                                   01640091
         BINDEC IAH                                                   01650091
```

Figure B-47 SMF42TI assembler source (part 3 of 11

```
         BINDEC IAI                                                    01660091
         BINDEC IAJ                                                    01670091
         PUT   PRINTDCB,PIA142L2                                       01680091
         PUT   PRINTDCB,PIA142L3                                       01690091
         BINDEC IAK                                                    01700091
         BINDEC IAL                                                    01710091
         BINDEC IAM                                                    01720091
         BINDEC IAN                                                    01730091
         BINDEC IAO                                                    01740091
         BINDEC IAP                                                    01750091
         BINDEC IAQ                                                    01760091
         BINDEC IAR                                                    01770091
         BINDEC IAS                                                    01780091
         BINDEC IAT                                                    01790091
         BINDEC IAU                                                    01800091
         BINDEC IAV                                                    01810091
         PUT   PRINTDCB,PIA142L4                                       01820091
         PUT   PRINTDCB,PIA142L5                                       01830091
         BINDEC IAW                                                    01840091
         BINDEC IAX                                                    01850091
         BINDEC IAY                                                    01860091
         BINDEC IAZ                                                    01870091
         BINDEC IBA                                                    01880091
         BINDEC IBB                                                    01890091
         BINDEC IBC                                                    01900091
         BINDEC IBD                                                    01910091
         BINDEC IBE                                                    01920091
         BINDEC IBF                                                    01930091
         DROP  R5                                                      01940091
         PUT   PRINTDCB,PIA142L6                                       01950091
         PUT   PRINTDCB,PIA142L7                                       01960091
         PUT   PRINTDCB,PRINTBLK                                       01970091
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS              01980091
         BCT   R10,IA142TRP                                            01990091
IC142PRP EQU   *                                                      02000091
         L     R8,SMF42IM4  OFFSET                                     02010091
*                    TO PARTITION FOR SINGE CACHE STRUCTURE           02020091
         LH    R10,SMF42IM6   NUMBER OF COPIES OF THE DSECT            02030091
         LH    R9,SMF42IM5    LENGTH OF BM SECTION                     02040091
         LTR  R10,R10                                                 02050091
         BZ    READ                                                   02060091
         XR    R14,R14        CLEAR HIGH ORDER                        02070091
         STH   R14,P42TIC1C   START THE COUNT AT 0                    02080091
         LR    R15,R9         LOAD LEGTH IN R9                        02090091
         DR    R14,R10         GET LENGTH OF SMF42OPA DSECTS          02100091
         LR    R9,R15                                                 02110091
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                 02120091
*                           THE START OF THE RECORD                  02130091
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                 02140091
IC142TRP DS    0H                                                     02150091
         LH    R14,P42TIC1C  GET CURRENT COUNT                        02160091
         LA    R14,1(R14)    ADD ONE                                  02170091
         STH   R14,P42TIC1C  PUT CURRENT COUNT                        02180091
         CVD   R14,DWORD                                              02190091
         OI    DWORD+7,X'0F'                                          02200091
```

Figure B-48 SMF42TI assembler source (part 4 of 11

```
         UNPK  P42TIC1#(7),DWORD+4(4)                                  02210091
         PUT   PRINTDCB,PIC142L1                                       02220091
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                          02230091
         USING SMF420IC,R5                                             02240091
         BINDEC ICA                                                    02250091
         MVC   P42ICB(26),SMF42ICB                                     02260091
         MVC   P42IDG(16),SMF42IDG                                     02270091
         BINDEC IDH                                                    02280091
         BINDEC ICD                                                    02290091
         BINDEC ICF                                                    02300091
         BINDEC ICG                                                    02310091
         BINDEC ICH                                                    02320091
         PUT   PRINTDCB,PIC142L2                                       02330091
         PUT   PRINTDCB,PIC142L3                                       02340091
         BINDEC ICI                                                    02350091
         BINDEC ICJ                                                    02360091
         BINDEC ICK                                                    02370091
         BINDEC ICL                                                    02380091
         BINDEC ICM                                                    02390091
         BINDEC ICN                                                    02400091
         BINDEC ICO                                                    02410091
         BINDEC ICP                                                    02420091
         BINDEC ICQ                                                    02430091
         BINDEC ICR                                                    02440091
         BINDEC ICS                                                    02450091
         PUT   PRINTDCB,PIC142L4                                       02460091
         PUT   PRINTDCB,PIC142L5                                       02470091
         BINDEC ICT                                                    02480091
         BINDEC ICU                                                    02490091
         BINDEC ICV                                                    02500091
         BINDEC ICW                                                    02510091
         BINDEC ICX                                                    02520091
         BINDEC ICY                                                    02530091
         BINDEC ICZ                                                    02540091
         BINDEC IDA                                                    02550091
         BINDEC IDB                                                    02560091
         BINDEC IDC                                                    02570091
         BINDEC IDD                                                    02580091
         BINDEC IDE                                                    02590091
         BINDEC IDF                                                    02600091
         DROP  R5                                                      02610091
         PUT   PRINTDCB,PIC142L6                                       02620091
         PUT   PRINTDCB,PIC142L7                                       02630091
         PUT   PRINTDCB,PRINTBLK                                       02640091
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               02650091
         BCT   R10,IC142TRP                                            02660091
IE142PRP EQU   *                                                       02670091
         L     R8,SMF42IM7  OFFSET                                     02680091
*                    TO SYSPLEX TOTALS LOCAL BM LRU STATS SUMM         02690091
*                    BELOW THE LINE                                   02700091
         LH    R10,SMF42IM9   NUMBER OF COPIES OF THE DSECT            02710091
         LH    R9,SMF42IM8    LENGTH OF ALL SMF420JA DSECTS            02720091
         LTR  R10,R10                                                  02730091
         BZ    READ     GO GET ANOTHER RECORD                         02740091
```

Figure B-49 SMF42TI assembler source (part 5 of 11

```
         XR    R14,R14        CLEAR HIGH ORDER                          02750091
         STH   R14,P42TIE1C   START THE COUNT AT 0                      02760091
         LR    R15,R9         LOAD LEGTH IN R9                          02770091
         DR    R14,R10         GET LENGTH OF SMF42OJA DSECTS            02780091
         LR    R9,R15                                                   02790091
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                   02800091
*                            THE START OF THE RECORD                   02810091
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   02820091
IE142TRP DS    OH                                                       02830091
         LH    R14,P42TIE1C  GET CURRENT COUNT                          02840091
         LA    R14,1(R14)    ADD ONE                                    02850091
         STH   R14,P42TIE1C  PUT CURRENT COUNT                          02860091
         CVD   R14,DWORD                                                02870091
         OI    DWORD+7,X'0F'                                            02880091
         UNPK  P42TIE1#(7),DWORD+4(4)                                   02890091
         PUT   PRINTDCB,PIE142L1                                        02900091
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                           02910091
         USING SMF42OIE,R5                                              02920091
         BINDEC IEA                                                     02930091
         MVC   P42IEB(26),SMF42IEB                                      02940091
         BINDEC IEC                                                     02950091
         BINDEC IED                                                     02960091
         BINDEC IEE                                                     02970091
         BINDEC IEF                                                     02980091
         BINDEC IEG                                                     02990091
         BINDEC IEH                                                     03000091
         BINDEC IEI                                                     03010091
         BINDEC IEJ                                                     03020091
         DROP  R5                                                       03030091
         PUT   PRINTDCB,PIE142L2                                        03040091
         PUT   PRINTDCB,PIE142L3                                        03050091
         PUT   PRINTDCB,PRINTBLK                                        03060091
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               03070091
         BCT   R10,IE142TRP                                             03080091
         B     READ                                                     03090091
IGNORE   DS    OH EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS        03100091
         B     READ                                                     03110091
FINI     DS    OH                                                       03120091
         SEGEND                                                         03130091
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,EODAD=FINI    03140091
DWORD    DS    D                                                        03150091
         ORG   DWORD                                                    03160091
         DC    C'12345678'                                              03170091
TIMET    DS    D                WORKAREA FOR TIME CONVERSION            03180091
TIMEX    DS    D                WORKAREA FOR TIME CONVERSION            03190091
TIMEF    DS    F                WORKAREA FOR TIME CONVERSION            03200091
PRINTBLK DC    CL133' '                                                 03210091
PRINTHDR DC    CL133'1SMF TYPE 42 S/TYPE 18 RECS. COLS USE SMF NAMES'   03220091
```

Figure B-50 SMF42TI assembler source (part 6 of 11

```
************* SMF42IA1 ************                                  03230091
PIA142L1 DC    CL133' SMF42IA1 TOTALS SET #:'                        03240091
         ORG   PIA142L1+26                                           03250091
P42TIAC# DC    CL8' '                                                03260091
         ORG                                                         03270091
P42TIACC DC    H'0'  TO COUNT THE NUMBER OF SETS                     03280091
PIA142L2 DC    CL133' '                                              03290091
         ORG   PIA142L2+1                                            03300091
P42TMEH  DC    CL8'HH:MM:SS '                                        03310091
         DC    CL1' '                                                03320091
P42DTEH  DC    CL8'YYYYDDD '                                         03330091
P42IAAH  DC    CL9'SMF42IAA'                                         03340091
P42IBGH  DC    CL18'SMF42IBG'                                        03350091
P42IBHH  DC    CL9'SMF42IBH'                                         03360091
P42IO1H  DC    CL18'SMF42IO1'                                        03370091
P42IADH  DC    CL9'SMF42IAD'                                         03380091
P42IAFH  DC    CL9'SMF42IAF'                                         03390091
P42IAGH  DC    CL9'SMF42IAG'                                         03400091
P42IAHH  DC    CL9'SMF42IAH'                                         03410091
P42IAIH  DC    CL9'SMF42IAI'                                         03420091
P42IAJH  DC    CL9'SMF42IAJ'                                         03430091
         ORG                                                         03440091
PIA142L3 DC    CL133' '                                              03450091
         ORG   PIA142L3+1                                            03460091
P42TME   DC    CL8' '                                                03470091
         DC    CL1' '                                                03480091
P42DTE   DC    CL8'YYDDDFI '                                         03490091
P42IAA   DC    CL9' '                                                03500091
P42IBG   DC    CL18' '                                               03510091
P42IBH   DC    CL9' '                                                03520091
P42IO1   DC    CL18' '                                               03530091
P42IAD   DC    CL9' '                                                03540091
P42IAF   DC    CL9' '                                                03550091
P42IAG   DC    CL9' '                                                03560091
P42IAH   DC    CL9' '                                                03570091
P42IAI   DC    CL9' '                                                03580091
P42IAJ   DC    CL9' '                                                03590091
         ORG                                                         03600091
PIA142L4 DC    CL133' '                                              03610091
         ORG   PIA142L4+1+8+1+8                                      03620091
P42IAKH  DC    CL9'SMF42IAK'                                         03630091
P42IALH  DC    CL9'SMF42IAL'                                         03640091
P42IAMH  DC    CL9'SMF42IAM'                                         03650091
P42IANH  DC    CL9'SMF42IAN'                                         03660091
P42IAOH  DC    CL9'SMF42IAO'                                         03670091
P42IAPH  DC    CL9'SMF42IAP'                                         03680091
P42IAQH  DC    CL9'SMF42IAQ'                                         03690091
P42IARH  DC    CL9'SMF42IAR'                                         03700091
P42IASH  DC    CL9'SMF42IAS'                                         03710091
P42IATH  DC    CL9'SMF42IAT'                                         03720091
P42IAUH  DC    CL9'SMF42IAU'                                         03730091
P42IAVH  DC    CL9'SMF42IAV'                                         03740091
         ORG                                                         03750091
```

Figure B-51 SMF42TI assembler source (part 7 of 11

```
PIA142L5 DC    CL133' '                                                 03760091
         ORG   PIA142L5+1+8+1+8                                         03770091
P42IAK   DC    CL9' '   SMF42IAK                                       03780091
P42IAL   DC    CL9' '   SMF42IAL                                       03790091
P42IAM   DC    CL9' '   SMF42IAM                                       03800091
P42IAN   DC    CL9' '   SMF42IAN                                       03810091
P42IAO   DC    CL9' '    SMF42IAO                                      03820091
P42IAP   DC    CL9' '    SMF42IAP                                      03830091
P42IAQ   DC    CL9' '    SMF42IAQ                                      03840091
P42IAR   DC    CL9' '    SMF42IAR                                      03850091
P42IAS   DC    CL9' '    SMF42IAS                                      03860091
P42IAT   DC    CL9' '    SMF42IAT                                      03870091
P42IAU   DC    CL9' '    SMF42IAU                                      03880091
P42IAV   DC    CL9' '    SMF42IAV                                      03890091
         ORG                                                            03900091
PIA142L6 DC    CL133' '                                                 03910091
         ORG   PIA142L6+1+8+1+8                                         03920091
P42IAWH  DC    CL9'SMF42IAW'                                            03930091
P42IAXH  DC    CL9'SMF42IAX'                                            03940091
P42IAYH  DC    CL9'SMF42IAY'                                            03950091
P42IAZH  DC    CL9'SMF42IAZ'                                            03960091
P42IBAH  DC    CL9'SMF42IBA'                                            03970091
P42IBBH  DC    CL9'SMF42IBB'                                            03980091
P42IBCH  DC    CL9'SMF42IBC'                                            03990091
P42IBDH  DC    CL9'SMF42IBD'                                            04000091
P42IBEH  DC    CL9'SMF42IBE'                                            04010091
P42IBFH  DC    CL9'SMF42IBF'                                            04020091
         ORG                                                            04030091
PIA142L7 DC    CL133' '                                                 04040091
         ORG   PIA142L7+1+8+1+8                                         04050091
P42IAW   DC    CL9' '   SMF42IAW                                       04060091
P42IAX   DC    CL9' '   SMF42IAX                                       04070091
P42IAY   DC    CL9' '   SMF42IAY                                       04080091
P42IAZ   DC    CL9' '   SMF42IAZ                                       04090091
P42IBA   DC    CL9' '    SMF42IBA                                      04100091
P42IBB   DC    CL9' '    SMF42IBB                                      04110091
P42IBC   DC    CL9' '    SMF42IBC                                      04120091
P42IBD   DC    CL9' '    SMF42IBD                                      04130091
P42IBE   DC    CL9' '    SMF42IBE                                      04140091
P42IBF   DC    CL9' '    SMF42IBF                                      04150091
         ORG                                                            04160091
************ SMF42IC1 ************                                      04170091
PIC142L1 DC    CL133' SMF42IC1 TOTALS SET #:'                           04180091
         ORG   PIC142L1+26                                              04190091
P42TIC1# DC    CL8' '                                                   04200091
         ORG                                                            04210091
P42TIC1C DC    H'0'  TO COUNT THE NUMBER OF SETS                        04220091
PIC142L2 DC    CL133' '                                                 04230091
         ORG   PIC142L2+1                                               04240091
P42TMEH2 DC    CL8'HH:MM:SS '                                           04250091
         DC    CL1' '                                                   04260091
P42DTEH2 DC    CL8'YYYYDDD '                                            04270091
P42ICAH  DC    CL9'SMF42ICA'                                            04280091
P42ICBH  DC    CL27'SMF42ICB'                                           04290091
```

Figure B-52 SMF42TI assembler source (part 8 of 11

```
P42IDGH  DC    CL18'SMF42IDG'                                    04300091
P42IDHH  DC    CL9'SMF42IDH'                                     04310091
P42ICDH  DC    CL9'SMF42ICD'                                     04320091
P42ICFH  DC    CL9'SMF42ICF'                                     04330091
P42ICGH  DC    CL9'SMF42ICG'                                     04340091
P42ICHH  DC    CL9'SMF42ICH'                                     04350091
         ORG                                                     04360091
PIC142L3 DC    CL133' '                                          04370091
         ORG   PIC142L3+1                                        04380091
P42TME2  DC    CL8' '                                            04390091
         DC    CL1' '                                            04400091
P42DTE2  DC    CL8' '                                            04410091
P42ICA   DC    CL9' '     SMF42ICA                               04420091
P42ICB   DC    CL27' '                                           04430091
P42IDG   DC    CL18' '                                           04440091
P42IDH   DC    CL9' '                                            04450091
P42ICD   DC    CL9' '                                            04460091
P42ICF   DC    CL9' '                                            04470091
P42ICG   DC    CL9' '                                            04480091
P42ICH   DC    CL9' '                                            04490091
         ORG                                                     04500091
PIC142L4 DC    CL133' '                                          04510091
         ORG   PIC142L4+1+8+1+8                                  04520091
P42ICIH  DC    CL9'SMF42ICI'                                     04530091
P42ICJH  DC    CL9'SMF42ICJ'                                     04540091
P42ICKH  DC    CL9'SMF42ICK'                                     04550091
P42ICLH  DC    CL9'SMF42ICL'                                     04560091
P42ICMH  DC    CL9'SMF42ICM'                                     04570091
P42ICNH  DC    CL9'SMF42ICN'                                     04580091
P42ICOH  DC    CL9'SMF42ICO'                                     04590091
P42ICPH  DC    CL9'SMF42ICP'                                     04600091
P42ICQH  DC    CL9'SMF42ICQ'                                     04610091
P42ICRH  DC    CL9'SMF42ICR'                                     04620091
P42ICSH  DC    CL9'SMF42ICS'                                     04630091
         ORG                                                     04640091
PIC142L5 DC    CL133' '                                          04650091
         ORG   PIC142L5+1+8+1+8                                  04660091
P42ICI   DC    CL9' '                                            04670091
P42ICJ   DC    CL9' '                                            04680091
P42ICK   DC    CL9' '                                            04690091
P42ICL   DC    CL9' '                                            04700091
P42ICM   DC    CL9' '                                            04710091
P42ICN   DC    CL9' '                                            04720091
P42ICO   DC    CL9' '                                            04730091
P42ICP   DC    CL9' '                                            04740091
P42ICQ   DC    CL9' '                                            04750091
P42ICR   DC    CL9' '                                            04760091
P42ICS   DC    CL9' '                                            04770091
         ORG                                                     04780091
```

Figure B-53 SMF42TI assembler source (part 9 of 11

```
PIC142L6 DC    CL133' '                                                 04790091
         ORG   PIC142L6+1+8+1+8                                         04800091
P42ICTH  DC    CL9'SMF42ICT'                                            04810091
P42ICUH  DC    CL9'SMF42ICU'                                            04820091
P42ICVH  DC    CL9'SMF42ICV'                                            04830091
P42ICWH  DC    CL9'SMF42ICW'                                            04840091
P42ICXH  DC    CL9'SMF42ICX'                                            04850091
P42ICYH  DC    CL9'SMF42ICY'                                            04860091
P42ICZH  DC    CL9'SMF42ICZ'                                            04870091
P42IDAH  DC    CL9'SMF42IDA'                                            04880091
P42IDBH  DC    CL9'SMF42IDB'                                            04890091
P42IDCH  DC    CL9'SMF42IDC'                                            04900091
P42IDDH  DC    CL9'SMF42IDD'                                            04910091
P42IDEH  DC    CL9'SMF42IDE'                                            04920091
P42IDFH  DC    CL9'SMF42IDF'                                            04930091
         ORG                                                            04940091
PIC142L7 DC    CL133' '                                                 04950091
         ORG   PIC142L7+1+8+1+8                                         04960091
P42ICT   DC    CL9' '                                                   04970091
P42ICU   DC    CL9' '                                                   04980091
P42ICV   DC    CL9' '                                                   04990091
P42ICW   DC    CL9' '                                                   05000091
P42ICX   DC    CL9' '                                                   05010091
P42ICY   DC    CL9' '                                                   05020091
P42ICZ   DC    CL9' '                                                   05030091
P42IDA   DC    CL9' '                                                   05040091
P42IDB   DC    CL9' '                                                   05050091
P42IDC   DC    CL9' '                                                   05060091
P42IDD   DC    CL9' '                                                   05070091
P42IDE   DC    CL9' '                                                   05080091
P42IDF   DC    CL9' '                                                   05090091
         ORG                                                            05100091
************* SMF42IE1 ************                                     05110091
PIE142L1 DC    CL133' SMF42IE1 TOTALS SET #:'                           05120091
         ORG   PIE142L1+26                                              05130091
P42TIE1# DC    CL8' '                                                   05140091
         ORG                                                            05150091
P42TIE1C DC    H'0'  TO COUNT THE NUMBER OF SETS                        05160091
PIE142L2 DC    CL133' '                                                 05170091
         ORG   PIE142L2+1                                               05180091
P42TMEH3 DC    CL8'HH:MM:SS '                                           05190091
         DC    CL1' '                                                   05200091
P42DTEH3 DC    CL8'YYYYDDD '                                            05210091
P42IEAH  DC    CL9'SMF42IEA'                                            05220091
P42IEBH  DC    CL27'SMF42IEB'                                           05230091
P42IECH  DC    CL9'SMF42IEC'                                            05240091
P42IEDH  DC    CL9'SMF42IED'                                            05250091
P42IEEH  DC    CL9'SMF42IEE'                                            05260091
P42IEFH  DC    CL9'SMF42IEF'                                            05270091
P42IEGH  DC    CL9'SMF42IEG'                                            05280091
P42IEHH  DC    CL9'SMF42IEH'                                            05290091
P42IEIH  DC    CL9'SMF42IEI'                                            05300091
P42IEJH  DC    CL9'SMF42IEJ'                                            05310091
         ORG                                                            05320091
```

Figure B-54 SMF42TI assembler source (part 10 of 11

```
PIE142L3 DC    CL133' '                                                 05330091
         ORG   PIE142L3+1                                               05340091
P42TME3  DC    CL8' '                                                   05350091
         DC    CL1' '                                                   05360091
P42DTE3  DC    CL8'YYDDDFIL'                                            05370091
P42IEA   DC    CL9' '   SMF42IEA                                       05380091
P42IEB   DC    CL27' '                                                  05390091
P42IEC   DC    CL9' '                                                   05400091
P42IED   DC    CL9' '                                                   05410091
P42IEE   DC    CL9' '                                                   05420091
P42IEF   DC    CL9' '                                                   05430091
P42IEG   DC    CL9' '                                                   05440091
P42IEH   DC    CL9' '                                                   05450091
P42IEI   DC    CL9' '                                                   05460091
P42IEJ   DC    CL9' '                                                   05470091
         ORG                                                            05480091
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133               05490091
SMFDSECT DSECT                                                          05500091
*        IFASMFR  (42) THIS DOES NOT EXPAND THE SUBTYPES AS IT SHOULD   05510091
         IGWSMF  SMF42_OI=YES                                           05520091
         END                                                            05530091
```

Figure B-55 SMF42TI assembler source (part 11 of 11)

Note 1: In the SMF42TI code, fields SMF42ICB and SMF42IEB have been slightly truncated in the output to improve the format. The records involved (Cache-structure name) are unlikely to have been defined to their full length on most systems.

Note 2: In the SMF42TI code, field SMF42IDF has been defined in the output and will be populated when the program runs. However, the addition of the output field for SMF42IDF (P42IDF) takes the line length over the defined 133 characters. This apparent error has been included to demonstrate the possibility of defining the length too long. The situation can be changed by changing the program to use longer print line definitions where they occur, and on the LRECL statement on the PRINTDCB MACRO definition.

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-56 into your PDS MHLRES1.SMF42TI.SOURCE as member SMF42TIJ. The result should contain 31 lines.

Run the job when the member has been created.

```
//MHLRES1L JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
/*JOBPARM S=*
//ASMHCL PROC
//ASM      EXEC PGM=ASMA90,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF42TI.SOURCE(SMF42TI)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TI.LOAD,DISP=(,CATLG,DELETE),
//      UNIT=SYSDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF42TI(R)
```

Figure B-56 SMF42TIJ JCL to assemble and link the program SMF42TI

This creates data set MHLRES1.SMF42TI.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TI.LOAD,DISP=(,CATLG,DELETE),

//           UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TI.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

Program SMF42TI alteration

The SMF records are generally TEXT or binary values, and when binary values, are usually fixed bin 32.

The program refers to each field four times:

- The SMF source field as mapped by the IGWSMF macro
- The location in the code where the field is copied or converted
- The storage location where the output line header is generated
- The storage location where the copied or converted data is stored

If text fields are changed, you need to take account of the length of the field. If the replacement is shorter than the field it replaces, the length does not have to be adjusted, but the output is longer than it need be. If the replacement field is longer than the field it replaces, the lengths must be adjusted to avoid overlay or truncation of storage. For example, the following simple changes can be made:

- Character changes

  If field SMF42I00 (which is currently reserved) was changed so that it contained useful information, and field SMF42IBG was no longer needed, all references to IBG could be changed to I00 and the program would change the headings and the source of the data when assembled and display the new information.

- Binary changes

  All currently defined fields in the IGWSMF macro for SMF record type 42 subtype 18 are used in the program, but as commented in step 2 above, this results in a longer line length than defined for printing.

  If field SMF42IDE was no longer needed, but SMF42IDF was required, if all references to IDE could be removed and the program would change the headings and the source of the data when assembled and display the new information.

SMF record type 42 subtype 19 data display program

Program SMG42TI displays the contents of selected fields of SMF record type 42 subtypes 19 data. It is not intended to provide a comprehensive report on RLS activity but rather to verify that data is being allocated below or above the 2 GB bar as required.

There are three steps to build the program, which needs to be done once, after which it can be executed several times. It is not necessary to have in-depth assembler experience, but familiarity with JCL is required.

Step 1: create a PDS/PDSE to hold the members

In this example the PDS is called MHLRES1.SMF42TJ.SOURCE. The LRECL/RECFM must be 80/FB. All other attributes can be chosen by you.

Step 2: store the program source in the PDS

Cut and paste the contents of Figure B-57 on page 548 and all subsequent contents through to Figure B-67 on page 558, one after the other, into member SMF42TJA. Set the EDIT session up in NUMBER OFF mode. The result should contain 498 lines.

```
         MACRO                                                          00010099
&NAME    SEGSTART                                                       00020099
&NAME    STM   14,12,12(13)           SAVE HIS REGS IN HIS SAVE AREA   00030099
R0       EQU   0                                                        00040099
R1       EQU   1                                                        00050099
R2       EQU   2                                                        00060099
R3       EQU   3                                                        00070099
R4       EQU   4                                                        00080099
R5       EQU   5                                                        00090099
R6       EQU   6                                                        00100099
R7       EQU   7                                                        00110099
R8       EQU   8                                                        00120099
R9       EQU   9                                                        00130099
R10      EQU   10                                                       00140099
R11      EQU   11                                                       00150099
RB       EQU   12                                                       00160099
R13      EQU   13                                                       00170099
R14      EQU   14                                                       00180099
R15      EQU   15                                                       00190099
         BALR  12,0                  SET UP ADDRESSABILITY            00200099
         USING *,12                  USE REG 12 AS BASE REG           00210099
         ST    13,SAVEREGS+4         SAVE @ OF HIS SAVEAREA IN MINE   00220099
         LA    03,SAVEREGS           LOAD @ OF MY SAVE AREA IN REG 3  00230099
         ST    03,8(13)              SAVE @ OF MY SAVE AREA IN HIS    00240099
         LR    13,03                 LOAD @ OF MY SAVE AREA IN REG 13 00250099
         MEND                                                           00260099
         MACRO                                                          00270099
&NAME2   SEGEND                                                         00280099
&NAME2   L     13,SAVEREGS+4         LOAD REG13 WITH @ OF HIS SAVE    00290099
         LM    14,12,12(13)          RESTORE REGS FROM HIS SAVEAREA   00300099
         XR    R15,R15                                                00310099
         BR    14                    RETURN TO CALLING RTN VIA REG 14 00320099
SAVEREGS DC    18F'0'                SET UP SAVE AREA                 00330099
         MEND                                                           00340099
         MACRO                                                          00350099
         BINDEC   &KEY                                                  00360099
         L     R7,SMF42&KEY                                           00370099
         CVD   R7,DWORD                                               00380099
         OI    DWORD+7,X'0F'                                          00390099
         UNPK  P42&KEY.(7),DWORD+4(4)                                 00400099
         MEND                                                           00410099
         MACRO                                                          00420099
         BINDECA  &KEY                                                  00430099
         L     R7,SMF2A&KEY                                           00440099
         CVD   R7,DWORD                                               00450099
         OI    DWORD+7,X'0F'                                          00460099
         UNPK  P2A&KEY.(7),DWORD+4(4)                                 00470099
         MEND                                                           00480099
SMF42TJA  SEGSTART                                                      00490099
*  THIS IS A SIMPLE PROGRAM TO DISPLAY THE CONTENTS OF VARIOUS OF       00500099
*  THE SMF TYPE 42 SUBTYPE 19 RECORDS, WHICH ARE OF THE SMS             00510099
*  BUFFER MANAGER LRU ACTIVITY                                          00520099
*  THE IFASMFDP PROGRAM MUST HAVE BEEN USED                             00530099
*  TO SELECT RECORDS FROM EITHER THE ACTIVE SMF 'MAN' DATASETS OR       00540099
*  OFF A PREVIOUSLY EXTRACTED COPY OF THE 'MAN' DATASETS.               00550099
*                                                                       00560099
```

Figure B-57 SMF42TJ assembler source (part 1 of 11)

```
*  THE STANDARD SMF RECORD MAPPING MACROS ARE USED.                      00570099
*  REGISTER EQUATES TO PARTS OF THE SMF TYPE 42 RECORD                   00580099
*  R3    START OF WHOLE RECORD                                           00590099
*  R4    START OF SMF42SJ SECTION                                        00600099
*  R5    START OF EACH SECTION IN BM LRU ACTIVITY GROUP                  00610099
*  R6    TRACE                                                           00620099
*  OTHER REGISTER USES                                                   00630099
*  R12   OVERALL BASE REGISTER                                           00640099
*  R7    RECORD TYPE/SUBTYPE CHECKING/WORKING                            00650099
*  R8    RECORD TYPE/SUBTYPE CHECKING/WORKING                            00660099
*  R9    USED FOR OFFSET LENGTH ON TRIPLETS                              00670099
*  R10   USED FOR BCT                                                    00680099
*  R11   USED FOR TIME CONVERSION                                        00690099
*  R14   USED FOR TIME CONVERSION                                        00700099
*  R15   USED FOR TIME CONVERSION                                        00710099
*                                                                        00720099
         OPEN SMFIN                QSAM GET LOCATE PROCESSING IS USED   00730099
         OPEN (PRINTDCB,(OUTPUT))  QSAM PUT MOVE   PROCESSING IS USED   00740099
         PUT  PRINTDCB,PRINTHDR                                         00750099
READ     GET SMFIN                                                      00760099
         LR    R3,R1               COPY PARAMETER POINTER               00770099
         USING SMF42,R3           -> SMF RECORD                         00780099
*   CHECK IF TYPE 02 WHICH CAN ALWAYS BE SELECTED                       00790099
         CLI   SMF42RTY,X'02' (SAME DISPLAEMENT SMF ADMIN RECORD 02)    00800099
         BE    IGNORE                                                   00810099
         CLI   SMF42RTY,X'2A'      *   CHECK IF TYPE 42                00820099
         BNE   IGNORE                                                   00830099
CHKSTYP1 DS    0H                                                       00840099
         CLI   SMF42STY+1,X'13'    *   CHECK IF SUBTYPE 19              00850099
         BNE   IGNORE                                                   00860099
* IS TYPE 42 SUBTYPE 19, SO EXTRACT DATA                                00870099
* FIRST EXTRACT THE RECORD TIME AND DATE AND CONVERT TO READABLE        00880099
* THEN  ESTABLISH ADDRESSIBILITY TO THE VARIOUS SECTIONS.               00890099
* GENERAL PROCESS IS LOAD R8 WITH OFFSET TO THE RELEVANT SECTION        00900099
* ADD R8 TO R3 THEN INDEX IT THOUGH R5.                                 00910099
* THEN THE DSECTS SHOULD ADDRESS THE SECTIONS, HOWEVER MANY THERE ARE   00920099
* DO THE TIME & DATE EXTRACTS NOW SO THE REGS CAN BE REUSED IF NEEDED   00930099
         MVC   TIMEF,SMF42TME      SAVE THE SMF TIME IN 100THS OF SECS  00940099
*                                                                       00950099
         XR    R14,R14              CLEAR HIGH END OF PAIR             00960099
         L     R15,TIMEF           COMPLETE EVEN/ODD PAIR CONTENTS     00970099
         LA    R11,100            START BY DIVIDING BY 100 TO GET SECS 00980099
         DR    R14,R11              DIVIDE                             00990099
* REMAINDER NOW IN R14 WHICH WE IGNORE                                  01000099
* QUOTIENT IN R15 - IE SECONDS WHICH WE CONVERT TO MINS + SECS          01010099
         XR    R14,R14               CLEAR HIGH END OF PAIR            01020099
         LA    R11,60              DIVIDE BY 60 TO GET MINS            01030099
         DR    R14,R11           DIVIDE TO GET MINUTES & SECONDS AS REM. 01040099
* REMAINDER NOW IN R14 WHICH IS SECONDS WHICH WE MUST SAVE              01050099
```

Figure B-58 SMF42TJ assembler source (part 2 of 11)

```
* QUOTIENT IN R15 - IE MINUTES FOR MORE PROCESSING                       01060099
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL            01070099
         OI    TIMET+7,X'0F'      FIX SIGN                              01080099
         UNPK  TIMEX+6(2),TIMET+6(2)  UNPACK SECONDS                    01090099
         MVI   TIMEX+5,C':'                                            01100099
         XR    R14,R14              CLEAR HIGH END OF PAIR              01110099
         LA    R11,60             DIVIDE BY 60 TO GET HOURS            01120099
         DR    R14,R11          DIVIDE TO GET HOURS & MINUTES AS REM.   01130099
* REMAINDER NOW IN R14 WHICH IS MINUTES WHICH WE MUST SAVE              01140099
* QUOTIENT IN R15 - IE HOURS WHICH WE MUST SAVE                         01150099
         CVD   R14,TIMET           CONVERT TO PACKED DECIMAL            01160099
         OI    TIMET+7,X'0F'      FIX SIGN                              01170099
         UNPK  TIMEX+3(2),TIMET+6(2)  UNPACK MINUTES                    01180099
         MVI   TIMEX+2,C':'                                            01190099
         CVD   R15,TIMET           CONVERT TO PACKED DECIMAL            01200099
         OI    TIMET+7,X'0F'      FIX SIGN                              01210099
         UNPK  TIMEX+0(2),TIMET+6(2)  UNPACK HOURS                      01220099
         MVC   P42TME1(8),TIMEX                                         01230099
         MVC   P42TME(8),TIMEX                                          01240099
         MVC   P2ATME1(8),TIMEX                                         01250099
         MVC   P2ATME(8),TIMEX                                          01260099
         UNPK  P42DTE(7),SMF42DTE(4)                                    01270099
         OI    P42DTE+3,X'F0'                                           01280099
         CLC   P42DTE(2),=CL2'01'                                       01290099
         BNE   *+10                                                     01300099
         MVC   P42DTE(2),=C'20'                                         01310099
         MVC   P2ADTE(7),P42DTE                                         01320099
         MVC   P42DTE1,P42DTE                                           01330099
         MVC   P2ADTE1,P42DTE1                                          01340099
         LA    R4,SMF42END  -> PRODUCT SECTION                          01350099
         USING SMF42SJ,R4                                               01360099
JNA42PRP EQU   *             PREPARE TO CYCLE                          01370099
         L     R8,SMF42JN1  OFFSET                                      01380099
*                    TO SYSPLEX TOTALS LOCAL BM LRU STATS SUMM          01390099
*                    BELOW THE LINE                                     01400099
         LH    R10,SMF42JN3   NUMBER OF COPIES OF THE DSECT             01410099
         LH    R9,SMF42JN2    LENGTH OF ALL SMF420JA DSECTS             01420099
         LTR  R10,R10                                                   01430099
         BZ    JPA42PRP GO ON TO THE LOCAL ONES                         01440099
         XR    R14,R14        CLEAR HIGH ORDER                          01450099
         STH   R14,P42TJNAC   START THE COUNT AT 0                      01460099
         LR    R15,R9         LOAD LEGTH IN R9                          01470099
         DR    R14,R10         GET LENGTH OF SMF420JA DSECTS            01480099
         LR    R9,R15                                                   01490099
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                   01500099
*                           THE START OF THE RECORD                     01510099
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   01520099
```

Figure B-59 SMF42TJ assembler source (part 3 of 11)

```
JNA42TRP DS    OH             CYCLE ROUND RECORDS                        01530099
         LH    R14,P42TJNAC  GET CURRENT COUNT                          01540099
         LA    R14,1(R14)    ADD ONE                                    01550099
         STH   R14,P42TJNAC  PUT CURRENT COUNT                          01560099
         CVD   R14,DWORD                                                01570099
         OI    DWORD+7,X'0F'                                            01580099
         UNPK  P42TJNA#(7),DWORD+4(4)                                   01590099
         PUT   PRINTDCB,PJNA42L1                                        01600099
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                           01610099
         USING SMF420JA,R5                                              01620099
         BINDEC JNA                                                     01630099
         BINDEC JN7                                                     01640099
         BINDEC JNG                                                     01650099
         BINDEC JNH                                                     01660099
         BINDEC JNI                                                     01670099
         BINDEC JNJ                                                     01680099
         BINDEC JNK                                                     01690099
         BINDEC JNL                                                     01700099
         BINDEC JNM                                                     01710099
         BINDEC JNN                                                     01720099
         DROP  R5                                                       01730099
         PUT   PRINTDCB,PJNA42L2                                        01740099
         PUT   PRINTDCB,PJNA42L3                                        01750099
         PUT   PRINTDCB,PRINTBLK                                        01760099
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               01770099
*        LA    R8,0(R8,R9)  ADVANCE TO THE NEXT TRIPLET (IF ANY)       01780099
         BCT   R10,JNA42TRP                                             01790099
JPA42PRP EQU   *                                                        01800099
         L     R8,SMF42JN4  OFFSET                                      01810099
*                    TO SYSPLEX TOTALS LOCAL BM LRU STATS SUMM          01820099
*                    BELOW THE LINE                                     01830099
         LH    R10,SMF42JN6   NUMBER OF COPIES OF THE DSECT             01840099
         LH    R9,SMF42JN5    LENGTH OF BM SECTION                      01850099
         LTR  R10,R10                                                   01860099
         BZ    JNA2APRP                                                 01870099
         XR    R14,R14        CLEAR HIGH ORDER                          01880099
         STH   R14,P42TJPAC   START THE COUNT AT 0                      01890099
         LR    R15,R9         LOAD LEGTH IN R9                          01900099
         DR    R14,R10         GET LENGTH OF SMF420PA DSECTS            01910099
         LR    R9,R15                                                   01920099
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                   01930099
*                            THE START OF THE RECORD                    01940099
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   01950099
```

Figure B-60 SMF42TJ assembler source (part 4 of 11)

```
JPA42TRP DS    0H                                                       01960099
         LH    R14,P42TJPAC  GET CURRENT COUNT                          01970099
         LA    R14,1(R14)    ADD ONE                                    01980099
         STH   R14,P42TJPAC  PUT CURRENT COUNT                          01990099
         CVD   R14,DWORD                                                02000099
         OI    DWORD+7,X'0F'                                            02010099
         UNPK  P42TJPA#(7),DWORD+4(4)                                   02020099
         PUT   PRINTDCB,PJPA42L1                                        02030099
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                           02040099
         USING SMF42OPA,R5                                              02050099
         BINDEC JPA                                                     02060099
         MVC   P42JPB(8),SMF42JPB                                       02070099
         BINDEC JP6                                                     02080099
         BINDEC JPG                                                     02090099
         BINDEC JPH                                                     02100099
         BINDEC JPI                                                     02110099
         BINDEC JP2                                                     02120099
         BINDEC JPJ                                                     02130099
         BINDEC JPK                                                     02140099
         BINDEC JPL                                                     02150099
         BINDEC JPM                                                     02160099
         BINDEC JPN                                                     02170099
         DROP  R5                                                       02180099
         PUT   PRINTDCB,PJPA42L2                                        02190099
         PUT   PRINTDCB,PJPA42L3                                        02200099
         PUT   PRINTDCB,PRINTBLK                                        02210099
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS               02220099
*        LA    R8,0(R8,R9)  ADVANCE TO THE NEXT TRIPLET (IF ANY)        02230099
         BCT   R10,JPA42TRP                                             02240099
JNA2APRP EQU   *            PREPARE TO CYCLE                          02250099
         L     R8,SMF2AJN1  OFFSET                                      02260099
*                  TO SYSPLEX TOTALS LOCAL BM LRU STATS SUMM           02270099
*                  BELOW THE LINE                                      02280099
         LH    R10,SMF2AJN3   NUMBER OF COPIES OF THE DSECT             02290099
         LH    R9,SMF2AJN2    LENGTH OF ALL SMF2A0JA DSECTS             02300099
         LTR  R10,R10                                                   02310099
         BZ    JPA2APRP GO ON TO THE LOCAL ONES                         02320099
         XR    R14,R14        CLEAR HIGH ORDER                          02330099
         STH   R14,P2ATJNAC   START THE COUNT AT 0                      02340099
         LR    R15,R9         LOAD LEGTH IN R9                          02350099
         DR    R14,R10         GET LENGTH OF SMF2A0JA DSECTS            02360099
         LR    R9,R15                                                   02370099
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                   02380099
*                            THE START OF THE RECORD                   02390099
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                   02400099
```

Figure B-61 SMF42TJ assembler source (part 5 of 11)

```
JNA2ATRP DS    OH            CYCLE                                        02410099
         LH    R14,P2ATJNAC  GET CURRENT COUNT                            02420099
         LA    R14,1(R14)    ADD ONE                                      02430099
         STH   R14,P2ATJNAC  PUT CURRENT COUNT                            02440099
         CVD   R14,DWORD                                                  02450099
         OI    DWORD+7,X'0F'                                              02460099
         UNPK  P2ATJNA#(7),DWORD+4(4)                                     02470099
         PUT   PRINTDCB,PJNA2AL1                                          02480099
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                             02490099
         USING SMF2AOJA,R5                                                02500099
         BINDECA JNA                                                      02510099
         BINDECA JN7                                                      02520099
         BINDECA JNG                                                      02530099
         BINDECA JNH                                                      02540099
         BINDECA JNI                                                      02550099
         BINDECA JNJ                                                      02560099
         BINDECA JNK                                                      02570099
         BINDECA JNL                                                      02580099
         BINDECA JNM                                                      02590099
         BINDECA JNN                                                      02600099
         DROP  R5                                                         02610099
         PUT   PRINTDCB,PJNA2AL2                                          02620099
         PUT   PRINTDCB,PJNA2AL3                                          02630099
         PUT   PRINTDCB,PRINTBLK                                          02640099
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS                 02650099
*        LA    R8,0(R8,R9)  ADVANCE TO THE NEXT TRIPLET (IF ANY)          02660099
         BCT   R10,JNA2ATRP                                               02670099
JPA2APRP EQU   *             PREPARE TO CYCLE                           02680099
         L     R8,SMF2AJN4  OFFSET                                        02690099
*                    TO SYSPLEX TOTALS LOCAL BM LRU STATS SUMM           02700099
*                    BELOW THE LINE                                      02710099
         LH    R10,SMF2AJN6   NUMBER OF COPIES OF THE DSECT               02720099
         LH    R9,SMF2AJN5    LENGTH OF BM SECTION                        02730099
         LTR  R10,R10                                                     02740099
         BZ    READ                                                       02750099
         XR    R14,R14        CLEAR HIGH ORDER                            02760099
         STH   R14,P2ATJPAC   START THE COUNT AT 0                        02770099
         LR    R15,R9         LOAD LEGTH IN R9                            02780099
         DR    R14,R10         GET LENGTH OF SMF2AOPA DSECTS              02790099
         LR    R9,R15                                                     02800099
         LA    R5,0(R3,R8)   USE THE OFFSET TO LOCATE                     02810099
*                            THE START OF THE RECORD                     02820099
         SR    R5,R9   BACK UP BY LENGTH OF 1 SECTION                     02830099
```

Figure B-62 SMF42TJ assembler source (part 6 of 11)

```
JPA2ATRP DS    OH            CYCLE                                        02840099
         LH    R14,P2ATJPAC  GET CURRENT COUNT                            02850099
         LA    R14,1(R14)    ADD ONE                                      02860099
         STH   R14,P2ATJPAC  PUT CURRENT COUNT                            02870099
         CVD   R14,DWORD                                                  02880099
         OI    DWORD+7,X'0F'                                              02890099
         UNPK  P2ATJPA#(7),DWORD+4(4)                                     02900099
         PUT   PRINTDCB,PJPA2AL1                                          02910099
         LA    R5,0(R5,R9)   ADVANCE TO DSECT                             02920099
         USING SMF2AOPA,R5                                                02930099
         BINDECA JPA                                                      02940099
         MVC   P2AJPB(8),SMF2AJPB                                         02950099
         BINDECA JP6                                                      02960099
         BINDECA JPG                                                      02970099
         BINDECA JPH                                                      02980099
         BINDECA JPI                                                      02990099
         BINDECA JP2                                                      03000099
         BINDECA JPJ                                                      03010099
         BINDECA JPK                                                      03020099
         BINDECA JPL                                                      03030099
         BINDECA JPM                                                      03040099
         BINDECA JPN                                                      03050099
         DROP  R5                                                         03060099
         PUT   PRINTDCB,PJPA2AL2                                          03070099
         PUT   PRINTDCB,PJPA2AL3                                          03080099
         PUT   PRINTDCB,PRINTBLK                                          03090099
*  LOOP BACK AT THIS POINT IF THERE ARE ANY MORE TRIPLETS                 03100099
*        LA    R8,0(R8,R9)  ADVANCE TO THE NEXT TRIPLET (IF ANY)          03110099
         BCT   R10,JPA2ATRP                                               03120099
         B     READ                                                       03130099
IGNORE   DS    0H EXIT WITH OUT WRITING IF NOT THE RIGHT RECORDS          03140099
         B     READ                                                       03150099
FINI     DS    0H                                                         03160099
         SEGEND                                                           03170099
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=(GL),EROPT=SKP,EODAD=FINI      03180099
PRINTDCB DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM),LRECL=133                 03190099
DWORD    DS    D                                                          03200099
         ORG   DWORD                                                      03210099
         DC    C'12345678'                                                03220099
TIMET    DS    D                WORKAREA FOR TIME CONVERSION              03230099
TIMEX    DS    D                WORKAREA FOR TIME CONVERSION              03240099
TIMEF    DS    F                WORKAREA FOR TIME CONVERSION              03250099
PRINTBLK DC    CL133' '                                                   03260099
PRINTHDR DC    CL133'1SMF TYPE 42 S/TYPE 19 RECS. COLS USE SMF NAMES'     03270099
*                                                                        03280099
```

Figure B-63 SMF42TJ assembler source (part 7 of 11)

```
************* SMF42JNA ************                                  03290099
PJNA42L1 DC    CL133' SMF42JNA SYSPLEX SET #:'                       03300099
         ORG   PJNA42L1+26                                           03310099
P42TJNA# DC    CL8' '                                                03320099
         ORG                                                         03330099
P42TJNAC DC    H'0'  TO COUNT THE NUMBER OF SETS                     03340099
*                                                                    03350099
PJNA42L2 DC    CL133' '                                              03360099
         ORG   PJNA42L2+1                                            03370099
P42TMEH1 DC    CL9'HH:MM:SS '                                        03380099
P42DTEH1 DC    CL8'YYYYDDD '                                         03390099
P42JNAH  DC    CL9'SMF42JNA'                                         03400099
P42JN7H  DC    CL9'SMF42JN7'                                         03410099
P42JNGH  DC    CL9'SMF42JNG'                                         03420099
P42JNHH  DC    CL9'SMF42JNH'                                         03430099
P42JNIH  DC    CL9'SMF42JNI'                                         03440099
P42JNJH  DC    CL9'SMF42JNJ'                                         03450099
P42JNKH  DC    CL9'SMF42JNK'                                         03460099
P42JNLH  DC    CL9'SMF42JNL'                                         03470099
P42JNMH  DC    CL9'SMF42JNM'                                         03480099
P42JNNH  DC    CL9'SMF42JNN'                                         03490099
         ORG                                                         03500099
*                                                                    03510099
PJNA42L3 DC    CL133' '                                              03520099
         ORG   PJNA42L3+1                                            03530099
P42TME1  DC    CL8' '                                                03540099
         DC    CL1' '                                                03550099
P42DTE1  DC    CL8'YYDDDFIL'                                         03560099
P42JNA   DC    CL9' '                                                03570099
P42JN7   DC    CL9' '                                                03580099
P42JNG   DC    CL9' '                                                03590099
P42JNH   DC    CL9' '                                                03600099
P42JNI   DC    CL9' '                                                03610099
P42JNJ   DC    CL9' '                                                03620099
P42JNK   DC    CL9' '                                                03630099
P42JNL   DC    CL9' '                                                03640099
P42JNM   DC    CL9' '                                                03650099
P42JNN   DC    CL9' '                                                03660099
         ORG                                                         03670099
```

Figure B-64 SMF42TJ assembler source (part 8 of 11)

```
************* SMF42JPA ************                                  03680099
PJPA42L1 DC    CL133' SMF42JPA LPAR SET #:'                          03690099
         ORG   PJPA42L1+26                                           03700099
P42TJPA# DC    CL8' '                                                03710099
         ORG                                                         03720099
P42TJPAC DC    H'0'  TO COUNT THE NUMBER OF SETS                     03730099
*                                                                    03740099
PJPA42L2 DC    CL133' '                                              03750099
         ORG   PJPA42L2+1                                            03760099
P42TMEH  DC    CL8'HH:MM:SS '                                        03770099
         DC    CL1' '                                                03780099
P42DTEH  DC    CL8'YYYYDDD '                                         03790099
P42JPAH  DC    CL9'SMF42JPA'                                         03800099
P42JPBH  DC    CL9'SMF42JPB'                                         03810099
P42JP6H  DC    CL9'SMF42JP6'                                         03820099
P42JPGH  DC    CL9'SMF42JPG'                                         03830099
P42JPHH  DC    CL9'SMF42JPH'                                         03840099
P42JPIH  DC    CL9'SMF42JPI'                                         03850099
P42JP2H  DC    CL9'SMF42JP2'                                         03860099
P42JPJH  DC    CL9'SMF42JPJ'                                         03870099
P42JPJK  DC    CL9'SMF42JPK'                                         03880099
P42JPJL  DC    CL9'SMF42JPL'                                         03890099
P42JPJM  DC    CL9'SMF42JPM'                                         03900099
P42JPJN  DC    CL9'SMF42JPN'                                         03910099
         ORG                                                         03920099
*                                                                    03930099
PJPA42L3 DC    CL133' '                                              03940099
         ORG   PJPA42L3+1                                            03950099
P42TME   DC    CL8' '                                                03960099
         DC    CL1' '                                                03970099
P42DTE   DC    CL8' '                                                03980099
P42JPA   DC    CL9' '                                                03990099
P42JPB   DC    CL9' '                                                04000099
P42JP6   DC    CL9' '                                                04010099
P42JPG   DC    CL9' '                                                04020099
P42JPH   DC    CL9' '                                                04030099
P42JPI   DC    CL9' '                                                04040099
P42JP2   DC    CL9' '                                                04050099
P42JPJ   DC    CL9' '                                                04060099
P42JPK   DC    CL9' '                                                04070099
P42JPL   DC    CL9' '                                                04080099
P42JPM   DC    CL9' '                                                04090099
P42JPN   DC    CL9' '                                                04100099
         ORG                                                         04110099
```

Figure B-65 SMF42TJ assembler source (part 9 of 11)

```
************* SMF2AJNA ************                                  04120099
PJNA2AL1 DC    CL133' SMF2AJNA SYSPLEX SET #:'                       04130099
         ORG   PJNA2AL1+26                                           04140099
P2ATJNA# DC    CL8' '                                                04150099
         ORG                                                         04160099
P2ATJNAC DC    H'0'  TO COUNT THE NUMBER OF SETS                     04170099
*                                                                    04180099
PJNA2AL2 DC    CL133' '                                              04190099
         ORG   PJNA2AL2+1                                            04200099
P2ATMEH1 DC    CL9'HH:MM:SS '                                        04210099
P2ADTEH1 DC    CL8'YYYYDDD '                                         04220099
P2AJNAH  DC    CL9'SMF2AJNA'                                         04230099
P2AJN7H  DC    CL9'SMF2AJN7'                                         04240099
P2AJNGH  DC    CL9'SMF2AJNG'                                         04250099
P2AJNHH  DC    CL9'SMF2AJNH'                                         04260099
P2AJNIH  DC    CL9'SMF2AJNI'                                         04270099
P2AJNJH  DC    CL9'SMF2AJNJ'                                         04280099
P2AJNKH  DC    CL9'SMF2AJNK'                                         04290099
P2AJNLH  DC    CL9'SMF2AJNL'                                         04300099
P2AJNMH  DC    CL9'SMF2AJNM'                                         04310099
P2AJNNH  DC    CL9'SMF2AJNN'                                         04320099
         ORG                                                         04330099
*                                                                    04340099
PJNA2AL3 DC    CL133' '                                              04350099
         ORG   PJNA2AL3+1                                            04360099
P2ATME1  DC    CL8' '                                                04370099
         DC    CL1' '                                                04380099
P2ADTE1  DC    CL8'YYDDDFIL'                                         04390099
P2AJNA   DC    CL9' '                                                04400099
P2AJN7   DC    CL9' '                                                04410099
P2AJNG   DC    CL9' '                                                04420099
P2AJNH   DC    CL9' '                                                04430099
P2AJNI   DC    CL9' '                                                04440099
P2AJNJ   DC    CL9' '                                                04450099
P2AJNK   DC    CL9' '                                                04460099
P2AJNL   DC    CL9' '                                                04470099
P2AJNM   DC    CL9' '                                                04480099
P2AJNN   DC    CL9' '                                                04490099
         ORG                                                         04500099
```

Figure B-66 SMF42TJ assembler source (part 10 of 11)

```
************* SMF2AJPA ************                                    04510099
PJPA2AL1 DC    CL133' SMF2AJPA LPAR SET #:'                           04520099
         ORG   PJPA2AL1+26                                            04530099
P2ATJPA# DC    CL8' '                                                 04540099
         ORG                                                          04550099
P2ATJPAC DC    H'0'  TO COUNT THE NUMBER OF SETS                      04560099
*                                                                     04570099
PJPA2AL2 DC    CL133' '                                               04580099
         ORG   PJPA2AL2+1                                             04590099
P2ATMEH  DC    CL8'HH:MM:SS '                                         04600099
         DC    CL1' '                                                 04610099
P2ADTEH  DC    CL8'YYYYDDD '                                          04620099
P2AJPAH  DC    CL9'SMF2AJPA'                                          04630099
P2AJPBH  DC    CL9'SMF2AJPB'                                          04640099
P2AJP6H  DC    CL9'SMF2AJP6'                                          04650099
P2AJPGH  DC    CL9'SMF2AJPG'                                          04660099
P2AJPHH  DC    CL9'SMF2AJPH'                                          04670099
P2AJPIH  DC    CL9'SMF2AJPI'                                          04680099
P2AJP2H  DC    CL9'SMF2AJP2'                                          04690099
P2AJPJH  DC    CL9'SMF2AJPJ'                                          04700099
P2AJPJK  DC    CL9'SMF2AJPK'                                          04710099
P2AJPJL  DC    CL9'SMF2AJPL'                                          04720099
P2AJPJM  DC    CL9'SMF2AJPM'                                          04730099
P2AJPJN  DC    CL9'SMF2AJPN'                                          04740099
         ORG                                                          04750099
*                                                                     04760099
PJPA2AL3 DC    CL133' '                                               04770099
         ORG   PJPA2AL3+1                                             04780099
P2ATME   DC    CL8' '                                                 04790099
         DC    CL1' '                                                 04800099
P2ADTE   DC    CL8' '                                                 04810099
P2AJPA   DC    CL9' '                                                 04820099
P2AJPB   DC    CL9' '                                                 04830099
P2AJP6   DC    CL9' '                                                 04840099
P2AJPG   DC    CL9' '                                                 04850099
P2AJPH   DC    CL9' '                                                 04860099
P2AJPI   DC    CL9' '                                                 04870099
P2AJP2   DC    CL9' '                                                 04880099
P2AJPJ   DC    CL9' '                                                 04890099
P2AJPK   DC    CL9' '                                                 04900099
P2AJPL   DC    CL9' '                                                 04910099
P2AJPM   DC    CL9' '                                                 04920099
P2AJPN   DC    CL9' '                                                 04930099
         ORG                                                          04940099
SMFDSECT DSECT                                                        04950099
*        IFASMFR  (42) THIS DOES NOT EXPAND THE SUBTYPES AS IT SHOULD  04960099
         IGWSMF  SMF42_OJ=YES                                         04970099
         END                                                          04980099
```

Figure B-67 SMF42TJ assembler source (part 11 of 11)

Step 3: store the JCL to assemble and link the source in the PDS

Cut and paste the contents of Figure B-68 into your PDS MHLRES1.SMF42TJ.SOURCE as member SMF42TlJ. The result should contain 30 lines.

Run the job when the member has been created.

```
//MHLRES1L JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
//ASMHCL PROC
//ASM      EXEC PGM=ASMA90,REGION=0M,
//             PARM='OBJECT,NODECK'
//SYSIN    DD  DSN=SYS1.SAMPLIB(IEFESO),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(10,2)),DCB=BLKSIZE=3120
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
//*
//LKED     EXEC PGM=HEWL,REGION=2048K,COND=(8,LE,ASM),
//             PARM='XREF,LIST,LET'
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=&&LOADMOD(IEFESO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,20,1))
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,5))
// PEND
//  EXEC ASMHCL
//ASM.SYSIN DD DISP=SHR,DSN=MHLRES1.SMF42TJ.SOURCE(SMF42TJA)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TJJ.PDS,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(1,1,1))
//LKED.SYSIN   DD *
 SETSSI 00001800
 NAME SMF42TJ(R)
```

Figure B-68 SMF42TJJ JCL to assemble and link the program SMF42TJ

This creates data set MHLRES1.SMF42TJ.LOAD.

Note: If this JCL needs to be rerun, change the lines:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TJ.LOAD,DISP=(,CATLG,DELETE),

//            UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

to read:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF42TJ.LOAD,DISP=SHR (,CATLG,DELETE),

//* UNIT=SYSDA,SPACE=(CYL,(1,1,1))
```

This is to stop it from trying to make the data set again.

Program SMF42TJ alteration

The SMF records are generally TEXT or binary values, and when binary values, are usually fixed bin 32.

The program refers to each field four times:

- The SMF source field as mapped by the IGWSMF macro
- The location in the code where the field is copied or converted
- The storage location where the output line header is generated
- The storage location where the copied or converted data is stored

If text fields are changed, you need to take account of the length of the field. If the replacement is shorter than the field it replaces, the length does not have to be adjusted, but the output is longer than it need be. If the replacement field is longer than the field it replaces, the lengths must be adjusted to avoid overlay or truncation of storage. For example, the following simple changes can be made:

- Character changes

  If field SMF42JPC (which is currently reserved) was changed so that it contained useful information, and field SMF42JPB was no longer needed, all references to JPB could be changed to JPC and the program would change the headings and the source of the data and display the new information. Note that this would change both the below and above the 2 GB bar fields.

- Binary changes

  If field SMF42JP6 was no longer needed, but SMF42JPO was required, if all references to JP6 were changed to JPO, the program would change the headings and the source of the data and display the new information. Note that this would change both the below and above the 2 GB bar fields.

  There are other fields that do not fit into the Fixed Bin 32 category, for example, Fixed bin(16) or double words, which would require more significant program changes.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 562. Note that some of the documents referenced here may be available in softcopy only.

- *z/OS Version 1 Release 7 Implementation*, SG24-6755
- *Partitioned Data Set Extended Usage Guide*, SG24-6106
- *VSAM Demystified*, SG24-6105
- *LOBs with DB2 for z/OS: Stronger and Faster*, SG24-7270

Other publications

These publications are also relevant as further information sources:

- *MVS Initialization and Tuning Guide,* SA22-7591
- *MVS Initialization and Tuning Reference,* SA22-7592
- *MVS System Commands,* SA22-7627
- *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426
- *Network File System Guide and Reference*, SG26-7417
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405
- *DFSMSrmm Application Programming Interface,* SC26-7403
- *DFSMSrmm Guide and Reference,* SC26-7404
- *DFSMS Managing Catalogs,* SC26-7409
- *z/OS Support for Unicode: Using Conversion Services*, SC33-7050
- *z/OS UNIX System Services Planning,* SC28-1890
- *z/OS IBM CS V2R10.0: IP Configuration Guide,* SC31-8725
- *z/OS Integrated Security Services Network Authentication Service Administration,* SC24-5926
- *z/OS Integrated Security Services Network Authentication Service Programming,* SC24-5927
- *z/OS CS, IPv6 Network and Application Design Guide*, SC31-8885
- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926
- *DFSMS Using the New Functions,* SC26-7473
- *DFSMSdfp Diagnosis*, GY27-7618

Online resources

These Web sites and URLs are also relevant as further information sources:

- SNIA Web site

  http://www.opengroup.org/snia-cimom/

- z/OS NFS Web site

  http://www-1.ibm.com/servers/eserver/zseries/zos/nfs/

- DHCP Web site

  http://www.faqs.org/rfcs/rfc2131.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

H

I

L

M

N

O

P

R

S

T

V

X

z/OS V1R8 DFSMS Technical Update

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

z/OS V1R8 DFSMS Technical Update

Understand the features and functions in z/OS DFSMS V1.8

Read implementation hints and tips

Learn DFSMS trends and directions

Each release of DFSMS builds upon the previous version to provide enhanced storage management, data access, device support, program management, and distributed data access for the z/OS platform in a system-managed storage environment.

This IBM Redbooks publication provides a technical overview of the functions and enhancements in z/OS V1R8 DFSMS. It provides you with the information that you need to understand and evaluate the content of this DFSMS release, along with practical implementation hints and tips. Also included are enhancements that were made available through an enabling PTF that has been integrated into z/OS DFSMS V1R8.

This book was written for storage professionals and system programmers who have experience with the components of DFSMS. It provides sufficient information so that you can start prioritizing the implementation of new functions and evaluating their applicability in your DFSMS environment.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7435-00 ISBN 073848573X