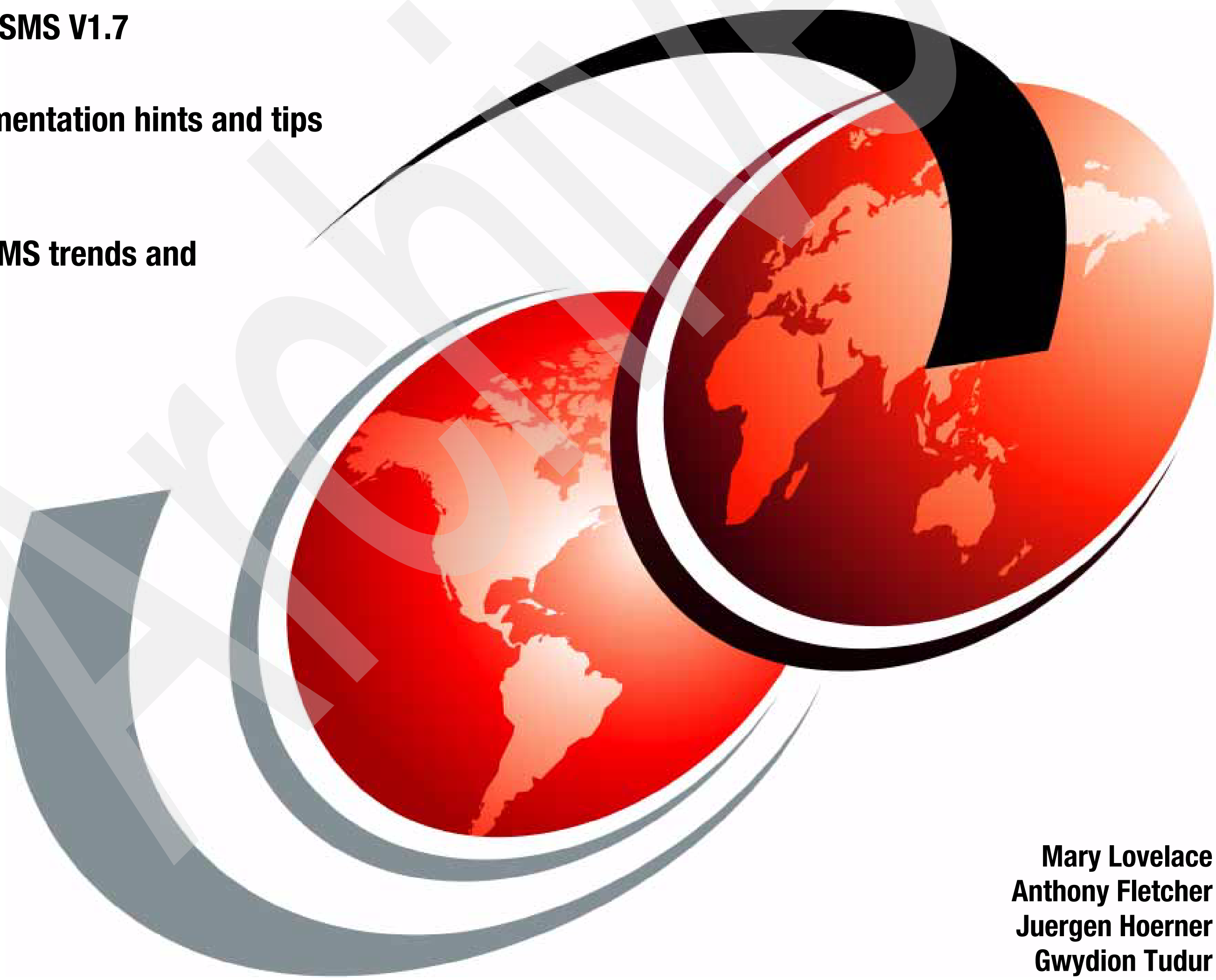
IBM

z/OS V1R7 DFSMS Technical Update

Understand the features and functions in z/OS DFSMS V1.7

Get implementation hints and tips

Learn DFSMS trends and directions



Mary Lovelace
Anthony Fletcher
Juergen Hoerner
Gwydion Tudur

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS V1R7 DFSMS Technical Update

December 2006

SG24-7225-00

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

First Edition (December 2006)

This edition applies to Version 1, Release 7, of IBM z/OS V1.7 DFSMS (product number 5655-G52).

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

Redbooks (logo) ™
ibm.com®
z/Architecture®
z/OS®
z/VM®
AIX®
CICS®
DB2®
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSMSrmm™
DFSORT™
FlashCopy®
Hiperspace™
IBM®
Language Environment®
Maestro™
MVS™
MVS/DFP™
MVS/ESA™
OS/390®
Redbooks™
RACF®
RMF™
RS/6000®
TotalStorage®
WebSphere®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

NOW, and the Network Appliance logo are trademarks or registered trademarks of Network Appliance, Inc. in the U.S. and other countries.

Java, JavaBeans, JavaMail, JDK, JVM, Portmapper, Solaris, Sun, WebNFS, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds 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® Redbook provides a technical overview of the functions and enhancements in z/OS V1R7 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 V1R5 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 redbook

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

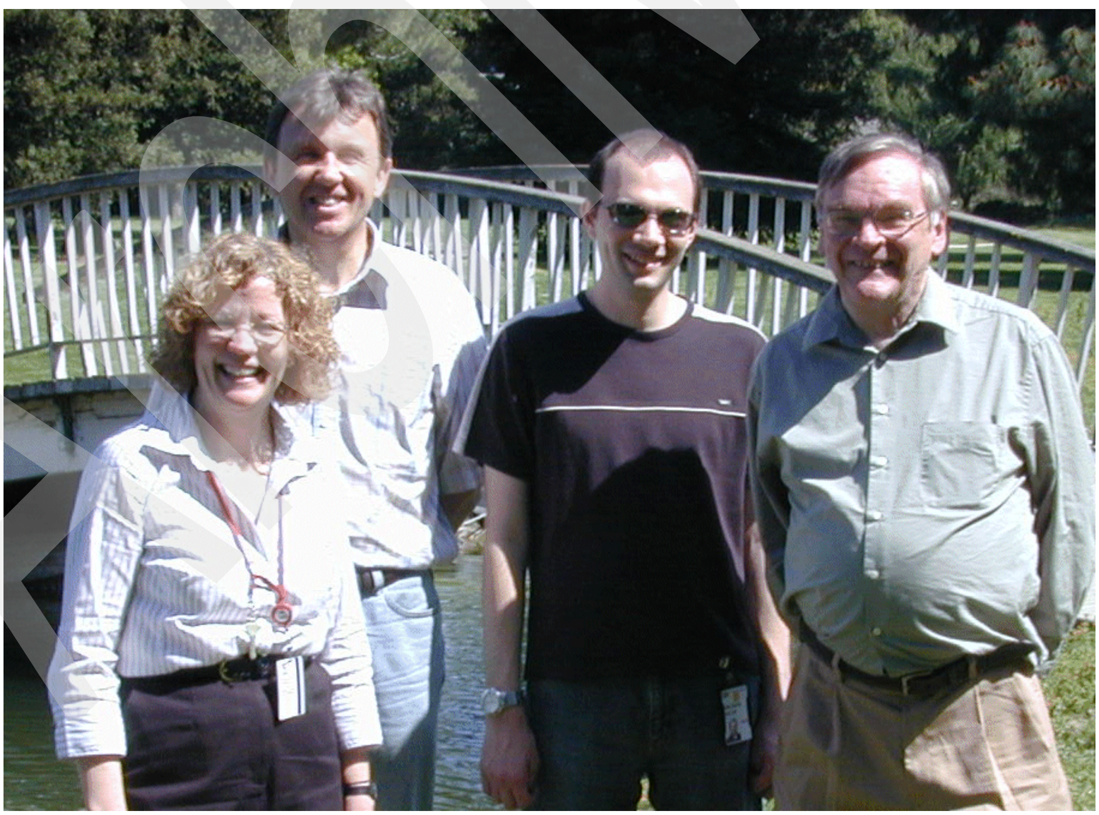


Figure 1 The team (from left to right): Mary, Juergen, Gwydion, Anthony

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 Ondemand Infrastructure Services, based in New Zealand but working for New Zealand and Australia. He has 35 years of experience in z/OS, OS/390®, and their predecessors and related components, both as a client of IBM and with IBM Global Services. He is a team leader for the mainframe operations of three diverse clients in the banking, airline, 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 Anthony has a working knowledge of RACF®. He also has experience in installing non-IBM products for the GSD platform. Anthony can be contacted at fletcha@nz1.ibm.com.

Juergen Hoerner is an IT Specialist working for IBM TSS Mainz, Germany. He has 13 years of experience in supporting IBM OS/390 Software. His areas of expertise include z/OS DFP, DFSMS, NFS, UNIX® System Services, and DFS/SMB. In this book he wrote the chapter on Network File Service.

Gwydion Tudur is an IT Specialist working in the z/OS data management team at IBM Hursley software development laboratory in Hampshire, UK. He has three years of experience working with DFSMS, and also develops applications on the LAMP (Linux®, Apache, MySQL, Perl) open source Web platform. He holds a degree in Computing from Imperial College, London, UK. His main areas of expertise include DFSMS, DFSMShsm, and DFSMSdss™.

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

Bob Haimowitz
Rich Conway
International Technical Support Organization

Stephan Branch
Michael Bull
Mike Madrid
Terry Menendez
Mike Woods
IBM San Jose

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'll team with IBM technical professionals, Business Partners and/or clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll 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 redbook form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@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

What is new in DFSMS V1R7

This chapter provides:

- A summary of what was new in z/OS V1R6 DFSMS
- An overview of what is new in z/OS V1R7 DFSMS

Details of the new functions and enhancements in DFSMS V1R7 are provided in later chapters.

1.1 What was new in DFSMS V1R6

This section provides an overview of the enhancements introduced in DFSMS V1R6. The enhancements are:

- z/OS V1R6 SMS enhancements
  - SMS PAV parallel access volume
  - Multiple level security (MLS) labels in ACS automatic class selection routines
- IBM TotalStorage Enterprise Tape System 3592 enhancements
  - Support for Model J new media types
  - MEDIA6 - 300-GB capacity WORM (Write Once - Read Many)
  - MEDIA7 - 60-GB capacity Read/Write
  - Media 8 - 60-GB capacity WORM
- PDSE enhancements
  - PDSE restartable address space
- Catalog enhancements
  - MODIFY CATALOG,TAKEDUMP
  - MODIFY CATALOG,RESTART
- DFSMShsm enhancements
  - Secondary Space Management multiple tasks
- DFSMSdss enhancements
  - COPY and RESTORE commands REPLACEUNCONDITIONAL
  - COPY and DEFRAG commands FCTOPPRCPRIMARY
  - SNAPX option cross memory extension
- DFSMSrmm enhancements
  - Application programming interface enhancements
  - RMMplex
  - Client/server support
  - ISPF Dialog enhancements
- Advanced copy services enhancements
  - New support for Open System (fixed block) volumes in FlashCopy® and PPRC relationships.
  - New support for Inband FlashCopy requests to be issued remotely through an existing PPRC connection.
  - New Incremental FlashCopy option copies only data that has changed since the last FlashCopy request.
  - New FlashCopy mode, NOCOPY2COPY.
  - New FlashCopy to PPRC Primary support allows the target of a FlashCopy request to be designated as a primary PPRC device.
- OAM enhancements
  - Support for 3592 Model J drives

Note: There were no enhancements to DFSMSdfp™ in z/OS V1R6.

For a full list of all user interface changes in z/OS V1R6, see *z/OS V1R6.0 Summary of Message and Interface Changes,* SA22-7505. For details of enhancements in z/OS V1R6 refer to *z/OS V1R6 DFSMS Technical Guide*, SG24-6651.

1.1.1 DFSMSdfp

This section lists the user interface changes in the DFSMSdfp component of DFSMS V1R6.

DCOLLECT command

Changed output: displays the new Storage Class value for parallel access volume capability.

VARY SMS,PDSE1,ACTIVATE

New command in DFSMS V1.6: brings up the restartable SMSPDSE1 address space if previously closed down.

VARY SMS,PDSE1,ANALYSIS

New parameter PDSE1: carries out analysis of the SMSPDSE1 address space. This is in addition to the original command to carry out analysis of the SMSPDSE address space.

VARY SMS,PDSE1,FREELATCH

New parameter PDSE1: releases any latch that is reported by the analysis command. This is in addition to the original command to free a latch on the SMSPDSE address space.

VARY SMS,PDSE1,MONITOR

New parameter PDSE1: modifies processing of the monitor for SMSPDSE1. This is in addition to the original command to vary processing for SMSPDSE.

VARY SMS,PDSE1,RESTART

New command: shuts down and restarts the restartable SMSPDSE1 address space.

ISMF panel ACS Test Case Define/Alter page 1

New field: SECLABEL can be specified.

ISMF panel Storage Class Define/Alter

New field: parallel access volume capability value set.

ISMF panel Storage Display

New field: parallel access volume capability output values.

ISMF panel Storage Class List

New field: parallel access volume capability output values.

ISMF panel Storage Class List Print

New field: parallel access volume column 20.

ISMF panel Storage Class List Sort

New field: parallel access volume column 20.

ISMF panel Storage Class List View

New field: parallel access volume column 20.

Naviquest panel ACBDFLG1

Changed: can specify SECLABEL.

Naviquest panel ACBJBAS1

Changed: can specify PAVCAP.

SMS ACS read-only variables

New: contains the default seclabel value from the RACF profile for the user or the data set, or is null if the SECABEL class is not active.

SMS constructs

New field: parallel access volume capability.

1.1.2 DFSMSdss

This section lists the user interface changes in the DFSMSdss component of DFSMS V1.6.

SNAPX PARM

New: SNAPX can be used when invoking the DFSMSdss Cross Memory Application Interface (API) using the system macros LINK, ATTACH, and CALL.

Old: The SNAPX parameter was only allowed on JCL invocations of the DFSMSdss Cross Memory API (this use is still allowed).

COPY DATASET REPLACEUnconditional keyword

New: the REPLACEUnconditional keyword, for use with the RENAMEUnconditional keyword.

ISMF panel Copy Command Entry

New field: Replace Unconditional.

ISMF panel Entry (copy line operator for volumes)

New field: Replace Unconditional.

ISMF panel Data Set Restore Entry

New field: Replace Unconditional.

ISMF panel Restore Command Entry

New field: Replace Unconditional.

ISMF panel Restore Entry (logical restore line operator)

New field: Replace Unconditional.

1.1.3 DFSMShsm

This section lists the user interface changes in the DFSMShsm component of DFSMS V1.6.

SETSYS command Secondary Space Management

New: The MAXSSMTASKS parameter has been added.

1.1.4 DFSMSrmm

This section lists the user interface changes in the DFSMSrmm component of DFSMS V1.6.

ADDVOLUME command for support of the IBM 3592 WORM tapes

Changed: MEDIATYPE updated to include MEDIA5/ETC, MEDIA6/EWTC, MEDIA7/EETC, and MEDIA8/EEWTC.

CHANGEVOLUME command for support of the IBM 3592 WORM tapes

Changed: MEDIATYPE updated to include MEDIA5/ETC, MEDIA6/EWTC, MEDIA7/EETC, and MEDIA8/EEWTC.

CHANGEVOLUME command for support of client server

Added: CONFIRMRELEASE(SCRATCH).

LISTCONTROL

Output changed: client/server related fields.

SEARCHVOLUME command for support of the IBM 3592 WORM tapes

Changed: MEDIATYPE updated to include MEDIA5/ETC, MEDIA6/EWTC, MEDIA7/EETC, and MEDIA8/EEWTC.

ISPF panels to support client/server

New and additional panels.

ISPF panels to improve ISPF usability

New and additional panels.

EXITS Client/server support

Changed: EDGUX200 volume release processing.

REXX variables support for user customization

Additions and changes: several variables available on several subcommands.

UTILITIES client/server support

Changes: utilities EDGBKUP, EDGHSKP, and EDGUTIL.

1.2 Summary of z/OS V1R6 DFSMS parameter changes

This section lists the DFSMS V1.6 parameter changes in support of the DFSMS V1.6 functions.

1.2.1 IGDSMSxx in support of PDSE

The following changes were required to implement PDSE restartable address space.

SMS parameter changes to options

The following changes to SMS options were introduced to support the PDSE restartable address space SMSPDSE1. The global address SMSPDSE is always required, but

SMSPDSE1 is optional. In order to allow for different values for SMSPDSE and SMSPDSE1, some parameters have been given synonyms. The function, in respect to SMSPDSE, remains the same whether the original or the synonym is used. BMFTIME has the synonym PDSE_BMFTIME.

Note that, contrary to the published documentation, the SMS option SMF_TIME(YES) does not override values set with PDSE_BMFTIME:

- HSP_SIZE has the synonym PDSE_HSP_SIZE.
- LRUCYCLES has the synonym PDSE_LRUCYCLES.
- LRUTIME has the synonym PDSE_LRUTIME.

SMS parameter additional options

The following additions to SMS options were introduced to enable and define the PDSE restartable address space SMSPDSE1:

- PDSE_RESTARTABLE_AS(YES)
- PDSE1_BMFTIME

Note that, contrary to the published documentation, the SMS option SMF_TIME(YES) does not override values set with PDSE-BMFTIME:

- PDSE1_HSP_SIZE
- PDSE1_LRUCYCLES
- PDSE1_LRUTIME
- PDSE1_MONITOR

1.2.2 DEVSUPxx PARMLIB member options

System-wide DEVSUPxx PARMLIB member options are included to control and manage media type selection for:

- Stand-alone environment:
  - ENFORCE_DC_MEDIA=ALLMEDIATY ensures that data class media policies are enforced for all stand-alone, non-specific mounts.
  - ENFORCE_DC_MEDIA=MEDIA5PLUS ensures that data class media policies are enforced for all stand-alone, non-specific mounts for any of the four 3592 tape cartridge media types.

> **Note:** The ENFORCE_DC_MEDIA option can be used in conjunction with the tape management systems pooling support to ensure that the correct media type is mounted. Before setting this option, determine how data class is used in the stand-alone environment to ensure that a data class with the correct media type specification is selected and that any tape management system pool assigned to the allocation request matches the media type specified in the assigned data class.

- Manual Tape Library (MTL) non-specific volume requests:
  - MTL_NO_DC_WORM_OK indicates that the installation is managing WORM media allocations through the tape management system pooling support, instead of data class. This option results in OPEN accepting MTL WORM scratch mounts when no media type is identified via data class.

- Automatic Tape Library (ATL) partitioning:
  - MEDIA5=*xxxx*
  - MEDIA6=*xxxx*
  - MEDIA7=*xxxx*
  - MEDIA8=*xxxx*

  Where *xxxx* is a four-digit hexadecimal category code.

1.3 Overview of what is new in z/OS V1R7 DFSMS

This section summarizes the enhancements made in various components in DFSMS V1.7.

1.3.1 DFSMSdfp

DFSMSdfp V1R7 has been enhanced 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

For details refer to Chapter 2, “DFSMSdfp V1R7 enhancements” on page 11.

1.3.2 DFSMSdss

DFSMSdss has been enhanced to support the new large format data sets that are introduced in 2.1, “Large format data sets” on page 12.

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 listed 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.

For details refer to Chapter 5, “DFSMSdss enhancements” on page 127.

1.3.3 DFSMShsm

DFSMShsm V1R7 has been enhanced 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 wildcards with HMIGRATE
- Saving LRECL of migrated data sets in the MCD
- New recycle processing option for connected sets
- AUDIT MEDIACONTROLS resume
- Dump encryption support

For details refer to Chapter 4, “DFSMShsm enhancements” on page 109.

1.3.4 DFSMSrmm

DFSMSrmm V1R7 provides 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

For details refer to Chapter 6, “DFSMSrmm V1R7 enhancements” on page 137.

1.3.5 OAM

DFSMSdfp V1R7 provides the following enhancements to OAM:

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

For details refer to Chapter 3, “OAM enhancements” on page 95.

1.4 NFS

NFS V1R7 provides the following enhancements to NFS:

- Miscellaneous small enhancements
- Serviceability improvements
- IPV6 support
- DHCP (Dynamic IP) support for clients
- NFS server locking interoperability
- NLM integration
- LFS support for NFS V4
- Security overview
- Reduced message traffic - compound RPC
- NFS Server V4 protocol

For details refer to Chapter 7, “NFS enhancements” on page 201.

2

DFSMSdfp V1R7 enhancements

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

- Large format data sets
- Device support address space (DEVMAN)
- DFSMS subchannel set support
- REPRO MERGECAT FROMKEY/TOKEY enhancement
- Catalog enhancements
  - VVDS implicit size specification and verification
  - 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
  - SMS ACS message processing enhancements
- Extended Remote Copy (XRC) Plus
- DEVSERV QLIB command
- PDSE enhancements (These were introduced with V1R6, but have 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
  - VSAM attributes removal reminder: Imbed and Replicate

2.1 Large format data sets

Large format data sets are sequential and can have greater than 65,535 tracks per volume. Before z/OS V1R7, most sequential data sets were limited to 65,535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. QSAM, BSAM, and EXCP access methods all support large format data sets, with some limitations for EXCP programs, and for BSAM programs that use the NOTE and POINT macros.

Large format data sets reduce the need to use multiple volumes for single data sets, especially very large ones like spool data sets, dumps, logs, and traces. Large format data sets can be either cataloged or uncataloged, SMS-managed or not. Unlike extended-format data sets, which also support greater than 65,535 tracks per volume, large format data sets are compatible with EXCP and do not need to be SMS-managed.

Large format data sets are not required to have greater than 65,535 tracks at initial allocation, but once allocated in large format they can expand to greater than 65,535 tracks.

2.1.1 Planning to use large format data sets

Before implementing large format data sets you must consider the following:

- The supported environment and how it has to be set up - see “Supported environments” on page 12.
- Compatibility with lower-level systems of new operands in the IGDSMSxx member of SYS1.PARMLIB - see “Large format data set coexistence with lower level systems” on page 22.
- Maintenance required on lower-level systems - see “Maintenance support in lower-level systems for large format data sets” on page 23.

2.1.2 Supported environments

Use of the full capability of large format data sets requires the use of z/OS V1R7. Limited use in prior level systems, in certain cases, is possible provided that the data set is not over 65,535 tracks on any volume that a data set is allocate on.

For use with z/OS V1R7, the BLOCKTOKENSIZE option has been added to SMS to allow the user to update applications to check for large format data set compatibility at data set OPEN time.

The BLOCKTOKENSIZE parameter of the IGDSMSxx member of SYS1.PARMLIB is provided to allow an installation to control use of large format data sets by applications.

> **Note:** Detailed information about validating or converting application programs for use with large format data sets can be found in the section headed “Modifying applications to use large format data sets” in Chapter 1 of the V1R7 version of *DFSMS Using the New Functions,* SC26-7473.

z/OS V1R7 supports large format data sets in many areas including:

- Standalone dump.
- IPCS.
- JES2 and JES3 spools.
- ISPF.
- TSO/E ALLOCATE command.
- SMS, DFSORT™, IDCAMS REPRO, IEBGENER, IEBDG, and IEBCOPY all support them and are unaffected by the BLOCKTOKENSIZE option in SYS1.PARMLIB.
- DFSMShsm supports them for user data sets and for the journal.
- DFSMSrmm supports them for user data sets and for temporary data sets for inventory management.
- If BLOCKTOKENSIZE(NOREQUIRE) is in effect, then generally programs that use QSAM or use BSAM without NOTE or POINT can be expected to work. Whether they work depends on factors such as whether they use the DS1LSTAR field, the DEBMTRK field, or call a track conversion routine.

Note: Detailed information about validating or converting application programs for use with large format data sets can be found in Chapter 1 of the V1R7 version of *DFSMS Using the New Functions,* SC26-7473.

z/OS V1R7 does not support large format data sets in the following areas:

- Virtual I/O.
- System dump.
- Password data sets.
- The BDAM access method.
- The TSO COPY command.
- LISTDSI and the equivalent REXX function may display the wrong value.
- The high-level languages do not support them when the new BLOCKTOKENSIZE(REQUIRE) option in the IGDSMSxx member of PARMLIB, which is the default, is in effect.

2.1.3 Setting up support for large format data sets

For complete information about initializing SMS, you can refer to "Initializing SMS through the IGDSMSxx member" in the V1R7 editions of the following manuals for detailed information:

- *DFSMSdfp Storage Administration Reference,* SC26-7402
- *MVS Initialization and Tuning Reference,* SA22-7592

The change in z/OS V1R7 specific to large format data sets is the addition of the BLOCKTOKENSIZE parameter, which has two versions:

- BLOCKTOKENSIZE(REQUIRE) specifies that BLOCKTOKENSIZE=LARGE must be specified on the DCBE macro. With the other application changes required to support large format data sets, this signifies that changes have been made to support large format data sets. Applications do not need to be modified, even if BLOCKTOKENSIZE(REQUIRE) is specified in IGDSMSxx if data sets meet the following conditions:

 – Data set has no more than 65,535 tracks on each volume
 – Is open for input using EXCP, BSAM, or QSAM or for update using BSAM or QSAM

- BLOCKTOKENSIZE(NOREQUIRE) specifies that applications may use large format data sets without having to make application changes as long as they meet the following conditions:
  – The access method is QSAM or it is BSAM without the NOTE or POINT macros.
  – The access method is BSAM with the NOTE or POINT macros (MACRF=xP is coded) and the data set has no more than 65,535 tracks on the volume and the OPEN option is INPUT or UPDAT.

Note: The BLOCKTOKENSIZE(REQUIRE) parameter is the default in the IGDSMSxx member of PARMLIB. Experience shows with many system options that are designed for compatibility, most installations eventually override the default. It is expected that it will not be long before most installations will want BLOCKTOKENSIZE(NOREQUIRE) to be in effect.

For compatibility in member IGDSMSxx across V1R7 and a down-level system, omitting BLOCKTOKENSIZE and therefore taking the default of REQUIRE is appropriate.

However, if you anticipate wanting to retain the REQUIRE option across any future change by IBM, you should code BLOCKTOKENSIZE(REQUIRE) once you have completed conversion to V1R7.

2.1.4 BLOCKTOKENSIZE specification and verification

In this section we show how to specify the BLOCKTOKENSIZE parameter and how to verify its setting.

Specification of BLOCKTOKENSIZE

In Figure 2-1 we show how to specify the BLOCKTOKENSIZE parameter of the IGDSMSxx member of PARMLIB. This is an example definition intended only to show the use of the BLOCKTOKENSIZE variable in context, and there is no significance to the other variables shown.

```
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)
    PDSE1_HSP_SIZE(256)
    PDSE1_BMFTIME(3600)
    TRACE(ON)
    SIZE(128K)
    TYPE(ALL)
    JOBNAME(*)
    ASID(*)
    SELECT(ALL)
    OAMPROC(OAM)
    OAMTASK(OAM)
    CICSVR_INIT(YES)
    CICSVR_DSNAME_PREFIX(DWWUSER.V3R1M0)
    LOG_OF_LOGS(IGWTVS.LOG.OF.LOGS)
    SYSNAME(SC63,SC64,SC65)
    TVSNAME(  63,  64,  65)
    TV_START_TYPE(WARM,WARM,WARM)
    BLOCKTOKENSIZE(REQUIRE)
    RLSFIXEDPOOLSIZE(SC70,0)
```

Figure 2-1 BLOCKTOKENSIZE specification

Verification of BLOCKTOKENSIZE value

You can obtain the system setting for BLOCKTOKENSIZE with the DISPLAY SMS,OPTIONS command.

Figure 2-2 shows part of an example of the SMS options display. The BLOCKTOKENSIZE value, in this case showing REQUIRE, is towards the end of the list.

```
000290  D SMS,OPTIONS
000090  IGD002I 14:18:43 DISPLAY SMS 197
000090  ACDS     = SYS1.SMS.ACDS
000090  COMMDS   = SYS1.SMS.COMMDS
000090  INTERVAL = 15          DINTERVAL = 150
000090  SMF_TIME = YES         CACHETIME = 3600
000090  CF_TIME = 1800         PDSE_RESTARTABLE_AS = NO
000090  PDSE_BMFTIME = 3600    PDSE1_BMFTIME = 3600
000090  PDSE_LRUTIME = 60      PDSE1_LRUTIME = 60
000090  PDSE_LRUCYCLES = 15   PDSE1_LRUCYCLES = 15
000090  LOCAL_DEADLOCK = 15    GLOBAL_DEADLOCK = 4
000090  REVERIFY = NO          DSNTYPE = PDS
000090  ACSDEFAULTS = NO       PDSESHARING = EXTENDED
000090  OVRD_EXPDT = NO        SYSTEMS = 8
000090  PDSE_HSP_SIZE = 0MB   PDSE1_HSP_SIZE = 0MB
000090  USE_RESOWNER = YES     RLS_MAX_POOL_SIZE = 100MB
000090  RLSINIT = YES          RLSTMOUT = 0
000090  COMPRESS = GENERIC     LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
000090  QTIMEOUT = 300         TVSNAME = 063
000090  AKP = 1000             TV_START_TYPE = WARM
000090  MAXLOCKS = (0,0)
000090  CICSVR_INIT = YES      CICSVR_DSNAME_PREFIX = DWWUSER.V3R1M0
000090  CICSVR_RCDS_PREFIX = DWW
000090  CICSVR_GRPNAME_SUFFIX = PROD
000090  CICSVR_ZZVALUE_PARM =
000090  CICSVR_UNDOLOG_CONTROL =
000090  CICSVR_UNDOLOG_PREFIX = DWW
000090  CICSVR_BACKOUT_CONTROL =
000090  CICSVR_GENERAL_CONTROL =
000090  Rls_MaxCfFeatureLevel = Z
000090  RlsAboveThebarMaxPoolSize = 0
000090  RlsFixedPoolSize = 0
000090  PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
000090  GDS_RECLAIM = YES      DSSTIMEOUT = 0
000090  BLOCKTOKENSIZE = REQUIRE
000090  IGD002I 14:18:43 DISPLAY SMS
000090  TRACE    = ON     SIZE = 128K     TYPE = ALL
000090    JOBNAME = *    ASID = *
```

Figure 2-2 Output from D SMS,OPTIONS

2.1.5 Data set allocation

In addition to the setting for DSNTYPE to allocate a large format data set, in z/OS V1R7 there are additional new values that can be specified with DSNTYPE on the JCL DD statement and through the dynamic allocation equivalent. These additional values do not designate new types of data sets. They are provided to give additional control over data set type specification, or in the case of BASIC to provide a way of specifying that a data set must not be a large format version.

DSNTYPE keyword options, ACS data class attribute, and read-only variable &DSNTYPE presented in ACS routines:

- LARGE - Refer to "Creating large format data sets" on page 17.
- BASIC - Refer to "Creating basic format data sets" on page 23.
- EXTREQ - Refer to "Specifying a DSNTYPE of EXTREQ or EXTPREF" on page 24.
- EXTPREF - Refer to "Specifying a DSNTYPE of EXTREQ or EXTPREF" on page 24.

Creating large format data sets

To create a large format data set you can use the explicit or implicit method:

- Explicit: Specify a DSNTYPE of LARGE on:
  - DD statement
  - TSO/E ALLOCATE command
  - IDCAMS ALLOCATE
  - Dynamic allocation (SV 99)
- Implicit: Assign the value LARGE to an SMS data class. This will take effect (for both SMS and non-SMS managed data sets) if no other DSNTYPE is specified on the data set allocation.

The large format data set must have a DSORG value of PS or PSU, or have no DSORG specified on allocation.

The DSNTYPE of LARGE becomes a permanent attribute of the associated DSCB (bit DS1Large in DS1FLAG1 field).

- You cannot add or remove the attribute for an existing data set.
- To change a data set to or from large format, the data set has to be deleted and recreated.

Allocation using JCL

Figure 2-3 shows an example of allocating a large format data set using the JCL DD statement DSNTYPE operand.

```
//MHLRES1E JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//LARGEDS  EXEC PGM=IEFBR14
//DDALLOC  DD  DISP=(,CATLG),UNIT=SYSALLDA,
//             DSN=MHLRES1.LARGDS.JCL,
//             SPACE=(TRK,(65600,1)),
//             DSNTYPE=LARGE
```

Figure 2-3 Large format data set allocation using JCL DD statement

Allocation using TSO

Figure 2-4 shows an example of allocating a large format data set using the TSO ALLOCATE command statement DSNTYPE operand. There is no response to this command if it works correctly. In this case the high-level qualifier, using standard TSO conventions, is the user ID of the user issuing the command.

```
alloc da(dsntype.large.tso) dsntype(large) new
```

Figure 2-4 Large format data set allocation using the TSO allocate command

Display of data set attributes using ISPF option 3.2

Figure 2-5 shows the result of an ISPF 3.2 display of data set MHLRES1.DSNTYPE.LARGE. This also demonstrates that a data set may be in large format while being less than 65,535 tracks.

```
Command ===>
                                                                   More:     +
Data Set Name . . . . : MHLRES1.DSNTYPE.LARGE

General Data                           Current Allocation
 Management class . . : **None**        Allocated tracks  . : 1
 Storage class  . . . : **None**        Allocated extents . : 1
  Volume serial . . . : SBOX20
  Device type . . . . : 3390
 Data class . . . . . : **None**       Current Utilization
  Organization  . . . : PS              Used tracks . . . . : 0
  Record format . . . : FBA             Used extents  . . . : 0
  Record length . . . : 133
  Block size  . . . . : 2660
  1st extent tracks . : 1
  Secondary tracks  . : 1
  Data set name type  : LARGE           SMS Compressible  :   NO

  Creation date . . . : 2006/03/24      Referenced date . . : 2006/03/24
  Expiration date . . : ***None***
```

Figure 2-5 ISPF display of data set showing data set name type LARGE

Display of data set attributes using ISPF option 3.4

Figure 2-6 shows the result of an ISPF 3.4 display of data sets with prefix MHLRES1.DSNTYPE, showing the PS-L attribute as well as other data sets that have the LARGE attribute.

```
DSLIST - Data Sets Matching MHLRES1                              Row 19 of 82
Command ===>                                                   Scroll ===> CSR

Command - Enter "/" to select action                  Dsorg  Recfm  Lrecl  Blksz
-------------------------------------------------------------------------------
         MHLRES1.DSNTYPE.LARGE                        PS-L  FBA      133   2660
         MHLRES1.DSNTYPE.LARGE.TSO                    PS-L  ?          0      0
         MHLRES1.DSS1P6.JCL                           PO    FB        80  27920
         MHLRES1.DSS1P6.PDS                           PO    U          0  32760
         MHLRES1.DSS1P6.UIM                           PO    FB        80  27920
         MHLRES1.DSS1P6.UIMEX                         PS    U          0  27998
         MHLRES1.DSS1P6.XMITBIN                       PS    FB        80   6160
         MHLRES1.DSS1P6EX.JCL                         PO    FB        80  27920
         MHLRES1.EXTPREF.DATA                               ?          0      0
         MHLRES1.EXTREQ.DATA                                ?          0      0
         MHLRES1.HSM.BACKUP                           PS    VB      2040  27998
         MHLRES1.JOBS.HSM.FAST.REPL                   PO    FB        80   3200
         MHLRES1.JOBS.JCL                             PO    FB        80   3200
         MHLRES1.LARGDS.JCL                           PS-L  ?          0      0
         MHLRES1.LARGDS1.JCL                          PS-L  ?          0      0
         MHLRES1.LARGDS2.JCL                          PS-L  ?          0      0
```

Figure 2-6 ISPF 3.4 list of data sets showing annotation for those with DSNTYPE LARGE

2.1.6 Application programming

After you have enabled support for large format data sets, you might need to modify some applications to work with them.

> **Note:** This section is oriented towards assembler coding. Large format data support by high-level languages is dependent on their current releases.

Modifying applications to use large format data sets

Applications can use the QSAM, BSAM, and EXCP access methods to access large format data sets. In many cases, depending on the access method and functions that the application uses, modifications are required. Take the following steps to modify your applications as appropriate:

1. Specify the BLOCKTOKENSIZE=LARGE parameter on the DCBE macro.

   This parameter signifies that the application is prepared to access large format data sets, and includes all the following modifications that apply. By default in z/OS V1R7 (unless the IGDSMSxx member of SYS1.PARMLIB specifies BLOCKTOKENSIZE(NOREQUIRE)), this parameter is required in order to open a large format data set except when the OPEN is for input only and the data set contains no more than 65,535 tracks on each volume. If the system programmer has activated the IGDSMSxx member of SYS1.PARMLIB with BLOCKTOKENSIZE(NOREQUIRE), then programs that do not have a DCBE with BLOCKTOKENSIZE=LARGE can work under the conditions described earlier in 2.1.3, “Setting up support for large format data sets” on page 13.

2. Change BSAM programs that use the NOTE or POINT macros.

   The BSAM NOTE and POINT macros normally use a four-byte field for the relative track and record number of a data block, in the format TTR0. To accommodate longer track numbers for large format data sets, programs that use these macros must specify BLOCKTOKENSIZE=LARGE on the DCBE macro. This causes the macros to provide and accept the track and record values in the format TTTR. The POINT macro also has a new RELNEXT parameter to position the data set after the record indicated in the NOTE value for large format data sets. Update the application program to these new track value formats and macro parameters as appropriate. If you update your program in this way and run it on z/OS V1R7 or later, then it also can run with any other DASD data set that is not large format and supports NOTE and POINT. We recommend making this change so that your programs will support the maximum number of types of data set. For more information see the NOTE and POINT macro descriptions in *DFSMS Macro Instructions for Data Sets*, SC26-7408.

3. Change programs that use or examine the DS1LSTAR field in the format 1 DSCB.

   The first two bytes of the DS1LSTAR field of the DSCB contains the track number of the last track used. If the DS1LARGE bit is on, then the data set is large format and the third (high-order) byte of the last-used track number is in the new byte DS1TTTHI at offset X'68' in the DSCB. Update the application program to use or examine this new field as well.

4. Change programs that use or examine the DEBNMTRK field in the data extent block (DEB) for an EXCP, BSAM, or QSAM DCB.

   The DEBNMTRK field of the DEB contains the number of tracks in the extent. This is a two-byte field. For large format data sets, an additional high-order byte of the track number (if any) may be contained in the DEBNmTrkHi field. Update the application program to use or examine this additional field as well.

5. Change QSAM, BSAM, BPAM or EXCP programs that use track address conversion routines CVTPRLTV and CVTPCNVT.

   These routines convert between absolute and relative track and record values, using a three-byte TTR0 format prior to z/OS V1R7. For large format data sets, these routines have new entry points that use three bytes for the track number, in TTTR format. The new entry points also work with all data sets that the old entry points work with. For details about the changes to these routines, see *DFSMS Advanced Services*, SC26-7400, and update any application programs that may use them for large format data sets.

6. Change programs that calculate DASD space usage.

   Programs that calculate DASD space usage should check the new DS1LARGE bit (X’08’) in the DSFLAG1 field of the DSCB, which is set when a large format data set is allocated. If the bit is on, the program should allow for the possibility that any extent except for a user label extent might exceed 65,535 tracks. A program that scans the output of IEHLIST LISTVTOC might not work as expected because the format of DS1LSTAR output was changed slightly. See *DFSMS Utilities*, SC26-7414, for details.

7. Change programs that examine SMF type 14 and 15 records.

   The SMF type 14 and 15 records have some slightly changed information for large format data sets. These two records have a field that contains the number of tracks released by DADSM for RLSE. For a sequential data set, SMF14NTU is a four-byte field that contains the relative track in TTR0 format of the last processed record. The first bit in the SMF14PG1 field is an indicator that SMF14NTU is in the format TTTR instead of TTR0. The bit name is SMF14LGE.

2.1.7 Diagnosing problems with large format data sets

The following ABEND code indicates that an OPEN or end-of-volume operation failed for a large format data set.

For updated information about these ABENDS, refer to the latest z/OS V1R7 version of the manual *MVS System Messages Vol 7 (IEB-IEE),* SA22-7637.

Abend 213-xx refer to message IEC143I

These ABENDS may occur on the z/OS V1R7 system or on a prior level system that has the appropriate coexistence maintenance installed. If the coexistence maintenances is not installed, results will be unpredictable.

- 213-10: OPEN failed because the data set type was large format sequential, but the access method was not QSAM, BSAM, or EXCP, or the DCB did not specify DSORG=PS.
- 213-14: OPEN failed because the data set type was large format sequential, but the application program could not access the whole data set on the volume. Both of the following are true:
  - The access method is either EXCP being opened for UPDAT or output processing or the access method is BSAM with the NOTE or POINT function requested and is being opened for output processing.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE.
- 213-15: OPEN failed because the data set type was large format sequential but the application program could not access the whole data set on the volume. All of the following are true:
  - The access method is EXCP, BSAM, or QSAM being opened for UPDAT or output processing.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE.
  - BLOCKTOKENSIZE(REQUIRE) is specified or defaulted in PARMLIB member IGDSMSxx.
- 213-16: OPEN failed because the data set type was large format sequential, but the application program could not access the whole data set on the volume. All of the following are true:
  - The access method is either EXCP or it is BSAM with the NOTE or POINT function requested.
  - The DCB is being opened for input processing.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE.
  - The data set has more than 65535 tracks allocated on the volume.
- 213-17: OPEN failed because the data set type was large format sequential, but the application program could not access the whole data set on the volume. All of the following are true:
  - The access method is EXCP, BSAM, or QSAM being opened for input processing.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE.
  - The data set has more than 65,535 tracks allocated on the volume.
  - BLOCKTOKENSIZE(REQUIRE) was specified or defaulted in PARMLIB member IGDSMSxx.

For Abend 737-xx refer to message IEC027I

- 737-44: EOV failed when switching to a volume after the first volume of a data set or any volume in a concatenation because the data set type is large format sequential but the application program could not access the whole data set on the volume. All of the following are true:
  - The access method is either EXCP or BSAM with the NOTE or POINT function requested.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE macro.
  - The data set has more than 65,535 tracks allocated on the volume.
- 737-45: EOV failed when switching to a volume after the first volume of a data set or any volume in a concatenation because the data set type is large format sequential but the application program could not access the whole data set on the volume. All of the following are true:
  - The access method is EXCP, BSAM, or EXCP.
  - BLOCKTOKENSIZE=LARGE is not specified on the DCBE macro.
  - The data set has more than 65,535 tracks allocated on the volume.
  - BLOCKTOKENSIZE(REQUIRE) is specified or defaulted in PARMLIB member IGDSMSxx.

2.1.8 Large format data set coexistence with lower level systems

There are restrictions on the use of large format data sets on lower level systems:

- On a z/OS system lower than V1R7, programs can open large format data sets only for input and only if the data set has no more than 65,535 tracks on each volume.
- Jobs that specify a DSNTYPE of LARGE or BASIC on a pre-V1R7 will fail.
- If an SMS active configuration data set is shared between a z/OS V1R7 system and a lower-level system, and a data class has a DSNTYPE value of LARGE or BASIC, the lower-level system will ignore that value.
- The IGDSMSxx member of SYS1.PARMLIB that contains the BLOCKTOKENSIZE parameter cannot be shared with a lower-level system.

Note: Services on lower-level systems such as ISPF, ISMF, IEHLIST, and LISTDSI that display data set information may show incorrect amounts of space used by large format data sets.

In 2.1.9, “Maintenance support in lower-level systems for large format data sets” on page 23, we show the minimum level PTF to be installed for each prior release. If these PTFs are not installed, results will be unpredictable.

2.1.9 Maintenance support in lower-level systems for large format data sets

Table 2-1 shows the minimum level PTF to be installed for each prior release of DFSMS to support large format data sets.

Table 2-1 DFSMSdfp large format data set support for PTFs

Feature and reason PTFs are needed	z/OS V1R4 PTFs	z/OS V1R5 PTFs	z/OS V1R6 PTFs
DFSMSdfp: z/OS V1R7 supports large format sequential data sets. These PTFs cause pre-z/OS V1R7 to issue ABEND 213-14 or 213-16 if a program attempts to open data sets that cannot work on those systems.	UA15869 UA16550	UA15870 UA16551	UA15871 UA16552
ISPF: Allows pre-z/OS V1R7 systems to intelligently display the number of tracks in a large format sequential data set. If the number of tracks is greater than 64 KB, the string >64K and a larger explanatory message are written to the panel.	UA19928	UA19929	UA19930
ISPF: Provides an appropriate message when a user attempts to open a large format sequential data set from a pre-z/OS V1R7 release. (Only the standard English language PTF is shown. There are co-requisites for certain languages and uppercase English.)	UA19931	UA19932	UA19933

2.2 Data set allocation

In addition to the setting for DSNTYPE to allocate a large format data set, in z/OS V1R7 there are additional new values that can be specified with DSNTYPE on the JCL DD statement and though the dynamic allocation equivalent. These additional values do not designate new types of data sets. They are provided to give additional control over data set type specification, or in the case of BASIC to provide a way of specifying that a data set must not be a large-format version.

The DSNTYPE keyword options, ACS data class attribute, and read-only variable &DSNTYPE presented in ACS routines are:

- LARGE - Refer to “Creating large format data sets” on page 17.
- BASIC - Refer to “Creating basic format data sets” on page 23.
- EXTREQ - Refer to “Specifying a DSNTYPE of EXTREQ or EXTPREF” on page 24.
- EXTPREF - Refer to “Specifying a DSNTYPE of EXTREQ or EXTPREF” on page 24.

In this section we discuss the BASIC, EXTREQ, and EXTPREF values.

2.2.1 Creating basic format data sets

z/OS V1R7 introduces the new DSNTYPE value BASIC to designate a sequential data set that is neither extended format nor large format. The term *basic format data set* therefore refers to the old type of sequential data set. A basic format data set, besides being sequential, may not have any more than 65,535 tracks per volume.

To create a basic format data set the following methods are available:

- Explicit: Specify a DSNTYPE of BASIC on:
  - DD statement
  - TSO/E ALLOCATE command
  - IDCAMS ALLOCATE
  - Dynamic allocation (SV 99)

  > **Note:** The BASIC attribute can be used on a DD statement via dynamic allocation to override a data class DSNTYPE of LARGE, EXTREQ, or EXTPREF.

- Implicit: Omit any value for DSNTYPE in the SMS data class. You cannot specify the value BASIC for DSNTYPE in a data class.

Normally, a basic format data set will have a DSORG value of PS or PSU, or have no DSORG specified on allocation. There is no specific attribute of the associated DSCB.

> **Restriction:** You cannot add the large format data set attribute for an existing data set. To change a data set to or from basic format, the data set has to be deleted and recreated.

Specifying a DSNTYPE of EXTREQ or EXTPREF

z/OS V1R7 introduces the new DSNTYPE values EXTREQ and EXTPREF to designate an extended format data set. Previously, these attributes could only be specified in a data class as P (preferred) or R (required) for a data set type of EXT.

To create an extended format data set you can use the explicit or implicit method. To use an explicit method, specify a DSNTYPE of EXTREQ or EXTPREF on:

- DD statement
- TSO/E ALLOCATE command
- IDCAMS ALLOCATE
- Dynamic allocation (SV 99)

The difference between EXTREQ and EXTPREF is:

- EXTREQ specifies that the allocation is required to be in extended format, and the allocation should fail if that is not possible.
- EXTPREF specifies that the allocation is preferred to be in extended format, but if the necessary resources are not available, then the data set should be allocated as BASIC.

New values for DSNTYPE in ACS routines

The &DSNTYPE read-only variable passed to ACS routines can have new values:

- LARGE - caused by the user coding a DSNTYPE of LARGE, being picked up from a LIKE data set, or by the data class supplying it.
- BASIC - caused by the user coding a DSNTYPE of LARGE, being picked up from a LIKE data set. The value BASIC cannot be specified in a data class, but omission of DSNTYPE is the equivalent of specifying BASIC.
- EXR - caused by the user coding a DSNTYPE of EXTREQ, being picked up from a LIKE data set, or by the data class supplying it.
- EXC - caused by the user coding a DSNTYPE of EXTPREF, being picked up from a LIKE data set, or by the data class supplying it.

2.3 Device support address space (DEVMAN)

z/OS V1R7 now uses a device support address space (DEVMAN). It is started during an IPL, and it is cancelable, non-swapable, and is restartable with the START command. z/OS uses the new address space to:

- Capture CTRACE information for CVAF events for first failure data capture.
- Capture CTRACE information for DADSM create events for first failure data capture.

This replaces the CVAF trace output to GTF. CVAF or DADSM CTRACE events must be enabled.

2.3.1 Implementation of DEVMAN

The started device support address space will be identified in the system as DEVMAN. This name will be seen in system displays and will be used as the target address space name for modify commands.

You do not need to do anything for normal operation of DEVMAN. It will start automatically at IPL time. You may, however, take advantage of the new component trace (CTRACE) facilities as required. A number of commands are provided to support DEVMAN.

Cancel DEVMAN

You can cancel DEVMAN using the CANCEL DEVMAN command. After you cancel DEVMAN you should restart it.

Restart DEVMAN

You do not start DEVMAN normally. It is started as part of the IPL process. However, if DEVMAN is cancelled you use the start command to restart DEVMAN, as shown in Figure 2-7. Nothing else is required.

```
s devman
```

Figure 2-7 Restarting DEVMAN

Modify DEVMAN commands

There are several MOFDIFY commands in support of DEVMAN.

QUERY VERSION

You can find the FMID and PF version of DEVMAN by using the MODIFY DEVMAN,QUERY=VERSION command.

In Figure 2-8 we show the MODIFY DEVMAN,QUERY=VERSION command and response. In this case no PTFs have been applied.

```
F DEVMAN,QUERY=VERSION
DMO0008I DEVICE MANAGER FMID=HDZ11K0  PTF=NONE
```

Figure 2-8 MODIFY DEVMAN,QUERY=VERSION

DUMP=DUMP1

You can force an SVC dump of the DEVMAN address space and its associated data spaces and trace table using the MODIFY DEVMAN,DUMP=DUMP1 command. The operand DUMP1 is the only choice available.

In Figure 2-9 we show the results of using the MODIFY DEVMAN,DUMP=DUMP1 command. The result is a data set that you can analyze using IPCS.

```
F DEVMAN,DUMP=DUMP1
IEA794I SVC DUMP HAS CAPTURED: 953
DUMPID=002 REQUESTED BY JOB (DEVMAN  )
DUMP TITLE=COMPONENT=DEVICE MANAGER,COMPID=DF133,ISSUER=DMOVS00
           1,JOBNAME=DEVMAN
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00012)
IEF196I         DSN (DUMP.D060404.H16.SC63.DEVMAN.S00002        )
IEF196I         STORCLAS (SCDUMP) MGMTCLAS (MCDB22) DATACLAS (
)
IEF196I         VOL SER NOS= SBOXAF
IEF196I IEF237I 8603 ALLOCATED TO SYS00112
IEF196I IEF285I   SYS1.LINKLIB                               KEPT
IEF196I IEF285I   VOL SER NOS= Z17RC1.
IEF196I IGD104I DUMP.D060404.H16.SC63.DEVMAN.S00002         RETAINED,
IEF196I DDNAME=SYS00012
IEA611I COMPLETE DUMP ON DUMP.D060404.H16.SC63.DEVMAN.S00002 960
DUMPID=002 REQUESTED BY JOB (DEVMAN  )
FOR ASID (000F)
INCIDENT TOKEN: SANDBOX  SC63     04/04/2006 16:04:48
```

Figure 2-9 MODIFY DEVMAN,DUMP=DUMP1

TRACE=FLUSH and TRACE=CLEAR

Once you have started CTRACE (see 2.3.2, “Start component trace (CTRACE)” on page 26), you may need to force the records out or clear them.

You can force the component trace records in the DEVMAN address space and its associated data spaces and trace table to be written to the CTRACE external writer (FLUSH) or deleted (CLEAR).

You use the command MODIFY DEVMAN,TRACE=FLUSH to have the records written to the external writer.

You use the command MODIFY DEVMAN,TRACE=CLEAR to have all the component trace records deleted.

2.3.2 Start component trace (CTRACE)

You can set CTRACE up to capture information in the DEVMAN address space and to write it to an external writer. The procedures to follow are slightly different depending on which mode you want to use.

To have the CTRACE just capture to the DEVMAN address space refer to “CTRACE start without using the external writer” on page 27.

To have CTRACE write to the external writer refer to 2.3.4, “CTRACE start using the CTRACE external writer” on page 28.

CTRACE start without using the external writer

You must start CTRACE, then respond to prompts to indicate what events you want traced. The trace component name for DEVMAN is SYSDMO.

The possible responses to the prompt for options are:

- DADSM1 for DADSM events
- CVAF1 for CVAF events

Either or both can be specified.

You must respond with the names of the events that you want traced, in this case DADSM1 and CVAF1.

In Figure 2-10 we show the TRACE,CT,ON,COMP=SYSDMO command and response to activate both DADSM1 and CVAF1.

```
 TRACE CT,ON,COMP=SYSDMO
*306 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 306,OPTIONS=(DADSM1,CVAF1),END
 IEE600I REPLY TO 306 IS;OPTIONS=(DADSM1,CVAF1),END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 506
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Figure 2-10 TRACE ON command for SYSDMO

You can use the DISPLAY TRACE command to check whether the SYSDMO trace is active.

In Figure 2-11 we show the DISPLAY TRACE command and response. In this case we are interested in the status of SYSDMO, which shows as ON.

```
 DISPLAY TRACE
 IEE843I 18.33.00  TRACE DISPLAY 541
         SYSTEM STATUS INFORMATION
  ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K)
  COMPONENT MODE  COMPONENT MODE  COMPONENT MODE  COMPONENT MODE
  --------------------------------------------------------------
  CSF       MIN   SYSGRS    MIN   SYSJES2   SUB   SYSANTOO  MIN
  I8F1      SUB   SYSRRS    MIN   SYSSPI    OFF   SYSJES    SUB
  ID9A      SUB   CL6631    ON    SYSHZS    ON    SYSSMS    OFF
  CSQXMQBA  OFF   SYSDLF    MIN   SYSOPS    ON    SYSXCF    ON
  SYSLLA    MIN   SYSXES    ON    SYSAPPC   OFF   SYSTTRC   OFF
  SYSTCPDA  SUB   SYSRSM    OFF   IRLH      SUB   SYSAOM    OFF
  SYSVLF    MIN   SYSTCPIP  SUB   SYSLOGR   ON    SYSOMVS   ON
  SYSWLM    MIN   SYSTCPIS  SUB   SYSTCPRE  SUB   SYSIOS    MIN
  ID8A      SUB   SYSANTMN  MIN   CL5632    ON    SYSDMO    ON
  SYSIEFAL  ON
```

Figure 2-11 DISPLAY TRACE command to determine which traces are active

2.3.3 Disable component trace (CTRACE)

To disable component trace, the special events DADSM0 and CVAF0 are provided. These are recognized within the DEVMAN code as indicators that tracing is to be disabled.

To change from DADSM1 or CVAF1 you must issue the start CTRACE command, then respond with the special prompts to disable tracing.

The trace component name for DEVMAN is SYSDMO. The function is turned off by using the special event names DADSM0 and CVAF0.

The possible responses to the prompt for options are:

- DADSM**0** for DADSM events
- CVAF**0** for CVAF events

Either or both can be specified.

In Figure 2-12 we show the TRACE CT command followed by the reply that disables CTRACE on DASDSM and CVAF by using the special event identifiers DADSM0 and CVAF0.

```
 TRACE CT,ON,COMP=SYSDMO
*338 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 338,OPTIONS=(DADSM0,CVAF0),END
 IEE600I REPLY TO 338 IS;OPTIONS=(DADSM0,CVAF0),END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 635
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Figure 2-12 Disabling CTRACE events

2.3.4 CTRACE start using the CTRACE external writer

To send CTRACE data to the CTRACE external writer, steps are similar to capturing the data to the DEVMAN address space, but you must start the CTRACE external writer and indicate that it is to be used. You must:

- Set up the CTWTR procedure.
- Start the CTRACE external writer.
- Start CTRACE, then respond to prompts to indicate that you want to use the writer and to specify what events you want traced.

The trace component name for DEVMAN is SYSDMO.

The external writer required prompt is WTR=CWTR.

The possible responses to the prompt for options are:

- DADSM1 for DADSM events
- CVAF1 for CVAF events

Either or both can be specified.

2.3.5 Set up CTRACE external writer CTWTR

CTWTR is the name of the JCL procedure to be used. In Figure 2-13 we show an example of the CTWTR procedure set up to record to the single data set MHLRES4.TEST.CTRACE3.

```
//CTWTR   PROC
//*  PROC USED FOR CAPTURING  CTRACE OUTPUT TO AN EXTERNAL  *//
//*  WRITER ... CHANGE THE OUTPUT DATA SETS AS REQUIRED TO  *//
//*  A MAX OF 16                                            *//
//*                                                         *//
//*                                                         *//
//*                                                         *//
//IEFPROC  EXEC PGM=ITTTRCWR
//TRCOUT03 DD  DSNAME=MHLRES4.TEST.CTRACE3,UNIT=3390,
//         SPACE=(CYL,100),DISP=(NEW,KEEP),DSORG=PS
```

Figure 2-13 CTWTR CTRACE external writer procedure

2.3.6 Start CTRACE external writer

You use the TRACE CT,WTRSTART command to start the CTRACE external writer.

In Figure 2-14 we show the START CT,WTRSTART,WTR=CTWTR command and response.

```
 TRACE CT,WTRSTART=CTWTR
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 591
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
 IEF196I         1 //CTWTR    JOB MSGLEVEL=1
 IEF196I         2 //STARTING EXEC CTWTR
 IEF196I  STMT NO. MESSAGE
 IEF196I         2 IEFC001I PROCEDURE CTWTR WAS EXPANDED USING SYSTEM
 IEF196I LIBRARY SYS1.PROCLIB
 IEF196I         3 XXCTWTR   PROC
 IEF196I           XX*  PROC USED FOR CAPTURING  CTRACE OUTPUT TO AN
 IEF196I EXTERNAL  *//
 IEF196I           XX*  WRITER ... CHANGE THE OUTPUT DATA SETS AS
 IEF196I REQUIRED TO  *//
IEF196I           XX*  A MAX OF 16
IEF196I       *//
IEF196I           XX*
IEF196I       *//
IEF196I           XX*
IEF196I       *//
IEF196I           XX*
IEF196I       *//
IEF196I         4 XXIEFPROC  EXEC PGM=ITTTRCWR
IEF196I         5 XXTRCOUT03 DD  DSNAME=MHLRES4.TEST.CTRACE3,
        UNIT=3390,
IEF196I           XX         SPACE=(CYL,100),DISP=(NEW,KEEP),DSORG=PS
IEF196I           XX*TRCOUT02 DD  DSNAME=MHLRES4.TEST.CTRACE2,
        UNIT=3390,
IEF196I           XX*        SPACE=(CYL,100),DISP=(NEW,KEEP),DSORG=PS
IEF196I         5 IGD01007I &DSNTYPE =
 IEF196I         5 IGD01009I MC ACS GETS CONTROL &ACSENVIR=ALLOC
 IEF196I         5 IGD01010I SG ACS GETS CONTROL &ACSENVIR=ALLOC
 IEF403I CTWTR - STARTED - TIME=20.03.42 - ASID=0095 - SC63
 IEF196I IEF236I ALLOC. FOR CTWTR CTWTR
 IEF196I IGD101I SMS ALLOCATED TO DDNAME (TRCOUT03)
 IEF196I         DSN (MHLRES4.TEST.CTRACE3                        )
 IEF196I         STORCLAS (STANDARD) MGMTCLAS (MCDB22) DATACLAS (
 IEF196I )
 IEF196I         VOL SER NOS= MHLS1A
 ITT110I INITIALIZATION OF CTRACE WRITER CTWTR COMPLETE.
```

Figure 2-14 TRACE CT,WTRSTART command

2.3.7 Start CTRACE with CTRACE external writer

Once the CTRACE writer is running you start CTRACE with the command TRACE CT,ON,COMP=SYSDMO (the same as without the CTRACE writer). Respond to the prompt with the options:

- WTR=CTWTR
- OPTIONS=(CVAF1,DADSM1),END

Note: The OPTIONS would be the same as when not using the CTRACE writer.

In Figure 2-15 we show the command and response to the message, this time adding WTR=CTWTR and using the same options as before.

```
 TRACE CT,ON,COMP=SYSDMO
*308 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 308,WTR=CTWTR,OPTIONS=(CVAF1,DADSM1),END
 IEE600I REPLY TO 308 IS;WTR=CTWTR,OPTIONS=(CVAF1,DADSM1),END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 636
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Figure 2-15 CTRACE with reply to use CTRACE WRITER

You can see the status of the DEVMAN CTRACE by using the command D TRACE,COMP=SYSDMO.

In Figure 2-16 we show the D TRACE,COMP=SYSDMO command with its response. It shows that the SYSDMO trace is on, that it is tracing the VCAF1 and DADSM1 events, and that it is using the CTRACE writer.

```
 D TRACE,COMP=SYSDMO
 IEE843I 20.23.26  TRACE DISPLAY 643
         SYSTEM STATUS INFORMATION
  ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K)
   COMPONENT     MODE BUFFER HEAD SUBS
  -------------------------------------------------------------
   SYSDMO        ON
      ASIDS      *NOT SUPPORTED*
      JOBNAMES   *NOT SUPPORTED*
      OPTIONS    CVAF1,DADSM1
      WRITER     CTWTR
```

Figure 2-16 DISPLAY TRACE,COMP=SYSDMO

2.3.8 Disable CTRACE when using CTRACE external writer

To disable CTRACE when using the CTRACE external writer:

- Issue the command TRACE CT,ON,COMP=SYSDMO.
- Reply to the ITT006A console message with OPTIONS=(DADSM0,CVAF0),END.

  The special options DADSM0 and CVAF0 will tell DEVMAN to disable tracing.

In Figure 2-17 we show the commands and responses to change the events being monitored to DADSM0 and CVAF0, and verify the change.

```
 TRACE CT,ON,COMP=SYSDMO
*309 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 309,OPTIONS=(DADSM0,CVAF0),END
 IEE600I REPLY TO 309 IS;OPTIONS=(DADSM0,CVAF0),END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 651
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
 D TRACE,COMP=SYSDMO
 IEE843I 20.31.38  TRACE DISPLAY 659
         SYSTEM STATUS INFORMATION
  ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K)
   COMPONENT     MODE BUFFER HEAD SUBS
  --------------------------------------------------------------
   SYSDMO        ON
      ASIDS      *NOT SUPPORTED*
      JOBNAMES   *NOT SUPPORTED*
      OPTIONS    DADSM0,CVAF0
      WRITER     CTWTR
```

Figure 2-17 CTRACE event change to DADSM0 and CVAF0

2.3.9 Stop CTRACE external writer

Once the CTRACE is no longer recording DADSM1 and CVAF1 events, you can shut down the CTRACE external writer with the TRACE CT,WTRSTOP=CTWTR command.

In Figure 2-18 we show the CTRACE external writer being closed down.

```
 TRACE CT,WTRSTOP=CTWTR
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.
 IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 665
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
 ITT111I CTRACE WRITER CTWTR TERMINATED BECAUSE OF A WTRSTOP REQUEST.
 -                                         --TIMINGS (MINS.)--
    ----PAGING COUNTS---
000290  -CTWTR           IEFPROC     00     20    .00    .00  33.57    261
         0      0      0      0     0     1
000010  IEF404I CTWTR - ENDED - TIME=20.37.17 - ASID=0095 - SC63
000290  -CTWTR    ENDED.  NAME-                    TOTAL CPU TIME=   .00
        TOTAL ELAPSED TIME= 33.57
000290  IEF196I IEF142I CTWTR CTWTR - STEP WAS EXECUTED - COND CODE 0000
000290  IEF196I IGD104I MHLRES4.TEST.CTRACE3                        RETAINED,
000290  IEF196I DDNAME=TRCOUT03
000290  IEF196I IEF373I STEP/IEFPROC /START 2006093.2003
000290  IEF196I IEF374I STEP/IEFPROC /STOP  2006093.2037 CPU    0MIN 00.01SEC
000290  IEF196I SRB    0MIN 00.00SEC VIRT    56K SYS   188K EXT     604K SYS
000290  IEF196I 9336K
000290  IEF196I IEF375I  JOB/CTWTR   /START 2006093.2003
000290  IEF196I IEF376I  JOB/CTWTR   /STOP  2006093.2037 CPU    0MIN 00.01SEC
000290  IEF196I SRB    0MIN 00.00SEC
```

Figure 2-18 CTRACE external writer closedown

2.3.10 Analyze the CTRACE data using IPCS

You can most easily analyze the CTRACE data by using IPCS.

2.4 DFSMS subchannel set support

There is a 64-K subchannel limit for UCBs in z/Series software. In z/OS V1R7, using the DFSMS subchannel support enhancement, Parallel Access Volume (PAV) users can define a second set of subchannels to be used with PAV to redefine aliases into the second subchannel set. This frees up device numbers on the first subchannel set for use as additional base devices. PAV users can increase the number of available devices. Device numbers can be duplicated in the same channel subsystem by being in both subchannel sets. PAV alias devices must be defined to an alternative subchannel set using HCD or HCM.

2.4.1 Restrictions

Some restrictions are:

- This support is limited to devices that support alternate subchannel sets.
- An IODF defined with PAV alias UCBs in an alternate subchannel set cannot be shared with a processor that does not support multiple subchannel sets.
- The four-digit device number limit remains unchanged.

2.4.2 Defining devices

PAV aliases must be defined to an alternative subchannel set using HCD or HCM. For more information refer to the manuals *z/OS Hardware Configuration Definition User's Guide*, SC33-7988, and *z/OS and z/VM Hardware Configuration Manager User's Guide*, SC33-7989.

2.4.3 Displaying PAV device address status

You can use the updated DEVSERV QPAVS command to learn which subchannel set a device address is associated with, and you can, if your system has more than one subchannel set, restrict the display to a particular subchannel set.

Refer to *MVS System Commands,* SA22-7627, for the full list of options on the DEVSERV command.

We provide examples in this section to illustrate the change in format to accommodate the addition of the subchannel set. In z/OS V1R7 the subchannel set is always provided when requested, even if the system has only one subchannel.

DEVSERV QPAVS display - one device address

You can display just once device address, which might be a base address or a PAV address, or you can extend the command to find all addresses, base or PAV, that relate to whatever you asked for.

In Figure 2-19 we show the DS QPAVS command and response with one BASE device address D00C. Only information relating to that device address is returned. The subchannel set that the unit address is on is shown as an additional digit before the device address in the UNIT NUM column.

```
DS QPAVS,D00C
IEE459I 14.45.46 DEVSERV QPAVS 030
     HOST                                SUBSYSTEM
 CONFIGURATION                         CONFIGURATION
---------------                   ---------------------
 UNIT                                  UNIT    UA
 NUM. UA  TYPE        STATUS      SSID  ADDR.   TYPE
----- --  ----        ------      ----  ----    ----------
0D00C 0C  BASE                    89E0   0C     BASE
****      1 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 2-19 DEVSERV QPAVS with base address specified

DEVSERV QPAVS display - base device address and related information

You can extend the command to find all addresses, base or PAV, that relate to whatever device address you asked for without knowing any more about the equipment, by using the VOLUME parameter.

In Figure 2-20 we show the DS QPAVS command and response with a BASE device address D00C specified, and request additional information via the VOLUME parameter. Information relating to the base address is returned as before, as is information to a PAV address associated with that base address. The subchannel is shown on the PAV address and on the base address as before. If the PAV address had been defined on a different subchannel, it would be shown. There may be PAV addresses with different subchannels relating back to the same base address.

```
DS QPAVS,D00C,VOLUME
IEE459I 14.54.21 DEVSERV QPAVS 135
     HOST                             SUBSYSTEM
 CONFIGURATION                      CONFIGURATION
---------------                  ---------------------
 UNIT                                  UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
----- --  ----        ------     ----  ----    ----------
0D00C 0C  BASE                   89E0   0C     BASE
0D07D 7D  ALIAS-D00C             89E0   7D     ALIAS-0C
****      2 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 2-20 DEVSERV QPAVS with base address and VOLUME parameter

DEVSERV QPAVS display - PAV address plus related information

You can specify a PAV address in the DEVSERV command, in which case, if the VOLUME parameter is specified, it will return information about the base address.

In Figure 2-21 we show the DS QPAVS command and response with a PAV device address D07D specified, and request additional information via the VOLUME parameter. The resulting display is the same, in this case, as the display of the BASE device address because there is only one PAV address associated with that base address.

```
DS QPAVS,D07D,VOLUME
IEE459I 15.05.35 DEVSERV QPAVS 183
     HOST                             SUBSYSTEM
 CONFIGURATION                      CONFIGURATION
---------------                  ---------------------
 UNIT                                  UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
----- --  ----        ------     ----  ----    ----------
0D00C 0C  BASE                   89E0   0C     BASE
0D07D 7D  ALIAS-D00C             89E0   7D     ALIAS-0C
****      2 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 2-21 DEVSERV QPAVS with PAV address and VOLUME parameter

DEVSERV QPAVS display - base address with many PAV addresses

You can specify a PAV address in the DEVSERV command, in which case, if the VOLUME parameter is specified, it will return information about the base address and any other PAV addresses that may be related to that base address.

In Figure 2-22 we show the DS DEVSERV command specifying device address D000. This is a base device address that has several PAV addresses associated with it, all on subchannel 0.

```
DS QPAVS,D000,VOLUME
IEE459I 15.11.07 DEVSERV QPAVS 596
     HOST                             SUBSYSTEM
 CONFIGURATION                      CONFIGURATION
---------------                  ---------------------
 UNIT                                  UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
----- --  ----        ------     ----  ----    ----------
0D000 00  BASE                   89E0   00     BASE
0D071 71  ALIAS-D000             89E0   71     ALIAS-00
0D074 74  ALIAS-D000             89E0   74     ALIAS-00
0D079 79  ALIAS-D000             89E0   79     ALIAS-00
0D07A 7A  ALIAS-D000             89E0   7A     ALIAS-00
0D07E 7E  ALIAS-D000             89E0   7E     ALIAS-00
0D07F 7F  ALIAS-D000             89E0   7F     ALIAS-00
0D080 80  ALIAS-D000             89E0   80     ALIAS-00
0D081 81  ALIAS-D000             89E0   81     ALIAS-00
0D082 82  ALIAS-D000             89E0   82     ALIAS-00
0D083 83  ALIAS-D000             89E0   83     ALIAS-00
0D086 86  ALIAS-D000             89E0   86     ALIAS-00
0D089 89  ALIAS-D000             89E0   89     ALIAS-00
0D0E2 E2  ALIAS-D000             89E0   E2     ALIAS-00
0D0E5 E5  ALIAS-D000             89E0   E5     ALIAS-00
0D0ED ED  ALIAS-D000             89E0   ED     ALIAS-00
0D0EE EE  ALIAS-D000             89E0   EE     ALIAS-00
****     17 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 2-22 DEVSERV QPAVS with PAV address, which is one of many

2.5 REPRO MERGECAT FROMKEY/TOKEY enhancement

This enhancement enables you to repair damaged catalogs by targeting a range of catalog keys in a REPRO MERGECAT command. This makes it easier to maintain catalogs and recover from problems.

This ability is provided through the addition of two new keywords, FROMKEY and TOKEY, for use with the MERGECAT parameter of the REPRO command. FROMKEY specifies the starting key, and TOKEY specifies the ending key of the range. Using the MERGECAT FROMKEY and TOKEY parameters, you can copy unbroken segments of a damaged catalog to a new catalog. You can specify the range with specific or generic keys.

Note: It is important to remember that REPRO MERGECAT is a MOVE function. This has *not* changed. Any catalog entries that are successfully processed will no longer be in the source catalog. This means that recovery from failing processes should not include deleting the target catalog until all moved entries have been restored to the target catalog.

2.5.1 Generic keys

You can specify generic keys for either or both of FROMKEY and TOKEY. When using generic keys the folowing conditions apply:

- Copying stops after the last record whose key matches that portion of the key you specified.
- If you specify a key longer than the one defined for the data set, the data set is not copied.
- If the specified key is not found, copying ends at the next lower key.

2.5.2 FROMKEY/TOKEY use

The REPRO MERGECAT command is modified by the addition of the FROMKEY and TOKEY operands. You can use the new operands in the TSO command REPRO MERGECAT and in BATCH JCL using PGM=IDCAMS.

The FROMKEY and TOKEY may be both specific or both generic, and one may be specific and the other generic in any one execution of the REPRO MERGECAT command.

We show the form of the command for TSO, and the form of the BATCH JCL and the results of running the BATCH JCL.

TSO

We show examples of the commands for use of specific and of generic keys. Executed examples are shown under the BATCH JCL section.

In Figure 2-23 we show use of the TSO MERGECAT command with specific keys MHLCAT1.DS1 and MHLCAT1.DS7. This copies the catalog entries for all data sets starting with MHLCAT1.DS1 and ending with MHLCAT7.DS7. No entries from before MHLCAT1.DS1 (such as MHLCAT1.ANY. DATASET) or from after MHLCAT7.DS7 (such as MHLCAT8.ANY.DATASET) will be copied.

```
REPRO MERGECAT INDATASET(‘CATALOG.MERGECAT.TESTA‘)
OUTDATASET(‘CATALOG.MERGECAT.TESTB’) FROMKEY(MHLCAT1.DS1) TOKEY(MHLCAT7.DS7)
```

Figure 2-23 Use of TSO form of REPRO MERGECAT with specific keys

In Figure 2-23 we show use of the TSO MERGECAT command with generic keys MHLCAT1.A.* and MHLCAT7.G.*. This is intended to copy entries for any data sets with names starting with prefix MHLCAT1.B and ending with names with prefix MHLCAT7.G. No entries from before MHLCAT1.B (such as MHLCAT1.A) or after MHLCAT7.G (such as MHLCAT7.H) will be copied.

```
REPRO MERGECAT INDATASET(‘CATALOG.MERGECAT.TESTA‘)
OUTDATASET(‘CATALOG.MERGECAT.TESTB’) FROMKEY(MHLCAT1.B.*) TOKEY(MHLCAT7.G.*)
```

Figure 2-24 Use of TSO form of REPRO MERGECAT with generic keys

BATCH JCL

In this section we show examples of the use of specific and of generic keys in batch JCL.

In Figure 2-25 we show a list of data sets that are catalogued in USERCAT CATALOG.MERGECAT.TESTA.

```
 MHLCAT1                                                       *ALIAS
         MHLCAT1.ANY.DATASET                                           SBOX20
         MHLCAT1.DS1.TEST1                                             SBOX20
         MHLCAT1.DS1.TEST2                                             SBOX20
         MHLCAT1.DS1.TEST3                                             SBOX20
         MHLCAT1.DS3.TEST1                                             SBOX20
         MHLCAT1.DS3.TEST2                                             SBOX20
         MHLCAT1.DS3.TEST3                                             SBOX20
         MHLCAT1.DS5.TEST1                                             SBOX20
         MHLCAT1.DS5.TEST2                                             SBOX20
         MHLCAT1.DS5.TEST3                                             SBOX20
         MHLCAT1.DS7.TEST1                                             SBOX20
         MHLCAT1.DS7.TEST2                                             SBOX20
         MHLCAT1.DS7.TEST3                                             SBOX20
         MHLCAT1.DS8.TEST1                                             SBOX20
         MHLCAT1.DS8.TEST2                                             SBOX20
         MHLCAT1.DS8.TEST3                                             SBOX20
         MHLCAT1.MCAT.TEST1                                            SBOX20
         MHLCAT1.MCAT.TEST2                                            SBOX20
         MHLCAT1.MCAT.TEST3                                            SBOX20
         MHLCAT1.MCAT.TEST4                                            SBOX20
MHLCAT8                                                        *ALIAS
         MHLCAT8.ANY.DATASET                                           SBOX20
```

Figure 2-25 List of data sets for MERGECAT demonstration

In Figure 2-26 we show use of the BATCH JCL program IDCAMS with specific keys MHLCAT1.DS1 and MHLCAT1.DS7. This is intended to copy the catalog entries for all data sets starting with MHLCAT1.DS1 and ending with MHLCAT1.DS7. No entries from before MHLCAT1.DS1 (such as MHLCAT1.ANY. DATASET) or from after MHLCAT1.DS7 (such as MHLCAT1.DS8.* or MHLCAT1.MCAT.* or MHLCAT8.ANY.DATASET) will be copied.

```
//MHLRES1C JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//MRGCAT   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INDD     DD  DISP=SHR,DSN=CATALOG.MERGECAT.TESTA
//OUTDD    DD  DISP=SHR,DSN=CATALOG.MERGECAT.TESTB
//SYSIN DD *
 REPRO MERGECAT -
 INFILE(INDD) -
 OUTFILE(OUTDD) -
 FROMKEY(MHLCAT1.DS1) -
 TOKEY(MHLCAT1.DS7)
/*
```

Figure 2-26 Use of BATCH form of REPRO MERGECAT with specific keys

In Figure 2-27 we show part of the SDSF listing from the job to copy selected catalog entries between catalogs CATALOG.MERGECAT.TESTA to CATALOG.MERGECAT.TESTB. It shows that 12 data set entries were moved.

```
 SDSF OUTPUT DISPLAY MHLRES1C JOB17371  DSID     4 LINE 12      COLUMNS 02- 81
IEF376I  JOB/MHLRES1C/STOP  2006090.1816 CPU    0MIN 00.03SEC SRB    0MIN
00.00S
IDCAMS  SYSTEM SERVICES                                           TIME:
18:16:10

 REPRO MERGECAT -
 INFILE(INDD) -
 OUTFILE(OUTDD) -
 FROMKEY(MHLCAT1.DS1) -
 TOKEY(MHLCAT1.DS7)
IDC01460I THE NUMBER OF ENTRIES MERGED WAS 12
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

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

Figure 2-27 SDSF listing of part of the job output using MERGECAT

In Figure 2-28 we show that of all the data sets originally listed with High Level Qualifiers (HLQ) MHLCAT1 and MHLCAT8, only those before and after the specified range remain.

```
 MHLCAT1                                                        *ALIAS
        MHLCAT1.ANY.DATASET                                             SBOX20
        MHLCAT1.DS8.TEST1                                               SBOX20
        MHLCAT1.DS8.TEST2                                               SBOX20
        MHLCAT1.DS8.TEST3                                               SBOX20
        MHLCAT1.MCAT.TEST1                                              SBOX20
        MHLCAT1.MCAT.TEST2                                              SBOX20
        MHLCAT1.MCAT.TEST3                                              SBOX20
        MHLCAT1.MCAT.TEST4                                              SBOX20
MHLCAT8                                                        *ALIAS
        MHLCAT8.ANY.DATASET                                             SBOX20
```

Figure 2-28 TSO ISPF 3.4 listing of data sets under MHLCAT1 and MHLCAT8 after specific keys run

In Figure 2-29 we show that CATALOG.MERGECAT.TESTB has the selected entries from CATALOG.MERGECAT.TESTA, and it does not have those that were intentionally omitted.

```
 SDSF OUTPUT DISPLAY MHLRES1C JOB17375  DSID     4 LINE 10      COLUMNS 02- 81
 COMMAND INPUT ===>                                           SCROLL ===> CSR
IEF375I  JOB/MHLRES1C/START 2006090.1820
IEF376I  JOB/MHLRES1C/STOP  2006090.1820 CPU    OMIN 00.01SEC SRB    OMIN
00.00S
IDCAMS  SYSTEM SERVICES                                             TIME:
18:20:08

 LISTCAT LVL(MHLCAT1) CATALOG(CATALOG.MERGECAT.TESTB)
IDCAMS  SYSTEM SERVICES                                             TIME:
18:20:08
                             LISTING FROM CATALOG -- CATALOG.MERGECAT.TESTB
NONVSAM ------- MHLCAT1.DS1.TEST1
NONVSAM ------- MHLCAT1.DS1.TEST2
NONVSAM ------- MHLCAT1.DS1.TEST3
NONVSAM ------- MHLCAT1.DS3.TEST1
NONVSAM ------- MHLCAT1.DS3.TEST2
NONVSAM ------- MHLCAT1.DS3.TEST3
NONVSAM ------- MHLCAT1.DS5.TEST1
NONVSAM ------- MHLCAT1.DS5.TEST2
NONVSAM ------- MHLCAT1.DS5.TEST3
NONVSAM ------- MHLCAT1.DS7.TEST1
NONVSAM ------- MHLCAT1.DS7.TEST2
NONVSAM ------- MHLCAT1.DS7.TEST3

 LISTCAT LVL(MHLCAT8) CATALOG(CATALOG.MERGECAT.TESTB)
IDC3012I ENTRY MHLCAT8. NOT FOUND
IDC3007I ** VSAM CATALOG RETURN CODE IS 8
IDC1566I ** MHLCAT8 NOT LISTED
```

Figure 2-29   IDCAMS listing of contents of CATALOG.MERGECAT.TESTB

In Figure 2-30 we show the contents of USERCAT CATALOG.MERGECAT.TESTA.

```
 MHLCAT1                                                 *ALIAS
         MHLCAT1.A.TEST1                                         SBOX20
         MHLCAT1.A.TEST2                                         SBOX20
         MHLCAT1.A.TEST3                                         SBOX20
         MHLCAT1.ANY.DATASET                                     SBOX20
         MHLCAT1.B.TEST1                                         SBOX20
         MHLCAT1.B.TEST2                                         SBOX20
         MHLCAT1.B.TEST3                                         SBOX20
         MHLCAT1.C.TEST1                                         SBOX20
         MHLCAT1.C.TEST2                                         SBOX20
         MHLCAT1.C.TEST3                                         SBOX20
         MHLCAT1.DS1.TEST1                                       SBOX20
         MHLCAT1.DS1.TEST2                                       SBOX20
         MHLCAT1.DS1.TEST3                                       SBOX20
         MHLCAT1.DS3.TEST1                                       SBOX20
         MHLCAT1.DS3.TEST2                                       SBOX20
         MHLCAT1.DS3.TEST3                                       SBOX20
         MHLCAT1.DS5.TEST1                                       SBOX20
         MHLCAT1.DS5.TEST2                                       SBOX20
         MHLCAT1.DS5.TEST3                                       SBOX20
         MHLCAT1.DS7.TEST1                                       SBOX20
         MHLCAT1.DS7.TEST2                                       SBOX20
         MHLCAT1.DS7.TEST3                                       SBOX20
         MHLCAT1.DS8.TEST1                                       SBOX20
         MHLCAT1.DS8.TEST2                                       SBOX20
         MHLCAT1.DS8.TEST3                                       SBOX20
         MHLCAT1.MCAT.TEST1                                      SBOX20
         MHLCAT1.MCAT.TEST2                                      SBOX20
         MHLCAT1.MCAT.TEST3                                      SBOX20
         MHLCAT1.MCAT.TEST4                                      SBOX20
MHLCAT7                                                 *ALIAS
         MHLCAT7.A.TEST1                                         SBOX20
         MHLCAT7.A.TEST2                                         SBOX20
         MHLCAT7.A.TEST3                                         SBOX20
         MHLCAT7.E.TEST1                                         SBOX20
         MHLCAT7.E.TEST2                                         SBOX20
         MHLCAT7.E.TEST3                                         SBOX20
         MHLCAT7.G.TEST1                                         SBOX20
         MHLCAT7.G.TEST2                                         SBOX20
         MHLCAT7.G.TEST3                                         SBOX20
         MHLCAT7.H.TEST1                                         SBOX20
         MHLCAT7.H.TEST2                                         SBOX20
         MHLCAT7.H.TEST3                                         SBOX20
MHLCAT8                                                 *ALIAS
         MHLCAT8.ANY.DATASET                                     SBOX20
```

Figure 2-30 Listing of contents of CATALOG.MERGECAT.TESTA

In Figure 2-31 we show use of the BATCH JCL program IDCAMS with generic keys MHLCAT1.B.* and MHLCAT7.G.*. This is intended to copy entries for any data sets with names starting with prefix MHLCAT1.B and ending with names with prefix MHLCAT7.G. No entries from before MHLCAT1.B (such as MHLCAT1.A) or after MHLCAT7.G (such as MHLCAT7.H.* or MHLCAT8.*.*) will be copied.

```
//MHLRES1C JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
//MRGCAT   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INDD     DD  DISP=SHR,DSN=CATALOG.MERGECAT.TESTA
//OUTDD    DD  DISP=SHR,DSN=CATALOG.MERGECAT.TESTB
//SYSIN DD *
 REPRO MERGECAT -
 INFILE(INDD) -
 OUTFILE(OUTDD) -
 FROMKEY(MHLCAT1.B.*) -
 TOKEY(MHLCAT7.G.*)
/*
```

Figure 2-31 JCL for BATCH form of REPRO MERGECAT with generic keys

In Figure 2-32 we show part of the output from running REPRO MERGECAT using generic keys. It shows that 34 data set catalog entries were copied.

```
 SDSF OUTPUT DISPLAY MHLRES1C JOB17409  DSID     4 LINE 12      COLUMNS 02- 81
 COMMAND INPUT ===>                                             SCROLL ===> CSR
IEF376I  JOB/MHLRES1C/STOP  2006090.1920 CPU    0MIN 00.06SEC SRB    0MIN
00.00S
IDCAMS  SYSTEM SERVICES                                            TIME:
19:20:38

 REPRO MERGECAT -
 INFILE(INDD) -
 OUTFILE(OUTDD) -
 FROMKEY(MHLCAT1.B.*) -
 TOKEY(MHLCAT7.G.*)
IDC01460I THE NUMBER OF ENTRIES MERGED WAS 34
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

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

Figure 2-32 Partial SDSF display of output from REPRO MERGECAT job with generic keys

Figure 2-33 shows the output of a TSO ISPF 3.4 listing of all data sets under MHLCAT1, MHLCAT7, and MHLCAT8 after the generic keys job has been run. The data sets remain in CATALOG.MERGECAT.TESTA and are as expected.

```
 MHLCAT1                                                         *ALIAS
        MHLCAT1.A.TEST1                                                 SBOX20
        MHLCAT1.A.TEST2                                                 SBOX20
        MHLCAT1.A.TEST3                                                 SBOX20
        MHLCAT1.ANY.DATASET                                             SBOX20
MHLCAT7                                                         *ALIAS
        MHLCAT7.H.TEST1                                                 SBOX20
        MHLCAT7.H.TEST2                                                 SBOX20
        MHLCAT7.H.TEST3                                                 SBOX20
MHLCAT8                                                         *ALIAS
        MHLCAT8.ANY.DATASET                                             SBOX20
```

Figure 2-33 Listing of all data sets under MHLCAT1, MHLCAT7, and MHLCAT8 after generic keys job

In Figure 2-34 we show edited output captured by SDSF of an IDCAMS job to list the contents of USERCAT CATALOG.MERGECAT.TESTB. It shows, as expected, that selected data sets with HLQ MHLCAT1 and MHLCAT7 have been copied but none from MHLCAT8.

```
 SDSF OUTPUT DISPLAY MHLRES1C JOB17414  DSID     2 LINE 0      COLUMNS 02- 81
LISTCAT LVL(MHLCAT1) CATALOG(CATALOG.MERGECAT.TESTB)
IDCAMS  SYSTEM SERVICES                                           TIME:
19:34:25
                             LISTING FROM CATALOG -- CATALOG.MERGECAT.TESTB
NONVSAM ------- MHLCAT1.B.TEST1
NONVSAM ------- MHLCAT1.B.TEST2
NONVSAM ------- MHLCAT1.B.TEST3
NONVSAM ------- MHLCAT1.C.TEST1
NONVSAM ------- MHLCAT1.C.TEST2
NONVSAM ------- MHLCAT1.C.TEST3
NONVSAM ------- MHLCAT1.DS1.TEST1
NONVSAM ------- MHLCAT1.DS1.TEST2
NONVSAM ------- MHLCAT1.DS1.TEST3
NONVSAM ------- MHLCAT1.DS3.TEST1
NONVSAM ------- MHLCAT1.DS3.TEST2
NONVSAM ------- MHLCAT1.DS3.TEST3
NONVSAM ------- MHLCAT1.DS5.TEST1
NONVSAM ------- MHLCAT1.DS5.TEST2
NONVSAM ------- MHLCAT1.DS5.TEST3
NONVSAM ------- MHLCAT1.DS7.TEST1
NONVSAM ------- MHLCAT1.DS7.TEST2
NONVSAM ------- MHLCAT1.DS7.TEST3
NONVSAM ------- MHLCAT1.DS8.TEST1
NONVSAM ------- MHLCAT1.DS8.TEST2
NONVSAM ------- MHLCAT1.DS8.TEST3
NONVSAM ------- MHLCAT1.MCAT.TEST1
NONVSAM ------- MHLCAT1.MCAT.TEST2
NONVSAM ------- MHLCAT1.MCAT.TEST3
LISTCAT LVL(MHLCAT7) CATALOG(CATALOG.MERGECAT.TESTB)
IDCAMS  SYSTEM SERVICES                                           TIME:
19:34:25
                             LISTING FROM CATALOG -- CATALOG.MERGECAT.TESTB
NONVSAM ------- MHLCAT7.A.TEST1
NONVSAM ------- MHLCAT7.A.TEST2
NONVSAM ------- MHLCAT7.A.TEST3
NONVSAM ------- MHLCAT7.E.TEST1
NONVSAM ------- MHLCAT7.E.TEST2
NONVSAM ------- MHLCAT7.E.TEST3
NONVSAM ------- MHLCAT7.G.TEST1
NONVSAM ------- MHLCAT7.G.TEST2
NONVSAM ------- MHLCAT7.G.TEST3
.
```

Figure 2-34 IDCAMS listing after REPRO MERGECAT job using generic keys

2.6 Catalog enhancements

In this section we discuss the two catalog enhancements available with z/OS V1R7:

- VSAM volume data set (VVDS) size specification for implicit VVDS allocation - see “VVDS implicit size specification and verification” on page 45.
- Catalog access tuning - see “Catalog access tuning” on page 46.

2.6.1 VVDS implicit size specification and verification

You can still define the size of a VVDS explicitly. This support has not changed. This new support allows you to specify the size of the VVDS when it is allocated implicitly.

With the new support you can specify the amount of space to be used to allocate a VSAM volume data set (VVDS) when the allocation is done implicitly. The default value for an implicit VVDS allocation is TRK(10,10), which may not be adequate for the size of your volumes and number of data sets you are creating. Now, using the implicit VVDS space quantity enhancement, you can specify the amount of space in tracks that you want to use when a VVDS is defined implicitly.

You specify the implicit space values through the MODIFY CATALOG command. The format is:

```
MODIFY CATALOG VVDSSPACE(primary,secondary)
```

In Figure 2-35 we show use of the MODIFY CATALOG command and the response to set new primary and secondary values for the VVDS on a volume that does not already have a VVDS allocated.

```
F CATALOG,VVDSSPACE(15,10)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Figure 2-35 MODIFY CATALOG VVDSSPACE

> **Note:** The VVDS implicit size specification is retained over a catalog restart, but not over an IPL. You will need to issue the MODIFY CATALOG VVDSSPACE command automatically if you want this to be a permanent setting.

2.6.2 VVDS implicit size verification

In Figure 2-36 we show the status of the implicit VVDS size specification in effect where it appears as DEFAULT VVDS SPACE.

```
F CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 012
*CAS*****************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ11K0                              *
*  CATALOG ADDRESS SPACE ASN = 0035                                 *
*  SERVICE TASK UPPER LIMIT  =  180                                 *
*  SERVICE TASK LOWER LIMIT  =   60                                 *
*  HIGHEST # SERVICE TASKS   =   21                                 *
*  CURRENT # SERVICE TASKS   =   21                                 *
*  MAXIMUM # OPEN CATALOGS   = 1,024                                *
*  ALIAS TABLE AVAILABLE     = YES                                  *
*  ALIAS LEVELS SPECIFIED    = 1                                    *
*  SYS% TO SYS1 CONVERSION   = OFF                                  *
*  CAS MOTHER TASK           = 007FF540                             *
*  CAS MODIFY TASK           = 007A2DA8                             *
*  CAS ANALYSIS TASK         = 007A2948                             *
*  CAS ALLOCATION TASK       = 007A2B78                             *
*  VOLCAT HI-LEVEL QUALIFIER = SYS1                                 *
*  NOTIFY EXTENT             =   80%                                *
*  DEFAULT VVDS SPACE        = ( 15, 10) TRKS                       *
*  ENABLED FEATURES          = DSNCHECK DELFORCEWNG SYMREC          *
*  ENABLED FEATURES          = UPDTFAIL AUTOTUNING                  *
*  DISABLED FEATURES         = VVRCHECK                             *
*  INTERCEPTS                = (NONE)                               *
*CAS*****************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Figure 2-36 MODIFY CATALOG REPORT

For more information about the VSAM volume data set (VVDS), explicit and implicit definitions, see “Defining a VVDS” in the manual *DFSMS Managing Catalogs,* SC26-7409.

2.6.3 Catalog access tuning

z/OS v1R7 implements an auto-tuning function to adjust some of the default parameter values to improve performance. It does this by temporarily adjusting the number of data and index buffers and VSAM strings for the catalog on the current system. You are advised through message IEC391I what the temporary values are. You can then use IDCAMS ALTER against the catalog to make the values permanent.

The auto-tuning function is automatic, but you can disable it or subsequently re-enable it by specifying AUTOTUNING on the DISABLE and ENABLE operands of the MODIFY CATALOG command.

We discuss the following topics:

- Auto-tuning message IEC391I
- IDCAMS ALTER
- AUTOTUNING DISABLE and ENABLE

Catalog auto-tuning message IEC391I

If the catalog function determines that adjustments to the catalog parameters would improve performance, it makes the adjustments and issues message IEC391I, as shown in Figure 2-37, for the catalog CATALOG.HSM.

```
IEC391I CATALOG CATALOG.HSM HAS BEEN AUTOTUNED TO:
IEC391I BUFNI:     4 BUFND:     4 STRNO:     2
```

Figure 2-37 Message IEC391I

Where:

BUFND Shows the number of *new* index buffers assigned to the catalog
BUFNI Shows the number of *new* data buffers assigned to this catalog
STRNO Shows the number of strings assigned to this catalog

In Figure 2-38 on page 48 we show part of the output from the LISTCAT command before any changes are made to the catalog parameters. Of interest are the values for:

BUFND 0
BUFNI 0
STRNO 2

This indicates that the recommendation from auto-tune is to increase both BUFND and BUFNI from 0 to 4, but that the STRNO can stay at 2. The value of 0 is a default, and the actual values in use are related to the STRNO value. For BUFND the default is STRNO+1 and for BUFNI the default is STRNO+2. The auto-tune values are therefore compatible with the calculated defaults for this catalog.

```
LISTC ENT(CATALOG.HSM) CAT(CATALOG.HSM) ALL
IDCAMS  SYSTEM SERVICES                                                TIME: 17:43:38
                             LISTING FROM CATALOG -- CATALOG.HSM
CLUSTER ------- 00000000000000000000000000000000000000000000
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2000.199
       RELEASE----------------2     EXPIRATION------0000.000
       BWO STATUS--------(NULL)     BWO TIMESTAMP-----(NULL)
       BWO---------------(NULL)
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
     ASSOCIATIONS
       DATA-----CATALOG.HSM
       INDEX----CATALOG.HSM.CATINDEX
   DATA ------- CATALOG.HSM
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2000.199
       RELEASE----------------2     EXPIRATION------0000.000
       ACCOUNT-INFO-----------------------------------(NULL)
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
     ASSOCIATIONS
       CLUSTER--00000000000000000000000000000000000000000000
     ATTRIBUTES
       KEYLEN----------------45     AVGLRECL------------4086     BUFSPACE-------
       RKP--------------------9     MAXLRECL-----------32400     EXCPEXIT-------
       BUFND------------------0     STRNO------------------2
       SHROPTNS(3,4)      SPEED     UNIQUE           NOERASE     INDEXED       N
       UNORDERED        NOREUSE     SPANNED        NOECSHARE     ICFCATALOG
     STATISTICS
       REC-TOTAL--------------0     SPLITS-CI--------------0     EXCPS----------
       REC-DELETED----------196     SPLITS-CA--------------0     EXTENTS--------
       REC-INSERTED-----------0     FREESPACE-%CI---------10     SYSTEM-TIMESTAM
       REC-UPDATED------------0     FREESPACE-%CA---------10          X'00000000
       REC-RETRIEVED----------0     FREESPC-----------757760
     ALLOCATION
       SPACE-TYPE------CYLINDER     HI-A-RBA----------757760
        SPACE-PRI--------------1     HI-U-RBA----------757760
       SPACE-SEC--------------1
     VOLUME
       VOLSER------------SBXHS1     PHYREC-SIZE--------10240     HI-A-RBA-------
       DEVTYPE------X'3010200F'     PHYRECS/TRK------------5     HI-U-RBA-------
       VOLFLAG------------PRIME     TRACKS/CA-------------15
       EXTENTS:
       LOW-CCHH-----X'000A0000'     LOW-RBA----------------0     TRACKS---------
       HIGH-CCHH----X'000A000E'     HIGH-RBA----------757759
   INDEX ------ CATALOG.HSM.CATINDEX
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2000.199
       RELEASE----------------2     EXPIRATION------0000.000
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
IDCAMS  SYSTEM SERVICES                                                TIME: 17:43:38
                             LISTING FROM CATALOG -- CATALOG.HSM
     ASSOCIATIONS
       CLUSTER--00000000000000000000000000000000000000000000
     ATTRIBUTES
       KEYLEN----------------45     AVGLRECL---------------0     BUFSPACE-------
       RKP--------------------9     MAXLRECL------------1017     EXCPEXIT-------
       BUFNI------------------0
       SHROPTNS(3,4)   RECOVERY     UNIQUE           NOERASE     NOWRITECHK
       NOREUSE       ICFCATALOG
     STATISTICS  (* - VALUE MAY BE INCORRECT)
       REC-TOTAL--------------0*    SPLITS-CI--------------0*     EXCPS----------
       REC-DELETED------------0*    SPLITS-CA--------------0*    EXTENTS--------
       REC-INSERTED-----------0*    FREESPACE-%CI----------0     SYSTEM-TIMESTAM
       REC-UPDATED------------0*    FREESPACE-%CA----------0          X'00000000
       REC-RETRIEVED----------0*    FREESPC------------33792*
     ALLOCATION
       SPACE-TYPE---------TRACK     HI-A-RBA-----------33792
       SPACE-PRI--------------1     HI-U-RBA------------1024
       SPACE-SEC--------------1
     VOLUME
       VOLSER------------SBXHS1     PHYREC-SIZE---------1024     HI-A-RBA-------
       DEVTYPE------X'3010200F'     PHYRECS/TRK-----------33     HI-U-RBA-------
       VOLFLAG------------PRIME     TRACKS/CA--------------1
       EXTENTS:
       LOW-CCHH-----X'0000000C'     LOW-RBA----------------0     TRACKS---------
       HIGH-CCHH----X'0000000C'     HIGH-RBA-----------33791
```

Figure 2-38   LISTCAT output for catalog CATALOG.HSM

IDCAMS ALTER

You can make the recommended values permanent by use of the IDCAMS ALTER command, but this is not necessary, as the default values will be dynamically updated as appropriate.

CATALOG AUTOTUNING DISABLE and ENABLE

You can, should it be necessary, disable the auto-tuning function. It is unlikely that you would need to do this unless requested to do so for support reasons. You can change the status of the AUTOTUNING function by using the MODIFY CATALOG,DISABLE function. You can determine the status of the AUTOTUNING function by using the MODIFY CATALOG,REPORT command.

AUTOTUNING DISABLE

In Figure 2-39 we show the command and the response to the command to disable AUTOTUNING.

```
F CATALOG,DISABLE(AUTOTUNING)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Figure 2-39 MODIFY CATALOG to disable AUTOTUNING

You can enable AUTOTUNING in a similar manner by using the MODIFY CATALOG,ENABLE(AUTOTUNING) command.

AUTOTUNING status

You can determine the status of AUTOTUNING by using the MODIFY CATALOG,REPORT command.

In Figure 2-40 we show the output from the MODIFY CATALOG,REPORT command. Note that AUTOTUNING is shown as one of the disabled features.

```
F CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 625
*CAS*************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ11K0                          *
*  CATALOG ADDRESS SPACE ASN = 0035                             *
*  SERVICE TASK UPPER LIMIT  =  180                             *
*  SERVICE TASK LOWER LIMIT  =   60                             *
*  HIGHEST # SERVICE TASKS   =   21                             *
*  CURRENT # SERVICE TASKS   =   21                             *
*  MAXIMUM # OPEN CATALOGS   = 1,024                            *
*  ALIAS TABLE AVAILABLE     = YES                              *
*  ALIAS LEVELS SPECIFIED    = 1                                *
*  SYS% TO SYS1 CONVERSION   = OFF                              *
*  CAS MOTHER TASK           = 007FF540                         *
*  CAS MODIFY TASK           = 007A2DA8                         *
*  CAS ANALYSIS TASK         = 007A2948                         *
*  CAS ALLOCATION TASK       = 007A2B78                         *
*  VOLCAT HI-LEVEL QUALIFIER = SYS1                             *
*  NOTIFY EXTENT             =   80%                            *
*  DEFAULT VVDS SPACE        = ( 15, 10) TRKS                   *
*  ENABLED FEATURES          = DSNCHECK DELFORCEWNG SYMREC      *
*  ENABLED FEATURES          = UPDTFAIL                         *
*  DISABLED FEATURES         = VVRCHECK AUTOTUNING              *
*  INTERCEPTS                = (NONE)                           *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Figure 2-40 CATALOG REPORT showing AUTOTUNING status

2.7 VSAM data set extent constraint removal

In previous releases, there was a 255-extent limit for VSAM data sets and, for striped VSAM data sets, a limit of 255 extents per stripe. Now, using the VSAM extent constraint removal enhancement, these extent limits are removed for SMS-managed volumes if the extent constraint removal parameter in the data class is set to Y (yes). For non SMS-managed volumes, the previous extent limits still apply. A VSAM data set can be expanded to 123 extents per volume. This is unchanged from previous releases.

2.7.1 Planning to use VSAM data set extent constraint removal

You must take the following into consideration:

- The supported environment and how it has to be set up - see “Supported environments for VSAM data set extent constraint removal” on page 51.
- Compatibility with lower-level systems of new operands in the IGDSMSxx member of SYS1.PARMLIB - see “Coexistence of VSAM data set extent removal with lower-level systems” on page 58.

- Maintenance required on lower-level systems - see “Maintenance on lower-level systems for VSAM data set extent removal” on page 58.

2.7.2 Supported environments for VSAM data set extent constraint removal

You can only use VSAM data sets with more than 255 extents in the following circumstances:

- z/OS V1R7 systems
- Extent removal indicator set to Y in the appropriate data class

You cannot use data sets with more than 255 extents in the following circumstances:

- Pre-z/OS V1R7 systems
- Non-SMS managed VSAM data sets

2.7.3 Setting up VSAM data set extent constraint removal

You must decide which data sets should be allowed to extend to over 255 extents.

You control over which data sets may extend over 255 extents through the VSAM extent control removal flag in the data class.

> **Note:** Any data set that is shared with a pre-z/OS V1R7 system must not be allowed to extend to over 255 extents, so these data sets must not be associated with a data class in which you set the flag.

You must set the extent constraint removal parameter in each data class where you want this function to be applicable, using the ISMF panels. The parameter appears on the panels as EXT CON REMOVAL.

- Use the Data Class Define/Alter panel to set the parameter in a data class.
- From the ISMF main panel, select **Data Class** and you will see the DATA CLASS APPLICATION SELECTION PANEL. On this panel, as shown in Figure 2-41, we show that we specified, for this example, Data Class WELCHRLS and option 4, and we then pressed Enter to bring up the next panel.

```
 DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Data Class Name  . . WELCHRLS  (For Data Class List, fully or partially
                                 specified or * for all)

Select one of the following options  :
  4  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


Use ENTER to Perform Selection;
```

Figure 2-41 ISMF initial panel to set EXT CON REMOVAL

In Figure 2-42 we show the first of four panels. On this panel we select DOWN and repeat on subsequent panels until we reach page 4 of 4.

```
 DATA CLASS ALTER                 Page 1 of 4
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

To ALTER Data Class, Specify:
  Description ==>
              ==>
  Recfm . . . . . . . . . .            (any valid RECFM combination or blank)
  Lrecl . . . . . . . . . .            (1 to 32761 or blank)
  Space Avgrec  . . . . . .            (U, K, M or blank)
        Avg Value . . . . .            (0 to 65535 or blank)
        Primary . . . . . .            (0 to 999999 or blank)
        Secondary . . . . .            (0 to 999999 or blank)
        Directory . . . . .            (0 to 999999 or blank)
  Retpd or Expdt  . . . . .            (0 to 9999, YYYY/MM/DD or blank)
  Volume Count  . . . . . . 1          (1 to 59 or blank)
  Add'l Volume Amount . . .            (P=Primary, S=Secondary or blank)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 2-42 ISMF DATA CLASS ALTER page 1 of 4 (for EX CON REMOVAL)

In Figure 2-43 we show page 4 of 4 where extent constraint removal is specified as Y. Press Enter to confirm the setting.

```
 DATA CLASS ALTER                 Page 4 of 4
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

To ALTER Data Class, Specify:

  Shareoptions Xregion  . . .     (1 to 4 or blank)
               Xsystem  . . .     (3, 4 or blank)
  Reuse . . . . . . . . . . . N   (Y or N)
  Initial Load  . . . . . . . R   (S=Speed, R=Recovery or blank)
  BWO . . . . . . . . . . . .     (TC=TYPECICS, TI=TYPEIMS, NO or blank)
  Log . . . . . . . . . . . .     (N=NONE, U=UNDO, A=ALL or blank)
  Logstream Id  . . . . . . .
  FRlog . . . . . . . . . . .     (A=ALL, N=NONE, R=REDO, U=UNDO or blank)
  RLS CF Cache Value  . . . . A   (A=ALL, N=NONE, U=UPDATESONLY)
  RLS Above the 2-GB Bar  . . N   (Y or N)
  Extent Constraint Removal   Y   (Y or N)

Use ENTER to perform Verification; Use UP Command to View previous Panel;
```

Figure 2-43 ISMF DATA CLASS ALTER page 2 of 4 (for EX CON REMOVAL)

Once the data class has been set up as required, you can see whether the parameter is set for a given data class on the following panels.

Data Class display

From the data class display:

- From the ISMF main panel, select **Data Class** and you will see the DATA CLASS APPLICATION SELECTION panel. On this panel, as shown in Figure 2-41 on page 52, we show that we specified, for this example, data class WELCHRLS and option 4, and we then pressed Enter to bring up the next panel.
- In Figure 2-44 we have selected option 2 - Display. Press Enter to display the first DATA CLASS DISPLAY panel.

```
 DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Data Class Name  . . WELCHRLS  (For Data Class List, fully or partially
                                 specified or * for all)

Select one of the following options  :
  2  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


Use ENTER to Perform Selection;
```

Figure 2-44 ISMF DATA CLASS DISPLAY application selection

In Figure 2-45 we show the first of four pages in the display. We pressed DOWN, looking at the values until we found extent constraint removal on page 4.

```
 DATA CLASS DISPLAY                  Page 1 of 4
Command ===>

CDS Name  . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

Description :

Recfm  . . . . . . . . . :
Lrecl  . . . . . . . . . :
Space Avgrec . . . . . . :
      Avg Value  . . . . :
      Primary  . . . . . :
      Secondary  . . . . :
      Directory  . . . . :
Retpd Or Expdt . . . . . :
Volume Count . . . . . . : 1
Add'l Volume Amount  . . :


Use DOWN Command to View next Panel;
```

Figure 2-45 ISMF DATA CLASS DISPLAY page 1 of 4

Figure 2-46 shows page 4 of 4 where extent constraint removal is shown as YES.

```
 DATA CLASS DISPLAY                  Page 4 of 4
Command ===>

CDS Name  . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

Reuse  . . . . . . . . . . . : NO
Initial Load . . . . . . . . : RECOVERY
BWO  . . . . . . . . . . . . :
Log  . . . . . . . . . . . . :
Logstream Id . . . . . . . . :
FRlog  . . . . . . . . . . . :
RLS CF Cache Value . . . . . : ALL
RLS Above the 2-GB Bar . . . : YES
Extent Constraint Removal  . : YES





Use UP Command to View previous Panel;
```

Figure 2-46 ISMF DATA CLASS DISPLAY page 4 of 4

If you want to find a data class that has extent constraint removal, you can look through all of the data class definitions using the data class list function.

Data Class List

For the data class list, from the ISMF main panel, select **Data Class** and you will see the DATA CLASS APPLICATION SELECTION panel. On this panel, as shown in Figure 2-47, we show that we specified, for this example, Data Class * and option 1 - list, and we then pressed Enter to bring up the next panel.

```
 DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Data Class Name  . . *         (For Data Class List, fully or partially
                                 specified or * for all)

Select one of the following options  :
  1  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


Use ENTER to Perform Selection;
```

Figure 2-47 ISMF initial panel to list data classes

In Figure 2-48 we show the first of many possible panels. You could page down or to the right. We paged right to find the panel containing column 45 EXT CON REMOVAL. We then paged down looking for the setting YES.

```
 DATA CLASS LIST
Command ===>                                                  Scroll ===> CSR
                                                        Entries 1-10 of 34
                                                        Data Columns 3-9 of 46
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       DATACLAS                                                  AVG
     OPERATOR   NAME     RECORG  RECFM  LRECL  KEYLEN  KEYOFF  AVGREC  VALUE
    ---(1)----  --(2)--- -(3)--  -(4)-  -(5)-  -(6)--  -(7)--  -(8)--  -(9)-
                BIGHFS   --      ----   -----     ---   -----  -       -----
                DB2STRIP LS      ----   -----     ---   -----  -       -----
                DB8ALDS  --      ----   -----     ---   -----  -       -----
                DB8XD    --      ----   -----     ---   -----  -       -----
                DCEF     --      ----   -----     ---   -----  -       -----
                DCGDG    --      ----      30     ---   -----  -       -----
                DCPDSE   --      ----   -----     ---   -----  -       -----
                DC3592   --      ----   -----     ---   -----  -       -----
                DC54236  --      ----   -----     ---   -----  -       -----
                DEFAULT  --      ----   -----     ---   -----  -       -----
```

Figure 2-48 ISMF DATA CLASS LIST panel 1 of 2

In Figure 2-49 we show two data classes that have YES in column 45 - WELCHECR and WELCHRLS.

```
 DATA CLASS LIST
Command ===>                                                  Scroll ===> CSR
                                                      Entries 31-34 of 34
                                                      Data Columns 45-46 of 46
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       DATACLAS EXT CON  RLS ABOVE
     OPERATOR   NAME     REMOVAL  THE BAR
    ---(1)----  --(2)--- -(45)--  --(46)---
                TEST01   NO       NO
                WELCHECR YES      NO
                WELCHLRG NO       NO
                WELCHRLS YES      YES
```

Figure 2-49 ISMF DATA CLASS LIST panel 2 of 2

Similar displays can be generated using the other data class list options:

- Data class list print
- Data class list sort
- Data class list view

2.7.4 Coexistence of VSAM data set extent removal with lower-level systems

There are restrictions on the use of more than 255 data set extents on lower-level systems:

- On a z/OS system lower than V1R7 data sets with greater than 255 extents cannot be used.
- Results of attempting to use such a data set on lower-level systems will be unpredictable. It is essential that maintenance be installed on the lower-level systems to detect and prevent access to the data set.

2.7.5 Maintenance on lower-level systems for VSAM data set extent removal

In Table 2-2 we show the minimum PTFs that must be applied to lower-level systems.

Table 2-2 DFSMSdfp VSAM data set extent constraint removal PTFs

Feature and reason PTFs are needed	z/OS V1R4 PTFs	z/OS V1R5 PTFs	z/OS V1R6 PTFs
As part of VSAM extent constraint removal in z/OS V1R7, these PTFs prevent pre-z/OS V1R7 systems from opening data sets that have more than 255 extents. Also, if a data set is open and a z/OS V1R7 job extends it past 255 extents, the close on the pre-z/OS V1R7 system fails. These PTFs also prevent conversion of a data set having more than 255 extents from being converted by DFSMSdss to a non-SMS managed data set on pre-z/OS V1R7 releases.	UA14882 UA16223 UA16796	UA14883 UA16224 UA16797	UA14884 UA16225 UA16798

2.8 VSAM RLS 64-bit data buffers

VSAM record-level sharing (VSAM RLS) is an extension to VSAM that provides direct record-level sharing of VSAM data sets (as opposed to CI-level sharing) from multiple address spaces across multiple systems. VSAM RLS uses the z/OS coupling facility for cross-system locking, local buffer invalidation, and cross-system data caching. Primary users of VSAM RLS include high-volume applications that access VSAM data sets, such as CICS® applications. Now, with z/OS V1R7, you can optionally specify that VSAM RLS uses 64-bit addressable virtual storage for data buffers. Doing so can help you avoid possible buffer space constraints and potentially improve performance for your high-transaction applications.

2.8.1 Planning to use VSAM RLS 64-bit data buffers

You must take the following into consideration:

- The supported environment and how it has to be set up - see “Supported environments for VSAM RLS 64-bit data buffers” on page 59.
- Compatibility with lower-level systems of new operands in the IGDSMSxx member of SYS1.PARMLIB - see “Coexistence with lower-level systems for VSAM RLS 64-bit data buffers” on page 73.
- Maintenance required on lower-level systems - see “Maintenance on lower-level systems for VSAM RLS 64-bit data buffers” on page 73.

2.8.2 Supported environments for VSAM RLS 64-bit data buffers

You can only use VSAM 64-bit data buffers on a z/OS V1R7 system.

The IGDSMSxx member of SYS1.PARMLIB must specify parameter RlsAboveTheBarMaxPoolSize with a value of between 500 MB and 2 terabytes.

The data class for the data set must specify RlsAboveThe Bar=Y.

2.8.3 Setting up VSAM RLS 64-bit data buffers

To implement VSAM RLS 64-bit data buffers perform the following steps:

1. Use ISMF to create or modify a data class to set the RLS ABOVE THE BAR attribute. See “Modify a data class to set the RLS ABOVE THE BAR attribute to Y” on page 59.
2. Allocate or associate a data set with the appropriate data class - see “Associate required VSAM data sets to the data class” on page 61.
3. Update member IGDSMSxx of SYS1.PARMLIB with RLS parameters - see Figure 2-58 on page 64.

Modify a data class to set the RLS ABOVE THE BAR attribute to Y

From the ISMF main panel, select **DATA CLASS** and you will see the DATA CLASS APPLICATION SELECTION panel. On this panel, as shown in Figure 2-50, we show that we specified, for this example, Data Class WELCHRLS and Option 4, and we then pressed Enter to bring up the next panel.

```
 DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                                (1 to 44 character data set name or 'Active' )
  Data Class Name  . . WELCHRLS  (For Data Class List, fully or partially
                                  specified or * for all)

Select one of the following options  :
  4 1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

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 2-50 ISMF initial panel to set RLS ABOVE THE BAR

In Figure 2-51 we show the first of four panels. On this panel we made no selections, but selected DOWN repeatedly until we reached page 4 of 4.

```
 DATA CLASS ALTER                 Page 1 of 4
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

To ALTER Data Class, Specify:
  Description ==>
              ==>
  Recfm . . . . . . . . . .            (any valid RECFM combination or blank)
  Lrecl . . . . . . . . . .            (1 to 32761 or blank)
  Space Avgrec  . . . . . .            (U, K, M or blank)
        Avg Value . . . . .            (0 to 65535 or blank)
        Primary . . . . . .            (0 to 999999 or blank)
        Secondary . . . . .            (0 to 999999 or blank)
        Directory . . . . .            (0 to 999999 or blank)
  Retpd or Expdt  . . . . .            (0 to 9999, YYYY/MM/DD or blank)
  Volume Count  . . . . . . 1          (1 to 59 or blank)
  Add'l Volume Amount . . .            (P=Primary, S=Secondary or blank)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 2-51 ISMF DATA CLASS ALTER Page 1 of 4

In Figure 2-52 we show page 4 of 4, where RLS Above the 2-GB Bar is specified as Y. We pressed Enter to continue.

```
 DATA CLASS ALTER                 Page 4 of 4
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : WELCHRLS

To ALTER Data Class, Specify:

  Shareoptions Xregion  . . .     (1 to 4 or blank)
               Xsystem  . . .     (3, 4 or blank)
  Reuse . . . . . . . . . . . N   (Y or N)
  Initial Load  . . . . . . . R   (S=Speed, R=Recovery or blank)
  BWO . . . . . . . . . . . .     (TC=TYPECICS, TI=TYPEIMS, NO or blank)
  Log . . . . . . . . . . . .     (N=NONE, U=UNDO, A=ALL or blank)
  Logstream Id  . . . . . . .
  FRlog . . . . . . . . . . .     (A=ALL, N=NONE, R=REDO, U=UNDO or blank)
  RLS CF Cache Value  . . . . A   (A=ALL, N=NONE, U=UPDATESONLY)
  RLS Above the 2-GB Bar  . . Y   (Y or N)
  Extent Constraint Removal   N   (Y or N)

Use ENTER to perform Verification; Use UP Command to View previous Panel;
```

Figure 2-52 ISMF DATA CLASS ALTER page 4 of 4

Associate required VSAM data sets to the data class

In Figure 2-53 we show selection of the VSAM utilities and specification of the data set name to be allocated as MHLRES1.testrls.vsam. Press Enter to continue.

```
 Data Set Utility
Option ===> v

    A Allocate new data set                C Catalog data set
    R Rename entire data set               U Uncatalog data set
    D Delete entire data set               S Short data set information
blank Data set information                 V VSAM Utilities

ISPF Library:
   Project  . . MHLRES1          Enter "/" to select option
   Group  . . . testrls          /  Confirm Data Set Delete
   Type . . . . vsam

Other Partitioned, Sequential or VSAM Data Set:
   Data Set Name . . .
   Volume Serial . . .           (If not cataloged, required for option "C")

Data Set Password  . .           (If password protected)
```

Figure 2-53 ISPF option 3.2 VSAM data set allocate to using data class WELCHRLS step1 of 5

Figure 2-54 shows the selection of:

- Process Request 1. Define
- Data Type 3. Cluster.

Press Enter to continue.

```
 VSAM Utilities
Command ===>
                                                        More:     +
 Process Request                 Data Type
 1  1. Define                    3   1.  Alias
    2. Delete                        2.  Alternate Index
    3. Information (Listcat)         3.  Cluster
                                     4.  Generation Data Group
                                     5.  Non-VSAM
                                     6.  Page Space
                                     7.  Path
                                     8.  User Catalog
                                     9.  Data     *
                                     10. Index    *
                                     11. NVR      **
                                     12. Truename **
                                     13. VVR      **
                      * Listcat Only
                     ** Delete Only
```

Figure 2-54 ISPF option 3.2 VSAM data set allocate to using data class WELCHRLS step 2 of 5

In Figure 2-55 the cluster name is shown as MHRES1.TESTRLS.VSAM, as expected. The following options are specified:

- Space Units 1. Cylinders
- Primary Quantity 2
- Secondary Quantity 0
- Data Class WELCHRLS

  WELCHRLS is the data class defined to have 64-bit data buffers.

```
 Define Cluster
Command ===>

                                               Enter "/" to select option
                                               /  Edit IDCAMS command
                                               /  Browse errors only
 Cluster Name  . . . MHLRES1.TESTRLS.VSAM
                                                                   More:     +
                          Cluster Level Information:

 Space Units  . . . . . . . 1  1. Cylinders   Primary Quantity . . . 2
                               2. Tracks      Secondary Quantity . . 0
                               3. Records
                               4. Kilobytes
                               5. Megabytes
 Volumes  . . . . . . . . .         . . .          . . .         . . .
 Buffer Space . . . . . . .
 Control Interval Size  . .
 Data Class . . . . . . . . WELCHRLS
 Management Class . . . . .
 Storage Class  . . . . . .
 Free Space CI-Percent  . .                 CA-Percent
 Record Size   Average  . .                 Maximum  . .
 Share Options  Region  . .    (1,2,3,4)  System . . .    (3,4)
 Keys          Length  . .                 Offset . . .
 Days              For . .             or  To . . . . .              (YYYY DDD)


 Enter any of the following keywords:

 Noerase or Erase . . . . .
 Norecatalog or Recatalog
 Noreuse or Reuse . . . . .
 Nonspanned or Spanned  . .
 Recovery or Speed  . . . .
 Unique, Suballocation  . .
 Indexed,NOnindexed,Linear or NUmbered  . . .

 Owner  . . . . . . . . . .
 Model  . . . . . . . . . .
```

Figure 2-55 IISPF option 3.2 VSAM data set allocate using data class WELCHRLS step 3 of 5

Note: At the top of this panel in the right top corner, we used a slash (/) to specify that we want to edit the IDCAMS command. This is so that we see the command for illustration.

Press Enter to continue.

In Figure 2-56 we see the predefined IDCAMS commands ready to execute. Specify **exec** on the command line to have the command executed, and press Enter to continue.

```
 Columns 00001 00072
Command ===> exec                                              Scroll ===> CSR

    Instructions:

      Enter EXECute command to issue request.

      Enter CANcel, END, or RETURN command to cancel request.
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001  /* IDCAMS COMMAND */
000002  DEFINE CLUSTER (NAME(MHLRES1.TESTRLS.VSAM) -
000003         CYLINDERS(2 0) -
000004         DATACLASS(WELCHRLS) -
000005         ) -
000006         DATA (NAME(MHLRES1.TESTRLS.VSAM.DATA) -
000007         ) -
000008         INDEX (NAME(MHLRES1.TESTRLS.VSAM.INDEX) -
000009         )
```

Figure 2-56 ISPF option 3.2 VSAM data set allocate to using data class WELCHRLS step 4 of 5

The panel in Figure 2-57 is similar to the previous panel. The annotation `Return Code 0` appears in the upper right-hand corner.

```
                                                             Return Code 0
Command ===>                                                  Scroll ===> CSR

    Instructions:

      Enter EXECute command to issue request.

      Enter CANcel, END, or RETURN command to cancel request.
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001  /* IDCAMS COMMAND */
000002  DEFINE CLUSTER (NAME(MHLRES1.TESTRLS.VSAM) -
000003         CYLINDERS(2 0) -
000004         DATACLASS(WELCHRLS) -
000005         ) -
000006         DATA (NAME(MHLRES1.TESTRLS.VSAM.DATA) -
000007         ) -
000008         INDEX (NAME(MHLRES1.TESTRLS.VSAM.INDEX) -
000009         )
```

Figure 2-57 ISPF option 3.2 VSAM data set allocate to using data class WELCHRLS step1 of 5

Update member IGDSMSxx of SYS1.PARMLIB (only on V1R7 systems)

Specify the parameter RlsAboveTheBarMaxPoolSize to a value between 500 MB and 2 terabytes. If you specify less than 500 MB, VSAM RLS 64-bit data buffers will not be assigned. Each system in a sysplex can have the same or a different value.

Optionally, specify the parameter RlsFixedPoolSize to set an amount of real storage to be permanently fixed for RLS data buffers. The value set may be adjusted internally not to exceed 80% of available real storage to avoid compromising other system functions.

Figure 2-58 shows an example of parameters to set RlsAboveTheBarMaxPoolSize for system SC70 to 500. The minimum value allowed for this parameter is 500. The parameter to set the RlsFixedPoolSize for system SC70 is set to 50. This value may be adjusted internally if it would cause more than 80% of available real storage to be fixed.

```
RLSABOVETHEBARMAXPOOLSIZE(SC70,500)
 RLSFIXEDPOOLSIZE(SC70,50)
```

Figure 2-58 IGDSMSxx parameters to set RLS 64-bit values

2.8.4 Monitoring the 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.

Note: The SMF data gathering for RLS was not included in z/OS V1R7 as originally scheduled. You can, therefore, only verify that a particular data set is enabled to use 64-bit data buffers by checking that it is associated with the correct data class. It is planned to be included in z/OS V1R8.

The original intent was that you could 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 *DFSMS Using the New Functions,* SC26-7473.

- 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.
- Subtype 18 (CF Cache Partition Usage) provides statistics on the use of coupling facility cache Storage Classes.
- Subtype 19 (LRU record) provides statistics for data buffers above and below 2-gigabytes.

2.8.5 Check data classes enabled for 64-bit buffers

You can use the ISMF Data Class List panel to see which data classes have the attribute RLS ABOVE THE BAR.

From the ISMF main menu, select option 4 - Data Class.

On the DATA CLASS APPLICATION SELECTION panel, specify the name of the CDS in use and an asterisk (*) for the data class name (see Figure 2-59). Press Enter to continue.

```
 DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Data Class Name  . . *         (For Data Class List, fully or partially
                                 specified or * for all)

Select one of the following options  :
  1  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

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 2-59 ISMF DATA CLASS APPLICATION SELECTION

The next panel is the DATA CLASS LIST panel. Scroll to the extreme right until you see the heading `RLS ABOVE THE BAR`. Scroll down until you find a YES in the RLS ABOVE THE BAR column.

Figure 2-60 shows the data class named WELCHRLS with the RLS ABOVE THE BAR attribute.

```
 DATA CLASS LIST
Command ===>                                                        Scroll ===> CSR
                                                            Entries 27-34 of 34
                                                            Data Columns 45-46 of 46

CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE        DATACLAS EXT CON   RLS ABOVE
     OPERATOR    NAME     REMOVAL   THE BAR
    ---(1)----   --(2)--- -(45)--   --(46)---
                 RMMCDS   NO        NO
                 SERVSMF  NO        NO
                 SHARE33  NO        NO
                 STRIPE   NO        NO
                 TEST01   NO        NO
                 WELCHECR YES       NO
                 WELCHLRG NO        NO
                WELCHRLS NO        YES
```

Figure 2-60 DATA CLASS LIST panel showing RLS ABOVE THE BAR entry

2.8.6 Check data sets enabled for 64-bit buffers

You can use the ISMF Data Set panel for each data class to find the data sets in that data class.

Figure 2-61 through to Figure 2-64 on page 69 show a selection of data sets that are using the data class WELCHRLS, as shown in Figure 2-60 on page 66.

```
 DATA SET SELECTION ENTRY PANEL    OTHER VALUES PRESENT
Command ===>

For a Data Set List, Select Source of Generated List  . . 2  (1 or 2)

  1  Generate from a Saved List         Query Name To
       List Name  . .                   Save or Retrieve

  2  Generate a new list from criteria below
       Data Set Name . . . **
       Enter "/" to select option   /  Generate Exclusive list
       Specify Source of the new list  . . 2    (1 - VTOC, 2 - Catalog)
       1 Generate list from VTOC
          Volume Serial Number . . .            (fully or partially specified)
          Storage Group Name . . . .            (fully specified)
       2 Generate list from Catalog
          Catalog Name . . .
          Volume Serial Number . . .            (fully or partially specified)
          Acquire Data from Volume . . . . . . . N  (Y or N)
          Acquire Data if DFSMShsm Migrated  . . N  (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
```

Figure 2-61   Panel 1 of 4 to generate list of data sets using data class WELCHRLS

On the DATA SET SELECTION ENTRY panel (see Figure 2-61) specify the following options:

- Select source of generated List . . 2.
- Data set name . . . **.
- Enter a slash (/) to select option Generate Exclusive list.

> **Note:** It is necessary to specify the slash (/) to select Generate Exclusive list. Otherwise, the subsequent selection filters do not take effect.

Do not press Enter at this point. Scroll using the DOWN command to the next selection panel.

Continue to scroll down through the panels until you find the one where you can enter the data class name. In our environment we used the data class WELCHRLS (see Figure 2-62, “Panel to specify data class” on page 68) in the form:

```
Data Class Name . . . . EQ      WELCHRLS
```

Press Enter to continue.

```
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                        DATA SET SELECTION ENTRY PANEL             Page 4 of 5
Command ===>

To limit the List, Specify a value or range of values in any of the following:
                             Rel Op   Value      Value      Value     Value
                             ------  --------   --------   --------  --------
    Allocation Unit  . . . .
    CF Cache Set Name  . . .
    CF Monitor Status  . . .
    CF Status Indicator  . .
    Change Indicator . . . .
    Compressed Format  . . .
    Data Class Name  . . . . EQ      WELCHRLS
    Data Set Environment . .
    Data Set Name Type . . .
    Data Set Organization
    (1 to 8 Values)     ===>
    DDM Attributes . . . . .
    Device Type  . . . . . .
    (1 to 8 Values)     ===>
Use ENTER to Perform Selection; Use UP/DOWN Command for other Selection Panel;
```

Figure 2-62 Panel to specify data class

You should then see a panel of the form shown in Figure 2-63.

```
 DATA SET LIST                 REDISPLAYED LIST
Command ===>                                                  Scroll ===> HALF
                                                        Entries 1-1 of 1
Enter Line Operators below:                             Data Columns 3-5 of 39

     LINE                                  ALLOC     ALLOW    % NOT
     OPERATOR          DATA SET NAME       SPACE     USED     USED
    ---(1)----  ------------(2)------------ --(3)---  --(4)---  -(5)-
                MHLRES1.RLS.JCL            --------  --------    ---
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----
```

Figure 2-63 Initial data set list from data class list

You can confirm that the data set is associated with data class WELCHRLS by paging to the right until you find the column headed DATA CLASS NAME, as in Figure 2-64.

```
 DATA SET LIST
Command ===>                                                  Scroll ===> HALF
                                                      Entries 1-1 of 1
Enter Line Operators below:                           Data Columns 23-25 of 39

     LINE                                  LAST BACKUP  CHG      DATA
     OPERATOR          DATA SET NAME           DATE     IND   CLASS NAME
    ---(1)----  ------------(2)------------ ---(23)----  (24)  ---(25)---
                MHLRES1.RLS.JCL             ----------   ---   WELCHRLS
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----
```

Figure 2-64 Data set list from data class list showing data class name

2.8.7 Operating support of VSAM RLS 64-bit data buffers

You can use operator commands to modify the size and usage of 64-bit data buffers, and check the level of the SMSVSAM server.

For detailed information about the use of system commands refer to *MVS System Commands,* SA22-7627.

2.8.8 Displaying settings for 64-bit data buffers

Use the DISPLAY SMS,SMSVSAM command to display the status of the settings for 64-bit data buffers. Figure 2-65 shows the results of issuing the DISPLAY SMS,SMSVSAM command. In particular, you must verify that CACHE FEATURE LEVEL B is available.

```
D SMS,SMSVSAM
 IEE932I 551
 IGW420I DISPLAY SMS,SMSVSAM
 DISPLAY SMS,SMSVSAM - SERVER STATUS
   SYSNAME:  SC63       AVAILABLE ASID: 0009 STEP: SmsVsamInitComplete

 DISPLAY SMS,SMSVSAM - JOB STATUS
   SUBSYSTEMS CONNECTED:        3 BATCH:        4

 DISPLAY SMS,SMSVSAM - LOCK TABLE STATUS (IGWLOCK00)
   CONNECT STATUS:
     SYSNAME:  SC63     ACTIVE            RSN: 02010407 RbldNotActive

   COMPOSITE STATUS:
     ORIGINAL STRUCTURE: NOT VOLATILE       NOT FAILURE ISOLATED


   STRUCTURE STATUS:
   SYSNAME: SC63      Simplex

 DISPLAY SMS,SMSVSAM - SMF RECORD 42 STATUS
                     SMF_TIME  CF_TIME ----- SUB-TYPE  SUMMARY -----
                                             15  16  17  18  19
   SYSNAME: SC63      YES- 4    1800- 4     YES YES YES YES YES

    SMSVSAM SMF 42 RECORDS ARE NOT WRITTEN FROM THIS SYSTEM.


 DISPLAY SMS,SMSVSAM - GLOBAL CACHE FEATURE PARMLIB VALUES
   MAXIMUM CF CACHE FEATURE LEVEL = Z

 DISPLAY SMS,SMSVSAM - CACHE FEATURE CODE LEVEL VALUES
   SYSNAME:  SC63
    CACHE FEATURE CODE LEVEL      = ABMx

 CACHE FEATURE LEVEL DESCRIPTION:
   Z  = No Caching Advanced functions are available
   A  = Greater than 4K Caching code is available
   B = VSAM/RLS 64-Bit Above the 2GB Bar processing capable
   Mx = IGW500I Open Sphere Messsage Status:
      M0 = Suppress All Open Sphere IGW500I Messages
      M1 = Display All Open Sphere IGW500I Messages
      M2 = Display First Open Sphere IGW500I message in sysplex
      M3 = All Open Sphere IGW500I message in sysplex to HardCopy
      M4 = First Open Sphere IGW500I message in sysplex to HardCopy
```

Figure 2-65 DISPLAY SMS,SMSVSAM command

This display shows that SMF type 42 records are not being written for this system. To have SMF type 42 records written you must issue the **vary sms,monds** command.

In Figure 2-66 we show the **v sms,monds(mhlres1.vsamrls.*),ON** command and the results.

```
V SMS,MONDS(MHLRES1.VSAMRLS.*),ON
IEE932I 785
IGW663I DFSMS CF MONITORING HAS BEEN ACTIVATED
FOR VSAM RLS DATASETS WHICH MATCH THE FOLLOWING
MASK SPECIFICATION: MHLRES1.VSAMRLS.*
```

*Figure 2-66 VARY SMS,MONDS(MHLRES1.VSAMRLS.**

Any data set with a name beginning with MHLRES1.VSAMRLS will have statistics recorded by SMF as type 42 subtype 16.

2.8.9 Controlling the use of 64-bit data buffers

You can change the amount of storage used by VSAM RLS 64-bit data buffers through the SETSMS RLSABOVETHEBARMAXPOOLSIZE command. In Figure 2-67 we show the use of the command to set the size to 500 (units are in MB) and the response received.

```
SETSMS RLSABOVETHEBARMAXPOOLSIZE(SC63,500)
IGW467I DFSMS RlsAboveTheBarMaxPoolSize 833
PARMLIB VALUE CHANGED ON SYSTEM: SC63
OLD VALUE: 0  1
NEW VALUE: 500  1
```

Figure 2-67 SETSMS RLSABOVETHEBARMAXPOOLSIZE command

The ability to use 64-bit storage data buffers can be displayed by setting the amount of storage to 0. If this parameter is set back to 0 then RLSFIXEDPOOLSIZE should also be set to 0 to avoid dedicating real storage that will not be used.

2.8.10 Specifying the use of fixed real storage for 64-bit data buffers

You can change the amount of fixed real storage used by VSAM RLS 64-bit data buffers through the SETSMS RLSFIXEDPOOLSIZE command.

In Figure 2-68 we show the use of the command to set the size to 50 (units are in MB), and the response received.

```
SETSMS RLSFIXEDPOOLSIZE(SC63,050)
IGW467I DFSMS RlsFixedPoolSize 879
PARMLIB VALUE CHANGED ON SYSTEM: SC63
OLD VALUE: 0  1
NEW VALUE: 50  1
```

Figure 2-68 SETSMS RLSFIXEDPOOLSIZE command

The amount of VSAM RLS 64-bit real storage reserved can be set back to 0 to release it.

2.8.11 Display of VSAM RLS 64-bit data buffer information

You can use the DISPLAY SMS,OPTIONS command to show the values for the parameters that relate to the VSAM RLS 64-bit data buffer feature. The applicable parameters are

- Rls_MaxCfFeatureLevel = Z
- RlsAboveThebarMaxPoolSize = 500
- RlsFixedPoolSize = 50

In Figure 2-69 we show the results of issuing the DISPLAY SMS,OPTIONS command.

```
D SMS,OPTIONS
 IGD002I 15:51:38 DISPLAY SMS 882
 ACDS     = SYS1.SMS.ACDS
 COMMDS   = SYS1.SMS.COMMDS
 INTERVAL = 15          DINTERVAL = 150
 SMF_TIME = YES         CACHETIME = 3600
 CF_TIME = 1800         PDSE_RESTARTABLE_AS = NO
 PDSE_BMFTIME = 3600    PDSE1_BMFTIME = 3600
 PDSE_LRUTIME = 60      PDSE1_LRUTIME = 60
 PDSE_LRUCYCLES = 15   PDSE1_LRUCYCLES = 15
 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 = 0MB
 USE_RESOWNER = YES     RLS_MAX_POOL_SIZE = 100MB
 RLSINIT = YES          RLSTMOUT = 0
 COMPRESS = GENERIC     LOG_OF_LOGS = IGWTVS.LOG.OF.LOGS
 QTIMEOUT = 300         TVSNAME = 063
 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 = 500
 RlsFixedPoolSize = 50
 PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
 GDS_RECLAIM = YES      DSSTIMEOUT = 0
 BLOCKTOKENSIZE = REQUIRE
 IGD002I 15:51:38 DISPLAY SMS
 TRACE    = ON     SIZE = 128K     TYPE = ALL
   JOBNAME = *    ASID = *
```

Figure 2-69 DISPLAY SMS,SMSVSAM

2.8.12 Coexistence with lower-level systems for VSAM RLS 64-bit data buffers

There is no support for 64-bit data buffers on lower-level systems:

- On a z/OS system lower than V1R7 data sets with greater than 255 extents cannot be used.
- If an SMS active configuration data set is shared between a z/OS V1R7 system and a lower-level system, and a data class specifies RlsAboveTheBar=Y, then that will be ignored on the lower-level system.
- The IGDSMSxx member of SYS1.PARMLIB that contains either of the following cannot be shared with a lower-level system:
  - RlsAboveTheBarMaxPoolSize
  - RlsFixedPoolSize

2.8.13 Maintenance on lower-level systems for VSAM RLS 64-bit data buffers

There is no specific maintenance required on lower-level systems since the VSAM RLS data buffers continue to be allocated as 31-bit buffers. We do, however, recommend that the PTFs for support of VSAM 255 extent removal be applied, as specified in Table 2-2 on page 58.

2.9 SMS volume and ACS allocation test enhancements

In this section we discuss the SMS enhancements with z/OS V1R7, which include:

- Volume status change command

  Prior to z/OS V1R7, it was not possible to change the status of a volume in NOTCON (not connected) status to any other status (for example, ENABLE or QUIESCE) without updating the SCDS and activating the new configuration. With z/OS V1R7 you can issue the V SMS,VOLUME command to alter the volume status from NOTCON to any other status. For further details see 2.9.1, “SMS volume status change from NOTCON” on page 74.

- New volume selection messages and traces

  In z/OS V1R7 you can use new functions for processing SMS-managed data sets that assist you in diagnosing the reasons for unexpected volume selection situations. This new function provides:

  - Summarized and detailed messages on request
  - Externalized DADSM failure reasons and diagnostic codes in summarized analysis messages
  - Externalized volume selection analysis data in SMS trace
  - New trace data for SMS-managed and non-SMS-managed VSAM allocations

  For further details see 2.9.2, “SMS ACS message processing enhancements” on page 74.

- ACS allocation test environment

  ACS services can be invoked by various callers representing different environments, such as ALLOC, RECALL, and RECOVER. Frequently, ACS services need to know who the caller is since processing may need to be different for the different environments. Some ISV products currently invoke ACS services with no specific environment, causing unpredictable results. With z/OS V1R7 a new value ALLOCTST has been provided for the ACS variable &ACSENVIR. Once the ISV programs are modified to set this variable, your ACS routines will be able to distinguish the new allocation test environment from the environment that represents the normal systems allocation environment. For further details see 2.9.3, “ACS allocation test environment” on page 81.

2.9.1 SMS volume status change from NOTCON

Prior to z/OS V1R7, to change the status of a volume from NOTCON (not connected) status, the ISMF command had to be used, followed by activation of the SCDS.

With z/OS V1R7 you can use the VARY SMS,VOLUME command to change the status from NOTCON directly.

In Figure 2-70 we show the command and response to vary volume MLD50C to ENABLE status on system SC63.

```
V SMS,VOLUME(MLD50C,SC63),ENABLE
IGD010I VOLUME (MLD50C,SC63 ) STATUS IS NOW ENABLED
```

Figure 2-70 VARY SMS,VOLUME to enable a volume

> **Note:** As with all uses of the VARY SMS,VOLUME command, this is only a change to the ACTIVE environment. To make a change permanent, the SCDS must be updated and activated.

Migration and coexistence considerations

There are no coexistence or maintenance considerations with respect to lower-level systems.

2.9.2 SMS ACS message processing enhancements

In addition to the summarized analysis messages issued by SMS when volume selection fails for an SMS-managed data set, in z/OS Release V1R7, SMS optionally provides more information to help you analyze the reasons why selection might have failed. The summary analysis messages provide diagnostic information about the number of volumes rejected for each reason during volume selection. However, there are many cases in which you need more detailed failure information to determine why volume selection has failed, or messages about successful allocations to determine why a data set was not allocated as expected. This enhancement provides the following functions:

- Summarized and detailed analysis messages on request
- DADSM failure reasons and diagnostic codes in summarized analysis messages
- Adding volume selection data to SMS trace data
- Adding new trace data for SMS and non-SMS managed VSAM allocations with more-complete information

Planning to use volume selection messages and traces

You must take account into consideration the following:

- The number of messages that could be issued once the enhanced message processing is turned on - see “Setting up volume selection messages and traces” on page 75.
- Compatibility with lower-level systems of new operands in the IGDSMSxx member of SYS1.PARMLIB - see “Coexistence with lower-level systems” on page 81; and maintenance required on lower-level systems - see “Maintenance for lower-level systems” on page 81.

Setting up volume selection messages and traces

The settings that you use to set up the message processing enhancements will affect the number of additional messages produced.

For full details on the concepts of volume selection messages and traces refer to:

- *z/OS and z/OS.e Planning for Installation*, GA22-7504
- *DFSMSdfp Storage Administration Reference,* SC26-7402
- *MVS Initialization and Tuning Reference,* SA22-7592

There are new parameters in the IGDSMSxx member of SYS1.PARMLIB that control the issuance of summarized and detailed volume selection analysis messages when creating a new SMS-managed or non-SMS-managed data set.

You must customize member SMS with appropriate values for VOLSELMSG and TRACE operands. To do this, do either or both of the following:

- Add the VOLSELMSG parameter and appropriate keyword to the IGDSMSxx member of SYS1.PARMLIB.
- Use the SETSMS command.

The default for VOLSELMSG is OFF, while the default for TRACE is ON with additional values that come into effect.

You should check the current settings for VOLSELMSG and TRACE so that you can, if necessary, adjust the current values and set up what you require. You can use the DISPLAY SMS,VOLSELMSG, as shown in “Checking status of VOLSELMSG settings” on page 77.

Initialization

There are SETSMS and SYS1.PARMLIB IGDSMSxx parameters that support the new volume selection messages and traces.

Trace and VOLSELMSG option parameters

TRACE (ON or OFF) SMS tracing is to be turned on or off.

Default values, if no trace values are specified in SYS1.PARMLIB, consist of:

- TRACE (ON)
- SIZE (128K)
- TYPE (ERROR)
- JOBNAME (*), which means all jobs
- ASID (*), which means all address spaces
- SELECT (ALL)

> **Note:** Final values consist of the sum of the defaults (where not overridden), the values in SYS1.PARMLIB, and those added by SETSMS SELECT or removed by SETSMS DESELECT.

The following parameters specify the size of the trace table, the type of errors to be traced, the job name or ASID to be traced, and the particular events that are to be selected (turned on) for tracing or deselected (turned off). The TRACE, SELECT, and DESELECT parameters apply only to the system on which the operator issues the SETSMS command.

- SIZE(*nnnnnn*, or *nnnnnn*K, or *nnn*M)

  Specifies the size of the trace table in kilobytes. If you omit K or M, the default unit is K. The default value is 128 K. The maximum is 255000 K or 255 M. This value is rounded up to the nearest 4 K.

Note: If you specify a size that is different from the previously used value, a new trace data area is built, the old trace data area is deleted, and no trace data is saved. If, however, the size you specify is the same as the previously used size, no new trace data area is built. If you specify zero, the existing trace data area is deleted without being replaced.

- SELECT(*option*[,*option*]...)

  SMS is to add one or more specific events to those that are to be traced. If tracing had been turned off for these events, SMS turns it back on for the specified events. The default is SELECT(ALL). SELECT only adds events. It does not delete any events. Use DESELECT to turn off one or more events. Both SELECT and DESELECT affect only the system on which you issue the SETSMS command. Both SELECT and DESELECT are only available for the life of the IPL.

- DESELECT(*option*[,*option*]...)

  SMS is to delete one or more events from the list of traced events.

- TYPE(ERROR *or* ALL)

  This specifies how you want to trace events and issue volume selection analysis messages.

  – ERROR - Specify ERROR to trace error events (TRACE(ON)) and issue volume selection analysis messages (VOLSELMSG(ON)) on failure allocations. The default value from SMS initialization is ERROR.

  – ALL - Specify ALL to trace all events (TRACE(ON)) and issue volume selection analysis messages (VOLSELMSG(ON)) on all allocations.

- VOLSELMSG(ON|OFF),(0|nnnnn|ALL)

  This allows you to control volume selection analysis messages issued when you create or extend a SMS-managed data set to a new volume. These analysis messages are written to the hardcopy log and the joblog.

  – ON|OFF - controls whether SMS volume selection analysis messages are being issued. The default is OFF.

  – 0|nnnnn|ALL - controls whether detailed analysis messages are being issued and the number of volumes to be included in them. The default is 0. Only summarized analysis messages will be issued.

    - nnnnn - indicates the number of volumes to be included in the message with a range of 0 to 65535.

    - ALL - indicates that all volumes used for volume selection will be included in detailed analysis messages.

Note: TRACE and VOLSELMSG are affected by the TYPE, JOBNAME, ASID, STEPNAME, and DSNAME operands. TRACE is also affected by SELECT and DESELECT.

- If you specify VOLSELMSG(nnnnn|ALL), with nnnnn having a value greater then 0 along with TYPE(ALL), you must also specify one of the following parameters to limit the number of detailed analysis messages issued:
  - JOBNAME
  - ASID
  - STEPNAME
  - DSNAME

  When all volumes are to be included, volumes are listed by storage group. If only a subset of volumes is to be included, volumes are listed in volume selection preference order with no association to storage group. The system can issue an excessive number of analysis messages to the spool when the following conditions occur:
  - The job or address space creates or extends numerous SMS-managed data sets.
  - Many volumes are to be included in the analysis messages.
- JOBNAME(jobname or *)

  Specify JOBNAME to limit tracing (TRACE(ON)), issue volume selection messages (VOLSELMSG(ON)), or both, to particular jobs. Specify an asterisk (*) to select all jobs. If you specify JOBNAME, omit ASID.
- ASID(asid or *)

  Specify ASID to limit tracing (TRACE(ON)) or issue volume selection messages (VOLSELMSG(ON)) to particular address spaces. The default is to permit them for all address spaces. Specify an asterisk (*) to select all address spaces. If you specify ASID, omit JOBNAME.
- STEPNAME(stepname | *)

  Limits the number of issued volume selection analysis messages activated by VOLSEGMSG(ON) to either a certain stepname or all stepnames. The default is to issue volume selection analysis messages on all stepnames.
- DSNAME(dsname | *)

  Limits the number of issued volume selection analysis messages activated by VOLSEGMSG(ON) to either a certain data set or all data set names. The default is to issue volume selection analysis messages on all data set names. For a VSAM data set, this is the cluster's entry name.

Checking status of VOLSELMSG settings

There are a number of options on the VOLSELMSG parameter.

In Figure 2-71 we show the DISPLAY SMS,VOLSELMSG command and response. Note that in addition to the VOLSELMSG settings, the TRACE settings are also shown.

```
D SMS,VOLSELMSG
IGD002I 14:11:11 DISPLAY SMS 536
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
TRACE    = ON     SIZE = 128K     TYPE = ALL
  JOBNAME = *    ASID = *
```

Figure 2-71 DISPLAY SMS,VOLSELMSG

VOLSELMGS and TRACE are interrelated, as they are both influenced by the same SMS parameters. In this case, VOLSELMSG is OFF, in which case no other operands are applicable, and TRACE is ON, but the trace is not restricted by JOBNAME or ASID.

Examples of the use of VOLSELMSG enhanced messages

We show a data set allocation with VOLSELMSG turned off, then a similar allocation with VOLSELMSG turned on. In both cases the operation is restricted to JOBNAME=MHLRES1E.

In Example 2-1, using SDSF, we show part of the output from job MHLRES1E running while VOLSELMSG is turned off. Messages IGD01007I, IGD01009I, and IGD0101I are issued from the ACS routines. You can contrast this with Example 2-4 on page 79 where, besides messages IGD01007I, IGD01009I, and IGD0101 as before, additional information messages are issued.

Example 2-1 Job MHLRES1E to allocate a data set with VOLSELMSG OFF

```
Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17083  DSID     2 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA **********************************
                   J E S 2  J O B  L O G  --  S Y S T E M  S C 6 3  --  N O D E

15.29.07 JOB17083 ---- THURSDAY,  30 MAR 2006 ----
15.29.07 JOB17083  IRR010I  USERID MHLRES1  IS ASSIGNED TO THIS JOB.
15.29.07 JOB17083  ICH70001I MHLRES1  LAST ACCESS AT 11:57:42 ON THURSDAY, MARCH
15.29.07 JOB17083  $HASP373 MHLRES1E STARTED - INIT 1    - CLASS A - SYS SC63
15.29.07 JOB17083  IEF403I MHLRES1E - STARTED - TIME=15.29.07 - ASID=002B - SC63
15.29.07 JOB17083  -                                        --TIMINGS (MINS.)--
15.29.07 JOB17083  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
15.29.07 JOB17083  -MHLRES1E          LARGEDS     00      7    .00    .00    .00
15.29.07 JOB17083  IEF404I MHLRES1E - ENDED - TIME=15.29.07 - ASID=002B - SC63
15.29.07 JOB17083  -MHLRES1E ENDED.  NAME-                    TOTAL CPU TIME=
15.29.07 JOB17083  $HASP395 MHLRES1E ENDED
------ JES2 JOB STATISTICS ------
  30 MAR 2006 JOB EXECUTION DATE
            6 CARDS READ
           43 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
-------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17083  DSID     2 LINE 19      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
            2 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //MHLRES1E JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
        2 //LARGEDS  EXEC PGM=IEFBR14
        3 //DDALLOC  DD  DISP=(,CATLG),UNIT=SYSALLDA,
          //             DSN=MHLRES1.LARGDS1.JCL,
          //             SPACE=(TRK,(65600,1)),
          //             DSNTYPE=LARGE
 STMT NO. MESSAGE
-
        3 IGD01007I &DSNTYPE = LARGE
        3 IGD01009I MC ACS GETS CONTROL &ACSENVIR=ALLOC
        3 IGD01010I SG ACS GETS CONTROL &ACSENVIR=ALLOC
ICH70001I MHLRES1  LAST ACCESS AT 11:57:42 ON THURSDAY, MARCH 30, 2006
```

```
IEF236I ALLOC. FOR MHLRES1E LARGEDS
IGD101I SMS ALLOCATED TO DDNAME (DDALLOC )
        DSN (MHLRES1.LARGDS1.JCL                         )
        STORCLAS (STANDARD) MGMTCLAS (MCDB22) DATACLAS (        )
        VOL SER NOS= MLD20C
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17083  DSID     4 LINE 12      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
IEF142I MHLRES1E LARGEDS - STEP WAS EXECUTED - COND CODE 0000
IGD104I MHLRES1.LARGDS1.JCL                         RETAINED,  DDNAME=DDALLOC
IEF373I STEP/LARGEDS /START 2006089.1529
IEF374I STEP/LARGEDS /STOP  2006089.1529 CPU    0MIN 00.00SEC SRB    0MIN 00.00S
IEF375I  JOB/MHLRES1E/START 2006089.1529
IEF376I  JOB/MHLRES1E/STOP  2006089.1529 CPU    0MIN 00.00SEC SRB    0MIN 00.00S
******************************** BOTTOM OF DATA ********************************
```

In Example 2-2 we show the command to set VOLSELMSG on for all volumes in the pool, and to be applicable to job MHLRES1E only.

Example 2-2   SETSMS VOLSELMSG command to turn the function on

```
SETSMS VOLSELMSG(ON,ALL),JOBNAME(MHLRES1E)
IEE712I SETSMS   PROCESSING COMPLETE
```

In Example 2-3 we show the D SMS,VOLSELMSG command, with output. Note that the JOBNAME restriction now applies to the TRACE function as well.

Example 2-3   DISPLAY SMS,VOLSELMSG after tuning the function on

```
D SMS,VOLSELMSG
 IGD002I 15:52:19 DISPLAY SMS 678
 VOLSELMSG = (ON,ALL)    TYPE = ALL    JOBNAME = MHLRES1E
     ASID = *  STEPNAME = *
     DSNAME = *
 TRACE    = ON     SIZE = 128K     TYPE = ALL
   JOBNAME = MHLRES1E    ASID = *
```

In Example 2-4 we show the partial SDSF output from job MHLRES1E run while VOLSELMSG is on and applicable to job MHLRES1E. Note that besides the messages IGD01007I, IGD01009I, and IGD0101 issued from the ACS routines, a block of additional messages has been issued, as highlighted.

Example 2-4   Job MHLRES1E to allocate data set with VOLSELMSG (ON,ALL) and JOBNAME(MHLRES1E)

```
Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17084  DSID     2 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA **********************************
                   J E S 2  J O B  L O G  --  S Y S T E M  S C 6 3  --  N O D E

15.32.48 JOB17084 ---- THURSDAY,  30 MAR 2006 ----
15.32.48 JOB17084  IRR010I  USERID MHLRES1  IS ASSIGNED TO THIS JOB.
```

```
15.32.48 JOB17084  ICH70001I MHLRES1  LAST ACCESS AT 15:29:07 ON THURSDAY, MARCH
15.32.48 JOB17084  $HASP373 MHLRES1E STARTED - INIT 1    - CLASS A - SYS SC63
15.32.48 JOB17084  IEF403I MHLRES1E - STARTED - TIME=15.32.48 - ASID=002B - SC63
15.32.48 JOB17084  IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA
15.32.48 JOB17084  IGD17386I VOLSELMSG(ON,ALL) TYPE(ALL) JOBNAME(MHLRES1E) ASID(
   642             STEPNAME(* ) DSNAME(* )
15.32.48 JOB17084  IGD17387I DS_TYPE(NONVSAM) SC(STANDARD) DC() GS(N) SPACE(3630
15.32.48 JOB17084  IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE
   644             WERE ELIGIBLE FOR VOLUME SELECTION.
   644             THE CANDIDATE STORAGE GROUPS WERE:SGMHL01
15.32.48 JOB17084  IGD17269I 3 VOLUMES WERE REJECTED BECAUSE THE SMS VOLUME STAT
15.32.48 JOB17084  IGD17269I 2 VOLUMES WERE REJECTED BECAUSE THEY WERE NOT ONLIN
15.32.48 JOB17084  IGD17269I 2 VOLUMES WERE REJECTED BECAUSE THE UCB WAS NOT AVA
15.32.48 JOB17084  IGD17385I =====DETAILED ANALYSIS MESSAGES ON DEFINING DATA SE
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17084  DSID     2 LINE 19      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
15.32.48 JOB17084  IGD17388I ==POOL SG SGMHL01
15.32.48 JOB17084  IGD17389I MLD30C(S,3868001D) MLD00C(N,3868001D) MLD40C(N,3868
15.32.48 JOB17084  IGD17389I MLD20C(N,2868001D) MHLS1A(N,0868001D) MHLS1E(R,,800
15.32.48 JOB17084  -                                            --TIMINGS (MINS.)--
15.32.48 JOB17084  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
15.32.48 JOB17084  -MHLRES1E          LARGEDS     00      7    .00    .00    .00
15.32.48 JOB17084  IEF404I MHLRES1E - ENDED - TIME=15.32.48 - ASID=002B - SC63
15.32.48 JOB17084  -MHLRES1E ENDED.  NAME-                     TOTAL CPU TIME=
15.32.48 JOB17084  $HASP395 MHLRES1E ENDED
------ JES2 JOB STATISTICS ------
  30 MAR 2006 JOB EXECUTION DATE
            6 CARDS READ
           57 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            4 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //MHLRES1E JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
        2 //LARGEDS  EXEC PGM=IEFBR14
        3 //DDALLOC  DD  DISP=(,CATLG),UNIT=SYSALLDA,
//             DSN=MHLRES1.LARGDS2.JCL,
          //             SPACE=(TRK,(65600,1)),
          //             DSNTYPE=LARGE
 STMT NO. MESSAGE
-
        3 IGD01007I &DSNTYPE = LARGE
        3 IGD01009I MC ACS GETS CONTROL &ACSENVIR=ALLOC
        3 IGD01010I SG ACS GETS CONTROL &ACSENVIR=ALLOC
ICH70001I MHLRES1  LAST ACCESS AT 15:29:07 ON THURSDAY, MARCH 30, 2006
IEF236I ALLOC. FOR MHLRES1E LARGEDS
IGD101I SMS ALLOCATED TO DDNAME (DDALLOC )
        DSN (MHLRES1.LARGDS2.JCL                        )
        STORCLAS (STANDARD) MGMTCLAS (MCDB22) DATACLAS (        )
        VOL SER NOS= MLD30C
IEF142I MHLRES1E LARGEDS - STEP WAS EXECUTED - COND CODE 0000
IGD104I MHLRES1.LARGDS2.JCL                        RETAINED,  DDNAME=DDALLOC
IEF373I STEP/LARGEDS /START 2006089.1532
```

```
IEF374I STEP/LARGEDS /STOP  2006089.1532 CPU    0MIN 00.00SEC SRB    0MIN 00.00S
IEF375I  JOB/MHLRES1E/START 2006089.1532
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY MHLRES1E JOB17084  DSID     4 LINE 17      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
IEF376I  JOB/MHLRES1E/STOP  2006089.1532 CPU    0MIN 00.00SEC SRB    0MIN 00.00S
```

In Example 2-5 we show the SETSMS VOLSELMSG(OFF,ALL) command, followed by a display of the VOLSELMSG status. Note that the JOBNAME is still set, though no longer applicable in VOLSELMSG, but also still set for TRACE where it is applicable.

Example 2-5   Command to turn VOLSELMSG OFF and display status

```
SETSMS VOLSELMSG(OFF,ALL),JOBNAME(MHLRES1E)
 IEE712I SETSMS   PROCESSING COMPLETE
 D SMS,VOLSELMSG
 IGD002I 17:37:38 DISPLAY SMS 709
 VOLSELMSG = (OFF,ALL)    TYPE = ALL    JOBNAME = MHLRES1E
     ASID = *  STEPNAME = *
     DSNAME = *
 TRACE    = ON     SIZE = 128K     TYPE = ALL
   JOBNAME = MHLRES1E    ASID = *
```

Coexistence with lower-level systems

There is no support for enhanced volume selection messages and traces on lower-level systems. The IGDSMSxx member of SYS1.PARMLIB that contains the following cannot be shared with a lower-level system:

```
VOLSELMSG
```

Maintenance for lower-level systems

There is no specific maintenance required on lower-level systems since the VOLSELMSG parameter is not supported on lower-level systems.

2.9.3 ACS allocation test environment

ACS services can be invoked by various callers representing different environments. Frequently, ACS services need to know who the caller is since their processing may differ for different callers or environments.

When invoked by converter/interpreter or by dynamic allocation, the ACS read-only variable ACEROENV representing this environment is set to ALLOC to indicate that this invocation is occurring during the allocation process. Some Independent Software Vendor (ISV) programs have hooks in this allocation process and may call ACS Services directly. Since no specific environment has been defined for this situation, which would not normally result in an actual allocation, logging through the ACS Services routines can become misleading.

In z/OS V1R7, a new valid environment to support test entry to the ACS routines is provided. This is primarily intended for use by ISV programs when they call ACS services to check what would happen if certain JCL and parameters were used.

When the ACS routines are driven, the user can check the environment variable, and may choose to treat, for example, entries in the ALLOCTST environment (the test entry) differently than the ALLOC environment (the production entry).

Support for this use of the ACEROENV is also provided through extensions to the Naviquest testing environment.

2.10 Extended Remote Copy (XRC) Plus

XRC Plus is an enhancement to XRC that provides a solution to the problem of how to mirror log data from the system logger for a disaster recovery system. The current solution requires using mirrored disk staging data sets, which forces a wait for a synchronous I/O to take place to the staging data set even if you are also using coupling facility structures. This can mean an unacceptable delay to your applications, which will wait for the log data to be written to both the CF structure and the disk staging data set.

The solution to this problem allows the system logger to use the coupling facility and write asynchronously to the disk staging data sets. XRC Plus provides the capability to mirror the volume that contains the system logger staging data sets and maintain consistency. By allowing the system logger to use the coupling facility, performance will be improved.

The new LOGPLUS keyword for the XADDPAIR command specifies that the primary volume of the volume pair is to be explicitly written to by the z/OS system logger. When using LOGPLUS, you can specify a single pair or a single pair and a utility pair, but the utility pair must be specified last. A unique storage control session number will be assigned to the primary volume. LOGPLUS is available only in DFSMS z/OS V1R7 and later, and is mutually exclusive with the SCSESSION keyword.

In Figure 2-72 we show use of the XADDPAIR command with the LOGPLUS operand.

```
XADDPAIR LOGXPLUS VOLUME(DRXRC1 DRXRC2 XRCP1C XRCUTL) ERRORLEVEL(SESSION)
LOGPLUS
```

Figure 2-72 XADDPIR command with LOGPLUS operand

2.11 DEVSERV QLIB command

The DEVSERV command has been enhanced to display tape library information. Use the QLIB option of the DEVSERV command to:

- Request a list of tape library subsystems that are defined to the host. Libraries are listed by serial number (library-ID).
- Request a list of devices within a library. Devices are listed by device number and the library port for each device is displayed.
- Validate the connection status of devices in a library, for example, devices that are connected to the host.
- Delete an improperly defined library control block in preparation for an IODF activate.
- Issue a diagnostic state save order to a library when requested by the IBM Support Center.

In Figure 2-73 we show use of the DEVSERV QLIB,LIST command to determine tape libraries on the system, followed by the DEVSERV QLIB,10435 command to list details of the library identified by serial number 10435.

```
DS QL,LIST
IEE459I 11.31.04 DEVSERV QLIB 465
The following are defined in the   ACTIVE configuration:
*10435

DS QL,10435
IEE459I 11.32.04 DEVSERV QLIB 467
The following are defined in the   ACTIVE configuration:
LIBID  PORTID       DEVICES
10435  03           0B90* 0B91* 0B92* 0B93*
```

Figure 2-73 Sample DEVSERV QLIB displays

For the syntax of the DEVSERV QLIB command and further examples, see *MVS System Commands,* SA22-7627.

2.12 PDSE enhancements

The Partitioned Data Set Extended (PDSE) function was extended in DFSMS V1R6, when a second restartable address space was added. It is delivered in z/OS V1R7 substantially as provided in z/OS V1R6, but, as a result of a number of problems that arose after release, a number of PTFs were required. It is included here because although the defaults should now result in acceptable performance, users should be aware of the impact of changes that they may make, or that are inherited from a prior level system where they were explicitly set and so override the defaults now shipped.

Users interested in the significant enhancements delivered for PDSE with z/OS V1R6 and carried through to z/OS V1R7 should refer to the IBM Redbook *Partitioned Data Set Extended Usage Guide*, SG24-6106.

You can find on-going information about the status of PDSE APARs in the informational APAR II14067.

This section describes the PDSE function, which was made available in DFSMS V1.6 by PTF and is included in DFSMS V1.7.

This information was originally published on the ITSO Web site as TIPS0567. The information has been reviewed and updated where necessary in this document, but none of the information in TIPS0567 has been invalidated, although some of the original fixing PTFs have been superseded or merged into the z/OS V1R7 base.

Refer to Appendix B, "APAR text" on page 349, for any APARs that might be applicable to PDSE.

2.12.1 PDSE Hiperspace problems, considerations, and parameter changes

Users of Partitioned Data Set Extended data sets may experience high CPU utilization in the SMSPDSE address space. This has caused a problem for many users.

The high CPU utilization problem accessing PDSEs is caused by the unexpected caching of a large volume of program objects. This high CPU utilization is not a problem for those PDSE users who have a Hiperspace™ size specified or for those users whose applications fit the caching model. This section provides some background, discusses the problem, and presents the solution.

PDSE Hiperspace (caching) considerations: background

PDSE Hiperspace represents a caching function implemented to improve PDSE performance in those cases where the same member (or members) are accessed repeatedly. System storage and the CPU cycles required for caching replace the overhead of repeatedly reading members from DASD. As a result, benefits obtained from using PDSE Hiperspace depend on the application and its PDSE member access patterns. (PDSE directories are automatically cached by the buffer management facility and are not part of this discussion.)

Heavily used PDSEs and those that stay open for long periods of time are the best candidates for PDSE Hiperspace caching. Be sure your application fits the caching model before putting the PDSE into the *must cache* Storage Class.

PDSE system data sets (for example, those in LNKLST) are usually managed by LLA/VLF. Any program object that is cached in Hiperspace and at the same time is also managed by LLA/VLF will, in time, be dropped from the Hiperspace, so it is better to prevent these program objects from going into the Hiperspace in the first place.

For SMS-managed PDSE data sets, the use of Hiperspace caching is controlled by the Millisecond Response Time (MSR) value in the associated Storage Class. The MSR values apply not just to PDSE caching, but have wider applicability to the DASD storage units. In the case of PDSE data sets, only an MSR value that translates to *must cache* causes member caching. MSR values that translate to *may cache* and *never cache* will not cause members to be cached. For example, a Storage Class with MSR set to a value of 1 indicates must cache and 999 means never cache. (Refer to “Millisecond response time” on page 86 for more information about MSR values and caching.) Ensure that SMS-managed PDSEs are associated with Storage Classes that have appropriate MSR settings. (Note that PDSE data sets shipped as part of the operating system are generally not SMS-managed, so they have no Storage Class and therefore cannot be cached.)

If the PDSE Hiperspace is to be used, then its use of CPU cycles, real storage, or both must be balanced against the benefit obtained.

Hiperspaces use system storage (real and virtual) and must be tuned to avoid excessive use at the expense of other system functions. Hiperspace pages are the last to be stolen in a real storage-constrained environment, so the size of the Hiperspace may have to be limited when real storage is constrained. In addition, there has always been a trade-off between CPU and I/O costs. Caching reduces I/O path and increases CPU utilization. It is a trade-off made by the installation and applications depending on the access patterns for that particular data set.

APARs that created and solved several PDSE Hiperspace problems

The recent problems with CPU utilization resulted from fixing a PDSE bug, which fixed PDSE Hiperspace caching. As the default, it abruptly started working again for almost all PDSE users.

PDSE Caching Features - original design. (PDSE caching is limited by the ESA Architecture.)

1. For the installation to enable PDSE caching at that time, there were two requirements:
   - In the IGDSMSXX PARMLIB parameter, HSP_SIZE had to be a non-zero value. The product was shipped with a default size of 256 meg.
   - In addition, the client had to have expanded store, which had to be bought separately.
   - Few clients bought expanded store (and probably not any small clients), so PDSE caching was not common.
2. In addition, PDSE caching could (and can) be controlled at the data set level by Storage Class. Even though caching was enabled at the installation level, it could be regulated by the Storage Class MSR value. It is believed that few clients took advantage of this Storage Class feature in the beginning.

z/Architecture® was introduced. This removed the limiting factor that ESA Architecture had, such as the need for expanded store. Expanded store was replaced by real storage in the new processors. All clients automatically had it. PDSE code was changed, via OW53245, to recognize and use the real storage for a Hiperspace. This had the effect of enabling all installations with z/Architecture to PDSE caching, since the default of 256 meg remained in effect. While in the past the client had to do something explicit to get caching (such as buy Expanded Store), it now came automatically.

Yet PDSE caching continued to be limited, blocked by a bug. A bug prevented any pages from being thrown away (that is, they never aged out). When the cache was full, further caching was blocked and caching stopped. This bug was fixed with OA06884, which was released 06/07/04. This fix had the effect of abruptly enabling caching for many clients. Soon after the release of this APAR, it was obvious that it was causing a problem with CPU usage. There were two attempts to reissue the fix with further hold information to warn the clients (via PE APARs OA08991, released 10/20/04, and finally OA10323, released 03/04/05).

In addition, Storage Class stopped controlling caching at some point. Another bug caused contents supervisor to ignore Storage Class and always cache, so when OA06884 was released all program objects were cached. This was fixed via APAR OA09162, released 01/13/05.

To resolve the PDSE issue with CPU utilization, APAR OA11068 (target release 06/15/05) changes the default to not enable caching for z/Architecture, by changing the default to HSP_SIZE=0 for z/Architecture systems.

The installation can enable caching by changing the HSP_SIZE= to a non-zero value. This requires an IPL for all releases except z/OS V1.6, in which it is possible to restart the PDSE restartable address space if this feature is enabled. Note that for z/OS V1.6 each of the two address spaces, SMSPDSE and SMSPDSE1, need their own specification for HSP_SIZE. In all cases the specification of HSP_SIZE requires a re-IPL except for the restartable address space (SMSPDSE1), whose HSP_SIZE can be changed by restarting the address space.

Storage Class continues to be available to control caching at a PDSE level (via the MSR parameter).

The maximum value of MSR=999 means NEVER CACHE. Decreasing values mean MAYBE CACHE or MUST CACHE, subject to the DASD control unit characteristics. MSR=1 ensures MUST CACHE. (MAYBE CACHE is treated as NEVER CACHE for PDSEs.) This decision is based on how the data set is accessed. If an application does a lot of opens and closes, the PDSE should not be cached. Members are cached only while the data set is open by at least one application on the system. Caching works best for PDSEs that are open a long time with frequent references to the same members.

APAR OA11068 also modifies the least recently used (LRU) default values. LRU values can be modified by the installation using the SETSMS command. This does not take an IPL. These values can be changed dynamically.

LRUTIME( ) specifies the number of seconds between running the LRU queue. LRUCYCLES( ) specifies how many times LRU must run before throwing out a page. Default LRU values shipped with the product in the past were LRUTIME(15) and LRUCYCLES(240). These defaults will remain for non-z/Architecture. For z/Architecture, however, this may be too often to search and too long to keep a page in cache.

For z/Architecture default LRU values will be changed to LRUTIME(60) and LRUCYCLES(15). This will cause the code to run the LRU queue once per minute, keeping pages in cache for 15 minutes before throwing them out. This should control some of the possible thrashing and reduce the frequency of searching the queue, thus reducing the CPU utilization.

The following sample SETSMS commands can be used to change the LRUCYCLES and LRUTIME values:

```
SETSMS PDSE_LRUCYCLES(nnn)
SETSMS PDSE1_LRUCYCLES(nnn)

SETSMS PDSE_LRUTIME(nn)
SETSMS PDSE1_LRUTIME(nn)
```

Clients can tune their systems for performance by tweaking these LRU values. SMF provides caching statistics by Storage Class.

SMF records can provide a list of Storage Classes that are not caching well, such as those where the hit ratio is low.

This APAR (OA11068) is marked HIPER for performance reasons with a ++HOLD ACTION and ++HOLD DOC because default parameters are being changed. It will prereq the PTFs associated with APAR OA09162 (Storage Class) and APAR OA10323 (fixes caching by throwing out aged pages). When this is installed, CPU utilization trade-offs will be under the control of the installation.

2.12.2 Reference notes

The following additional information relates to PDSEs and caching for your reference.

Millisecond response time

The millisecond response time (MSR) serves two purposes in SMS. First, it is used as the performance objective for selecting candidate volumes for new data set placement. During a new data set allocation, SMS searches for a volume that meets or closely matches this objective. If no volume satisfies the objective, then SMS attempts to find a volume that comes closest to matching it. If more than one MSR is explicitly or implicitly specified, the Storage Class and associated device MSRs are averaged and compared.

Second, if the data is placed on a volume attached through an IBM 3990 Storage Control with cache, and cache is enabled for that volume, the MSR is used to determine whether caching is mandatory (optional) or should be inhibited for the data set. This attribute does not apply to objects.

The PDSE implementation of MSR is different from the SMS implementation. PDSE data sets use the PDSE Hiperspaces for caching. The Hiperspace caching is either must cache or never cache.

For PDSE data sets, specify an MSR value of 1 for must cache. Specify an MSR value of 999 for never cache.

Refer to "Defining Performance Objectives" in *z/OS DFSMSdfp Storage Administration Reference*, SC26-7402, for detailed information about the affect of MSR on different storage device types and models.

Optional keywords for IGDSMSxx

`HSP_SIZE | PDSE_HSP_SIZE({nnn})`

Default: 0 for z/Architecture mode, 256 for non-z/Architecture mode

With the introduction of a second PDSE address space in z/OS 1.6, there are now two sets of PDSE Hiperspace parameters. For z/OS V1.6 and later, HSP_SIZE is synonymous with PDSE_HSP_SIZE and is associated with the SMSPDSE address space. HSP_SIZE specifies the size of the Hiperspace in megabytes that is used for PDSE member caching for SMSPDSE.

You can use the HSP_SIZE parameter to request up to 2047 MB for the PDSE Hiperspace. You can indicate that the Hiperspace is not to be created by setting HSP_SIZE to 0. If the Hiperspace is not created, the system will not cache PDSE members.

On systems that are running in z/Architecture mode, when OA11068 is installed, by default the PDSE Hiperspace is not created and PDSE member caching is disabled.

On non-z/Architecture mode systems that have expanded storage, by default the PDSE Hiperspace uses either 256 MB of expanded storage or one-half of the system's available expanded storage, whichever amount is lower. If no expanded storage is online to the system, the Hiperspace cannot be created and PDSE member caching is disabled.

If you specify a valid value for HSP_SIZE, the system uses it to create the PDSE Hiperspace at IPL time. The HSP_SIZE value remains in effect for the duration of the IPL.

If there is not enough appropriate storage available to satisfy the HSP_SIZE value, the system uses some portion of the available storage (up to the full amount) for the PDSE Hiperspace depending on the amount of caching activity in the system. The system stops caching PDSE members if the available storage becomes full.

Use the HSP_SIZE parameter with care. If you specify an HSP_SIZE value that is too low for normal PDSE Hiperspace usage, you can degrade PDSE performance. If you specify a value for HSP_SIZE that is too large, and there is contention for storage on the system, you can degrade performance of other components or applications in the system.

To determine the current HSP_SIZE value of the PDSE Hiperspace, use the DISPLAY SMS,OPTIONS command, or review the messages that are written to syslog when SMS is started.

You can examine SMF type 42, subtype 1, records to evaluate the effectiveness of a particular HSP_SIZE value.

`LRUCYCLES | PDSE_LRUCYCLES(nnn)`

Default: 15 for z/Architecture mode, 240 for non-z/Architecture mode

With the introduction of a second PDSE address space in z/OS V1.6, there are now two sets of PDSE parameters. For z/OS V1.6 and later, LRUCYCLES is synonymous with PDSE_LRUCYCLES and is associated with the SMSPDSE address space.

`PDSE_LRUCYCLES(nnn l 15)`

Default: 15

PDSE_LRUCYCLES is associated with the restartable SMSPDSE1 address space. Refer to the RESTARTABLE_PDSE_AS(YES) parameter for information about the SMSPDSE1 address space.

LRUCYCLES specifies the maximum number of times (5 to 240 cycles) that the least recently used (LRU) routine will pass over inactive buffers before making them available for reuse. While this parameter sets the maximum value, the LRU algorithm will dynamically change the actual number of times it passes over inactive buffers.

LRUCYCLES is related to LRUTIME. A change to the LRUCYCLES value introduced by this parameter will take effect on the next execution of the LRU routine. Most installations should use the default value (changed by APAR OA11068, as mentioned above). In some very high data rate environments you may want to tune this value. You should monitor the SMF 42 type 1 record to determine the amount of caching activity in the BMF data space. See *z/OS: MVS System Management Facilities (SMF)*, SA22 -7630, for information about the buffer management statistics recorded in SMF record type 42.

`LRUTIME|PDSE_LRUTIME(nnn)`

Default: 60 for z/Architecture mode, 15 for non-z/Architecture mode

With the introduction of a second PDSE address space in z/OS V1.6, there are now two sets of PDSE parameters. For z/OS 1.6 and later, LRUTIME is synonymous with PDSE_LRUTIME and is associated with the SMSPDSE address space. PDSE1_LRUTIME is associated with the restartable SMSPDSE1 address space. Refer to the RESTARTABLE_PDSE_AS(YES) parameter for information about the SMSPDSE1 address space.

LRUTIME specifies the number of seconds (5 to 60) that the system will wait between calls to the LRU routine. The LRU routine releases inactive buffers in the Hiperspace used to cache PDSE member data.

LRUTIME is related to LRUCYCLES. A change to the LRUTIME value will take effect on the next execution of the LRU routine. Most installations should use the default value (which is changed by OA11068). In some very high data rate environments you may want to tune this value. You should monitor the SMF 42 type 1 record to determine the amount of caching activity in the data space. See *z/OS: MVS System Management Facilities (SMF)*, SA22-7630, for information about the buffer management statistics recorded in SMF record type 42.

`BMFTIME|PDSE_BMFTIME(nnn)`

Default: 3600

With the introduction of a second PDSE address space in z/OS V1.6, there are now two sets of PDSE parameters. For z/OS V1.6 and later, BMFTIME is synonymous with PDSE_BMFTIME and is associated with the SMSPDSE address space. PDSE1_BMFTIME is associated with the restartable SMSPDSE1 address space. Refer to the RESTARTABLE_PDSE_AS(YES) parameter for information about the SMSPDSE1 address space.

BMFTIME specifies the number of seconds that SMS is to wait between recording SMF records for buffer management facility (BMF) cache use. You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds), and the default is 3600 (one hour).

For information about the buffer management statistics recorded in SMF record type 42, see *z/OS: MVS System Management Facilities (SMF)*, SA22-7630.

`PDSE1_HSP_SIZE({nnn|0})`

Default: 0

With the introduction of a second PDSE address space in z/OS V1.6, there are now two sets of PDSE parameters. For z/OS 1.6 and later, PDSE1_HSP_SIZE is associated with the restartable SMSPDSE1 address space. Refer to the RESTARTABLE_PDSE_AS(YES) parameter for information about the SMSPDSE1 address space.

PDSE1_HSP_SIZE specifies the size of the Hiperspace in megabytes that is used for PDSE member caching for SMSPDSE1.

You can use the PDSE1_HSP_SIZE parameter to request up to 2047 MB for the PDSE1 Hiperspace, or you can indicate that the Hiperspace is not to be created by setting PDSE1_HSP_SIZE to 0. If the Hiperspace is not created, the system will not cache PDSE members.

On systems that are running in z/Architecture mode, by default the PDSE1 Hiperspace is not created and PDSE member caching is disabled.

If you specify a valid value for PDSE1_HSP_SIZE, the system uses it to create the PDSE1 Hiperspace at IPL time. The PDSE1_HSP_SIZE value may be changed by a **`SET SMS=xx`** command. The new value will be used if the SMSPDSE1 address space is restarted.

If not enough of the appropriate storage is available to satisfy the PDSE1_HSP_SIZE value, the system uses some portion of the available storage (up to the full amount) for the PDSE1 Hiperspace, depending on the amount of caching activity in the system. The system stops caching PDSE members if the available storage becomes full.

Use the PDSE1_HSP_SIZE parameter with care. If you specify a PDSE1_HSP_SIZE value that is too low for normal PDSE Hiperspace usage, PDSE performance can degrade. If you specify a value for PDSE1_HSP_SIZE that is too large, and there is contention for storage on the system, it can degrade performance of other components or applications in the system.

To determine the current PDSE1_HSP_SIZE value of the PDSE Hiperspace, use the DISPLAY SMS,OPTIONS command, or review the messages that are written to syslog when SMS is started.

You can examine SMF type 42, subtype 1, records to evaluate the effectiveness of a particular PDSE1_HSP_SIZE value.

`SMF_TIME(YES|NO)`

Default: YES

SMF_TIME specifies whether DFSMS is to use SMF timing (that is, whether SMF type 42 records are to be created at the expiration of the SMF interval period, synchronized with SMF and RMF™ data intervals).

The following SMF record 42 subtypes are affected when you specify SMF_TIME(YES): 1, 2, 15, 16, 17, 18. If you record these subtypes, you can use SMF_TIME(YES) to synchronize SMF type 42 data with SMF and RMF data intervals.

Specifying SMF_TIME(YES) overrides the IGDSMSxx parameters CACHETIME and CF_TIME.

2.13 Function removals

This section describes the function that has been removed from DFSMSdfp V1.7

2.13.1 ISAM removal

All support for Indexed Sequential Access Method (ISAM) is removed with z/OS V1R7. ISAM data sets that have been converted to VSAM can still be accessed through the compatibility interface, but they must have been converted to VSAM data sets before accessing on a z/OS V1R7 system.

Diagnostic messages

In this section we list the new and changed messages in support of the ISAM removal:

`IEC134I` *`jjj,sss, ddn[,dsn]`* `Z/OS AS OF V1R7 DOES NOT ALLOW OPENING OF INDEXED SEQUENTIAL DATA SETS`

Explanation: The system can no longer open this data set because it is indexed sequential (ISAM). This message is followed by message `IEC143I and ABEND 213-1C`.

In the message text:

jjj	Indicates the job name.
sss	Indicates the step name.
dsn	Indicates the data set name.
ddn	The ddname.
[dsn]	The data set name. The dsn portion of this message will not be displayed when the associated ISAM data set is temporary.

`IEC143I 213-`*`rc`*`,`*`mod`*`,`*`jjj`*`,`*`sss`*`, `*`ddname`*`[-#],`*`dev`*`,`*`volser`*`,`*`dsname`*

Explanation: An error occurred during the processing of an OPEN macro instruction for a data set on a direct access device. This is probably a user error.

In the message text:

rc	Associates this message with system completion code 213 and with the return code.
mod	The name of the module in which the error was detected.
jjj	The job name.
sss	The step name.
ddname[-#]	The data definition (DD) name, followed by a concatenation number (#) if the DD statement is part of a concatenation and not the first DD statement in the concatenation.
dev	The device number.
volser	The volume serial number.
dsname	The data set name.

Explanations of the hexadecimal return codes follow:

Return	Code explanation
1C	An OPEN macro instruction was issued for an ISAM data set, and failed because ISAM is no longer supported. For OPEN to succeed, the data set must be converted to VSAM.

IEC614I *func* FAILED - RC *rc*, DIAGNOSTIC INFORMATION IS (*diaginfo*) *sss*, *ser*, *dsname*

Explanation: The specified DADSM function failed due to the error described by the function return code and the diagnostic information. The failure occurred in the step, operating on the volume, on the data set.

In the message text:

func	The function specified. In this case it will be CREATE.
rc	The return code.
diaginfo	The indicated information referring to diagnostics. In this case it will be 0409007E
sss	The step name.
ser	The volume serial number.
dsname	The data set name.

IEC614I is a general message issued when SMS is not active. For this specific case when it is to do with an attempt to create an ISAM dataset, the diagnostic return code is 0409007E.

IGD17039I ALLOCATION FAILED FOR DATA SET *dsname*, THE SYSTEM NO LONGER SUPPORTS CREATION OF INDEXED SEQUENTIAL DATA SETS.

Explanation: You tried to create an indexed sequential (ISAM) data set. SMS handled this request for a non-SMS volume. If SMS were not running, you would have seen message IEC614I. As of z/OS 1.7, the system no longer supports creation or opening of indexed sequential data sets.

In the message text:

dsname	The data set name.

System action: The request to create the data set fails.

Operator response: None.

Application programmer response: Use an earlier level of the system to convert the data set to another type and change the programs that use the data set. For example, convert the data set to VSAM and change the programs that use the data set to use the ISAM interface to a VSAM data set (See *DFSMS Using Data Sets,* SC26-7410) or use native VSAM.

This is a specific message relating to ISAM support removal issued when SMS is active.

2.13.2 JOBCAT and STEPCAT removal

z/OS V1R7 does not support the use of JOBCAT or STEPCAT statements in JCL, and does not support the use of the CATALOG parameter on IDCAMS DEFINE PAGESPACE unless RECATALOG is specified.

In z/OS V1R7 the use of JOBCAT and STEP JCL statements is no longer supported, and there is no option to allow them to be used. If an attempt is made to use either of these JCL statements, the job will fail

Messages

In lower-level systems the user could optionally specify that JOBCAT and STEPCAT statements were no longer supported. This was a transitional arrangement that allowed users to migrate away from this use.

If JCL is found that specifies JOBCAT and STEPCAT, the following message will be issued:

```
IEFC034I JOBCAT OR STEPCAT NOT PERMITTED
```

Explanation: A JOBCAT or STEPCAT statement was encountered in the current job but JOBCAT/STEPCAT statements are no longer permitted.

System action: The job fails.

Operator response: See the system programmer response.

Application programmer response: JOBCAT/STEPCAT support is no longer available. Edit the job to remove the need for JOBCAT/STEPCATs.

System programmer response: You should modify the job by removing the need for JOBCAT/STEPCAT.

2.13.3 CATALOG parameter conditional use in DEFINE PAGESPACE

The CATALOG parameter is only allowed on DEFINE PAGESPACE if the RECATALOG parameter is also specified.

If CATALOG is used without RECATALOG, the following messages will be issued:

```
IDC3171I CATALOG PARAMETER NOT ALLOWED ON DEFINE PAGESPACE UNLESS RECATALOG
SPECIFIED
```

Explanation: An attempt was made to perform a DEFINE PAGESPACE command with a CATALOG subparameter where RECATALOG was not specified. This is no longer allowed.

System action: The system ends the command with an error message and a condition code of 12.

Operator response: None.

Application programmer response: Remove the CATALOG parameter. When defining a new pagespace, the pagespace must be able to be found by the normal catalog search order. If an alias is needed to orient the new data set name to a specific user catalog, that alias should be defined before attempting the DEFINE PAGESPACE.

System programmer response: If the error recurs and the program is not in error, search problem reporting databases for a fix for this problem. If no fix exists, contact the IBM Support Center. Provide the JCL, the SYSOUT output for the job, and all printed output and output data sets related to the problem.

2.13.4 VSAM attributes removal reminder: Imbed and Replicate

IBM has announced that at some future stage support for VSAM data sets, including catalogs, that have the IMBED and REPLICATE attributes will be withdrawn.

The release when this will happen has not been announced, but the time will come. Catalogs are typically difficult to deal with, as they are usually in use.

One technique that has been used is to use IDCAMS REPRO MERGECAT to move selected entries from the old to the new catalog, then switch the alias pointers. This allowed the task of moving to a newly defined catalog to be managed in smaller units of work.

This process was only applicable to complete sets of data set names, and usually only applied to a single high level qualifier (HLQ), as only the FROMKEY keyword was available.

With the provision of the REPRO MERGECAT FROMKET/TOKEY enhancements (refer to 2.5, “REPRO MERGECAT FROMKEY/TOKEY enhancement” on page 36), and the use of the catalog multi-level alias feature, it is now possible to extend this technique to a subset of the data sets under a HLQ and one or more subsidiary qualifiers, and so allow manageable units of work to be defined.

3

OAM enhancements

This chapter describes the following enhancements to OAM in DFSMS V1.7:

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

3.1 Tape dispatcher delay

A new TAPEDISPATCHERDELAY parameter has been added to the SETOAM keyword in the CBROAMxx PARMLIB member to delay the processing of certain requests and minimize the demounting and remounting of tapes. The syntax of the new parameter is as follows:

```
TAPEDISPATCHERDELAY(nn)
```

This instructs OAM to wait for a specified number of seconds before demounting a tape volume, even if other work is available for the drive. The valid values for the delay are between 1 and 60 seconds. This delay allows time for a new read request to come into OAM that requires the currently mounted tape volume. If such a read request arrives within the delay interval, it will be dispatched immediately upon arrival. This should greatly reduce the amount of mounts and dismounts of tape volumes for applications that send multiple read requests for data on a given tape volume. If no read request for the currently mounted volume arrives within the delay interval, another request for a different tape volume is dispatched.

The OAM tape dispatcher will only delay processing of a unit of work for a specific period of time when all of the following conditions are met:

- A non-zero tape dispatcher delay value is specified with the TAPEDISPATCHERDELAY keyword on the SETOAM statement in the CBROAMxx PARMLIB member.
- A read request for an object on a currently mounted tape volume has just been completed.
- There is no request for the currently mounted tape volume waiting to be processed on the OAM tape dispatcher queue.
- The OAM tape dispatcher has found a request for another tape volume and is about to dispatch this unit of work.

3.2 Immediate recall to DB2

OAM allows objects to move within the storage hierarchy. However, this does not happen until the OAM Storage Management Component (OSMC) cycle is run for the storage group where an object resides. In many cases, when an object has been referenced, the chances that it will be referenced again are very high. Therefore, an object that has been written to removable media could be repeatedly referenced while on this media, which has significantly longer read response times than DASD.

Immediate recall to DB2® addresses the issue by immediately recalling objects currently residing on removable media to a DB2 table residing on DASD for a specified number of days. This allows subsequent requests to read the objects to be satisfied from DB2 DASD rather than another read from the tape or optical volume where the object was stored.

When an object is recalled, a full copy of the object is written to DB2 DASD, even if the OSREQ RETRIEVE request is for a partial object. The object's location field is updated with a new location value of R to indicate that the object has been recalled to DB2 DASD.

The object's pending action date is set to the current date plus the number of days the object is to be recalled for. Once the pending action date has elapsed, OSMC deletes the object from DB2 DASD and restores the object directory to point back to the original optical or tape location. The ACS routines are not invoked when an object is recalled to DB2 DASD, nor when the recalled object is subsequently processed by OSMC and restored to its original location on removable media.

An OSREQ CHANGE request for an object that has been recalled to DB2 DASD results in the pending action date being set to the current date. Therefore, the object will be restored back to its pre-recalled state on the next OSMC cycle. Since no other OSMC processing takes place at that time, the expected transition based on the OSREQ CHANGE will not occur. When it is restored its original location, an object's pending action date is set to a value based on the current management class assigned to that object.

3.2.1 Using OAM with DB2

This section provides tips on how to configure OAM to work with DB2 during the initial installation.

APF authorized libraries

The OSREQ macro requires that the DSNALI module, or DSNCLI in the case where OAM is running under CICS, be in an APF authorized library. Additionally, the libraries containing these modules must either be included in the LNKLST concatenation, or in STEPLIB in the OAM and OTIS procedures, and in any jobs that use OSREQ to store and access objects. If any of these modules do not reside in an APF authorized library, message CBR0401I will be issued with return code 16 and reason code D8010000 when an attempt is made to store or access an object.

In order to ensure that OAM is loading these modules from the intended library, issue the SETPROG LNKLST command as follows to test where these modules are located in the current LNKLST set:

```
SETPROG LNKLST,TEST,NAME=CURRENT,MODNAME=DSNALI
```

The list of APF authorized data sets can be displayed using the following command:

```
DISPLAY PROG,LNKLST
```

For more information about commands to APF authorize data sets and update the LNKLST concatenations, refer to *MVS System Commands,* SA22-7627.

z/OS V1R8 coexistence PTF

As of z/OS V1R8, OAM will be enhanced to use DB2's large object (LOB) support. z/OS V1R7 systems will not be able to use this support, but coexistence PTFs will be provided by APAR OA12683 for lower-level systems. If this PTF is installed on a pre-z/OS V1R8 system, migration job CBRSMR18, provided in SYS1.SAMPLIB, must be edited and submitted to add a new column, ODLOBFL, to the object directory to indicate whether a DASD resident object resides in a LOB storage structure.

During the installation process, CBRSMR18 should be submitted after creating the DB2 databases and tables, but before creating and binding the DB2 packages. If this job is not submitted, job CBRPBIND will fail with SQL code -206 as part of the installation process, and OSMC processing will fail once OAM is installed.

Refer to 8.5.1, “Binary large object support” on page 319, for more information about OAM's LOB support in z/OS V1R8.

Installation verification program

The OSREQ TSO command processor closely resembles the OSREQ macro. It is used to exercise the OSREQ macro interface and OAM, without having to explicitly use the OSREQ ACCESS and UNACCESS macros to connect and disconnect the macro to OAM. The purpose of the OSREQ TSO command processor is to verify object support after OAM installation. It cannot be used to store real data. For more information about the OSREQ TSO command processor, refer to *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426.

A sample installation verification program, CBRSAMIV, is provided in SYS1.SAMPLIB. This job uses the OSREQ TSO command processor to store, change, and retrieve dummy objects. The object and collection names in the job can be modified to meet your naming conventions, and additional parameters can be specified on the command to verify immediate recall to DB2. The new parameters on the OSREQ TSO command to support immediate recall to DB2 are introduced in “New OSREQ TSO command parameters” on page 102.

3.2.2 Activating immediate recall to DB2

Immediate recall to DB2 is activated in one of the following ways:

- Implicitly, using the SETOSMC statement in the CBROAMxx parmlib member
- Explicitly, by specifying the RECALL keyword on the OSREQ RETRIEVE request

If recall is required, OAM initiates a request to OSMC to write a full copy of the object to DB2 DASD at the same time as the OSR component is servicing the object retrieval. Object recalls are processed asynchronously from the retrieve request in order to avoid delaying the retrieve request while the recall processing occurs. This may result in subsequent retrieve requests for the same object that are issued prior to recall processing completing being satisfied from the tape or optical volume rather than DASD. In most cases, the removable media cartridge would still be mounted on the drive, reducing any delays on subsequent retrievals. To increase the chances of the volume still being mounted, refer to the new tape dispatcher delay function introduced earlier in 3.1, “Tape dispatcher delay” on page 96.

New SETOSMC statement keywords

Immediate recall parameters can be specified in the SETOSMC statement in the CBROAMxx PARMLIB member. Figure 3-1 shows the syntax of the new keywords that have been added for this purpose.

```
>>__SETOSMC__ _____________________ __ ___________________ __________>
             |_MAXRECALLTASKS(nnn)_|  |_RECALLOPTICAL(nnn)_|

>__ _________________ __ ________________ __ ____________ ___________>
   |_RECALLTAPE(nnn)_|  |_RECALLALL(nnn)_|  |_RECALLNONE_|

>__ ____________________________________________ ___________________>
   |_STORAGEGROUP(See STORAGEGROUP Subparameters)_|


>>__STORAGEGROUP(objsg__ ___________________ __ ________________ ___>
                        |_RECALLOPTICAL(nnn)_|  |_RECALLTAPE(nnn)_|

>__ ________________ __ ____________ __ ___________________ __ )____><
   |_RECALLALL(nnn)_|  |_RECALLNONE_|  |_RECALLOFF( _ON__ )_|
                                                  |_OFF_|
```

Figure 3-1 New SETOSMC keywords for immediate recall

The keywords are:

- RECALLOPTICAL(*nnn*)

  This keyword specifies the number of days that objects residing on optical devices are recalled to DB2 DASD for before being restored to their original location. An object's pending action date is set to the current date plus the number of days specified. Valid values are between 0 and 255, with 0 being the default. A value of 0 indicates that the recall is for the current day only. This keyword can be specified at the global or storage group level, and applies to implicit recalls only.

- RECALLTAPE(*nnn*)

  This keyword specifies the number of days that objects residing on tape devices are recalled to DB2 DASD for before being restored to their original location. An object's pending action date is set to the current date plus the number of days specified. Valid values are between 0 and 255, with 0 being the default. A value of 0 indicates that the recall is for the current day only. This keyword can be specified at the global or storage group level, and applies to implicit recalls only.

- RECALLALL(*nnn*)

  This keyword specifies the number of days that objects residing on either optical or tape devices are recalled to DB2 DASD for before being restored to their original location. An object's pending action date is set to the current date plus the number of days specified. Valid values are between 0 and 255, with 0 being the default. A value of 0 indicates that the recall is for the current day only. This keyword can be specified at the global or storage group level, and applies to implicit recalls only.

- RECALLNONE

  This keyword specifies that objects residing on optical or tape devices should not be recalled to DB2 when retrieved. This can be specified at the global or storage group level, and applies to implicit recalls only.

- RECALLOFF(ON | OFF)

  This keyword specifies whether objects residing on optical or tape devices should be recalled to DB2 when retrieved. A value of OFF is the default and indicates that recalls are enabled. A value of ON indicates that recalls are disabled. This keyword can only be specified at the storage group level, and allows recall processing to be disabled at the storage group level regardless of whether the RECALL parameter is specified on the OSREQ RETRIEVE request.

- MAXRECALLTASKS(*nnn*)

  This keyword specifies the maximum number of recall tasks that can be run concurrently. Valid values are between 0 and 255, with 0 being the default. A value of 0 indicates that no recall operations are to be run. This keyword can only be specified at the global level, and applies to both implicit and explicit recalls.

Tip: Because OSMC recall tasks can consume considerable resources, we recommend that the MAXRECALLTASKS keyword on the SETOSMC statement is used to limit the number of recall tasks that can run simultaneously. Note that MAXRECALLTASKS must be set to a non-zero value to activate recall to DB2.

Figure 3-2 shows an example of a SETOSMC statement that limits the maximum number of concurrent recall tasks to two, sets the amount of time that objects recalled from optical or tape devices should be kept in DB2 to 15 days, and disables object recalls for the storage group MHLOBJ0.

```
SETOSMC MAXRECALLTASKS(2)
        RECALLALL(15)
        STORAGEGROUP(MHLOBJ0 RECALLOFF(ON))
```

Figure 3-2 Enabling immediate recall with the SETOSMC statement

These parameters can also be changed dynamically using the MODIFY OAM,UPDATE command. The keywords used to update the parameters using MODIFY OAM command are different than those used in the SETOSMC statement. Table 3-1 shows the valid MODIFY OAM keywords and their associated SETOSMC keywords.

Table 3-1 Valid MODIFY OAM,UPDATE keywords and their associated SETOSMC keywords

MODIFY OAM,UPDATE keyword	SETOSMC keyword
RECALLA	RECALLALL
RECALLO	RECALLOPTICAL
RECALLT	RECALLTAPE
RECALLN	RECALLNONE
RECALLF	RECALLOFF
MAXRECAL	MAXRECALLTASKS

To set a parameter at the global level, the command should be issued as follows:

`F OAM,UPDATE,SETOSMC,ALL,`*`keyword`*`,`*`value`*

To set a parameter at the storage group level, the command should be issued as follows:

`F OAM,UPDATE,SETOSMC,`*`storagegroup`*`,`*`keyword`*`,`*`value`*

Figure 3-3 shows an example of a command to update the maximum number of concurrent recall tasks, and the message issued by OAM when the command is executed.

```
F OAM,UPDATE,SETOSMC,ALL,MAXRECAL,5
CBR1074I Update successful for SETOSMC parameter MAXRECAL, new value
5, scope ALL. The previous value was 2.
```

Figure 3-3 Updating the number of maximum concurrent recall tasks

The current values of the immediate recall parameters at the global or storage group level can also be displayed by issuing the MODIFY OAM,DISPLAY command as follows:

```
F OAM,DISPLAY,SETOSMC,ALL
F OAM,DISPLAY,SETOSMC,storagegroup
```

Figure 3-4 shows an example of the output from this command.

```
F OAM,DISPLAY,SETOSMC,ALL
CBR1075I GLOBAL value for BACKUP1 is MHLOBJ1
CBR1075I GLOBAL value for BACKUP2 is MHLOBJ2
CBR1075I GLOBAL value for CYCLEW is STRTONLY
CBR1075I GLOBAL value for MAXRECAL is 5
CBR1075I GLOBAL value for RECALLO is 15, ON
CBR1075I GLOBAL value for RECALLT is 15, ON
CBR1075I GLOBAL value for CLEAROLD is NONE
```

Figure 3-4 Displaying the current immediate recall parameters

New OSREQ macro keywords

Two new optional keywords, RECALL and RETCODE2, have been added to the OSREQ RETRIEVE function to allow applications to recall objects residing on tape or optical media to DB2 DASD. The syntax of these keywords is shown in Figure 3-5.

```
>__ ___________________________________ _________________________>
   |_RECALL=_ _number_days_word__________ __|
             |_(number_days_word_pointer)_|

>__ _______________________________________ _____________________>
   |_RETCODE2=_ _return_code2_word__________ __|
               |_(return_code2_word_pointer)_|
```

Figure 3-5 New OSREQ RETRIEVE keywords

When the RECALL keyword is specified, the object being retrieved will be recalled to DB2 DASD for the number of days specified. *number_days_word* specifies a full word that indicates how many days a recalled object is to remain on DB2 before OSMC moves it back to its original location. The valid number of days that can be specified is between 0 and 255.

The RECALL keyword overrides any RECALLOPTICAL, RECALLTAPE, RECALLALL, or RECALLNONE settings defined on the SETOSMC statement in the CBROAMxx member of PARMLIB. However, the RECALL keyword is ignored if either MAXRECALLTASKS(0) or RECALLOFF(ON) is specified on the SETOSMC statement in the CBROAMxx member of PARMLIB.

The RETCODE2 keyword provides information about whether the retrieve request resulted in the object being scheduled for recall to DB2 DASD. Note that this is only the return code from the operation of scheduling the object recall, rather than an indication of the success of the recall operation itself. The return code is only valid if the retrieve request completed successfully. Table 3-2 shows the meaning of the possible return codes.

Table 3-2 Return codes for scheduling object recall

RETCODE2 value	Description
0	Either the object recall was successfully scheduled or the RECALL keyword was not specified, so no attempt was made to schedule a recall.
4	The RECALL keyword was not specified, but the object recall was successfully scheduled due to CBROAMxx PARMLIB member specifications.

RETCODE2 value	Description
8	An attempt to schedule a recall for the object was unsuccessful because OSMC=NO is specified on the OAM started procedure.
10	An attempt to schedule a recall for the object was unsuccessful because MAXRECALLTASKS(0) is specified in the CBROAMxx PARMLIB member.
12	An attempt to schedule a recall for the object was unsuccessful because RECALLOFF(ON) is specified in the CBROAMxx PARMLIB member.
14	An attempt to schedule a recall for the object was unsuccessful due to an unexpected scheduling error.
16	An attempt to schedule a recall for the object was unsuccessful because the retrieve was performed on an OAMplex member that does not support recall processing.

New OSREQ TSO command parameters

The OSREQ TSO command processor has been enhanced to accommodate the new OSREQ macro RECALL and RETCODE2 parameters when the retrieve function is invoked. The RECALL parameter can now be specified on the OSREQ TSO command as follows, and has the same effect as specifying the RECALL parameter on the OSREQ macro. If you wish to see the return code from scheduling the object recall to DB2 DASD, the new RETCODE2 parameter must be specified on the command:

`OSREQ RETRIEVE` *`collectionname objectname`* `RECALL(`*`days`*`) RETCODE2`

Figure 3-6 shows a sample command to retrieve and recall an object that currently resides on tape, and the output produced by OAM.

```
 OSREQ RETRIEVE MHLRES2.COLLECT MHLRES2.OBJECT RECALL(10) RETCODE2
CBR0400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ QUERY response time is 22 milliseconds.
CBR0440I OSREQ RETRIEVE successful. Return code = 00000000, reason code =
00000000, retcode2 = 00000000.
CBR0404I OSREQ RETRIEVE response time is 80824 milliseconds.
CBR0405I OSREQ RETRIEVE data rate is 11 kilobytes/second.
CBR0420I Data comparison for object MHLRES2.COLLECT MHLRES2.OBJECT successful.
```

Figure 3-6 OSREQ RETRIEVE command and output for object residing on tape

Figure 3-7, "OSREQ RETRIEVE command and output for recalled object" on page 103 shows another OSREQ command to retrieve the same object, issued a few minutes later, and the messages produced by OAM. Note that since the object was recalled to DB2 DASD as a result of issuing the command in Figure 3-6 on page 102, the response time is significantly lower and data rate significantly higher since the object is now being retrieved from DASD rather than tape.

```
 OSREQ RETRIEVE MHLRES2.COLLECT MHLRES2.OBJECT
CBR0400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ QUERY response time is 25 milliseconds.
CBR0400I OSREQ RETRIEVE successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ RETRIEVE response time is 18 milliseconds.
CBR0405I OSREQ RETRIEVE data rate is 56888 kilobytes/second.
CBR0420I Data comparison for object MHLRES2.COLLECT MHLRES2.OBJECT successful.
```

Figure 3-7 OSREQ RETRIEVE command and output for recalled object

3.2.3 SMF records

OAM SMF type 85 records have been enhanced to reflect information regarding immediate recall to DB2. A new subtype 38 record has been added to capture information specifically relating to the recall to DB2 DASD function, and OSREQ retrieve (subtype 3) and storage group processing (subtype 32) records have been updated. Refer to *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426, for more information about the SMF type 85 records.

3.2.4 Migration considerations

The OSREQ macro has grown from 96 bytes to 120 bytes in z/OS V1R7 OAM. As a result, applications that obtain storage explicitly for the OSREQ parameter list rather than using the list macro form (MF=L) of the OSREQ macro must ensure that they obtain a minimum of 120 bytes. Applications that use the list form must be re-assembled, and will automatically acquire the 120-byte parameter list in a modifiable area of storage in the caller's key.

3.2.5 Coexistence considerations

Coexistence PTFs are required on lower-level systems in an OAMplex that contains a system at z/OS V1R7. Systems with the coexistence PTFs installed will be able to recognize the new location value for objects that have been recalled to DB2, and will be able to retrieve the object from DB2 rather than from removable media. Objects that have been recalled to DB2 can be moved back to their original location on removable media during the OSMC cycle either on a z/OS V1R7 system or on a lower-level system. The RECALL keyword on the OSREQ macro will be ignored on pre-z/OS V1R7 systems.

Table 3-3 Coexistence PTFs for immediate recall to DB2

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08230	UA16993	UA16995	UA16996

3.3 Clear old location option

When an object is moved from an optical or tape volume to DB2 DASD during an OSMC cycle, the volser and sector or blockid location is retained in the object directory. This is because the transition to DB2 is usually temporary and the object will be moved back to optical or tape media at some point. While an optical or tape volume is associated with an object, that volume cannot be expired, even if the object is currently residing on DB2 DASD.

A new optional keyword, CLEAROLDLOC, has been added to the SETOSMC statement to instruct OAM to clear the original volser and sector or blockid location in the object directory for a given object when the object is moved by OSMC to DB2 DASD. This keyword can be specified at the global or storage group level. It should be used when it is known that objects that are moved from tape or optical devices to DB2 DASD are not usually moved back to removable media.

Old location values are only cleared when object transitions to DB2 DASD occur during an OSMC cycle. The old location values will not be cleared for an object that is recalled to DB2, regardless of whether the CLEAROLDLOC keyword is specified.

The syntax of the new keyword is as follows:

```
CLEAROLDLOC(OPT | TAPE | BOTH | NONE)
```

OPT This option specifies that an object's old location information in the object directory should be cleared when it is moved from optical media to DB2 DASD.

TAPE This option specifies that an object's old location information in the object directory should be cleared when it is moved from tape media to DB2 DASD.

BOTH This option specifies that an object's old location information in the object directory should be cleared when it is moved from either optical or tape media to DB2 DASD.

NONE This is the default. It specifies that an object's old location information should be left unchanged in the object directory when it is moved from optical or tape media to DB2 DASD.

The clear old location option may also be specified either globally or at the storage group level using the MODIFY OAM,UPDATE command, as follows:

```
F OAM,UPDATE,SETOSMC,ALL,CLEAROLD,mode
F OAM,UPDATE,SETOSMC,storagegroup,CLEAROLD,mode
```

Figure 3-8 shows an example of the command to specify at the global level that the old location of objects recalled from either optical or tape media should be cleared from the object directory, and the message issued by OAM when the command is executed.

```
F OAM,UPDATE,SETOSMC,ALL,CLEAROLD,BOTH
CBR1074I Update successful for SETOSMC parameter CLEAROLD, new value
BOTH, scope ALL. The previous value was NONE.
```

Figure 3-8 Setting the clear old location option

The current value of the clear old location option can be displayed either globally or at the storage group level using the MODIFY OAM,DISPLAY command, as follows:

```
F OAM,DISPLAY,SETOSMC,ALL
F OAM,DISPLAY,SETOSMC,storagegroup
```

3.4 Return to MVS scratch exit

OAM currently issues message CBR2165I to indicate that all knowledge of a given tape volume has been removed from OAM's tape volume inventory. Tape management systems, such as DFSMSrmm, can intercept this message as a signal to return the tape volume to MVS™ scratch status.

With z/OS V1R7, OAM provides the *OAM tape volume return to MVS scratch* dynamic exit point, CBRUXTVS_EXIT, which can be used to notify tape management systems that OAM has deleted an object tape from its database. The exit is invoked after OAM issues the CBR2165I message indicating that OAM has removed the tape volume from its inventory and released it for return to the MVS scratch pool. This is a notification-only exit in that the return code from the exit module has no effect on OAM's tape expiration processing.

3.4.1 Adding an exit routine to CBRUXTVS_EXIT

There is no need to provide an exit for this exit point if DFSMSrmm is used as the tape management system since the DFSMSrmm tape exit, EDGTVEXT, is always invoked directly by OAM whenever an OAM tape volume is returned to MVS scratch status. However, if any tape management system other than DFSMSrmm is used, the module name of the installation exit routine provided by the tape management system will need to be added to the new OAM dynamic exit point CBRUXTVS_EXIT.

The CBRUXTVS_EXIT exit point is implemented using the dynamic exit services, which allows multiple exit routines to be simultaneously associated with a single exit point. Exit routines can be managed in any of the following ways:

- The EXIT statement of the PROGxx PARMLIB member allows you to add exit routines to an exit, change the state of an exit routine, delete an exit routine for an exit, undefine an implicitly defined exit, and change the attributes of an exit. Refer to *MVS Initialization and Tuning Reference,* SA22-7592, for more information about the PROGxx PARMLIB member.
- The SETPROG EXIT operator command performs the same functions as the EXIT statement of the PROGxx parmlib member. Refer to *MVS System Commands,* SA22-7627, for more information about this command.
- The CSVDYNEX macro can be used to define exits to the dynamic exits facility, control their use within a program, and associate one or more exit routines with those exits. It can also be used to associate exit routines with the existing SMF and allocation exits, which have been defined to the dynamic exits facility. The CSVDYNEX macro provides a super-set of the functions available through the SETPROG EXIT operator command and the EXIT statement of the PROGxx parmlib member. Refer to *MVS Programming Authorized Assembler Services Reference Vol 1 (ALESERV-DYNALLOC)*, SA22-7609, for more information about the CSVDYNEX macro.

Figure 3-9 shows an example of the DFSMShsm ARCTVEXT module being added to the CBRUXTVS_EXIT exit using the EXIT command in the PROGxx PARMLIB member.

```
EXIT ADD EXITNAME(CBRUXTVX_EXIT) MODNAME(ARCTVEXT) STATE(ACTIVE)
```

Figure 3-9 Adding an exit routine to the CBRUXTVX_EXIT exit

3.4.2 Dynamically modifying the exit routines

If the exit modules associated with the CBRTUXTVS_EXIT exit need to be updated while OAM is running, we recommend that the following procedure be followed:

1. Change and recompile the exit module.
2. Update the library resident copy of the exit module, such as SYS1.LINKLIB.
3. Issue the LLA refresh command for the exit module.
4. Modify the current copy of the exit module to be inactive using the SETPROG command, as follows:

   `SETPROG EXIT,MODIFY,EXITNAME=CBRUXTVS_EXIT,MODNAME=modulename,STATE=INACTIVE`

5. Modify the new copy of the exit module to be active using the SETPROG command, as follows:

   `SETPROG EXIT,MODIFY,EXITNAME=CBRUXTVS_EXIT,MODNAME=modulename,STATE=ACTIVE`

Note: OAM continues its tape expiration processing regardless of the state of the exit. Therefore, in order to ensure that the tape management system is informed of any volumes released by OAM, we highly recommend that you quiesce any OAM processes that could release tape volumes while the state of the exit routines are being modified. OAM processes that release tape volumes include OSMC processing of object and object backup storage groups, MOVEVOL utilities with the DELETE or RECYCLE options specified, and RECOVERY utilities with the DELETE option specified.

3.4.3 Displaying associated exit routines

The DISPLAY PROG,EXIT command can be used as follows to display the exit routines that have been associated with the OAM return to MVS scratch dynamic exit:

`DISPLAY PROG,EXIT,EXITNAME=CBRUXTVS_EXIT`

Figure 3-10 shows the output from this command when the DFSMShsm ARCTVEXT module has been associated with this exit.

```
CSV461I 14.01.07 PROG,EXIT DISPLAY 248
EXIT             MODULE   STATE MODULE   STATE MODULE   STATE
CBRUXTVS_EXIT    ARCTVEXT   A
```

Figure 3-10 Displaying modules associated with the CBRUXTVS_EXIT exit

3.4.4 Writing the exit routine

The OAM return to MVS scratch installation exit allows the tape management system to accurately reflect OAM's object tape usage. When called, the exit routine should use the tape management system facilities to notify itself that OAM has deleted the tape from its database.

After the exit sends the notification to the tape management system, it returns control to OAM.

Parameter list

On entry to the exit routine, register 1 contains the address of the parameter list. The parameter list contains the information shown in Table 3-4.

Table 3-4 CBRUXTVS_EXIT parameter list

Offset	Length	Description
0	4	The address of an 8-byte data area, where the first six bytes contain the volume serial number of the volume having expired, followed by a 2-byte field.
4	4	The address of a full word binary return code.

Data area

The parameter list contains the address of the 8-byte data area described in Table 3-5. None of the information in the data area may be modified.

Table 3-5 CBRUXTVS_EXIT data area

Offset	Length or bit pattern	Description
0	6	The volume serial number of the expired tape volume
6	1 1... .... .1..	Flag byte one The tape volume is being purged Data sets on the tape volume are expiration date protected
7	1	Flag byte two

Return code

The exit routine may indicate return codes, as shown in Table 3-6. Note that OAM does not interrogate this return code. A non-zero return code has no effect on OAM's expiration processing of tape volumes. No error message is issued by OAM for a non-zero return code, but an error message is issued for an abend in the exit module.

Table 3-6 CBRUXTVS return codes

Return code	Meaning
Zero	The exit routine completed successfully.
Non-zero	The exit routine failed.

3.5 Enhanced MOVEVOL utility

OAM provides the move volume utility (MOVEVOL) to allow objects to be moved from a primary or backup source volume to one or more target volume. MOVEVOL can also recycle or delete tape and optical volumes in the OAM inventory after all objects have been successfully moved off the volume.

Currently, this utility can only process volumes belonging to either a primary or a backup object storage group. With z/OS V1R7, MOVEVOL can also process OAM scratch volumes. This allows scratch volumes to be removed from OAM's inventory.

There are no changes to how the MOVEVOL command is issued. To delete a scratch volume from OAM's inventory, issue the following command:

`F OAM,START,MOVEVOL,`*`volser`*`,DELETE`

Figure 3-11 shows the messages issued by OAM when processing this command.

```
F OAM,START,MOVEVOL,TST016,DELETE
CBR1000I OAM START command execution scheduled.
CBR9800I OAM Move Volume Delete starting for volumes TST016 and N/A.
CBR9883I MOVEVOL volume TST016 is a scratch volume.
CBR9858I Move Volume Utility status for volume TST016. Total: 0,
Attempted: 0, Successful: 0, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volumes TST016 and N/A.
CBR2165I Tape volume TST016 has had all objects expired or deleted
and can be returned to the MVS scratch pool.
```

Figure 3-11 Deleting a scratch tape from OAM

The RECYCLE option will be ignored for MOVEVOL commands that specify scratch volumes.

4

DFSMShsm enhancements

This chapter describes the following enhancements to DFSMShsm:

- 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 wildcards with HMIGRATE.
- Saving LRECL of migrated data sets in the MCD.
- New recycle processing option for connected sets.
- AUDIT MEDIACONTROLS resume.
- Dump encryption support.

4.1 Support for large format data sets

DFSMShsm has been enhanced to support the new large format sequential data sets that are introduced in 2.1, “Large format data sets” on page 12.

Large format data sets can have greater than 65,535 tracks per volume. Before z/OS V1R7, most sequential data sets were limited to 65,535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. Large format data sets are not required to have greater than 65,535 tracks at initial allocation, but once allocated in large format they can expand to greater than 65,535 tracks.

z/OS V1R7 also introduces the concept of a basic format data set to designate a sequential data set that is neither extended format nor large format. A basic format data set therefore may not have any more than 65,535 tracks per volume.

Large format data sets are supported by DFSMShsm in the same manner as basic format data sets in typical operations involving:

- Migration and recall
- Backup and recovery
- ABACKUP and ARECOVER

4.1.1 Large format journal data set

DFSMShsm also supports the use of a large format data set for its journal data set. A larger journal data set can allow more DFSMShsm activity to take place between journal backups and helps to minimize occurrences of journal full conditions.

Migrating to a large format journal data set

Once all systems in the HSMplex are at z/OS V1R7, the following procedure can be followed to migrate the journal data set to a large format data set:

1. Shut down all but one of the DFSMShsm subsystems in the HSMplex.
2. Either hold all DFSMShsm functions or set DFSMShsm to emergency mode.
3. Back up the control data sets using the BACKVOL CDS command. This creates a backup of the journal and then nulls the journal.
4. Stop the remaining DFSMShsm subsystem.
5. Rename the current journal data set.
6. Allocate the new large format journal data set, as shown in Figure 4-1 on page 111.
7. Ensure that the JOURNAL DD statement in the DFSMShsm startup procedure points to the new journal data set.
8. Restart the DFSMShsm hosts.

Figure 4-1 is an example of JCL to create a large format DFSMShsm journal data set.

```
//MHLRES2C JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
//S1      EXEC PGM=IEFBR14
//JRNL     DD  DSN=HSM.JRNL,DISP=(NEW,CATLG),VOL=SER=SBOXOA,
//             UNIT=3390,SPACE=(TRK,(70000),,CONTIG),
//             DSNTYPE=LARGE
```

Figure 4-1 Defining a large format HSM journal data set

Due to space restrictions, it may be impractical to rename the old journal data set and allocate the new journal. In this case, as an alternative to steps 5 and 6 you can delete the existing journal and then use IEBGENER to define the new journal data set, as shown in Figure 4-2.

```
//MHLRES2G JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//S1      EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DUMMY,RECFM=U,BLKSIZE=6556
//SYSUT2   DD  DSN=HSM.JRNL,DISP=(NEW,CATLG),VOL=SER=SBOXOA,
//             UNIT=3390,SPACE=(TRK,(70000),,CONTIG),
//             DSNTYPE=LARGE,RECFM=U,BLKSIZE=6556
//SYSIN    DD  DUMMY
```

Figure 4-2 Defining a large format HSM journal data set with IEBGENER

Should you need to fall back to using a basic format journal data set, the same procedure as above can be used to replace the current large format data set with a new basic format data set.

Note that if the used space in the journal grows as a consequence of the data set being allocated with a larger size, journal backups will take longer to complete due to DFSMShsm having to process a larger data set. This will increase the length of time that DFSMShsm functions are held during CDS backup.

Tip: Increasing the size of the journal may cause the journal backup to fail as the space allocated to its backup copies could no longer be enough. To prevent this from happening, rename your current journal backup data sets, and create new backup data sets of sufficient size. The backup data sets may be defined as large format data sets.

4.1.2 Coexistence considerations

A coexistence PTF is required so that lower-level releases of DFSMShsm will fail operations involving large format data sets.

Since large format data sets cannot be processed on hosts prior to z/OS V1R7, care must be taken when migrating data sets in a HSMplex where not all hosts are at this level. Any attempts to recall these data sets from a lower-level system will fail, which is especially problematic for implicit recalls. Similarly, attempts to recover large format data sets will fail on pre-z/OS V1R7 hosts, and large format data sets included in an ABARS aggregate can only be restored using ARECOVER if the recovery system is at z/OS V1R7.

In order to use a large format journal data set, all systems in the HSMplex must be at z/OS V1R7. The toleration PTF ensures that when a lower-level release of DFSMShsm attempts to open a large format journal data set, message ARC0509E is issued and DFSMShsm is stopped.

Table 4-1 Coexistence PTFs for HSM support of large format data sets

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08865	UA16954	UA16955	UA16956

4.2 Fast subsequent migration improvements

Fast subsequent migration (FSM) is the process that allows data sets that have been recalled from ML2 tape to be subsequently reconnected to the ML2 tape when the data set becomes eligible for migration again, as long as the data set has not been modified since being recalled.

Systems that have implemented tape mount management (TMM) have found that there are a large number of data sets on ML2 tapes that are not eligible for FSM, even though they were not modified after being recalled. The reason for this is that most TMM candidate data sets have a management class that specifies that the data sets do not require backup, and since the data sets have not been backed up, the change bit in the DSCB does not get turned off by the backup process. Prior to z/OS V1R7, DFSMShsm considered a data set to have been modified after being recalled if its change bit was on, therefore preventing the data set from being reconnected to the original ML2 tape.

To increase the number of migrated data sets eligible for fast subsequent migration, the function has been changed to use new indicators for reconnection eligibility. This enhanced implementation can increase the use of fast subsequent migration in the following ways:

- It allows fast subsequent migration to be used even if a product other than DFSMShsm is used to back up data sets.
- It allows DFSMShsm to reconnect data sets originally migrated to ML2 tape without a valid backup copy (such as TMM data). In previous releases, these data sets were not eligible for reconnection.

These fast subsequent migration improvements will apply automatically to all z/OS V1R7 systems. There is no need to make any configuration changes to DFSMShsm in order to use this enhancement.

Note: Data sets recalled prior to the implementation of z/OS V1R7 will not have their reconnection criteria changed.

4.2.1 Coexistence considerations

Coexistence PTFs are provided for lower-level systems. This will allow data sets recalled on a pre z/OS V1R7 system to be eligible for reconnection using the new criteria if the migration is performed on a z/OS V1R7 system.

Data sets recalled prior to the installation of the coexistence PTFs will be eligible for reconnection under the original criteria. See Table 4-2.

Table 4-2 Coexistence PTFs for HSM fast subsequent migration support

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08848	UA16944	UA16945	UA16946
OA09096	UA16860	UA16861	UA16862
OA08858	UA16849	UA16850	UA16851

4.3 Extended tape table of contents (TTOC)

Prior to DFSMS V1.7, DFSMShsm can write a maximum of 330,000 data set entries per tape volume. This limits the number of data sets that an installation can put on a tape and can prevent full utilization of high-capacity tapes. To enable better use of high-capacity tape volumes, the maximum number of data set entries that DFSMShsm can write per tape volume has been increased to 1,060,000. Note that this only affects newly created TTOC entries.

4.3.1 Migrating to extended TTOC

In order to use extended TTOCs, the offline control data set (OCDS) must be defined with a maximum record size of 6144. The following migration plan may be used to redefine the OCDS:

1. Shut down all but one of the DFSMShsm subsystems in the HSMplex.
2. Either hold all DFSMShsm functions or set DFSMShsm to emergency mode.
3. Back up the existing OCDS using the BACKVOL CDS command.
4. Define a new OCDS with a maximum record size of 6144, as shown in Figure 4-3 on page 114.
5. Stop the remaining DFSMShsm subsystem.
6. Use the REPRO command to copy the old OCDS into the newly defined OCDS, as shown in Figure 4-4 on page 114.
7. Ensure that the OFFCAT DD statement in the DFSMShsm startup procedure points to the new OCDS name.
8. Restart the DFSMShsm subsystems.

Tip: In order to ensure that the new OCDS you create will be large enough to contain the extended TTOC records for each tape volume managed by DFSMShsm, refer to the guidelines on the optimum OCDS size in the *DFSMShsm Implementation and Customization Guide,* SC35-0418, before deciding on the size of your new OCDS.

When the OCDS is defined with a record size of 6144, issue the new SETSYS EXTENDEDTTOC(Y) command to enable extended TTOCs. To have extended TTOCs enabled at DFSMShsm startup, add this command to your ARCCMDxx member. The SETSYS EXTENDEDTTOC(Y) command will fail if it is issued before the OCDS has been defined with a record size of 6144. See Figure 4-3.

```
//MHLRES2A JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//IDCAMS  EXEC PGM=IDCAMS
//HSMOCDS  DD  UNIT=3390,VOL=SER=SBXHS4,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  DEFINE CLUSTER(NAME(HSM.OCDS.EXTENDED) VOLUMES(SBXHS4) -
         CYLINDERS(100) FILE(HSMOCDS) -
         RECORDSIZE(1080 6144) FREESPACE(0 0) -
         INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
         SPEED BUFFERSPACE(530432) -
         UNIQUE NOWRITECHECK) -
         DATA(NAME(HSM.OCDS.EXTENDED.DATA) -
           CONTROLINTERVALSIZE(12288)) -
         INDEX(NAME(HSM.OCDS.EXTENDED.INDEX) -
           CONTROLINTERVALSIZE(2048))
/*
```

Figure 4-3 Defining a new OCDS with a maximum record length of 6144

```
//MHLRES2B JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  REPRO -
    INDATASET(HSM.OCDS) -
    OUTDATASET(HSM.OCDS.EXTENDED)
/*
```

Figure 4-4 Using REPRO to copy the contents of the OCDS

Should you need to fall back to writing a maximum of 330,000 data sets per tape, issue the SETSYS EXTENDEDTTOC(N) command. Note that this will only affect newly created TTOC entries.

4.3.2 Verification of extended TTOC

You can verify that extended TTOCs are enabled by issuing the QUERY SETSYS command. Figure 4-5 shows the output from this command when extended TTOCs have been enabled.

```
ARC0468I EXTENDEDTTOC=Y
```

Figure 4-5 QUERY SETSYS output when extended TTOC is enabled

4.3.3 Coexistence considerations

A coexistence PTF is required on lower-level systems. If a lower level of DFSMShsm detects that a z/OS V1R7 DFSMShsm instance in the same HSMplex is using an extended TTOC, then tape operations on that DFSMShsm will be inhibited and an ARC0130I message will be issued with new return code 19.

We recommend that you do not use extended TTOCs in a sysplex until all systems are upgraded to z/OS V1R7 to avoid inhibiting tape operations on any systems that are not at the z/OS V1R7 level.

Table 4-3 Coexistence PTFs for extended TTOC

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08863	UA16947	UA16948	UA16949

4.4 Removal of ABARS requirement for INCLUDE statement

ABARS currently requires that there is at least one data set in the INCLUDE list when backing up an aggregate group. However, it is often useful to have aggregates that contain only ALLOCATE and ACCOMPANY lists of data sets. These can be used to define a catalog and populate it with disk and tape data sets as well as define necessary GDG base definitions.

With this change, ABARS processing no longer requires you to specify an INCLUDE statement in the data set selection list. You need only specify allocation information (an ALLOCATE statement) or tape catalog information (an ACCOMPANY statement).

4.5 Cancellation of individual HSM tasks

This enhancement allows active DFSMShsm data movement tasks to be cancelled, including ABARS tasks that process in their own address space, without impacting other on-going DFSMShsm activity. Currently, active data movement tasks can only be terminated by bringing down the entire DFSMShsm address space.

APAR OA09344 makes this enhancement available on z/OS V1R3 through z/OS V1R7.

Table 4-4 PTFs for cancellation of DFSMShsm tasks

APAR	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA09344	UA20209	UA20210	UA20211	UA20212

4.5.1 QUERY command enhancements

The QUERY ACTIVE command now has an optional subparameter, TCBADDRESS, which requests that only active data movement tasks be listed. This command will also list the *task control block* (TCB) address of active data movement tasks, or the *secondary address space index* (SAS index) for active ABARS tasks. The command should be entered as follows:

```
F HSM,QUERY ACTIVE(TCBADDRESS)
```

Figure 4-6 shows the output from this command when a data set is being migrated and an ABACKUP task is running.

```
ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=2
ARC0162I MIGRATING DATA SET 041
ARC0162I (CONT.) MHLRES2.D1130.H17.SC64.#MASTER#.S00006.TRS FOR USER
ARC0162I (CONT.) MHLRES2, REQUEST 00000039 ON HOST 2 , TCB=08016168
ARC6020I BACKING UP AGGREGATE GROUP MHLRES2 FOR USER 042
ARC6020I (CONT.) MHLRES2 REQUEST 00000038 , SASINDEX=01
ARC0101I QUERY ACTIVE COMMAND COMPLETED ON HOST=2
```

Figure 4-6 QUERY ACTIVE(TCBADDRESS) output

Note: If CDS backup is active when the QUERY command is issued, more than one TCB may be returned since up to ten data movement tasks may be running in parallel.

4.5.2 CANCEL command enhancements

The CANCEL command now has two additional parameters, TCBADDRESS and SASINDEX, that allow active DFSMShsm data movement tasks to be cancelled. Note that this function is only intended for use when active tasks are hung. Cancelling a task that is still processing may have unintended consequences since the task may have moved on to another function between the time the TCB is obtained and used to cancel the task.

In order to cancel an active DFSMShsm data movement task, issue the CANCEL command with the TCB address of the task that was displayed in the output from the QUERY ACTIVE command.

`F HSM,CANCEL TCBADDRESS(`*`tcbaddress`*`)`

We strongly recommend that the following procedure be followed to cancel an active data movement task:

1. Hold the active DFSMShsm function.
2. Issue QUERY ACTIVE(TCBADDRESS) to ensure that the task to be cancelled is still running.
3. Issue CANCEL TCBADDRESS(*tcbaddress*) with the TCB address of the task to cancel.
4. Release the DFSMShsm function that was held.

If the DFSMShsm function is not held before the CANCEL command is issued there is a possibility that the CANCEL command itself may cause a task to hang.

Figure 4-7 shows the messages that are issued when an active migration task is cancelled.

```
ARC1001I MHLRES2.D1130.H17.SC64.#MASTER#.S00006.TRS MIGRATE FAILED, RC=0900,
ARC1001I (CONT.) REAS=0000
ARC1900I DFSMSHSM ABEND S000 OCCURRED PROCESSING REQUEST
```

Figure 4-7 Messages issued when cancelling an active migration task

Note: When cancelling a hung journal backup task, allow any active CDS backup tasks to run to completion first. If a CDS backup task is also hung, this should be cancelled before the hung journal backup task is cancelled.

When cancelling an active ABARS task, issue the CANCEL command with the secondary address space index that was provided by QUERY ACTIVE:

```
F HSM,CANCEL SASINDEX(sasindex)
```

The following procedure should be followed when cancelling an active ABARS task:

1. Hold ABACKUP or ARECOVER.
2. Issue QUERY ACTIVE(TCBADDRESS) to ensure that the task to be cancelled is still running.
3. Issue CANCEL SASINDEX(*sasindex*) with the secondary address space index of the task to cancel.
4. Release the DFSMShsm function that was held.

4.6 Using wildcards with HMIGRATE

Currently, processing a HMIGRATE command when using a wildcard data set filter (*) will generate error messages for any data set that is already migrated. With z/OS V1R7, DFSMShsm will bypass any data sets that are already migrated when the HMIGRATE command is specified with a wildcard data set filter, thereby eliminating the many error messages that were previously issued.

The behavior of the HMIGRATE command using wildcards in z/OS V1R7 is as follows:

- If the ML2 parameter is not specified, data sets that reside on Level 0 DASD will be migrated, and any data sets that are already migrated will not be processed.
- If the ML2 parameter is specified, data sets that reside on Level 0 DASD or ML1 DASD will be migrated, and any data sets residing on ML2 will not be processed.

4.7 Saving LRECL of migrated data sets in the MCD

Currently, a data set must be recalled in order to query the logical record length (LRECL). With z/OS V1R7, DFSMShsm the LRECL and a new *data set empty indicator* are added to the migrated data set record (MCD record) in the MCDS for migrated data sets. This allows the information to be queried without the need to recall the data set.

Both the LRECL and the data set empty indicator of a migrated data set can be queried using the access method services DCOLLECT function. DCOLLECT collects data in a sequential file that can be used as input to other programs or applications. When DCOLLECT is run with the MIGRATEDATA option, as shown in Figure 4-8, it will collect information about migrated data sets. Table 4-5 shows the position of these new fields in the DCOLLECT migrated data set record. This information differs from that published in *DFSMS Access Method Services for Catalogs,* SC26-73942.

```
//MHLRES2D JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM  L=999,SYSAFF=*
//DCOLLECT EXEC PGM=IDCAMS
//OUTDS    DD  DSN=MHLRES2.DCOLLECT.DATA,DISP=(NEW,CATLG),
//             SPACE=(CYL,(100,50),RLSE),RECFM=VB,LRECL=644
//MCDS     DD  DSN=HSM.MCDS,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DCOLLECT -
      OFILE(OUTDS) -
      MIGRATEDATA
/*
```

Figure 4-8 Using DCOLLECT to collect information about migrated data sets

Table 4-5 LRECL and data set empty fields in the DCOLLECT type M record

Offset from beginning of record	Type	Length	Description
234 (EA)	Character	4	LRECL of data set.
238 (EE)	Bitstring	1	Information flag 3. First bit will be set to 1 if data set is empty.

The FIXCDS command may also be used to display the MCD record for a migrated data set, which will contain the LRECL and data set empty indicator. The structure of the MCD record is described in *DFSMShsm Diagnosis*,GC52-1083.

```
FIXCDS D datasetname DISPLAY
```

The LIST command can be used with the new SELECT(EMPTY) option to display only those migrated data sets that were empty at the time of migration.

```
LIST DSNAME MCDS SELECT(EMPTY)
```

4.7.1 Coexistence considerations

If a data set is migrated by DFSMShsm on a z/OS V1R7 system, a FIXCDS DISPLAY command issued from a lower-level system will be able to display both the LRECL of a migrated data set and whether it is empty. However, a DCOLLECT command run on a lower-level system will not be able to display the new information.

If a data set is migrated by DFSMShsm on a lower-level system, the LRECL and data set empty information will not be stored in the MCD record. Consequently, a DCOLLECT or FIXCDS command will display binary zeros for the LRECL and data set empty indicator of such a data set.

4.8 New recycle processing option for connected sets

A *connected set* is a chain of volumes that become linked together when a data set starts on one volume and spans to one or more other volumes. The recycle function uses the PERCENTVALID parameter on the RECYCLE command to determine whether a volume or connected set is eligible for DFSMShsm recycle processing.

Currently, a connected set is only eligible for DFSMShsm recycle processing when:

- The first volume in the connected set contains a lower percentage of valid data than the PERCENTVALID threshold.
- The connected set as a whole contains a lower percentage of valid data than the PERCENTVALID threshold.

z/OS V1R7 introduces the option to bypass the criteria that the first volume in the connected set must meet the PERCENTVALID threshold. This is done by specifying the new keyword, CHECKFIRST, in the generic RECYCLE command. The format is:

```
RECYCLE PERCENTVALID(percent) CHECKFIRST(Y | N)
```

The recycle processing criteria is unchanged for the default value of CHECKFIRST(Y).

Specifying CHECKFIRST(N) will eliminate the criterion that the percentage of valid data on the first volume of the connected set must meet the PERCENTVALID threshold. Recycle processing will now check only the entire connected set's percentage of valid data against the PERCENTVALID threshold to determine its eligibility for recycle processing.

Note: Specifying CHECKFIRST(N) may increase the overall recycle processing time since DFSMShsm will always have to look at the percentage valid criteria for the entire connected set, rather than just the first volume in a connected set.

This enhancement will allow connected sets to be recycled that previously did not meet the recycle criteria. As a result, volumes that are part of a connected set could be recycled sooner, which will release more volumes to be used for subsequent processing or to return to the tape scratch pool.

4.9 AUDIT MEDIACONTROLS resume

Running AUDIT MEDIACONTROLS on a large tape volume can take more than a day. This can cause problems, as the CDS backup function requires exclusive use of the HSM control data sets, and it may be necessary to terminate the audit before it has finished in order to allow the backup to be performed.

An AUDIT command that is currently being processed will be terminated when any of the following events occur:

- Audit is held.
- DFSMShsm is stopped.
- DFSMShsm is set to emergency mode.

This enhancement allows an AUDIT MEDIACONTROLS command that was terminated prior to completion due to one of the above events to be restarted from the point where it was stopped. This has the potential to eliminate many hours of processing, as DFSMShsm will not have to start audit processing from the beginning of the tape once again.

APAR OA12976 makes this enhancement available on z/OS V1R3 through z/OS V1R7. Table 4-6 shows enabling APARs by FMID.

Table 4-6 PTFs for AUDIT MEDIACONTROLS resume

APAR	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA12976	UA21187	UA21188	UA21189	UA21190

4.9.1 Resuming audit processing

A new parameter, RESUME, has been added to the AUDIT command. To resume an audit on tape volume TST001 that was previously cancelled prior to completion, enter the command as follows:

```
AUDIT MEDIACONTROLS VOLUMES(TST001) RESUME FIX
```

Message ARC0877I will be issued, stating the volser and extension on which audit is starting.

```
ARC0877I AUDIT IS STARTING ON VOLUME TST001 ON EXTENSION 0000
```

Figure 4-9 Message issued when resuming an audit on a tape volume

Note that RESUME is only valid for migration and backup tapes, and only when the FIX parameter is also specified. It is not possible for an audit to resume after abends or I/O errors.

Audit resume processing may be affected if the contents of the tape being audited change between the audit terminating and being resumed.

- If data sets on the tape that have already been audited are recalled or reconnected, the valid block count will not reflect this when the resumed audit completes.
- If additional entries are added to the volume by migration or backup processing, the audit will resume at the interrupt point and continue through the new entries as usual. However, if the additional entries result in a situation where the TTOC is not consistent with the contents of the tape media, the RESUME parameter will be ignored and audit will add the missing TTOC entries. On completion of this audit, the AUDIT MEDIACONTROLS RESUME command can be issued again, and the audit will then resume from the point where it was first interrupted.

4.10 Dump encryption support

This enhancement allows DFSMShsm full volume dumps to be encrypted. The dump class definitions are used to indicate whether encryption is to be performed. Data sets and volumes recovered from encrypted dumps are automatically decrypted during the recovery process.

APAR OA13453 makes this enhancement available on z/OS V1R4 through z/OS V1R7. Table 4-7 shows enabling APAR by FMID.

Table 4-7 PTFs for dump encryption

APAR	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA13453	UA22728	UA22729	UA22730	UA22731

4.10.1 Enabling dump encryption

In order to enable dump encryption, the BCDS record size may need to be increased, and dump class definitions will need to be created or modified to specify that encryption is required.

Changing the BCDS record size

The size of the dump volume record (DVL record) in the BCDS has been increased with this enhancement. Therefore, if you plan on using a stacking value greater than 97 you will need to ensure that the maximum record size of the BCDS is at least 2093. DFSMShsm will fail a volume dump operation if the stacking value is greater than 97 and the BCDS record size is less than 2093. Note that if you use fast replication, or need to keep more than 29 backup versions for any data sets, your BCDS should already be allocated with a maximum record size of 6544, which is more than sufficient for this enhancement.

To increase the maximum record size of the BCDS, follow this procedure:

1. Shut down all but one of the DFSMShsm subsystems in the HSMplex.
2. Either hold all DFSMShsm functions or set DFSMShsm to emergency mode.
3. Back up the existing BCDS using the BACKVOL CDS command.
4. Define a new BCDS with a maximum record size of 2093 or greater, as shown in Figure 4-10.
5. Stop the remaining DFSMShsm subsystem.
6. Use the REPRO command to copy the old BCDS into the newly defined BCDS, as shown in Figure 4-11 on page 122.
7. Ensure that the BACKCAT DD statement in the DFSMShsm startup procedure points to the new BCDS name.
8. Restart the DFSMShsm subsystems.

```
//MHLRES2B JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//IDCAMS  EXEC PGM=IDCAMS
//HSMBCDS  DD  UNIT=3390,VOL=SER=SBXHS5,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  DEFINE CLUSTER(NAME(HSM.BCDS.LARGEREC) VOLUMES(SBXHS5) -
         CYLINDERS(100) FILE(HSMBCDS) -
         RECORDSIZE(334 2093) FREESPACE(0 0) -
         INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
         SPEED BUFFERSPACE(530432) -
         UNIQUE NOWRITECHECK) -
         DATA(NAME(HSM.BCDS.LARGEREC.DATA) -
           CONTROLINTERVALSIZE(12288)) -
         INDEX(NAME(HSM.BCDS.LARGEREC.INDEX) -
           CONTROLINTERVALSIZE(2048))
/*
```

Figure 4-10 Defining a new BCDS with a maximum record length of 2093

```
//MHLRES2R JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  REPRO -
    INDATASET(HSM.BCDS) -
    OUTDATASET(HSM.BCDS.LARGEREC)
/*
```

Figure 4-11 Using REPRO to copy the contents of the BCDS

Dump class definitions

The DEFINE DUMPCLASS command has been enhanced with the addition of the optional parameters shown in Figure 4-12 to allow encryption settings to be specified.

```
>------------------------------------------------------------|
   |            |-NO--|  |    |                            |
   |-HWCOMPRESS(|-YES-|)-|    |-ENCRYPT(-| Fragment D |-)-|


D:  ENCRYPT Optional Parameters:
 |------------------------------------------------->
   |-NONE-----------------------------------|
   |-RSA(keylabel)--------------------------|
   |-KEYPASSWORD(password)------------------|
                           |        |-16---|  |
                           |-ICOUNT(|-cnt--|)-|

>----------------------------|
   |      |-CLRAES128-|  |
   |-TYPE(|-CLRTDES---|)-|
          |-ENCTDES---|
```

Figure 4-12 New DUMPCLASS optional parameters

ENCRYPT is an optional parameter specifying whether the data to be dumped will be encrypted. The method of encryption is determined by the subparameter as follows:

- NONE

  This is the default value, and indicates that no data encryption should be performed for this dump class.

- RSA(*keylabel*)

  The RSA keyword should be specified when secure cryptographic hardware will be used to encrypt the data to be dumped. The required parameter of the RSA keyword identifies the label that references an existing RSA public key present in the ICSF PKDS. The RSA public key is used to encrypt a randomly generated data key, so that the encrypted data key can be stored on the output medium. The corresponding private key must be present at the recovery site for the data to be decrypted. The key label can consist of up to 64 characters, of which the first character must be alphabetic or national.

- KEYPASSWORD(*password*)

  This keyword should be specified to encrypt the data on a system without secure cryptographic hardware. The password, which will be used to generate a key for the data encryption, must contain between 8 and 32 characters. Acceptable characters are upper and lower case alphabetical characters, numerals, and the following characters: !@#$%¢&*-_=:<>?|{}.

- ICOUNT(*cnt*)

  If a password is used to encrypt the data, this keyword can be used to specify the number of iterations of the SHA-1 hash algorithm hash iterations to be performed on the password to generate the key. This number should be an integer between 1 and 10000. If a robust password of at least 32 random characters is specified, then the default value of 16 will be sufficient to provide reasonable security and performance. Higher iteration counts can be used to make weaker passwords more secure.

- TYPE(CLRAES128 | CLRTDES | ENCTDES)

  The TYPE keyword specifies the type of encryption that will be performed on the data. The possible values are:

  CLRAES128 Clear 128-bit Advanced Encryption Standard (AES) key. This is the default.

  CLRTDES Clear triple-length Data Encryption Standard (DES) key.

  ENCTDES Secure triple-length DES key. This option cannot be used when KEYPASSWORD is specified.

The HWCOMPRESS optional parameter specifies whether hardware-assisted compression should be performed on the data to encrypt.

- A value of YES specifies that hardware-assisted encryption should be performed before the data is encrypted. This option should not be used unless the dumped data is to be encrypted.
- A value of NO specifies that no hardware-assisted encryption should be performed. This is the default, and should be used when the dumped data is not to be encrypted so that tape hardware compression may be used.

Note: The same password and iteration count always generates the same encryption key. Therefore, if the same password and iteration count is used to encrypt many DFSMShsm full volume dumps, all the data in those dumps are protected by the same key. If the password is compromised, all the dumped data is vulnerable.

Figure 4-13 shows an example dump class definition using the new parameters. A volume dump targeted at this dump class will be compressed using hardware-assisted compression, before being encrypted using a clear triple-length DES key generated by 32 hash iterations of the given password.

```
DEFINE DUMPCLASS(MHLRES   -
  HWCOMPRESS(YES)         -
  ENCRYPT(                -
    KEYPASSWORD(PASSWORD) -
    ICOUNT(32)            -
    TYPE(CLRTDES))        -
  AUTOREUSE               -
  RETENTIONPERIOD(7)      -
  VTOCCOPIES(1)           -
  UNIT(3590-1)            -
  DISP('ENCRYPTED DUMP'))
```

Figure 4-13 Example dump class definition with encryption

The LIST DUMPCLASS command has been enhanced to display the additional information now contained in the dump class. The output of the LIST DUMPCLASS command for the dump class defined in Figure 4-13 is shown in Figure 4-14.

```
CLASS=MHLRES    UNIT=3590-1   REUSE=Y  RESTORE=N  RESET=N  DISABLE=N
DAY=**  FREQ=007  RETPD=000007  TAPEEXPDT=*******  VTOCCOPIES=001  STACK=01
DISP=ENCRYPTED DUMP
HWCOMP=YES  ENCRYPT=KEYPW  ENCRYPT_TYPE=CLRTDES    ICOUNT=00032
ENC_KEY=PASSWORD
ARC0140I LIST COMPLETED,       5 LINE(S) OF DATA OUTPUT
```

Figure 4-14 Output from LIST DUMPCLASS command

Note: All dump classes for a particular dump generation must have the same encryption and hardware compression settings. The dump operation will fail if this is not the case.

Additions to RECOVER command

The parameters that were used to encrypt a particular dump volume will be stored in the DVL record for that volume in the BCDS. DFSMShsm will use this information to automatically decrypt data recovered from encrypted dumps.

The RECOVER command has been enhanced to allow a label for an RSA private key that is present in the ICSF PKDS to be specified that will override what DFSMShsm would otherwise use to restore the encrypted dump. The new parameter may be used as follows:

`RECOVER * FROMDUMP(DUMPCLASS(`*`class`*`)) TOVOLUME(`*`volser`*`) UNIT(`*`unit`*`) RSA(`*`keylabel`*`)`

The key label can consist of up to 64 characters, of which the first character must be alphabetic or national. There is no need for this parameter to be specified when the same label for the same RSA private key exists in the ICSF PKDS as when the data was dumped.

Note: Encrypted dump volumes can still be decrypted automatically by DFSMShsm using information from the DVL record in the BCDS even if the encryption parameters in the dump class have been changed since the dump volume was created.

4.10.2 Validating dump encryption

When DFSMShsm is processing a full volume that is encrypted, message ARC515I will be issued, as shown in Figure 4-15.

```
ADR515I (001)-DDTFP(01), 2006.110 18:23:36 ENCRYPTION SERVICES TO BE
471
USED FOR DUMP DATA SET ON VOLSER TST028
```

Figure 4-15 Encrypting a DFSMShsm full volume dump

4.10.3 Migration and coexistence considerations

Before the APAR for this enhancement is installed, coexistence APAR OA13687 must be installed and active on all systems in the HSMplex. This APAR causes DFSMShsm to correctly fail a volume dump operation if the maximum BCDS record size is less than 2093 when the stack value for the dump is greater than 97.

DFSMShsm makes use of the DFSMSdss dump encryption support as described in 5.2, “Dump encryption” on page 131, to provide this enhancement. PTFs for APARs OA13300 and OA14991, which add the ability to encrypt DFSMSdss dumps, must be installed before this function can be used. In order for DFSMSdss to perform encryption the Encryption Facility DFSMSdss Encryption Feature (HCF773D) and the IBM Cryptographic Services Facility (HCR770B or later) must be installed. Table 4-8 and Table 4-9 list the required maintenance.

Table 4-8 Coexistence PTFs required for DFSMShsm dump encryption

APAR	HDZ11F0	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA13687	UA22712	UA22713	UA22714	UA22715	UA22716

Table 4-9 PTFs required for DFSMSdss dump encryption

APAR	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA13300	UA90212	UA90213	UA90214	UA90215
OA14991	UA24917	UA24918	UA24919	UA24920

5

DFSMSdss enhancements

This chapter describes the following enhancements to DFSMSdss:

- Large format data set support
- Dump encryption

5.1 Large format data set support

DFSMSdss has been enhanced to support the new large format data sets that are introduced in 2.1, “Large format data sets” on page 12.

Large format data sets can have greater than 65,535 tracks per volume. Before z/OS V1R7, most sequential data sets were limited to 65,535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. Large format data sets are not required to have greater than 65,535 tracks at initial allocation, but once allocated in large format they can expand to greater than 65,535 tracks.

z/OS V1R7 also introduces the concept of a basic format data set to designate a sequential data set that is neither extended format nor large format. A basic format data set therefore may not have any more than 65,535 tracks per volume.

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 listed 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.

5.1.1 Logical COPY

DFSMSdss logical data set COPY processes large format data sets as follows:

- When processing a large format sequential data set, and no target data set exists, DFSMSdss allocates a large format sequential data set to be used as the target.
- When processing a sequential data set of any format, and a preallocated large format target data set is found, this data set is used as the target as long as it is large enough to accommodate the used space in the source data set. If it is not large enough to accommodate the used space in the source, it is scratched and reallocated as a large format data set with sufficient space. This provides a way for existing basic format data sets to be upgraded to large format.
- When processing a large format sequential data set, and a preallocated basic format target data set is found, the following happens:
  - If the source data set has used fewer than 65,535 tracks, and the target data set has enough space allocated to accommodate the data from the source, the target data set is used and turned into a large format sequential data set. If the target data set does not have enough space allocated to accommodate the data from the source, it is

scratched and reallocated as a large format data set with enough space for the data from the source.

– If the source data set has used more than 65,535 tracks, and the source and target data sets have the same name, the target cannot be large enough to accommodate the data from the source. It is therefore treated as a preallocated target data set that is too small and the input data set is not copied. If the source and target have different names (RENAMEU and REPLACEU processing), the target data set is scratched and reallocated as a large format data set with enough space for the source data.

Figure 5-1 shows an example of the JCL used to copy a large format data set greater than 65,535 tracks to a preallocated basic format data set with a different name.

```
//MHLRES2C JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//COPY    EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 COPY DATASET(INCLUDE(MHLRES2.DSSCOPY.SOURCE)) -
      RENAMEU((MHLRES2.DSSCOPY.SOURCE,MHLRES2.DSSCOPY.TARGET)) -
      REPLACEU -
      ALLDATA(*) ALLEXCP -
      CATALOG
/*
```

Figure 5-1 DFSMSdss COPY from a large format data set to preallocated basic format data set

The preallocated data set is scratched and reallocated as a large format data set with sufficient space to accommodate the source data set. Figure 5-2 shows the DFSMSdss messages produced in this situation.

```
ADR442I (001)-PREDS(02), DATA SET MHLRES2.DSSCOPY.SOURCE PREALLOCATED WITH NEW
NAME MHLRES2.DSSCOPY.TARGET, IN CATALOG
                         UCAT.VSBOX01, ON VOLUME(S): MLD50C
ADR390I (001)-PREDS(01), DATA SET MHLRES2.DSSCOPY.TARGET WILL BE SCRATCHED FROM
MLD50C BECAUSE OF UNMATCHED SIZE. IT WILL BE
                         REALLOCATED
ADR431I (001)-DYNA (02), DATA SET MHLRES2.DSSCOPY.TARGET IN CATALOG
UCAT.VSBOX01 HAS BEEN DELETED
ADR711I (001)-NEWDS(01), DATA SET MHLRES2.DSSCOPY.SOURCE HAS BEEN ALLOCATED
WITH NEWNAME MHLRES2.DSSCOPY.TARGET USING STORCLAS
                         STANDARD, NO DATACLAS, AND MGMTCLAS MCDB22
```

Figure 5-2 Target data set reallocated as a large format data set

5.1.2 Logical and physical RESTORE

Logical and physical RESTORE has been enhanced to use large format sequential data sets as input, and to restore large format data sets.

DFSMSdss logical data set RESTORE processes large format data sets as follows:

- When restoring a large format data set and no preallocated target data set exists, DFSMSdss allocates a new large format data set to be used as the target.

- If a preallocated large format target data set exists, it is used as the target as long as it is large enough to contain the data from the source. Should the preallocated target data set's allocation be less than the used space in the data set being restored, the preallocated target is scratched and reallocated as a large sequential data set with adequate space.
- When restoring a large sequential data set, and a preallocated target data set is found that is not a large format sequential data set, the preallocated data set is used and turned into a large format data set if it is large enough to accommodate the data being restored. If the target data set does not have enough space allocated to accommodate the data being restored, it is scratched and reallocated as a large format sequential data set with adequate space.

DFSMSdss physical data set RESTORE processes large format data sets slightly differently, as follows:

- When restoring a large format data set and no preallocated target exists, DFSMSdss allocates a new large format sequential data set to restore the track data into.
- When restoring an unmovable large format sequential data set, and a preallocated data set exists with the same name, the preallocated data set is used as long as its extents, volume sequence numbers, and DSORG match that of the data set being restored.
- When restoring an unmovable sequential data set that is not large format, and a preallocated large format sequential data set with the same name exists, the preallocated data set is used if the extents, volume sequence numbers, and DSORG match. This has the effect of upgrading the unmovable sequential data set to a large format data set.
- When restoring a movable large format sequential data set, and a preallocated sequential data set exists with the same name, it is used as a target for the restore as long as it has enough allocated space to contain the restored data set's tracks. If the target data set does not have enough allocated space, it is scratched and reallocated as a large format sequential data set with the correct size.
- When restoring a movable sequential data set that is not large format, and a preallocated large format sequential data set with the same name exists, it is used as a target for the restore as long as the preallocated data set has enough allocated space to contain the restored data set's tracks. If the preallocated data set does not have a large enough allocation, it is scratched and reallocated as a large format data set with the correct allocation. This has the effect of upgrading a basic format data set to a large format sequential data set.

Note: If a preallocated basic format data set is used as the target for the physical restore of a large format data set, the preallocated data set will not be converted into large format. Therefore, in this situation, the data set being restored will be downgraded from large format to basic format. There is an APAR that changes this behavior so that the preallocated basic format data set is converted to a large format data set during the physical restore process.

5.1.3 COPYDUMP

DFSMSdss COPYDUMP can copy a large sequential dump data set to a sequential data set of any format for output. The output data set must be large enough to accommodate all the data from the input. This implies that if the dump data set has used more than 65,535 tracks, the output data set must be defined as a large format data set. However, a large format data set may be used as the output even if the dump data set being processed is not a large format data set. This allows basic format dump data sets to be upgraded to large format data sets.

5.1.4 Coexistence with lower-level systems

Coexistence PTFs are available for earlier systems to prevent DFSMSdss from attempting to process large format data sets on systems that are not capable of processing them correctly. Message ADR778E with reason code 17 is issued when an attempt is made to process a large sequential data set on these systems.

Figure 5-3 shows the message that is issued on a prior level of DFSMSdss when an attempt is made to process a large format data set.

```
ADR778E (001)-DTDSC(01), DATA SET WELCH.LARGE.DATASET WAS NOT SELECTED
BECAUSE
THE DATA SET TYPE IS NOT SUPPORTED IN THIS RELEASE,17
```

Figure 5-3 Unable to process large format data sets on pre-z/OS V1R7 systems

Table 5-1 DFSMSdss coexistence PTFs for large format data sets

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08286	UA16796	UA16797	UA16798

5.2 Dump encryption

DFSMSdss is now able to encrypt physical and logical dumps produced with the DUMP command. This enhancement applies to data set, track, and full volume dumps. Encrypted dumps will be decrypted when restored with the RESTORE command.

Only user data is encrypted during dump processing. This means that VTOC and VVDS tracks that are processed are not encrypted. Data set names, and other information in the VTOC, will appear unencrypted in the output dump data set.

APAR OA13300 makes this enhancement available on z/OS V1R4 through z/OS V1R7. PTFs for APAR OA14991 must also be installed, as it corrects a problem introduced by OA13300.

Table 5-2 PTFs required for DFSMSdss dump encryption

APAR	HDZ11G0	HDZ11H0	HDZ11J0	HDZ11K0
OA13300	UA90212	UA90213	UA90214	UA90215
OA14991	UA24917	UA24918	UA24919	UA24920

In order for this feature to operate, the Encryption Facility DFSMSdss Encryption Feature (HCF773D) and the IBM Cryptographic Services Facility (HCR770B or higher) must be installed.

This enhancement must be installed in order for DFSMShsm to provide full volume dump encryption, as described in 4.10, “Dump encryption support” on page 120.

For more information about DFSMSdss dump encryption refer to *DFSMSdss Storage Administration Guide,* SC26-0423, and *DFSMSdss Storage Administration Reference,* SC35-0424.

5.2.1 Creating encrypted dump data sets

New parameters have been added to the DFSMSdss DUMP command to allow dump encryption to be specified. The syntax for these parameters is shown in Figure 5-4.

```
>>__ ______________________________ _____________________________>
    |               _ENCRYPT(_CLRAES128_)_|
    |_RSA(label)_|______________________|
                 |_ENCRYPT( _CLRTDES_ )_|
                           |_ENCTDES_|

>>__ ________________________________________________________________>
    |                      _ICOUNT(16)_   _ENCRYPT(_CLRAES128_)_   |
    |_KEYPASSWORD(password)_|___________|_|______________________|__|
                           |_ICOUNT(n)__| |_ENCRYPT(_CLRTDES_)___|

>>__ ___________ ____________________________________________________>
    |_HWCOMPRESS_|
```

Figure 5-4 New DFSMSdss DUMP optional parameters

The parameters are:

- RSA(*keylabel*)

  The RSA keyword should be specified when secure cryptographic hardware will be used to encrypt the data to be dumped. The required parameter of the RSA keyword identifies the label that references an existing RSA public key present in the ICSF PKDS. The RSA public key is used to encrypt a randomly generated data key, so that the encrypted data key can be stored on the output medium. The corresponding private key must be present at the recovery site for the data to be decrypted. The key label can consist of up to 64 characters, of which the first character must be alphabetic or national.

- KEYPASSWORD(*password*)

  This keyword should be specified to encrypt the data on a system without secure cryptographic hardware. The password, which will be used to generate a key for the data encryption, must contain between 8 and 32 characters. Acceptable characters are upper and lower case alphabetical characters, numerals, and the following characters: !@#$%¢&*-_=:<>?|{}.

- ICOUNT(*n*)

  If a password is used to encrypt the data, this keyword can be used to specify the number of iterations of the SHA-1 hash algorithm hash iterations to be performed on the password to generate the key. This number should be an integer between 1 and 10000. If a robust password of at least 32 random characters is specified, then the default value of 16 will be sufficient to provide reasonable security and performance. Higher iteration counts can be used to make weaker passwords more secure.

- ENCRYPT(*type*)

  This keyword allows the type of encryption that will be used to be specified. The possible values are:

  CLRAES128 Clear 128-bit Advanced Encryption Standard (AES) key. This is the default.

  CLRTDES Clear triple-length Data Encryption Standard (DES) key.

ENCTDES Secure triple-length DES key. This option cannot be used when KEYPASSWORD is specified.

Each type of key is equally secure in regards to the data that appears in the output data set.

- HWCOMPRESS

  This keyword specifies that hardware-assisted compression should be performed on the data to encrypt to decrease the space occupied by the dump data set. We recommend that this keyword is specified whenever DFSMSdss dump encryption is performed. HWCOMPRESS cannot be specified at the same time as the COMPRESS keyword.

> **Note:** The same password and iteration count always generate the same encryption key. Therefore, if the same password and iteration count are used to encrypt many DFSMSdss dumps, all of the data from those DUMP jobs are protected by the same key. If the password is compromised, all of the dumped data is vulnerable.

Figure 5-5 shows a JCL example of a DFSMSdss job that will perform encryption on a logical data set dump. The dumped data will be compressed using hardware-assisted compression, before being encrypted using a clear triple-length DES key generated by 32 hash iterations of the given password.

```
//MHLRES2D JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//COPY    EXEC PGM=ADRDSSU
//OUT      DD  DSN=MHLRES2.DUMP,DISP=(NEW,CATLG),
//             SPACE=(TRK,(500,150),RLSE),
//             VOL=SER=MLD10B,UNIT=3390,STORCLAS=NONSMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DUMP DATASET(INCLUDE(MHLRES2.TEST.DS)) -
      OUTDD(OUT) -
      ALLDATA(*) ALLEXCP -
      KEYPASSWORD(PASSWORD) -
      ICOUNT(32) -
      ENCRYPT(CLRTDES) -
      HWCOMPRESS
/*
```

Figure 5-5 DFSMSdss dump encryption example

When this job is submitted, the ADR515I message shown in Figure 5-6 is issued by DFSMSdss to show that the dump data set will be encrypted.

```
ADR515I (001)-DTDSC(01), 2006.117 19:42:47 ENCRYPTION SERVICES TO BE
037
USED FOR DUMP DATA SET ON VOLSER MLD10B
```

Figure 5-6 DFSMSdss dump encryption message

5.2.2 Restoring encrypted dumps

Encrypted dump data sets can be restored using the DFSMSdss RESTORE command. New parameters have been added to the RESTORE command for this purpose. Their syntax is shown in Figure 5-7.

```
>>__ ______________________ __>
    |_KEYPASSWORD(password)_|

>>__ ___________ ____________>
    |_RSA(label)_|
```

Figure 5-7 New DFSMSdss RESTORE optional keywords

The KEYPASSWORD keyword is used to specify an 8 to 32 character password to decrypt the dump data set in order for the restore operation to complete successfully. The same password must be specified on the RESTORE command as was specified on the DUMP command to create the dump data set. Should the wrong password be specified, the dump data set will not be restored, and the ADR513E message shown in Figure 5-8 will be issued.

```
ADR513E (001)-TDDS (01), KEYPASSWORD TESTPASS IS INCORRECT. DATA WILL NOT BE
DECRYPTED
```

Figure 5-8 Message issued when incorrect password is specified

The RSA parameter allows a label for an RSA private key that is present in the ICSF PKDS to be specified. If the RSA keyword was specified on the DUMP command, there is no need for it to be specified on the RESTORE command when the same label for the same RSA private key exists in the ICSF PKDS. This keyword can be used to specify a different label for the RSA private key than was specified on the DUMP command.

Figure 5-9 shows a JCL example of a job to restore a single data set from a dump data set encrypted with a password.

```
//MHLRES2R JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
//             NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//COPY    EXEC PGM=ADRDSSU
//IN       DD  DSN=MHLRES2.DUMP,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 RESTORE DATASET(INCLUDE(MHLRES2.TEST.DS)) -
        INDDNAME(IN) -
        CATALOG -
        KEYPASSWORD(PASSWORD)
/*
```

Figure 5-9 DFSMSdss encrypted dump RESTORE example

5.2.3 Migration and coexistence considerations

Encrypted dumps can only be restored on other systems where a PTF for APAR OA13300 has been installed.

APAR OA13300 changes the format of the dump data set when extended format sequential data sets are dumped. Because of this, a PTF for APAR OA14991 must be installed in conjunction with a PTF for APAR OA13300. APAR OA14991 causes unencrypted dump data sets containing extended format sequential data sets to be produced in the current format. Therefore, an unencrypted dump produced on a system with a PTF for APAR OA13300 installed can be decrypted on any system, provided that a PTF for APAR OA14991 is also installed on the system where the DUMP job is run.

6

DFSMSrmm V1R7 enhancements

In this chapter we discuss the DFSMSrmm enhancements, which include:

- Issue DFSMSrmm TSO commands from the console
- Improved security control over DFSMSrmm functions
- Use of large format data sets
- Enterprise enablement

6.1 Issue DFSMSrmm TSO commands from the console

The MVS MODIFY command can now be used to issue an RMM TSO command from the console. The command output is returned to the console and system log.

`F DFRMM,CMD=command` is used to issue a DFSMSrmm TSO subcommand from the console and to have the command output returned to your console and the system log. In order to issue commands using the MODIFY command, you must be authorized to issue that command. If you use z/OS Security Server (RACF) to secure your operator console, the user ID of the logged-on operator is used to authorize the command. If you do not use z/OS Security Server (RACF) to secure your console, the user ID of the DFRMM started procedure is used to authorize the command.

In Figure 6-1 to Figure 6-4 on page 141 we show the command F DFRMM,CMD=LISTCONTROL ALL and results.

```
F DFRMM,CMD=LISTCONTROL ALL
Control record: 033
Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                 Update date = 2006/094    Update time = 11:26:23
Journal: Utilization =  43% (75% threshold)     STATUS: = ENABLED
CDS:     Utilization =  55%
Exit status:                             Options:
  EDGUX100 = NONE                          Stacked Volumes         =
NON
  EDGUX200 = NONE                          Extended Bin            =
DIS
Last backup:                             Last expiration processing:
  Date = 2003/274    Time = 20:47:56       Date = 2005/231    Time =
10:
Last journal backup:                     Last store update:
  Date = 2003/309    Time = 19:51:23       Date = 2003/302    Time =
14:
Last report extract:                     Last VRS processing:
  Date = 2005/231    Time = 10:49:12       Date = 2005/231    Time =
10:
Last scratch procedure:                  Last Catalog synchronize:
  Date =             Time =                Date =             Time =
Rack numbers       = 6248                Empty racks        = 4211
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 6-1 RMM TSO command issued from the console (part 1 of 4)

```
System options:
PARMLIB Suffix  = 02
Operating mode  = P    Retention period: Default = 0       Maximum =
NO
                                         Catalog = 6     hours
Control data set name      = RMM.CONTROL.DSET
Journal file data set name = RMM.JOURNAL.DSET
Journal threshold          = 75%
Catalog SYSID              = Notset
Scratch procedure name     = EDGXPROC
Backup procedure name      = EDGCDSBK
IPL date check  = N        Date format    = J       RACF support   = N
SMF audit       = 248      SMF security   = 249     CDS id         =
MAXHOLD value   = 100      Lines per page = 54      System ID      =
SC
BLP             = RMM      TVEXT purge    = EXPIRE  Notify         = N
Uncatalog       = Y        VRS job name   = 2       Message case   = M
MASTER overwrite= LAST     Accounting     = J       VRS selection  =
NE
VRS change      = INFO     VRSMIN action  = INFO    VRSMIN count   = 1
Disp DD name    = DISPDD   Disp msg ID    = EDG4054I
Retain by       = SET      Move by        = SET     CMDAUTH Owner  =
NO
PREACS          = NO       SMSACS         = NO      CMDAUTH Dsn    =
YE
Reuse bin       = CONFIRMMOVE                       Media name     =
34
                                                    Local tasks    =
10
PDA: ON
 Block count    = 255      Block size     = 27      Log            =
ON
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   =
YE
 Purge          = ASIS
```

Figure 6-2 RMM TSO command issued from the console (part 2 of 4)

```
Client/Server:
 Subsystem type = STANDARD Port           = 0
 Server                    Server tasks   = 0
  host name  =
  IP address =


NO INSTALLATION DEFINED SECURITY CLASSES

Volume Pools:
Pool   System   RA Ty Expdt Pool     Media    Description
                CF pe check name     name
------ -------- -- -- ----- -------- -------- ------------------------
-
                     Action Scratch Overwrite
                     ------ ------- ---------
DV*             N  S  O     DV3590   MEDIA3   3590 DUPLICATE POOL
                            AUTO   LAST
TST*            N  S  O     PL3590   MEDIA3   3590 POOL
                            AUTO   LAST
TS4*            N  S  O     PL3490   3490     3490 POOL
                            AUTO   LAST
*               N  S  O              3480     DEFAULT VOLUME POOL
                            AUTO   LAST

Mount/Fetch messages:
Message ID    ID  Volume Rack
------------  --- ------ ----
IEF233A       1   15     999

Reject prefixes:
Rack Prefix  Access type
-----------  -----------
ADS*         NONE
DB*          NONE
HE00*        NONE
J5T*         NONE
MXX*         NONE
M008*        NONE
M017*        NONE
M024*        NONE
M203*        NONE
R0008*       NONE
R0017*       NONE
R0024*       NONE
R020*        NONE
S0024*       NONE
S020*        NONE
THS*         NONE
```

Figure 6-3 RMM TSO command issued from the console (part 3 of 4)

```
Location definitions:
Location Def Mgtype  Ltype  Priority Medianames
-------- --- ------  -----  -------- ----------
          N          AUTO     4800
          N          MANUAL   4900
DISTANT   N          STORE     200
LOCAL     N          STORE     300
REMOTE    N          STORE     100
SHELF     N                   5000

Action   Status
-------- ---------
RETURN   Pending
SCRATCH  Complete
REPLACE  Unknown
INIT     Unknown
ERASE    Unknown
NOTIFY   Unknown


NO MOVES STATUS INFORMATION IS AVAILABLE

EDG1101I DFSMSrmm MODIFY COMMAND ACCEPTED
```

Figure 6-4 RMM TSO command issued from the console (part 4 of 4)

> **Note:** There may be some truncation or line output wrapping when using this function. The effective line length of SYSLOG is less than the 79-column standard on TSO. Whether this results in loss of significant information depends on the command. In the example output in Figure 6-1 on page 138 to Figure 6-4 on page 141, those lines that end with an equals sign (=) typically have been truncated, and they may also have information wrapped to the next line.

6.2 Improved security control over DFSMSrmm functions

New resource profiles are implemented in z/OS V1R7 DFSMSrmm to better control access to DFSMSrmm resources. It is no longer necessary to have CONTROL access to STGADMIN.EDG.MASTER to run daily DFSMSrmm tasks.

When checking authorization to use RMM subcommands and operands, DFSMSrmm checks in the following sequence:

1. CONTROL access to STGADMIN.EDG.MASTER. If the user is authorized, no further checking is performed.
2. Next, DFSMSrmm checks for specific subcommand operands and for each operand that requires specific authorization checks for the required access. If the resource is not protected, authorization continues with the next step. If the resource is protected, but the user is not authorized, the subcommand fails.
3. Finally, DFSMSrmm continues with ownership checks and RELEASE and FORCE checking, if required.

Because of the way authorization is checked, it is not necessary to have CONTROL access to STGADMIN.EDG.MASTER to perform many of the regular administrative tasks.

Since setting the appropriate authorization levels is an important activity, it needs to be done with care. If you wish to exercise some or all of the new profiles, review the information in Chapter 11 of the manual *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

6.3 Use of large format data sets

DFSMSrmm has been enhanced to support the new large format sequential data sets that are introduced in 2.1, "Large format data sets" on page 12. Large format data sets can have greater than 65,535 tracks per volume. Before z/OS V1R7, most sequential data sets were limited to 65,535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. Large format data sets are not required to have greater than 65,535 tracks at initial allocation, but once allocated in large format they can expand to greater than 65,535 tracks.

z/OS V1R7 also introduces the concept of a basic format data set to designate a sequential data set that is neither extended format nor large format. A basic format data set therefore may not have any more than 65,535 tracks per volume.

Large format data sets are supported by DFSMSrmm for the following:

- The JOURNAL - refer to "RMM JOURNAL and journal backup conversion to large format" on page 143.
- The journal backup file created by EDGBKUP or EDGHSKP - refer to "RMM JOURNAL and journal backup conversion to large format" on page 143.
- Temporary work files created by the DFRMM started task during inventory management when running the EDGHSKP utility.
- Temporary work file created by EDGUTIL during VERIFY and MEND processing.

6.3.1 Large format data set coexistence considerations

Coexistence maintenance must be installed on down-level systems to preserve data integrity by preventing the lower-level system from damaging the JOURNAL. There is no support for RMM JOURNAL processing on prior systems. Any attempt to use a data set that has been allocated in large format will result in an ABEND and the job or started task will fail to initialize. Any attempt to use a large format JOURNAL data set on a lower-level system that does not have the support maintenance applied is unpredictable.

> **Note:** The new data set does not have to be over 65,535 tracks for restrictions on use with down-level systems to apply

6.3.2 Migration considerations

Migration of permanent data sets to large format requires:

- Installation of supporting maintenance on any lower-level systems in the same sysplex. See "Maintenance on lower-level systems" on page 143.

- Allocation of new data sets and a restart of the started task. The procedures for allocating a replacement JOURNAL data set have not changed with large format data set support. The only change required is to add the DSNTYPE=LARGE when allocating the new JOURNAL data set, and on the JCL for the backup data set, specifically JOURNAL and journal backup data sets. Refer to “RMM JOURNAL and journal backup conversion to large format” on page 143.

Temporary work files will automatically use large format if specified in the JCL.

> **Note:** Large format can only be used on z/OS V1R7 systems. Refer to 6.3.1, “Large format data set coexistence considerations” on page 142.

6.3.3 Maintenance on lower-level systems

Table 6-1 shows the minimum level PTF to be installed for each prior release

Table 6-1 DFSMSrmm support PTFs required

Feature and reason PTFs are needed	z/OS V1R4 PTFs	z/OS V1R5 PTFs	z/OS V1R6 PTFs
DFSMSdfp: z/OS V1R7 supports large format sequential data sets. These PTFs cause pre-z/OS V1R7 to issue ABEND 213–14 or 213–16 if a program attempts to open data sets that cannot work on those systems (not specific to RMM).	UA15869 UA16550	UA15870 UA16551	UA15871 UA16552

6.3.4 RMM JOURNAL and journal backup conversion to large format

Your installation will have a procedure running regularly to back up the control data set (CDS) and clear the journal. This is the best procedure to use to allocate a replacement JOURNAL, and adjust the JCL for the allocation of the back-up data set. This allows you to use the same procedure that you would use to recover from an RMM failure that requires reversion to a specific CDS point.

1. Run the backup procedure.
2. Close DFSMSrmm down.
3. Rename the exiting journal data set to save it for rapid fallback.
4. Allocate the replacement data set using the JCL attribute DSNTYPE=LARGE.
5. Change the JCL in the backup procedure to specify DSNTYPE=LARGE (so that the next run makes a back-up in the appropriate format).
6. Restart DFSMShsm.

6.3.5 Recovery from JOURNAL problems

If the JOURNAL or journal backup is converted to large format too soon (for example, if a sharing system has not been upgraded), then a valid format data set will need to be obtained. How this can be done depends on the amount of processing that has taken place in the V1R7 system before the problem is detected.

No significant processing

If nothing significant has been done, then revert to the renamed copy created in step 3 of 6.3.4, “RMM JOURNAL and journal backup conversion to large format” on page 143.

Significant processing

If there have been significant changes, which will only have been on a z/OS V1R7 system, then a process similar to that used to convert to large format data set can be used:

1. Run the backup procedure.
2. Close DFSMSrmm down.
3. Delete the large format journal data set.
4. Rename the saved data set to the standard name (as used before the conversion started).
5. Remove the change (to add DSNTYPE=LARGE) made to the JCL in the backup procedure (so that the next run makes a back-up in the appropriate format).
6. Restart DFSMSrmm.

6.4 Enterprise enablement

DFSMSrmm started to expand to the enterprise in z/OS 1.6, first with enabling an RMM z/OS client, and also by adding the RMM class library for use of the RMM API from object-oriented languages. You could use the high-level language application programming interface from C/C++ and Java™ (using the JNI) code running on the same z/OS system as the DFSMSrmm subsystem. A series of calls to the application programming interface were necessary to run the subcommand and to receive the output.

In DFSMS V1.7, the direction of extending DFSMSrmm to the enterprise continues. Extending DFSMSrmm to the enterprise is being accomplished through enabling the object-oriented interface to the RMM API to be used via Web Services and creating a plug-in adapter for use with SNIA SMI-S CIMOM. For more information about the SNIA SMI-S standards go to the Web site:

http://www.opengroup.org/snia-cimom/

With z/OS V1R7 and the DFSMSrmm enterprise enablement enhancement, you can use the high-level language application programming interface as a Web service. This enables the API to be used from any system or platform that can run Java, C++, or any language that supports the Web services standards. Now it is as though the high-level language application programming interface is available as a locally callable program. A single call to the application programming interface to run a subcommand and receive all the output is all that is needed.

The Web service is provided as a programming interface so other applications that need tape management services can use RMM. The CIM agent support exploits this to access RMM from anywhere.

A sample Java Web service application, rmmSampleWSClient.java, is located in /usr/lpp/dfsms/rmm/. The sample code shows how the application programming interface can be used via a Web service.

For further information, see:

- *DFSMSrmm Application Programming Interface,* SC26-7403
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405

We cover the two aspects of the enhancements in z/OS V1R7 in 6.4.9, “DFSMSrmm CIM provider” on page 198.

6.4.1 WEB Service

In Figure 6-6 and Figure 6-58 on page 198 we show an overview of the Web service.

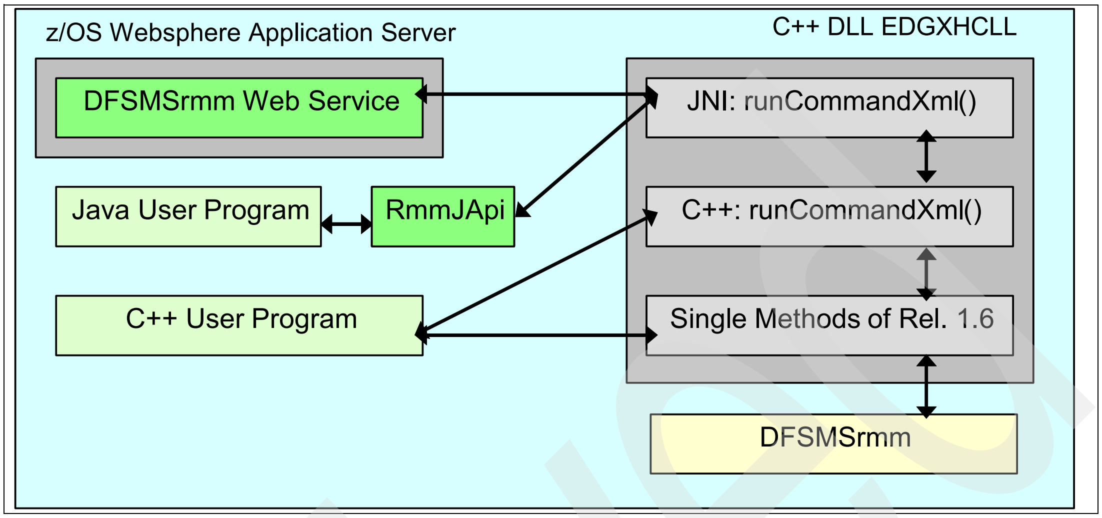


Figure 6-5 DFSMSrmm Web service overview

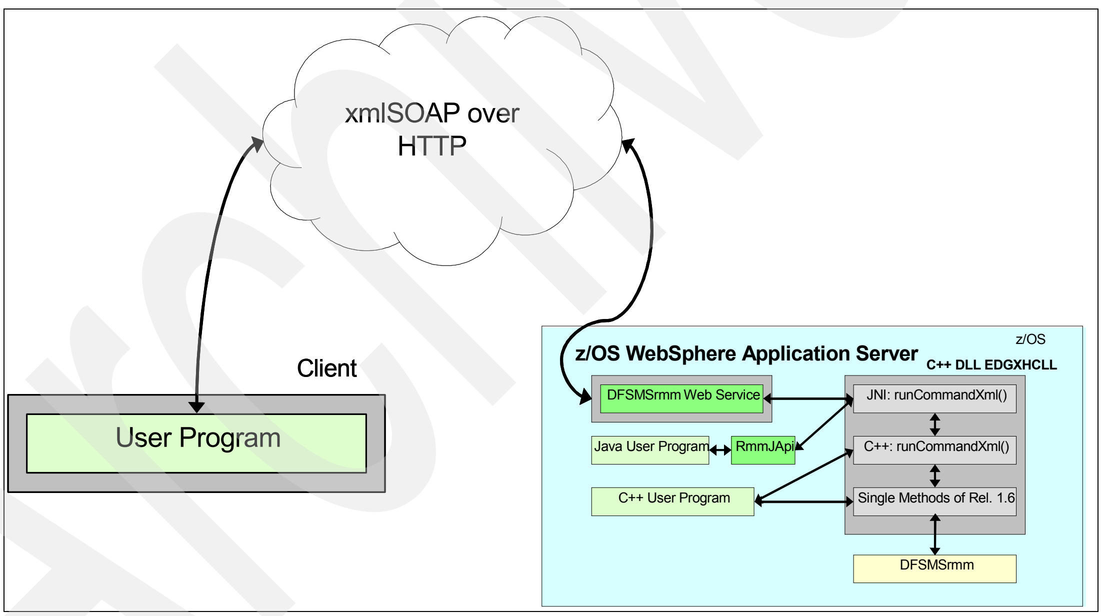


Figure 6-6 DFSMSrmm Web service overview (continued)

6.4.2 Prerequisites for DFSMSrmm WEB service installation

The prerequisites for DFSMSrmm Web service installation are:

- WebSphere® Application Server for z/OS V5.0.2 or later, or an equivalent.
- A fully functional Web service.
- When DFSMSrmm is installed, the resources necessary to populate tables in the WebSphere Application Server (WAS) are provided in the system HFS or zFS. You must install these resources in the appropriate WAS.

Preparation for implementation of DFSMSrmm WEB service

The steps required are summarized in a text file supplied as one of the resources provided with DFSMSrmm. When setting up the WAS interface, these resources are required. We use the Ishell function to locate the test file to demonstrate the directory structure that we search when populating the WAS structure. The resources are located in the /usr structure. To locate these resources, we did the following:

```
Start Ishell
```

In Figure 6-7 we show the Ishell panel with the path name specified as /usr and we also have specified the Directory pull-down.

```
 File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 ____________________________________________________________________________
                     UNIX System Services ISPF Shell

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                               More:     +
   /usr
   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________

EUID=0




Command ===> ____________________________________________________________
```

Figure 6-7 Ishell panel with pathname /usr and selecting Directory pull-down

In Figure 6-8 we show the directory pull-down with option1 - List directory specified.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 ─────┌────────────────────────────────┐─────────────────────────────────
      │ 1_ 1. List directory(L)...     │ ISPF Shell
      │    2. New(N)...                │
 Enter│    3. Attributes(A)...         │
      │    4. Delete(D)...             │
     -│    5. Rename(R)...             │
     -│    6. Copy to PDS(C)...        │
     -│    7. Copy from PDS(I)...      │n the command line.
      │    8. Print(P)                 │
 Retur│    9. Compare(M)...            │rent pathname.
      │   10. Find strings(F)...       │                      More:     +
    /u│   11. Set working directory(W) │
    __│   12. File system(U)...        │__________________________
    __└────────────────────────────────┘__________________________
    ______________________________________________________________

 EUID=0
```

Figure 6-8 Ishell directory pull-down showing List directory selected

Figure 6-9 is the panel showing selection of the lpp directory within the usr directory.

```
 File  Directory  Special_file  Commands  Help
──────────────────────────────────────────────────────────────────────────────
                              Directory List

Select one or more files with / or action codes.  If / is used also select an
action from the action bar otherwise your default action will be used.  Select
with S to use your default action.  Cursor select can also be used for quick
navigation.  See help for details.
EUID=0   /Z17RC1/usr/
  Type  Filename                                                  Row 1 of 11
_ Dir   .
_ Dir   ..
_ Dir   include
_ Dir   lib
_ Dir   local
/ Dir   lpp
_ Dir   mail
_ Dir   man
_ Dir   sbin
_ Dir   share
_ Dir   spool

Command ===> ________________________________________________________________
```

Figure 6-9 Ishell usr directory showing selection of lpp directory

Figure 6-10 is the panel showing the selection of the dfsms directory within the lpp directory. We had to page down to locate the dfsms directory.

```
 File  Directory  Special_file  Commands  Help
 ──────────────────────────────────────────────────────────────────────────────
                                Directory List

 Select one or more files with / or action codes.  If / is used also select an
 action from the action bar otherwise your default action will be used.  Select
 with S to use your default action.  Cursor select can also be used for quick
 navigation.  See help for details.
 EUID=0   /Z17RC1/usr/lpp/
   Type  Filename                                                 Row 12 of 67
 _ Dir   db2
 _ Syml  db2repl_08_01
 _ Syml  DB2WebQuery
 _ Dir   dce
 _ Dir   dfs
 / Dir   dfsms
 _ Dir   dmh
 _ Syml  egfv1r1
 _ Dir   eim
 _ Dir   EnterpriseDeveloper
 _ Dir   esa
 _ Dir   fw
 Command ===> _________________________________________________________________
```

Figure 6-10 Ishell lpp directory showing selection of dfsms directory

Figure 6-11 is the panel showing the selection of the rmm directory within the dfsms directory.

```
 File  Directory  Special_file  Commands  Help
 ──────────────────────────────────────────────────────────────────────────────
                                Directory List

 Select one or more files with / or action codes.  If / is used also select an
 action from the action bar otherwise your default action will be used.  Select
 with S to use your default action.  Cursor select can also be used for quick
 navigation.  See help for details.
 EUID=0   /Z17RC1/usr/lpp/dfsms/
   Type  Filename                                                   Row 1 of 5
 _ Dir   .
 _ Dir   ..
 _ Dir   bin
 _ Dir   nls
 / Dir   rmm


 Command ===> _________________________________________________________________
```

Figure 6-11 Ishell dfsms directory showing selection of rmm directory

Figure 6-12 is the panel showing the selection of the rmmwebs.txt file within the rmm directory.

```
 File  Directory  Special_file  Commands  Help
 ──────────────────────────────────────────────────────────────────────────
                                Directory List

 Select one or more files with / or action codes.  If / is used also select an
 action from the action bar otherwise your default action will be used.  Select
 with S to use your default action.  Cursor select can also be used for quick
 navigation.  See help for details.
 EUID=0   /Z17RC1/usr/lpp/dfsms/rmm/
   Type  Filename                                                  Row 1 of 13
 _ Dir   .
 _ Dir   ..
 _ Dir   config
 _ Dir   IBM
 _ File  rmmapi.ear
 _ File  rmmcim.txt
 _ File  rmmcimp.jar
 _ File  rmmcimp.mof
 _ File  rmmjapi.jar
 _ File  rmmmsgs.jar
 _ File  rmmSampleWSClient.java
 / File  rmmwebs.txt
 Command ===> _________________________________________________________________
```

Figure 6-12 Ishell rmm directory showing selection of rmmwebs.txt

Example 6-1 shows the contents of the rmmwebs.txt file as provided with DFSMSrmm V1.7. In subsequent paragraphs we show the use of this text as a checklist for installing the DFSMSrmm WEB interface.

Example 6-1 Ishell listing of rmmwebs.txt file

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000000 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
********************************* Top of Data **********************************
**********************************************************************
*                                                                    *
*  z/OS DFSMSrmm V1R7                                                *
*                                                                    *
* PROPRIETARY V3 STATEMENT                                           *
* Licensed Materials - Property of IBM                               *
* "Restricted Materials of IBM"                                      *
* 5694-A01                                                           *
* (C) Copyright IBM Corp. 1992,  2005                                *
* Status = HDZ11KO                                                   *
* END PROPRIETARY V3 STATEMENT                                       *
**********************************************************************

These instructions are provided to assist you to deploy
the RMM Web Service and to help you start to use it.
```

```
Initial version V1R7
====================

 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000021 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
-   The RMM Web Service is designed to enable usage of the
    RMM High Level Language API from any platform.

-   Any valid RMM TSO subcommand can be issued via Web Service,
    returning data in XML format.

-   A z/OS Websphere Application Server is needed to deploy
    the RMM Web Service.

-   You generate your own client application to access
    the RMM Web Service. A sample client is available in
    /usr/lpp/dfsms/rmm/rmmSampleWSClient.java.



Deployment of the RMM Web Service
=================================

1.) Start a browser, for example Microsoft Internet Explorer, and go to
    the Administrative Console of your z/OS Websphere Application
    Server. This can be done by entering http://x.xx.xxx.xxx:pppp/admin
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000042 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
    in the address field of the browser (where x.xx.xxx.xxx is the
    IP address and pppp is the port the server is running on).

2.) Login and choose 'Enterprise Applications',
    then 'Install New Application'.
    Choose 'Server path' and enter the location of the EAR file in HFS :
    '/usr/lpp/dfsms/rmm/rmmapi.ear'
    Then click 'Next'.

3.) Check 'Generate Default Bindings' and click 'Next'.
    On the next page enter the 'Application Name' you want your
    Web Service to use and click 'Next' again.
    Check 'Web Module' and click 'Next'.
    Then check 'Module' and click 'Next'.
    Click 'Finish'.
    You should see : 'Application xxxxxxx installed successfully'
    and you will be asked to 'Save to Master Configuration'.

4.) After you saved the configuration, choose 'Enterprise Applications'.
    In the list of applications you will now see your RMM Web Service.
    Check the box in front of it and click 'Start'.
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000063 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
```

```
    After the status symbol has changed to 'started', your RMM Web
    Service is available.



Usage of the RMM WebService Sample Client
=========================================

1.) Download the following file from z/OS (HFS) to your PC:
    /usr/lpp/dfsms/rmm/rmmSampleWSClient.java (SMP/E part name EDGSJWS1)
    This source file contains sample code to access the RMM Web Service.

    Use FTP with the following commands :
  OPEN your_server
  USER your_userid your_password
  ASCII
  GET /usr/lpp/dfsms/rmm/rmmSampleWSClient.java  _rmmSampleWSClient.java
  QUIT

    Check that special characters
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000084 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
         "Ý" (left square bracket)
         "¨" (right square bracket)
         "{" (left brace)
         "}" (right brace)
         "|" (vertical bar)
         "/" (slash)
         "\" (backslash)
    are downloaded correctly.

    If you are reading this text in a host session and the characters
    are not displayed correct, change your terminal emulator program
    to run with codepage=1047, since all text files were initially
    created with this codepage.

    In an editor on your workstation the downloaded file for example
    must have square-brackets in the line below, after 'byte' :
    +-----------------------------------------------------------------+
    public synchronized byteÝ¨ runCommandXml(byteÝ¨ b_cmd) ....
    +-----------------------------------------------------------------+
    In case the characters are incorrect, open an OMVS session on your
    host system, navigate to the HFS-directory where the file resides
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000105 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
    and execute the following translation utility from the command
    shell:

         iconv -f IBM-1047 -t IBM-xxx filein > fileout

    where IBM-xxx is the target codepage, that is compatible to your
    FTP program. If you don't know, try starting with IBM-924, which
```

```
    works well at our development location.
    +----------------------------------------------------------------+
    I   NOTE, that "filein" and "fileout" must NOT be identical !   I
    +----------------------------------------------------------------+

    After translation re-download the file again and ensure the
    characters are correct now.

    Rename the file to remove the leading underscore sign at the
    beginning of the file-name.

2.) Import the file into your development tool.

3.) Make sure the following libraries are added to your CLASSPATH:
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000126 Col 001 072
 Command ===>                                                  Scroll ===> PAGE

    Name          Version   Download location
    -----------------------------------------------------------------
    j2ee.jar       1.3.1    http://java.sun.com/j2ee/index.jsp
    soap.jar       1.0.0    http://www.apache.org/dyn/closer.cgi/ws/soap
    xercesImpl.jar 2.5.0    http://xml.apache.org/dist/xerces-j/
    mail.jar       1.3.1    http://java.sun.com/products/javamail/
                                                    downloads/index.html

4.) Build the program.

5.) Run the program specifying the required parameters as detailed
    below;
    As first parameter you need to pass the IP address, including the
    port of the server, which provides the Web Service.
    As second parameter you need to pass the RMM TSO subcommand, that
    you want to execute.
    As third parameter you need to pass the name of the file into which
    the resulting data is written.
    example:
    'java rmmSampleWSClient 9.10.111.122:9080
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 BROWSE -- /Z17RC1/usr/lpp/dfsms/rmm/rmmwebs.txt ---- Line 00000147 Col 001 072
 Command ===>                                                  Scroll ===> PAGE
                                  "LISTCONTROL ALL"  C:\temp\results.dat'
6.) (Note: This may be done via SMP/E instead)
    The Web Service makes use of a C++ DLL from the MVS Link List.
    It is a program object called EDGXHCLL.
    To make this DLL available for the Web Service, you need to install
    a link in the z/OS Websphere Application Servers' libpath.
    Go to the libpath in HFS, for example:
    /WebSphere/V5ROM2/AppServer/lib ,
    and type "ln -e EDGXHCLL libEDGXHCLL.so".
    This establishes an external link to the DLL in LINKLST.

7.) You are now ready to exploit the RMM Web Service.
    You may expand the sample source or write your own application based
    on the definitions and files contained in the RMM Enterprise ARchive
```

```
    file called rmmapi.ear.
******************************** Bottom of Data ********************************
```

6.4.3 Implementation of DFSMSrmm Web service

Contact the WAS administrator before starting this process to obtain the following information about resources that are available for your use:

- The Web address
- The port to use

DFSMSrmm Web service setup

To set up the DFSMSrmm Web service for the management of DFSMSrmm tasks, we started a Web browser to go to the administrative console of the z/OS WebSphere Application server. The specified Web address was 9.12.6.70, and the port number was 19080, specified in the Web browser (Internet Explorer®) as http:/9.12.60.70:19080/admin/. The /admin/ part of the address automatically changed to /ibm/console/.

The minimum level of WebSphere Application Server that is suitable for use by the DFSMSrmm Web interface is 5.0.2.

Note: The level of WebSphere Application Server used for this install was V6. This resulted in some of the steps and responses being slightly different than those documented in *DFSMSrmm Implementation and Customization Guide,* SC26-7405, and in the rmmwebs.txt file shown above. The differences were not sufficiently different to cause major confusion, mainly being a slightly different order of panels.

In Figure 6-13 we show the WebSphere Application Server Login screen. The user ID that is required is not a security control, it is a reference for tracking any changes that are made while using WAS to change configuration data.

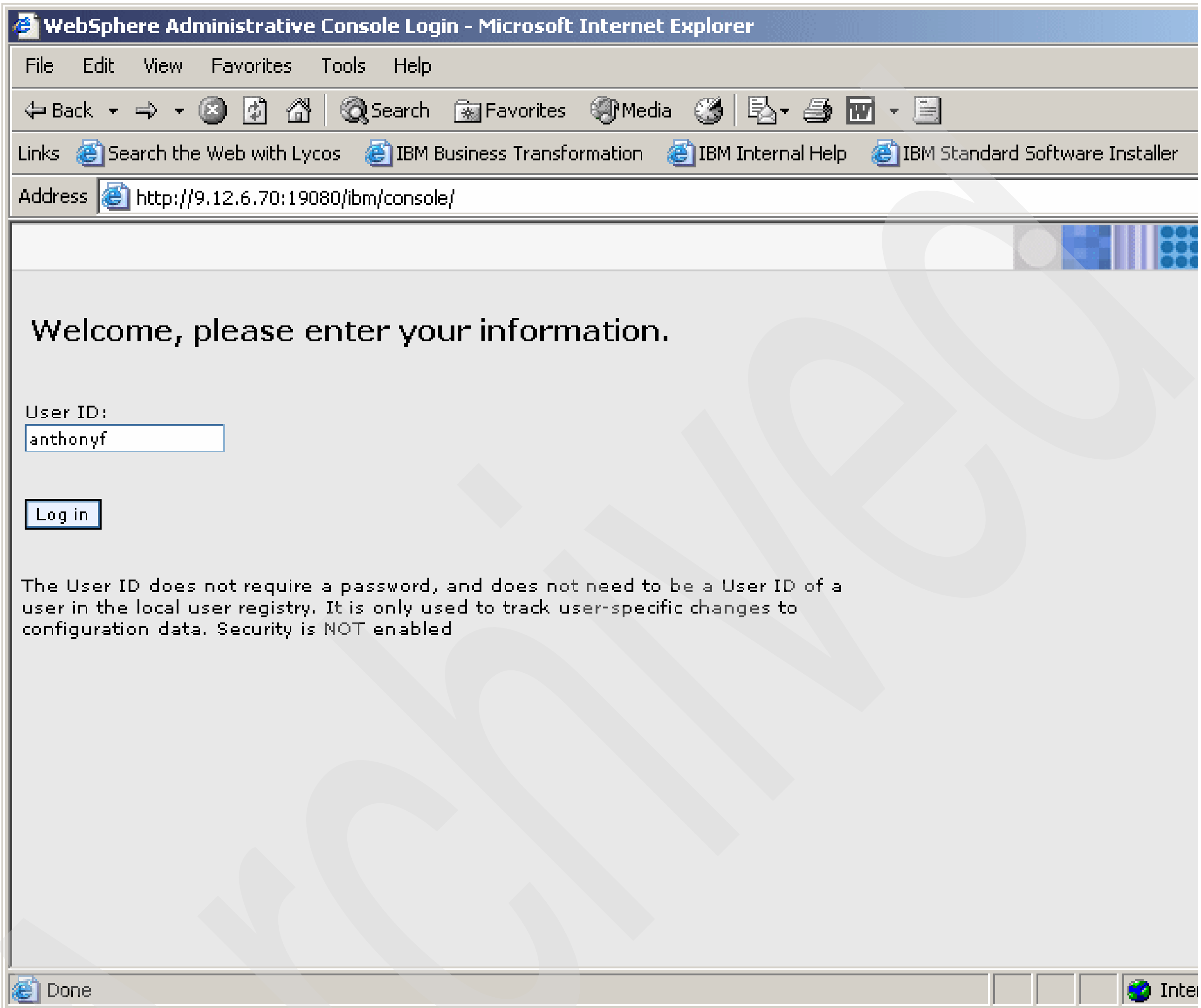


Figure 6-13 WAS login screen

In Figure 6-14 we show the WebSphere Application Server home page. It has been customized with the user ID entered on the login screen, so it now shows `Welcome anthonyf` in the upper left-hand portion of the panel.

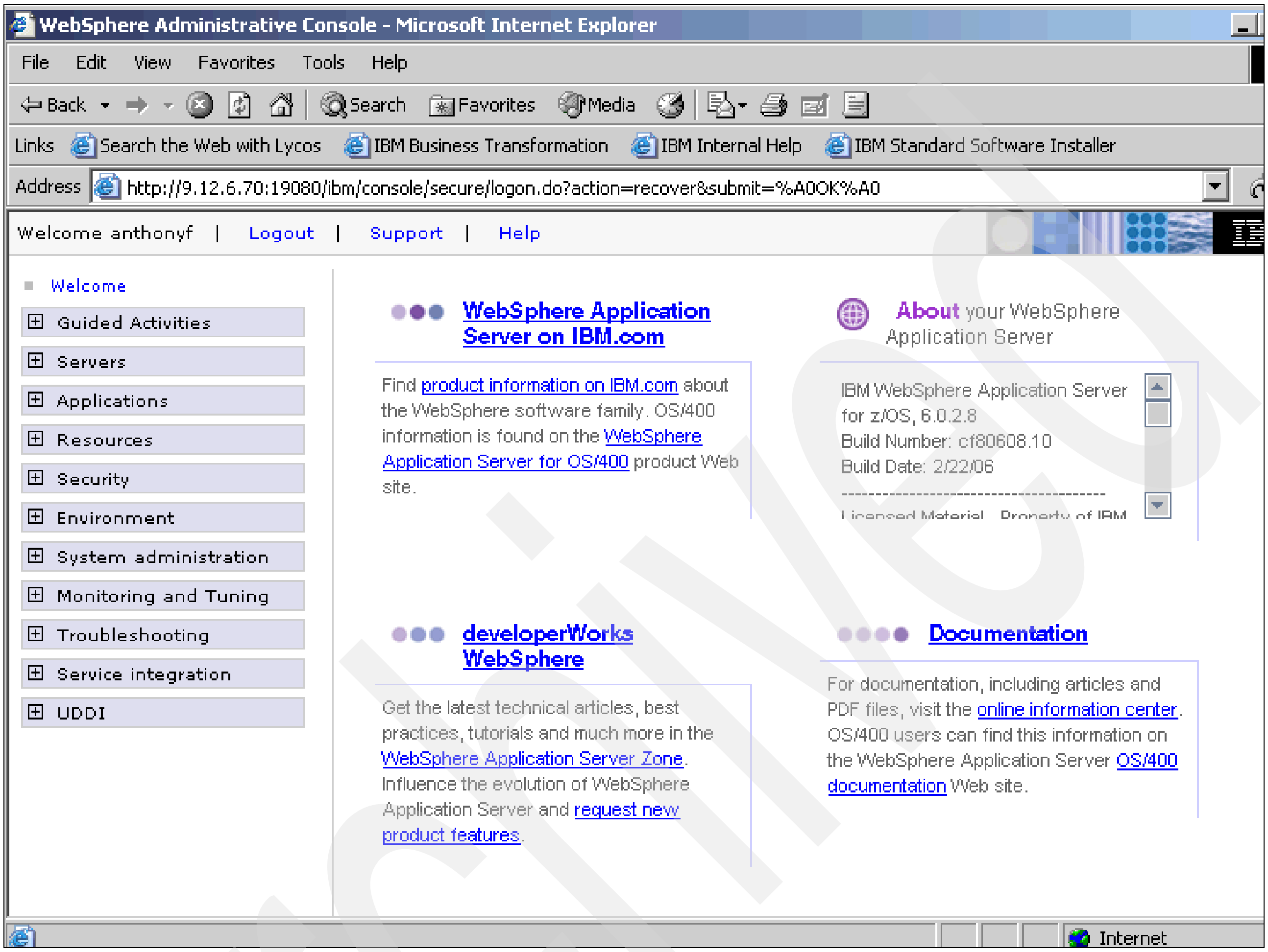


Figure 6-14 WAS home page

Figure 6-15 shows that the Applications menu task has been expanded. We want to install a new application, so select **Install New Application** from the expanded Applications selection.

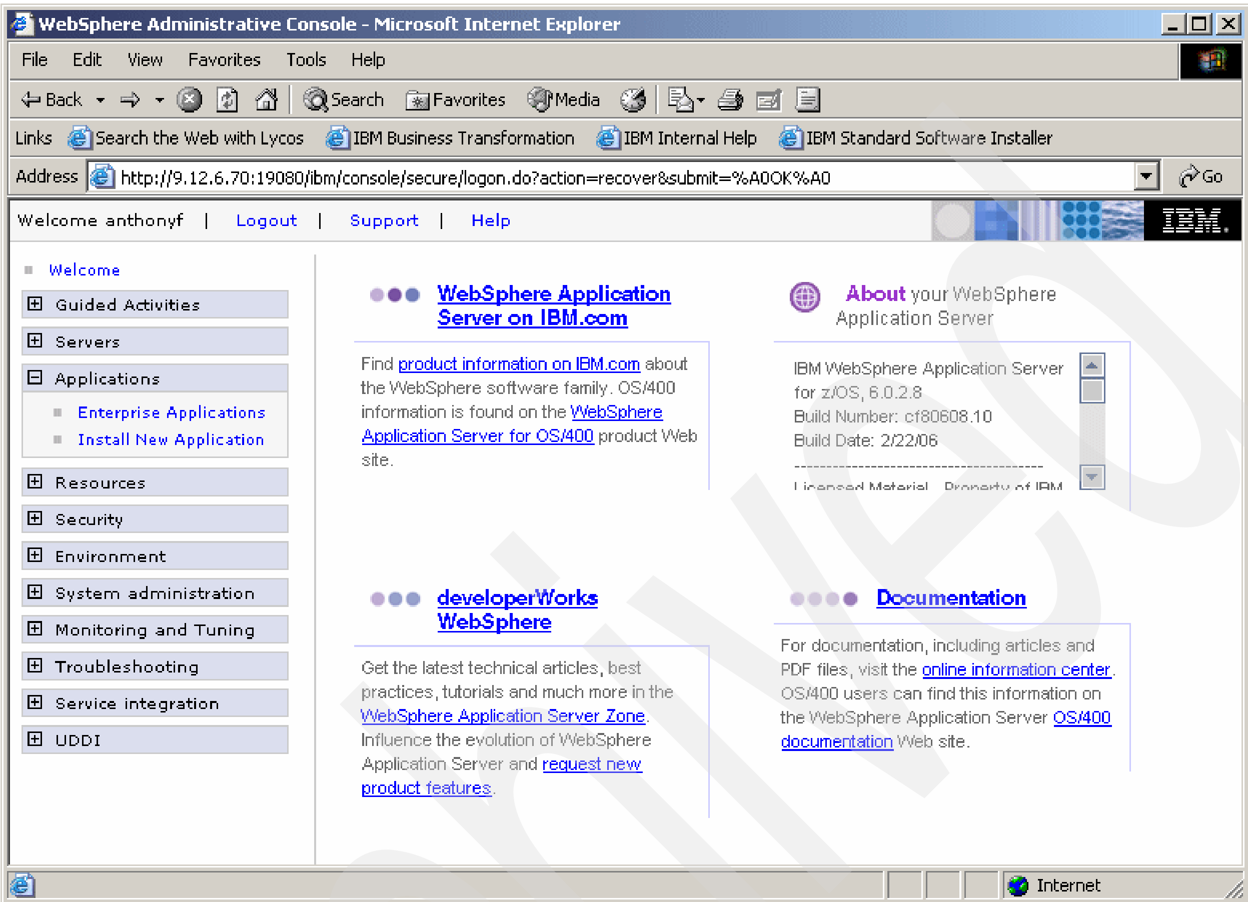


Figure 6-15 WAS activity selection panel with Applications expanded

Figure 6-16 shows the WAS Install New Application initial panel. You must locate the DFSMSrmm EAR file that contains the required resources and specify its name as the remote file system. To do this you could have typed it in directly from the information found when listing the contents of the /usr/lpp/dfsms/rmm/ file, as shown in Figure 6-12 on page 149. In this scenario we use the WAS facilities to locate the file.

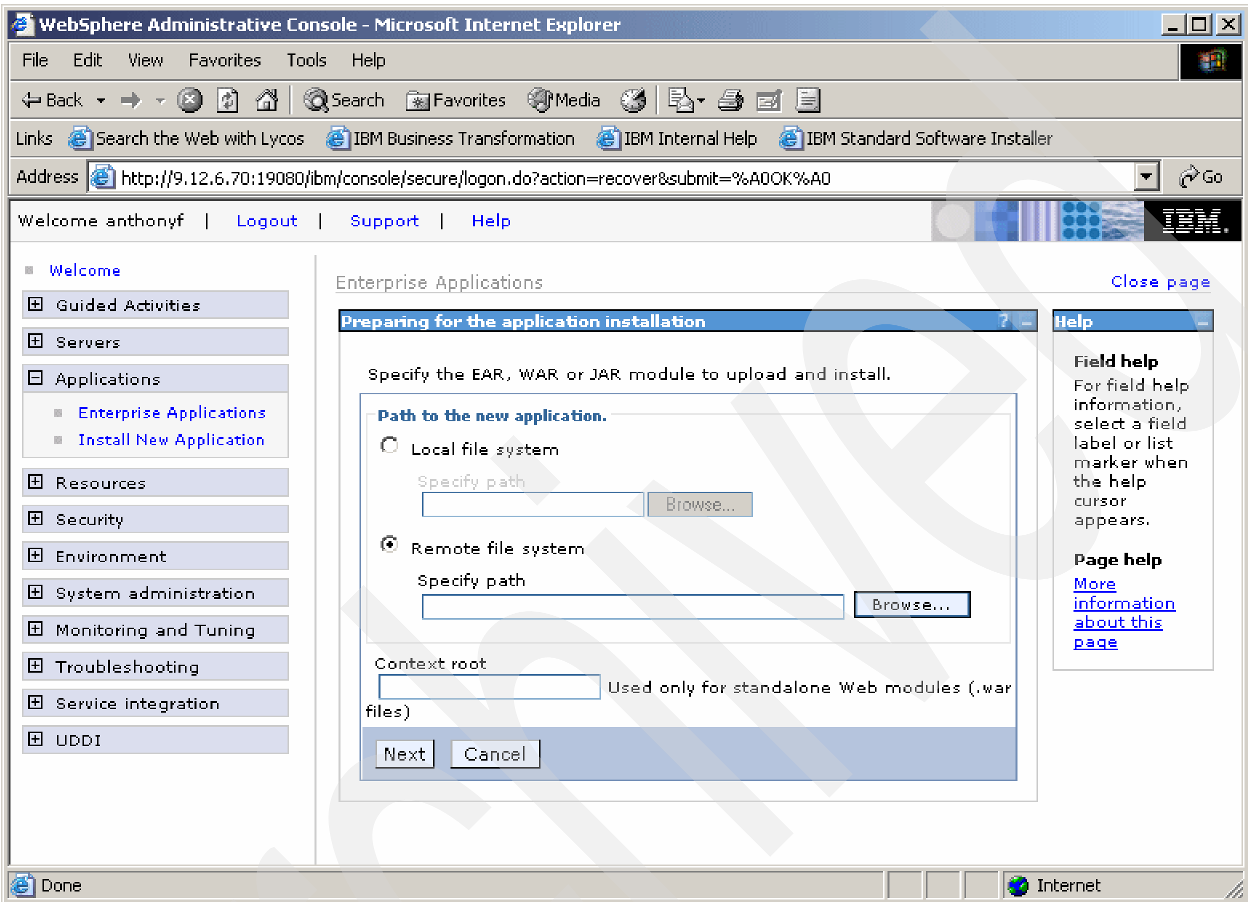


Figure 6-16 WAS New Application initial screen

Figure 6-17 shows the first directory of the WAS installed system, shown as nd6331.

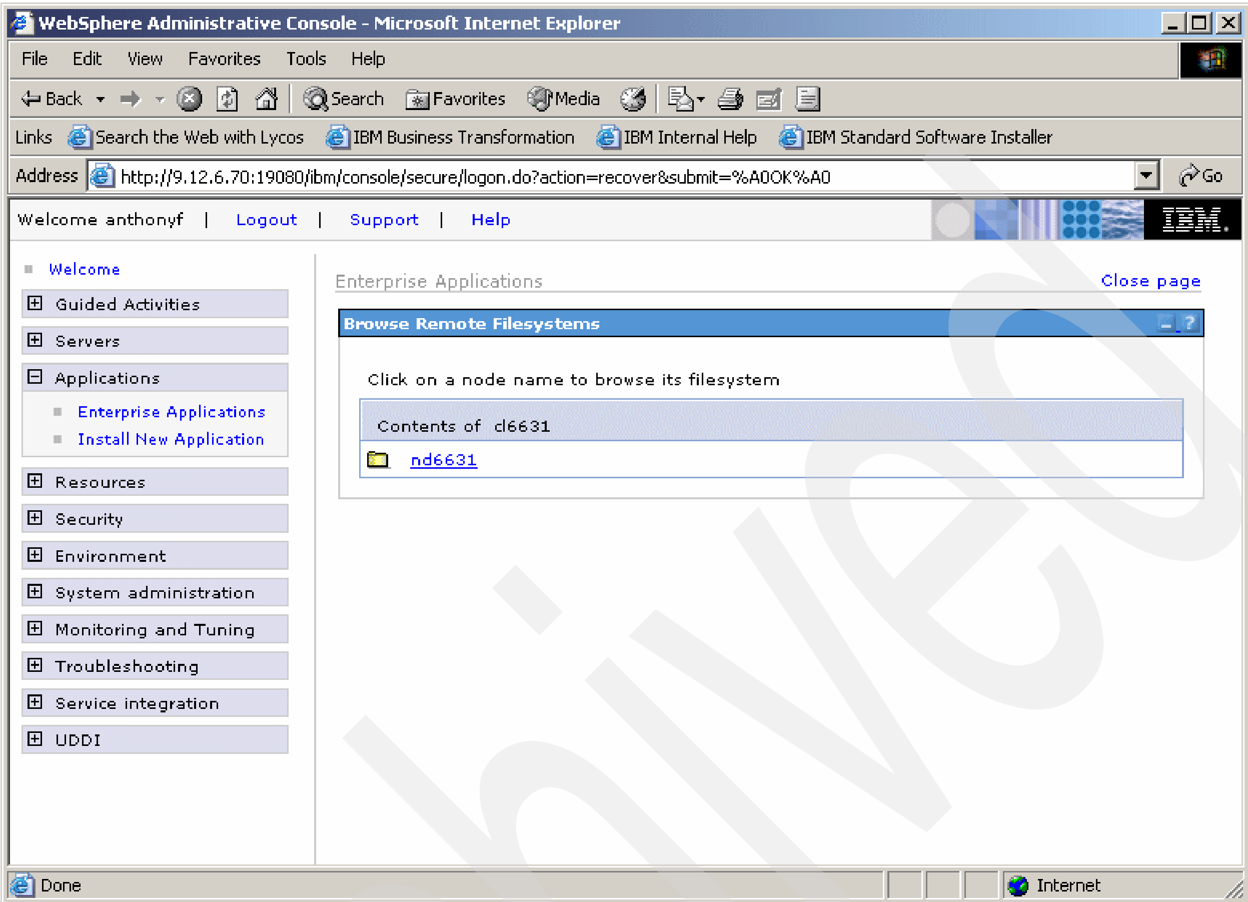


Figure 6-17 WAS remote file system directory (panel 1 of 10)

Figure 6-18 shows the WAS directory contents of nd6331. You must search the directory structure below the line shown as cellContext marked by the stylized L.

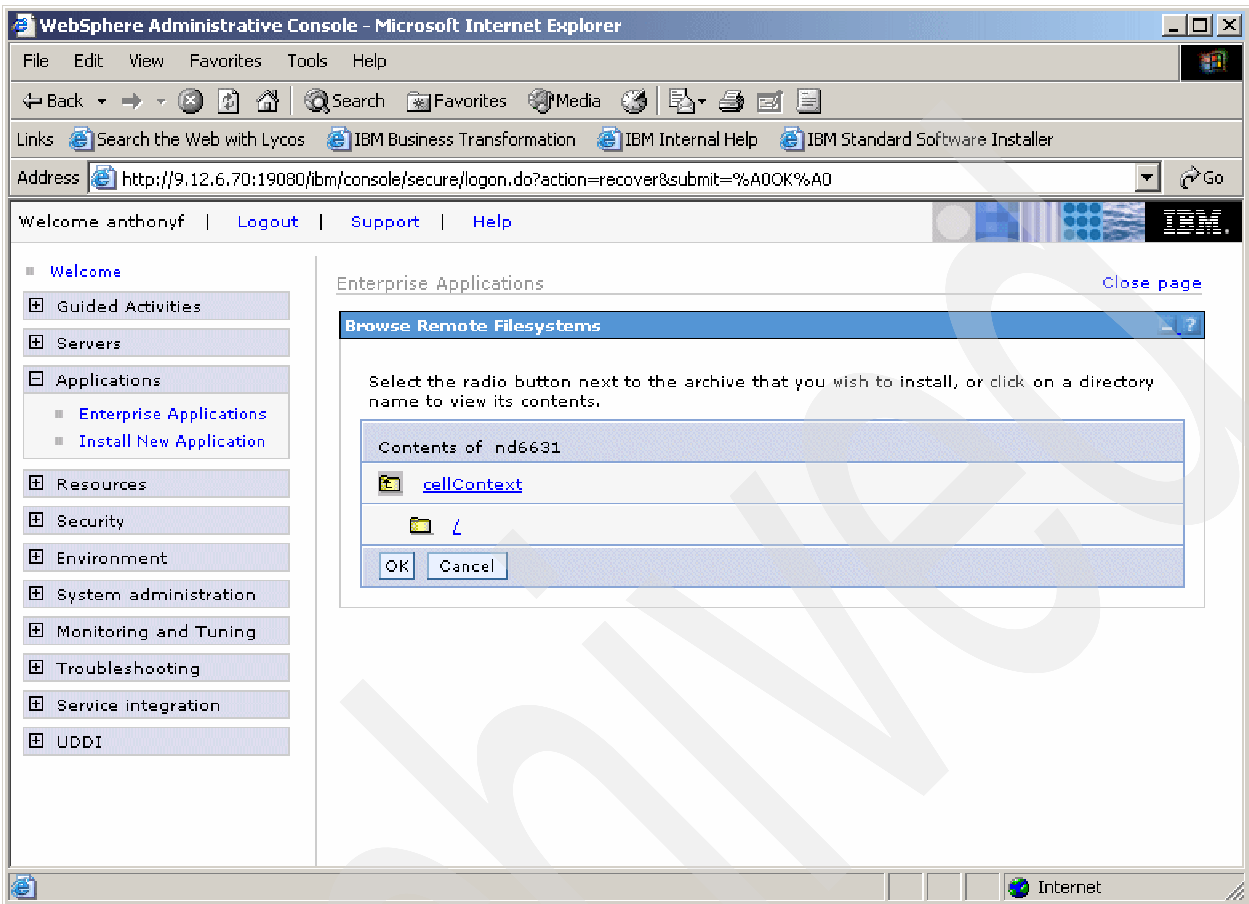


Figure 6-18 WAS remote file system directory (panel 2 of 10)

Figure 6-19 shows the first entries of the nd6631 structure. You must find the usr entry. This corresponds to the start of the directory structure that was followed in 6.4.2, “Prerequisites for DFSMSrmm WEB service installation” on page 145. Page down until the usr directory is located.

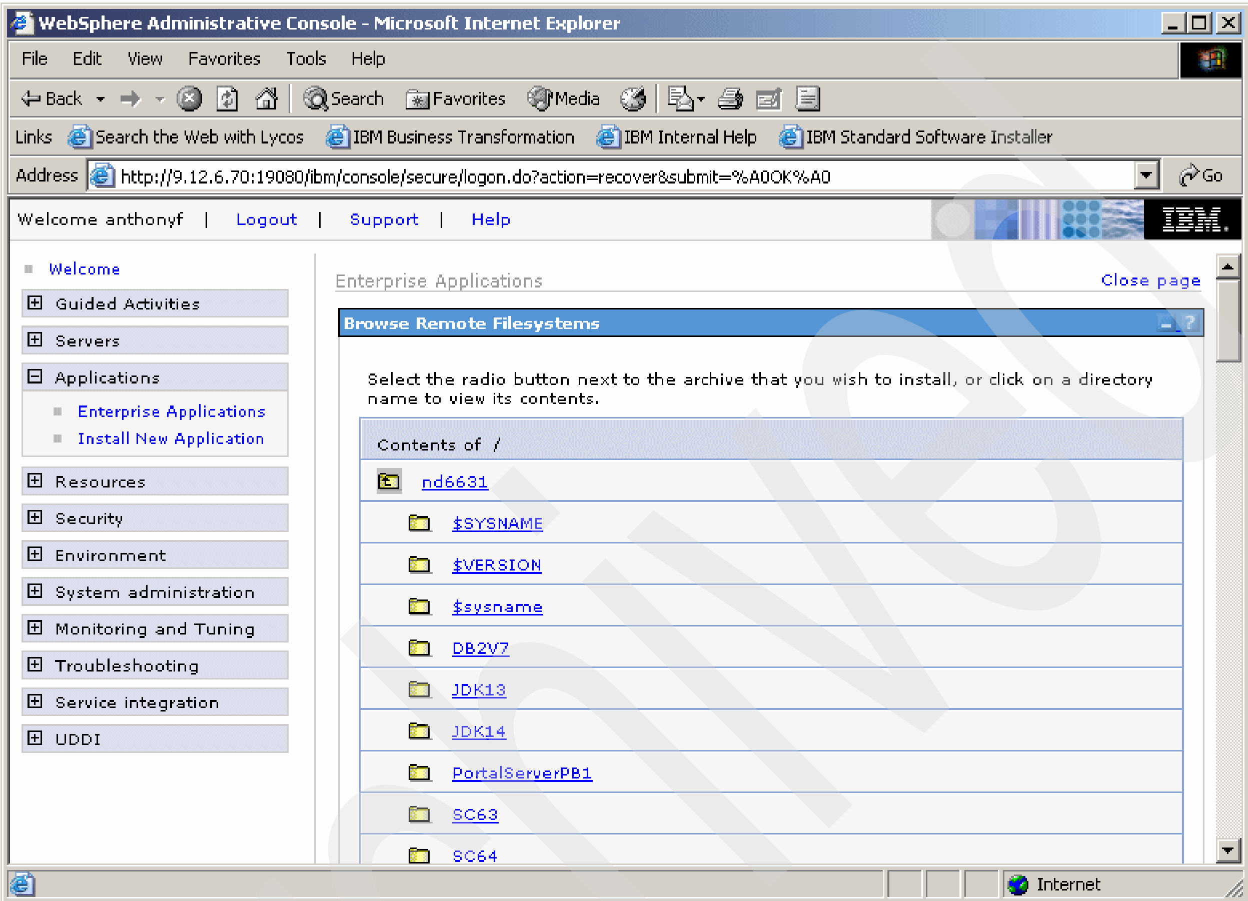


Figure 6-19 WAS remote file system directory (panel 3 of 10)

In Figure 6-20 we show the panel that contains the usr entry. The entries under nd6331 are not in any particular order. Find the usr entry and select it.

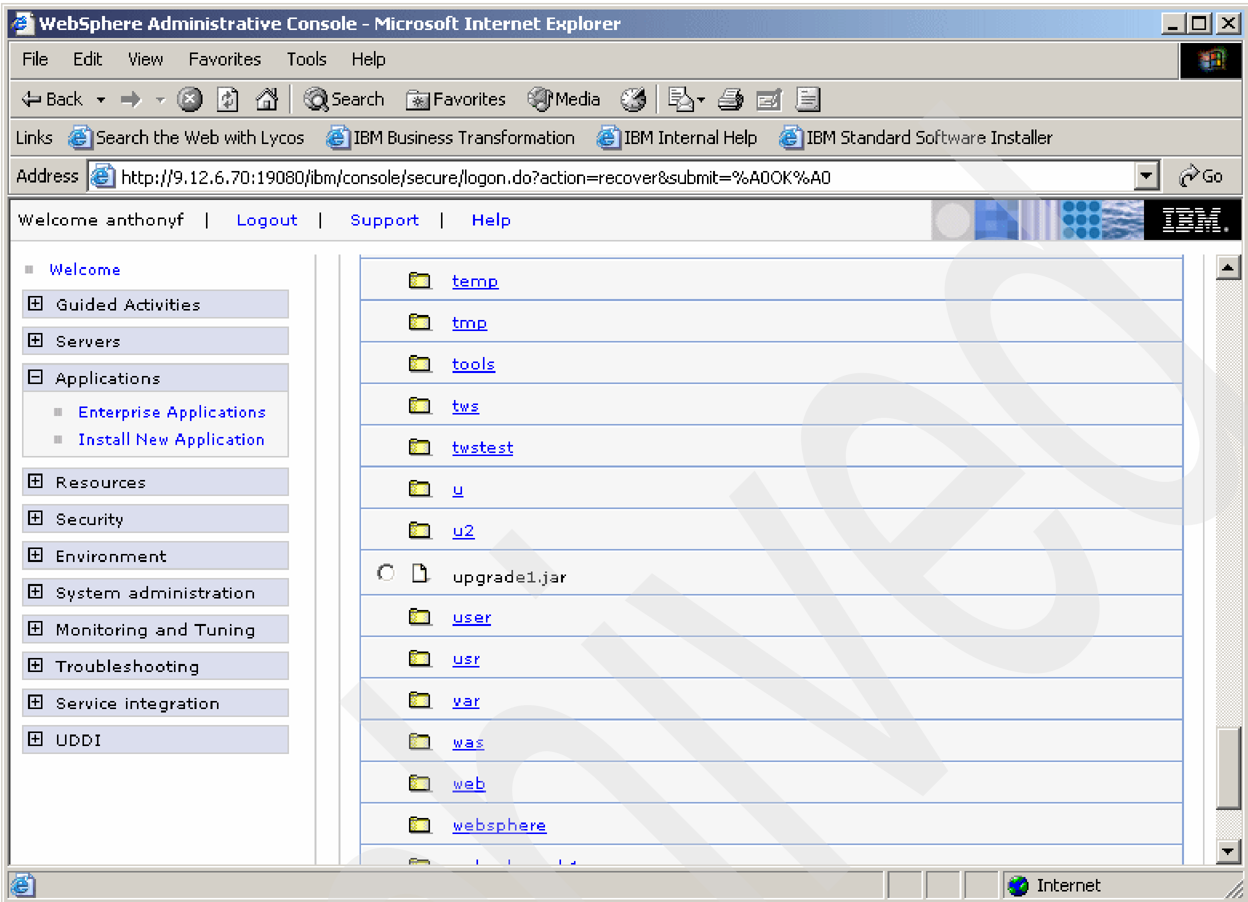


Figure 6-20 WAS remote file system directory showing usr entry (panel 4 of 10)

In Figure 6-21 we show the lpp directory under the usr directory. We paged down until we found the lpp directory and then selected it.

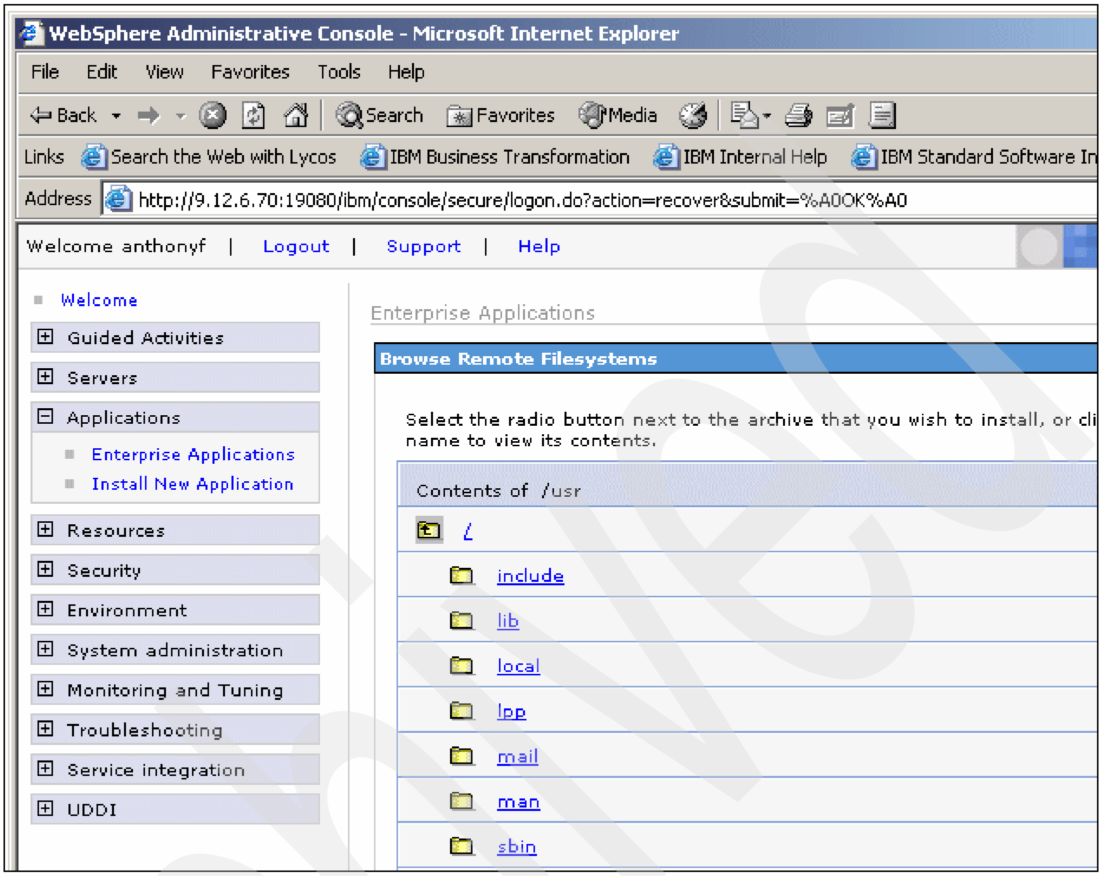


Figure 6-21 WAS remote file system directory showing lpp entry (panel 5 of 10)

Figure 6-22 shows the first part of the lpp directory structure. To locate the dfsms directory, page down until you locate it and then select it.

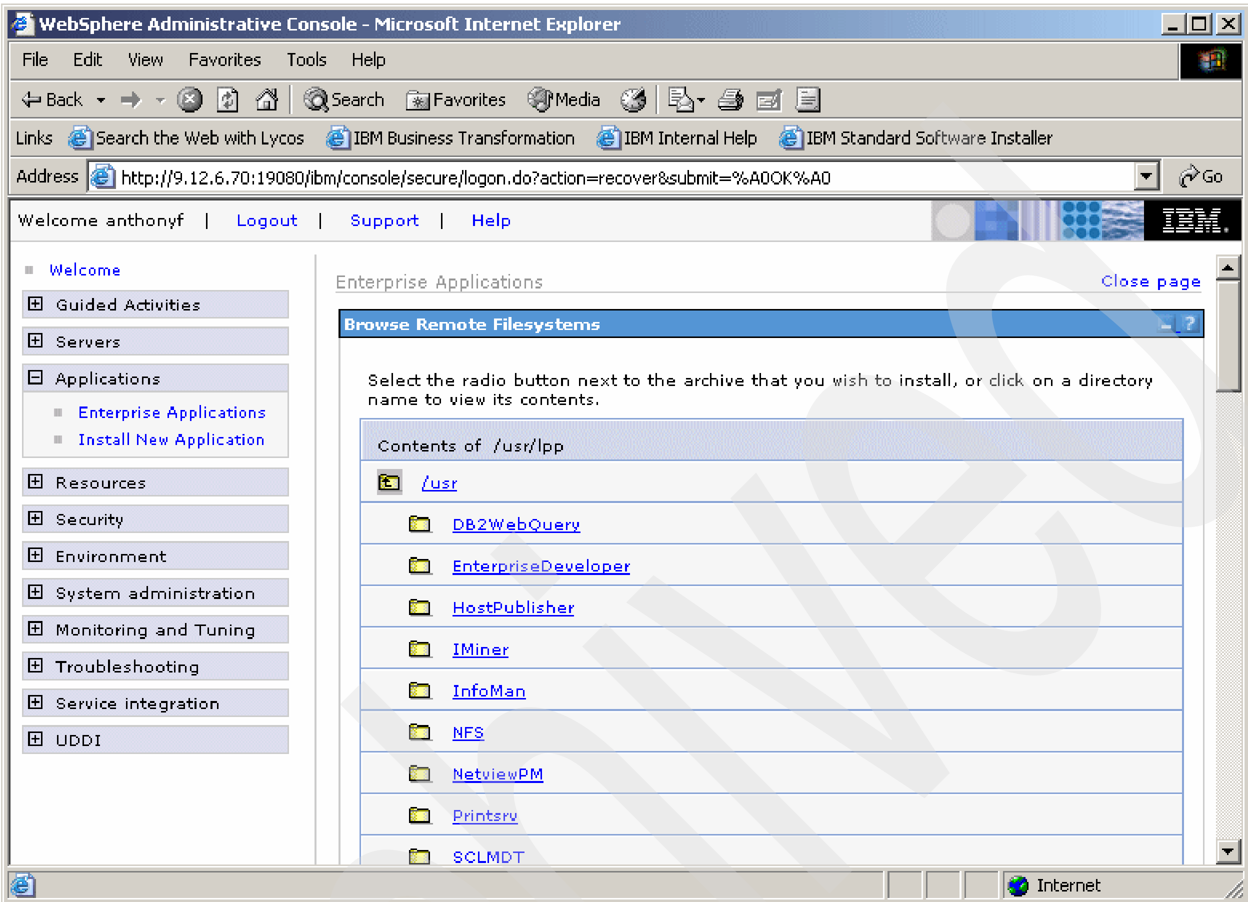


Figure 6-22 WAS remote file system directory showing lpp entry (panel 6 of 10)

Figure 6-23 shows the contents of the dfsms directory. Select the **rmm** directory.

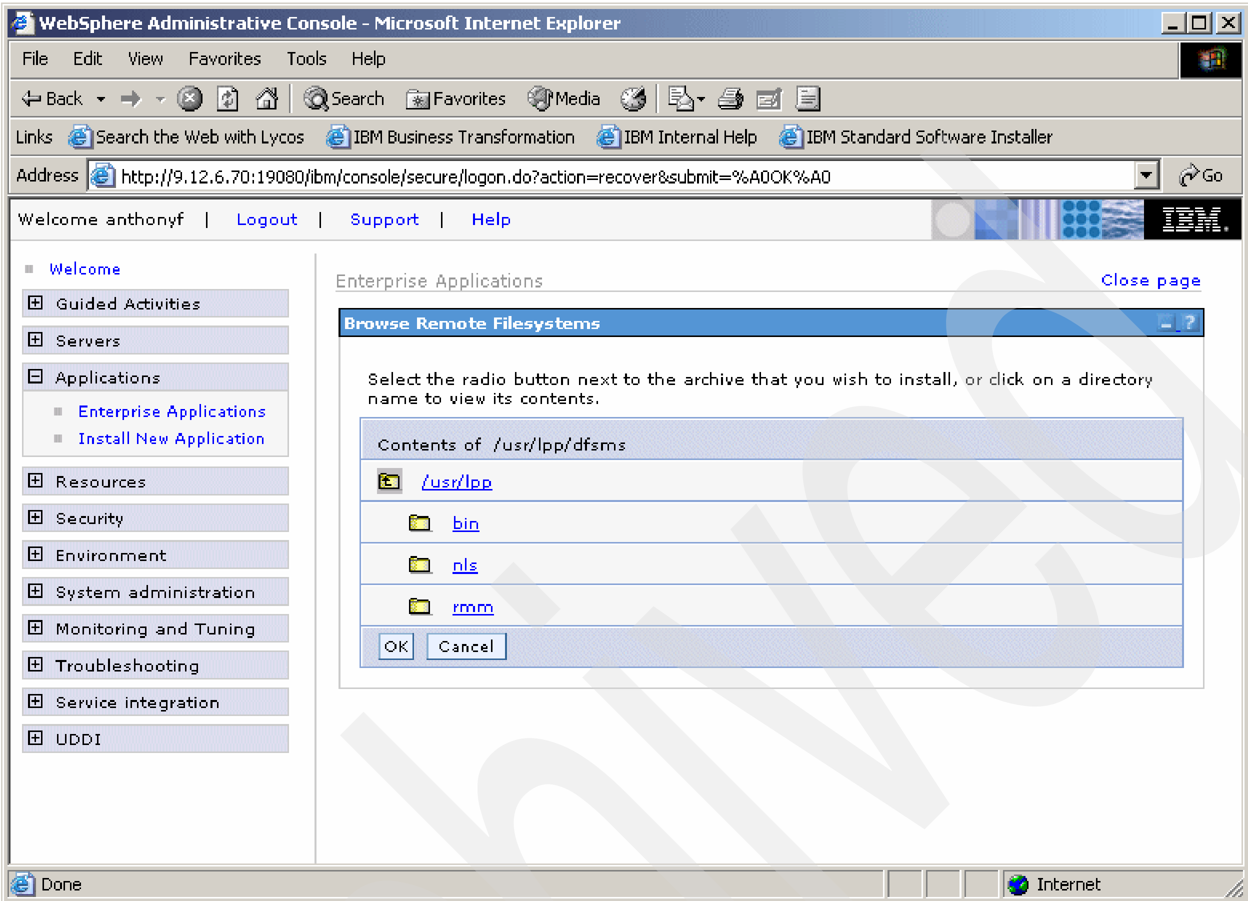


Figure 6-23 WAS remote file system directory showing dfsms entry (panel 7 of 10)

Figure 6-24 shows the rmm directory structure that contains, amongst other items, file rmmapi.ear, which is the resource file that we wish to install. To start the install we select the radio button next to rmmapi.ear and click **OK**.

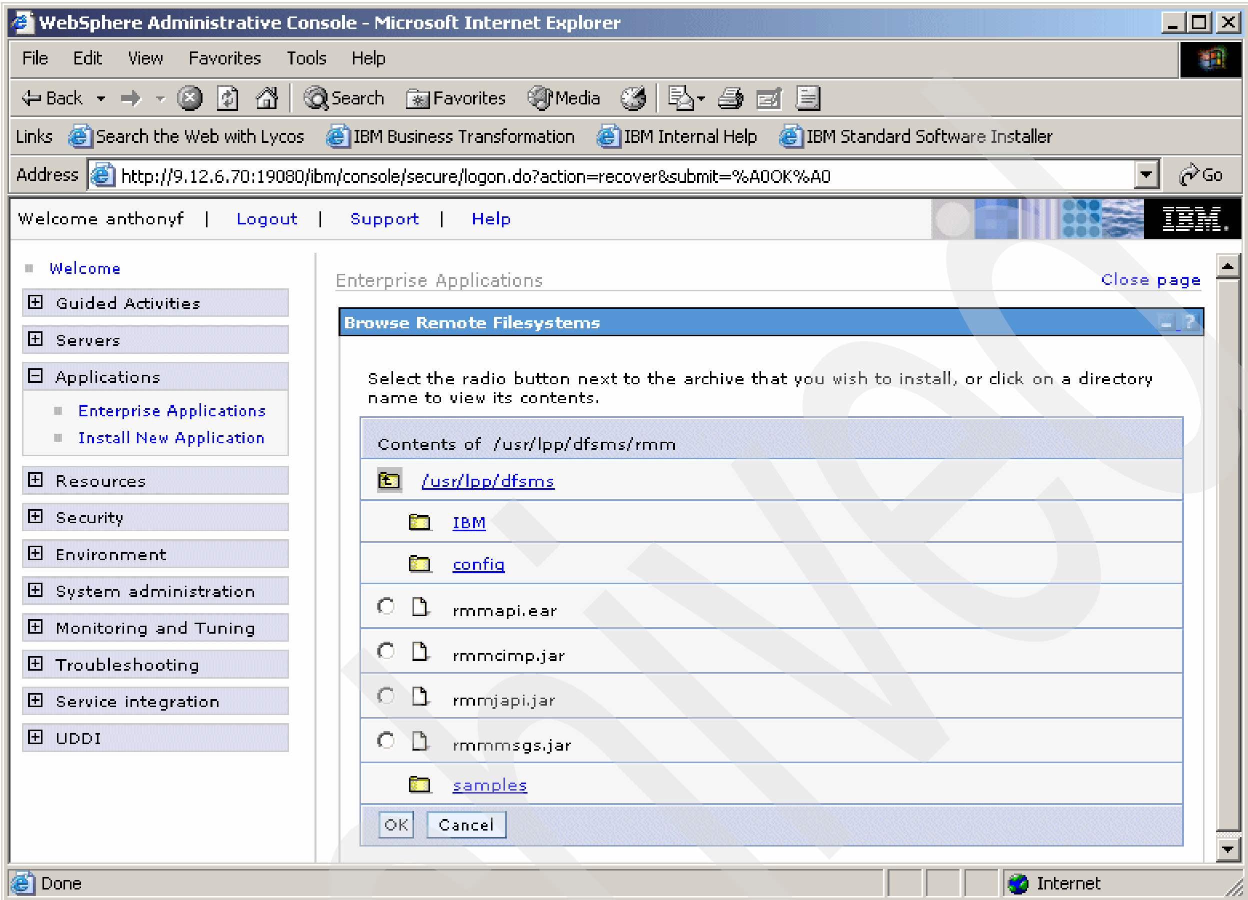


Figure 6-24 WAS remote file system directory showing dfsms entry (panel 8 of 10)

In Figure 6-25 we show the New Application Install panel with the remote file system path name filled in. Select **Next** to continue the install process.

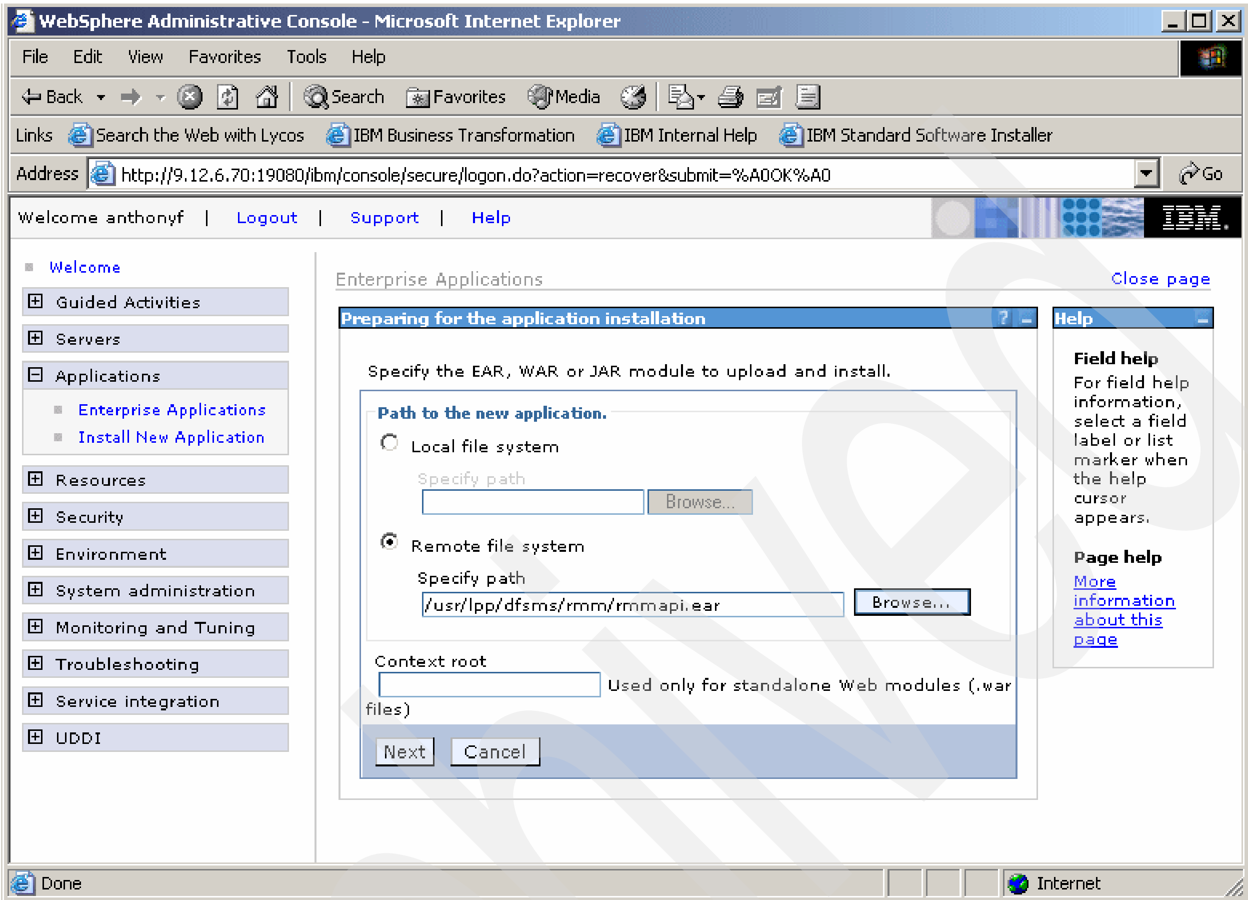


Figure 6-25 WAS remote file system directory showing dfsms entry (panel 9 of 10)

Figure 6-26 shows the next panel in the install the sequence. Select the option to **Generate Default Bindings** (which was not preselected), then click **Next** to continue.

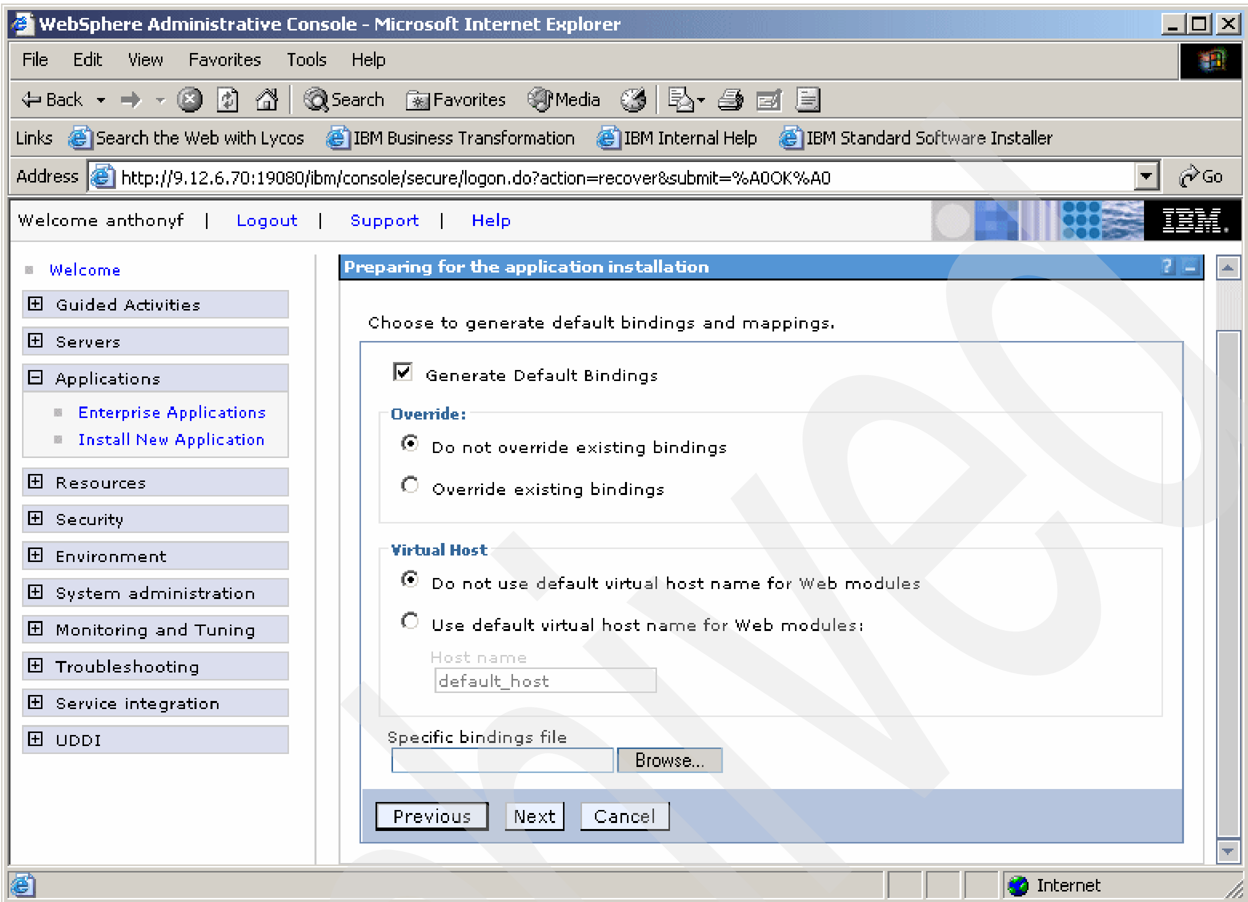


Figure 6-26 WAS remote file system install - generate default bindings

In the panel shown in Figure 6-27 we entered the application name as RmmApiEAR. Scroll down to the bottom of the panel and click **Next** to continue.

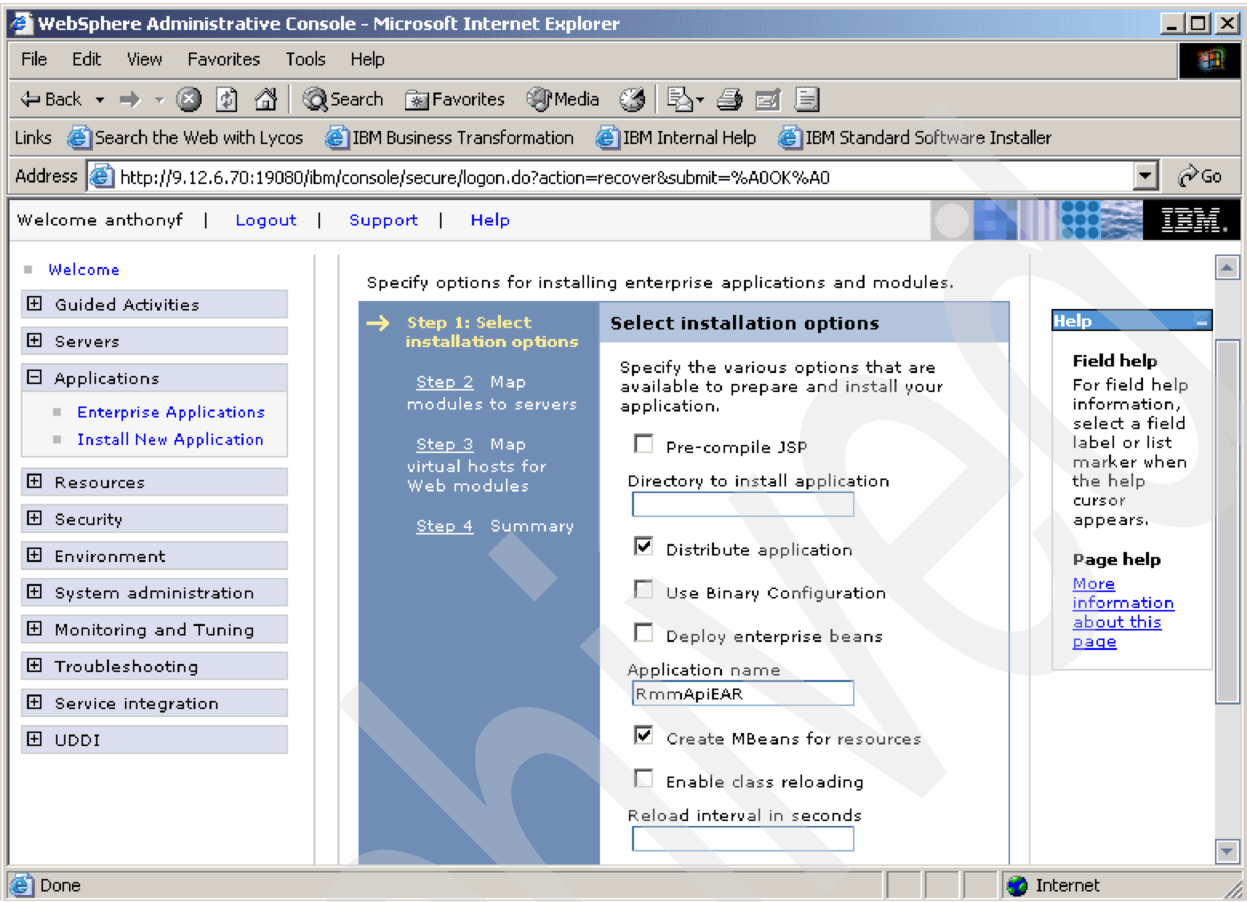


Figure 6-27 WAS remote file system install - specify application name as RmmApiEAR

Figure 6-28 shows the selection of the cluster that is *not* the portal, and the module name as RmmJApi. Click **Next** to continue.

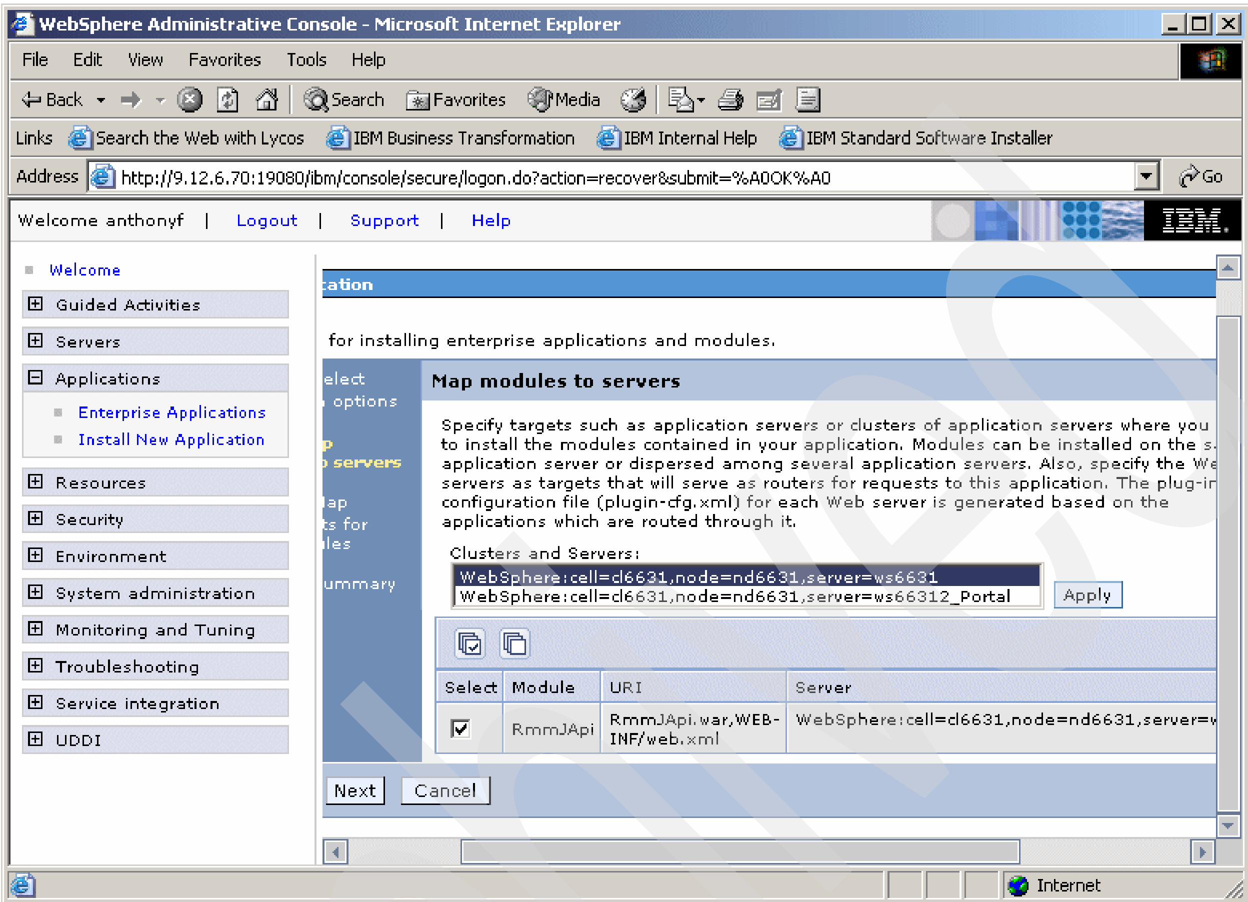


Figure 6-28 WAS remote file system install - specify cluster and module name

Figure 6-29 shows the specification of the Web module name as RmmJapi. Click **Next** to continue.

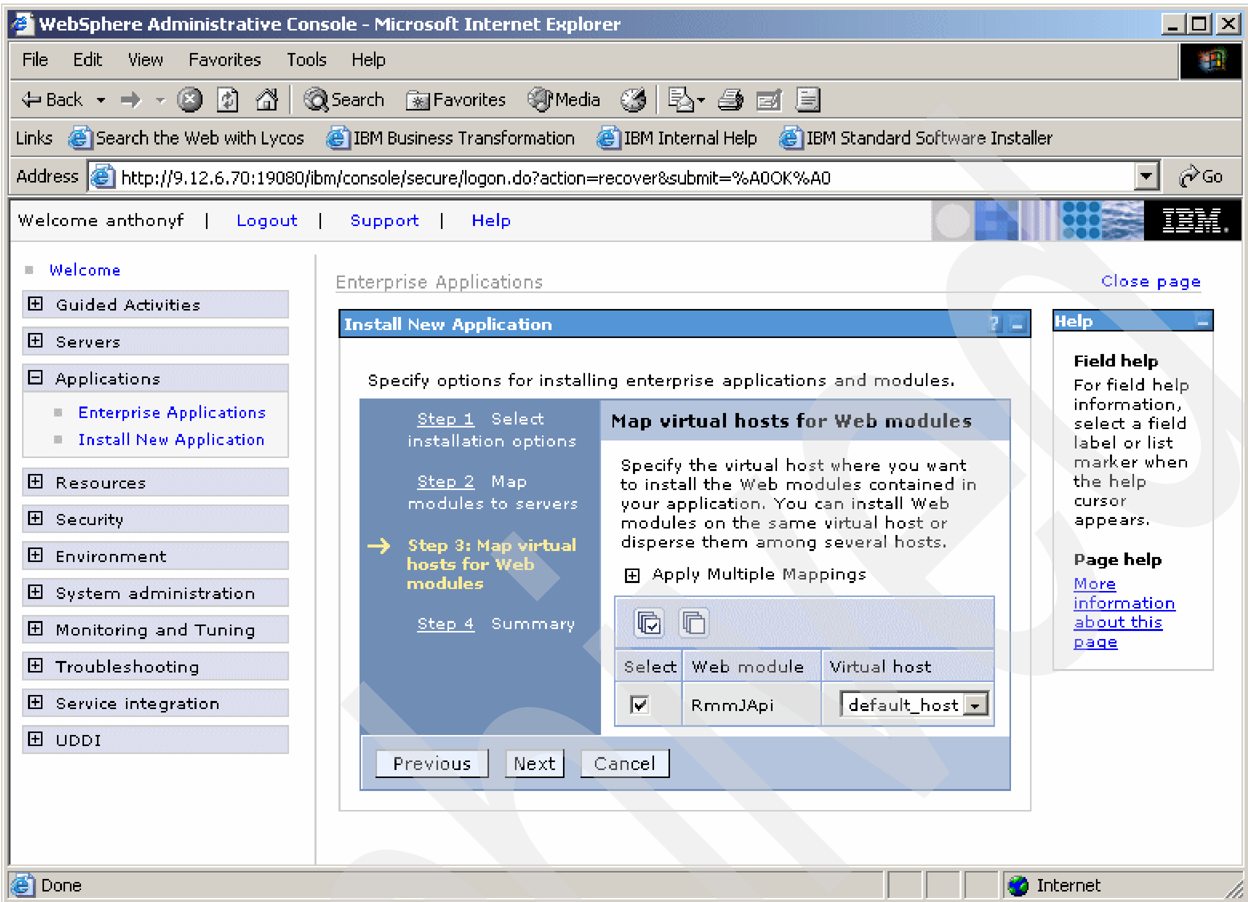


Figure 6-29 WAS Remote File System Install - specify Web name as RmmJApi

Figure 6-30 and Figure 6-31 on page 172 show the summary of the installation specifications. Select **Finish** to complete the install.

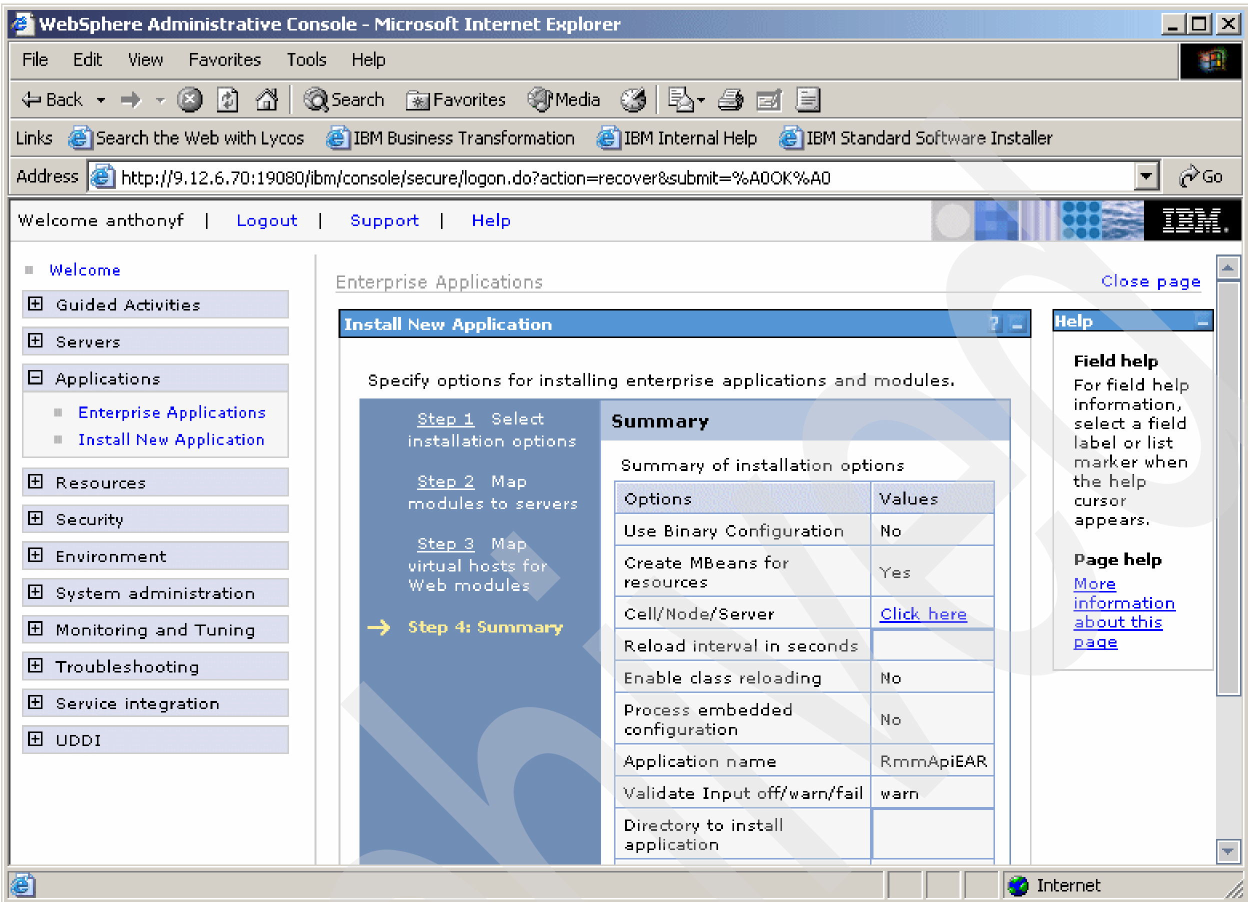


Figure 6-30 WAS remote file install summary - (panel 1 of 2)

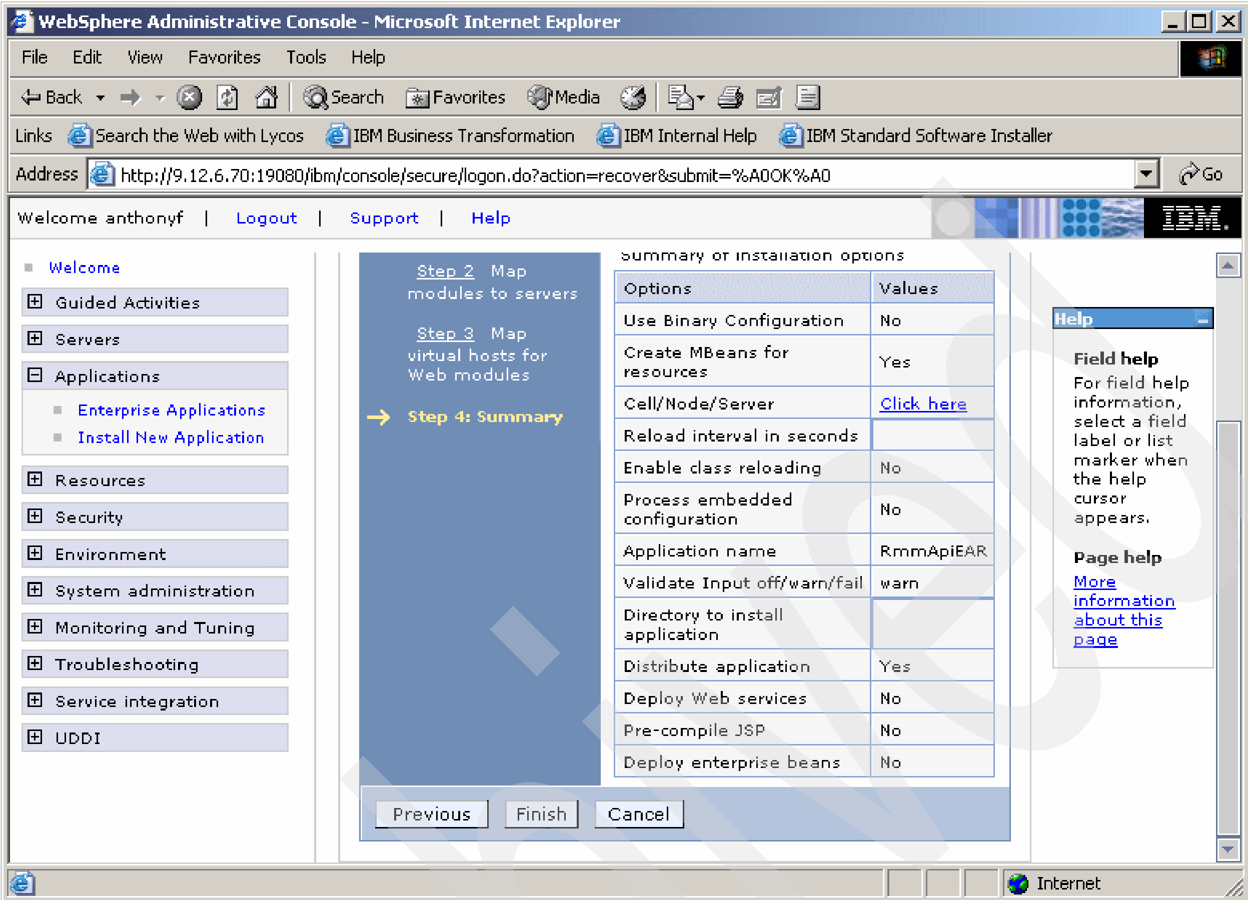


Figure 6-31 WAS Remote File Install Summary - (panel 2 of 2)

At this point be patient while the installation processing completes.

Figure 6-32 and Figure 6-33 on page 174 show the installation summary messages. At the end of the messages select the highlighted option to **Save to Master Configuration**.

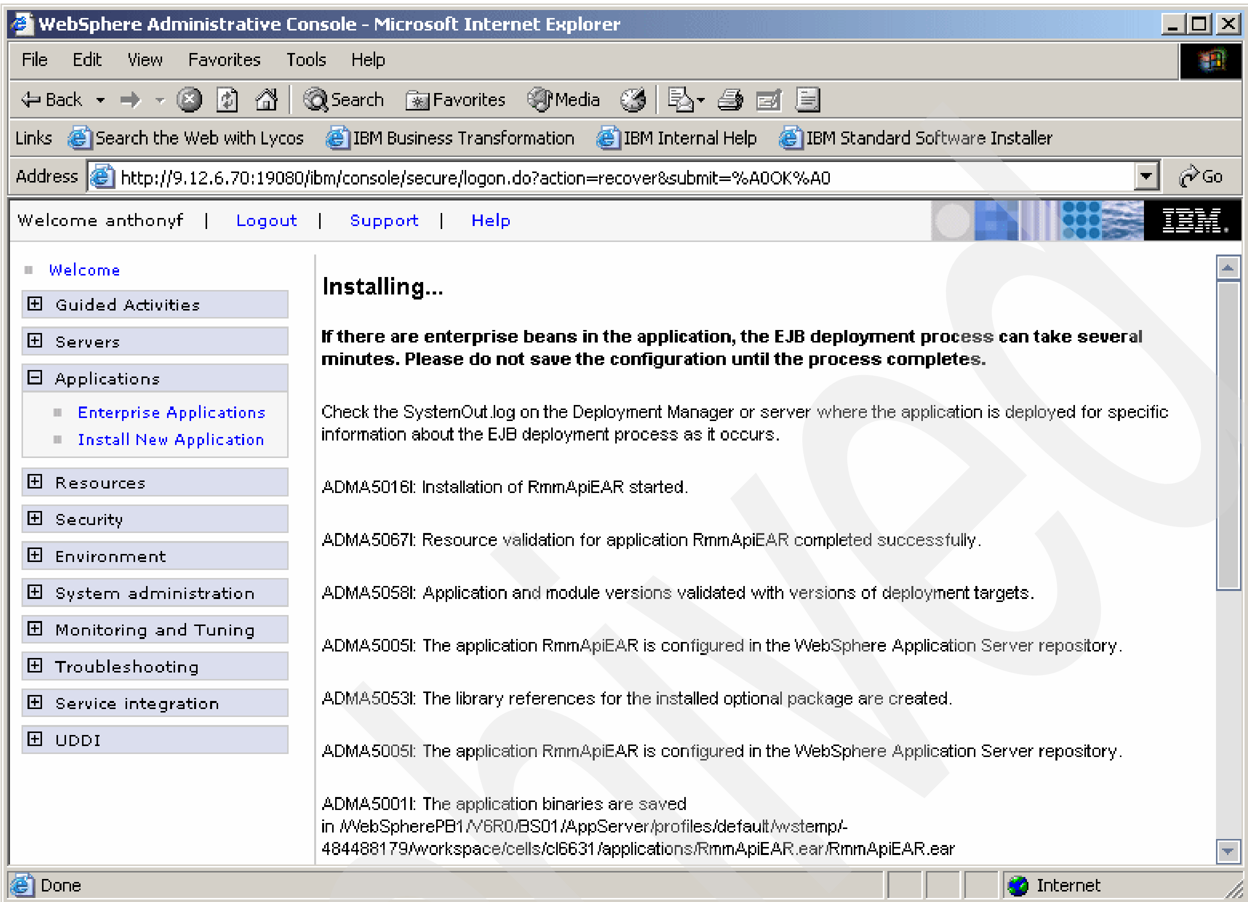


Figure 6-32 WAS application installation summary messages (panel 1 of 2)

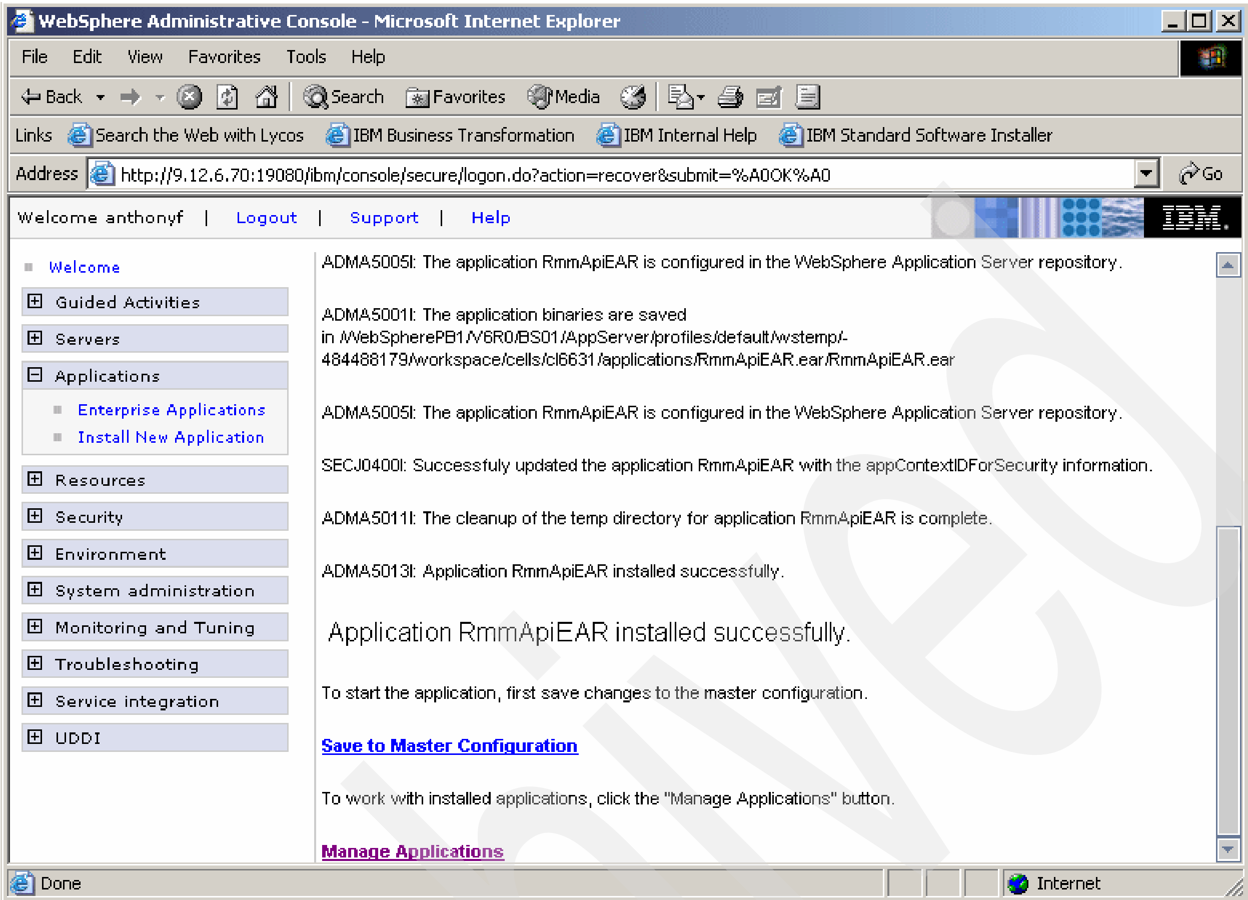


Figure 6-33 WAS application installation summary messages (panel 2 of 2)

Figure 6-34 shows the Save confirmation panel, which completes the installation process. Select **Save** to complete the install.

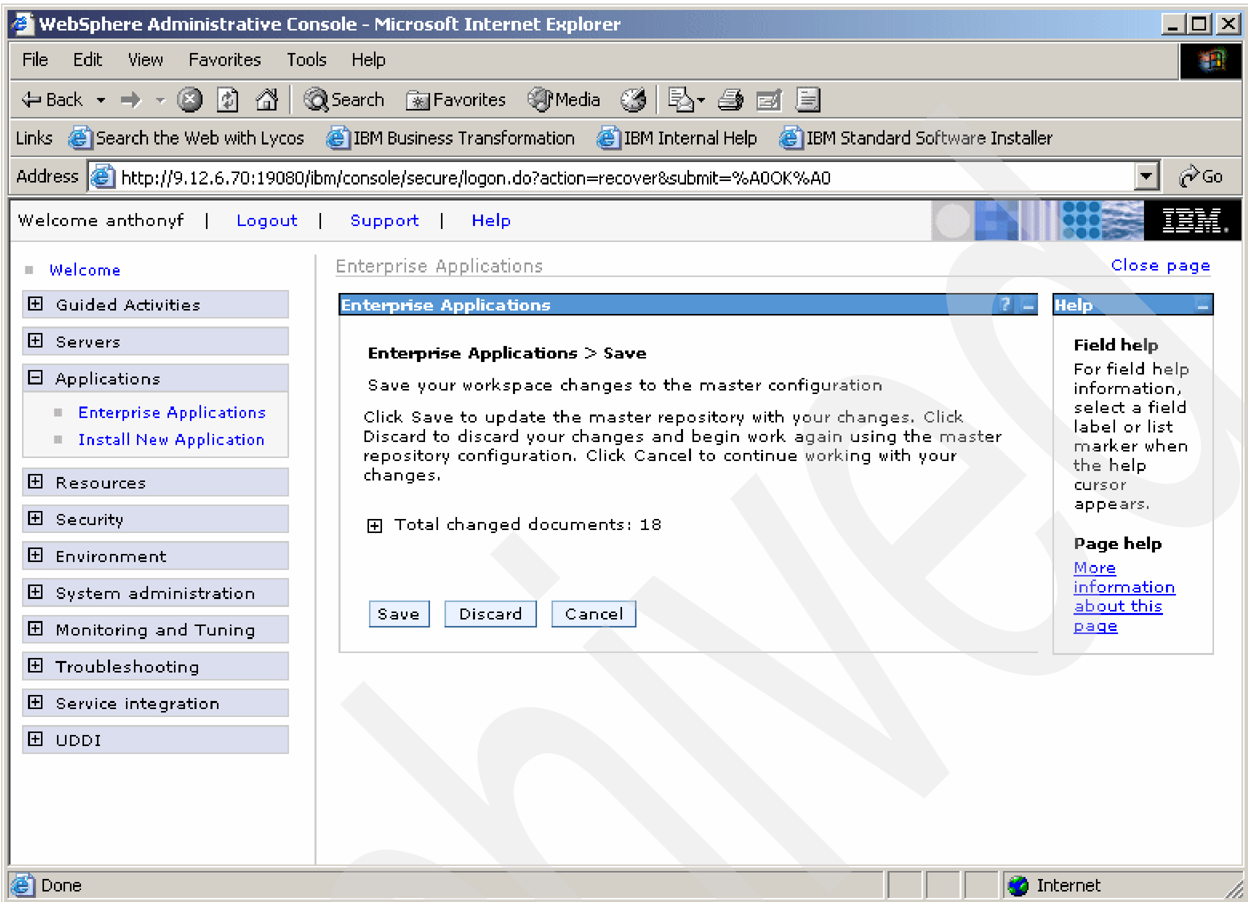


Figure 6-34 WAS application installation save confirmation panel

6.4.4 Implementing the DFSMSrmm Web service DLL

The DFSMSrmm Web service uses a C++ DLL from the MVS link list. The program object is called EDGXHCLL and is distributed in SYS1.SIEALNKE. To make this DLL available for the DFSMSrmm Web service you must install a link in the z/OS WebSphere Application Server's library path. To do this, use the OMVS command and, when initialized, use the CD command to set the current directory to the appropriate library file in the HFS or zFS. Once OMVS is in the correct directory, use the command **`ln -e EDGXHCLL libEDGXHCLL.so`** to establish the link. This establishes an external link to the DLL in LINKLIST.

In Figure 6-35 we show the OMVS screen where the command **cd /WebSpherePB1/V6R0/BS01/AppServer/lib** has been issued, and the command **ln -e EDGXHCLL libEDGXHCLL.so** is about to be issued.

```
IBM
Licensed Material - Property of IBM
5694-A01 (C) Copyright IBM Corp. 1993, 2005
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/usr/lpp/skrb/bin:/usr/lpp/dce/bin:/usr/lpp/ldap/bin:/usr/lpp/ldap/sbin:/usr/lp
p
/db2/db8a/jcc/bin:/usr/lpp/db2/db8a/bin:/usr/lpp/java/J1.4/bin:/bin:.
MHLRES1 @ SC63:/>cd /WebSpherePB1/V6R0/BS01/AppServer/lib
MHLRES1 @ SC63:/SC63/WebSpherePB1/V6R0/BS01/AppServer/lib>



 ===> ln -e EDGXHCLL libEDGXHCLL.so
```

Figure 6-35 OMVS screen showing cd command and ln command

> **Note:** The directory that contains the Web server code is installation dependent and must be ascertained from the WebSphere Application Server installer.

Once you have established the link, it will show up as a syml (symbolic link) entry in the directory. You can most easily use Ishell to verify that the link has been set up.

Figure 6-36 and Figure 6-37 show the specification of the WebSphere lib directory and the listing of the symbolic links. Page down to find libEDGHXCLL.so with the other lib entries.

```
 UNIX System Services ISPF Shell

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                            More:     +
   /WebSpherePB1/V6R0/BS01/AppServer/lib
   _________________________________________________________________
   _________________________________________________________________
   _________________________________________________________________

EUID=0





Command ===> ____________________________________________________________
```

Figure 6-36 Ishell specification of WebSphere lib directory

```
 Directory List

 Select one or more files with / or action codes.  If / is used also select an
 action from the action bar otherwise your default action will be used.  Select
 with S to use your default action.  Cursor select can also be used for quick
 navigation.  See help for details.
 EUID=0   /SC63/WebSpherePB1/V6R0/BS01/AppServer/lib/
   Type  Filename                                              Row 169 of 358
 _ Syml  libbboujsl_g.so
 _ Syml  libbboujuu.so
 _ Syml  libbboujuu_g.so
 _ Syml  libbbouph.so
 _ Syml  libbbouph_g.so
 _ Syml  libEDGXHCLL.so
 _ Syml  libivbtr30i.so
 _ Syml  libivbtr30i_g.so
 _ Syml  libpmiJvmpiProfiler.so
 _ Syml  libpmiJvmpiProfiler_g.so
 _ Syml  libsomorb.so
 _ Syml  libsomorb_g.so
 _ Syml  libsomsh.so
 Command ===> __________________________________________________________________
```

Figure 6-37 Ishell list of symbolic links, showing libEDGXHCLL.so

6.4.5 Using the DFSMSrmm Web service sample client

This section describes how to use the DFSMSrmm Web service. The sample client allows you to issue a DFSMSrmm command to a remote system from your workstation, and store the XML response in a file. You may modify this sample client, or implement your own based on the definitions contained in the enterprise archive file rmmapi.ear, in order to perform the functions that you require.

Prerequisites

Before you can use the DFSMSrmm Web service sample client, you must install the Java Development Kit (JDK™). This can be obtained at the following URL:

http://java.sun.com/j2se/1.5.0/download.jsp

You must also acquire the additional packages listed in Table 6-2, and place the JAR files in a directory that can be easily included in your classpath.

Table 6-2 Libraries required for the DFSMSrmm Web service client

Prerequisite	JAR file	Version used	Download location
Apache SOAP	soap.jar	2.3.1	http://www.apache.org/dyn/closer.cgi/ws/soap/
Apache Xerces XML parser	xercesImpl.jar	2.5.0	http://xml.apache.org/dist/xerces-j/
JavaMail™ API	mail.jar	1.3.3	http://java.sun.com/products/javamail/downloads/
JavaBeans™ Activation Framework	activation.jar	1.0.2	http://java.sun.com/products/javabeans/glasgow/jaf.html

Running the sample client

The following procedure can be used to run the sample client:

1. Download the sample client application source code using FTP from the z/OS hierarchical file system (HFS) directory to your workstation. The source code file is called /usr/lpp/dfsms/rmm/rmmSampleClient.java.
2. Compile the application using the Java compiler, making sure that the JAR files shown in Table 6-2 are included in your classpath. The following command can be used to compile the application if the JAR files are in the same directory as rmmSampleClient.java:

   ```
   javac -classpath .;soap.jar;xercesImpl.jar;mail.jar;activation.jar
   rmmSampleClient.java
   ```

3. Execute the compiled application using the Java Virtual Machine. You must specify the following three parameters on the command line:
   - The IP address and port of the WebSphere Application Server
   - A DFSMSrmm command to execute on the remote system
   - The name of a local file to which the command output should be stored

Example 6-2 shows how the DFSMSrmm Web service sample client could be used to run a LISTCONTROL ALL command on the remote system and store the command output in a file called result.xml.

Example 6-2 Sample Web service client

```
java -classpath .;soap.jar;xercesImpl.jar;mail.jar;activation.jar rmmSampleClient
9.10.111.122.9080 "LISTCONTROL ALL" result.xml
```

6.4.6 Security server preparation

When the request for service is received by the WebSphere Application Server, it acts on behalf of the remote user. The user ID that the server is running under must have access to the DFSMSrmm resources so that it can issue the command and collect the response. You must give WAS access to these resources, and as a consequence you then must define a method of ensuring that such access is validly used, by coding suitable authorization in the interface program. The sample application does not provide this authorization checking.

The resource rules are handled at the server, and any messages issued as a result of denial of service must be analyzed at the server.

If access is denied, a message to that effect is returned to the requestor.

6.4.7 Setting memory limit for returned XML data

DFSMSrmm can contain millions of data sets. This can result in gigabytes of data returned for a search command. If this data is retrieved through the Web service it may have to go through one or more of may environments such as C++, Java, SOAP, z/OS, USS, or Windows®, and some of these environments may not be able to handle this much data at once.

By default, the maximum amount of data returned from the DFSMSrmm Web service is 1 MB. This should be transferable in every standard environment. If the submitted command results in more data returned than 1 MB, you will get 1 MB of data plus an error message indicating that there is more data available:

```
EDG3921I: Insufficient storage for search processing RC = 4, RS = 10
```

To correct this situation, you can either submit a command that returns less data, or adjust the memory limit for the amount of returned data if your environment can handle more than 1 MB.

You can adjust the memory limit via the system property, RMM_XML_MAX_SIZE, that is passed to the Java Virtual Machine (JVM™). If you want your Web service to use this value, you must set it via the administrative console of your application server. In WebSphere Application Server V6.0.2, you find the JVM settings under Servers → Application Servers → your server name (not application name), and under Server Infrastructure select **Java and Process Management →Process Definition → Servant → Java Virtual Machine → Custom Properties**.

If you want, for example, to set the memory limit to 2 MB, you would enter the name RMM_XML_MAX_SIZE and the value 2000000.

Example of the process to change RMM_XML_MAX_SIZE

Figure 6-38 shows the Servers menu item expanded. Select **Application Servers**.

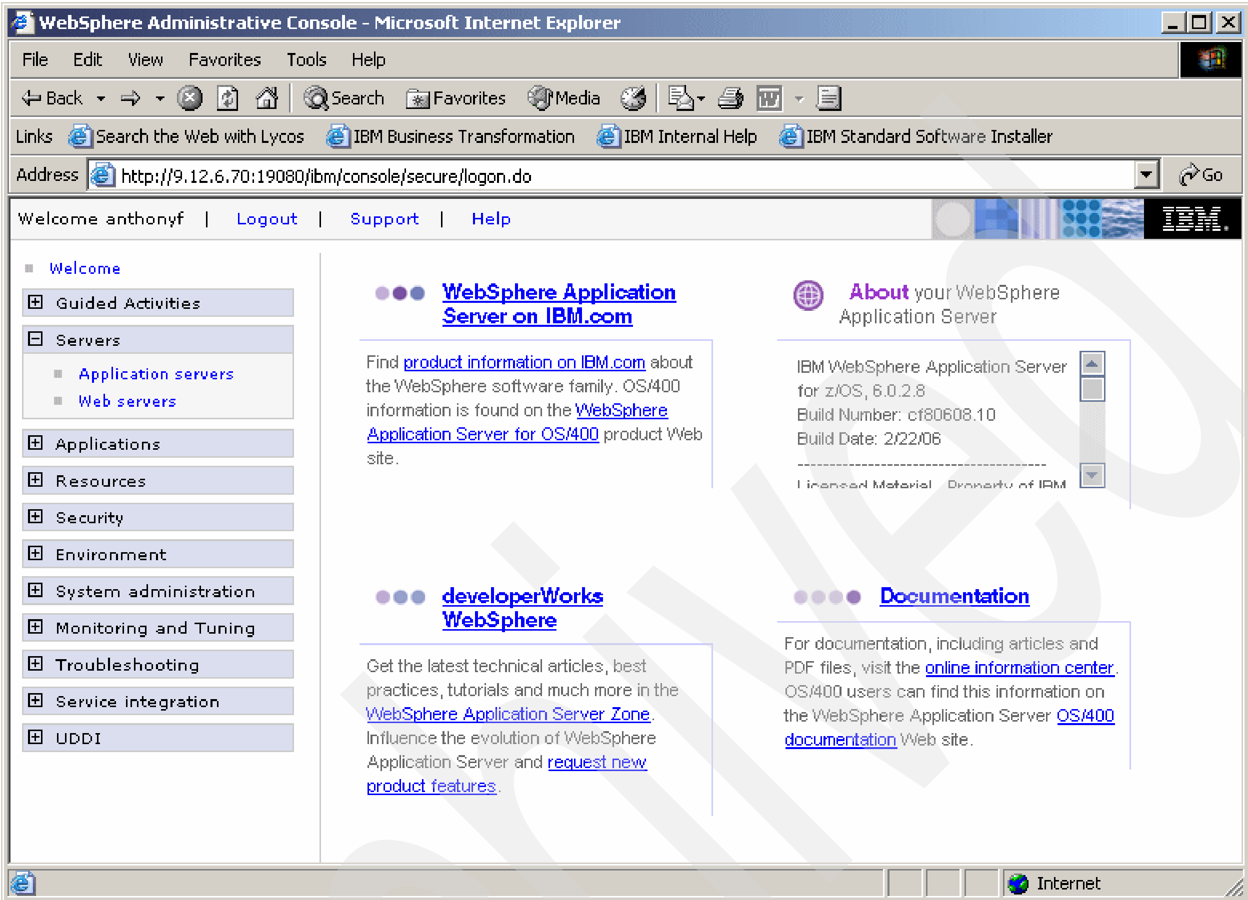


Figure 6-38 WAS showing servers menu expanded

Figure 6-39 shows the selection of server ws6631.

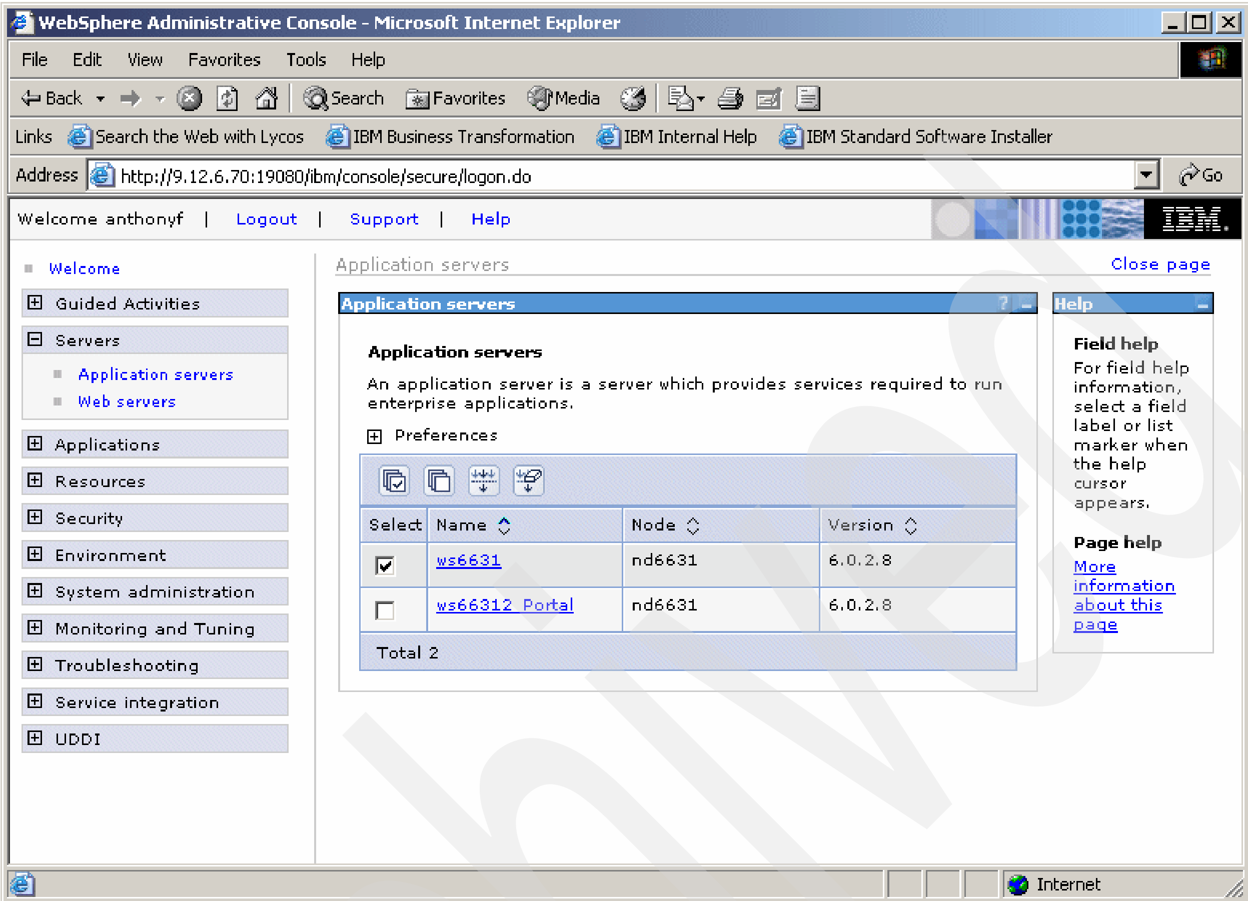


Figure 6-39 WAS application servers selection panel showing ws6631 selected

Figure 6-40 shows the first part of the panel to manage settings. Make sure that the Runtime tab is set. You must locate the JVM settings, so page down until the Server Infrastructure part is located on the right-hand side. Select the radio button for Java and Process Management.

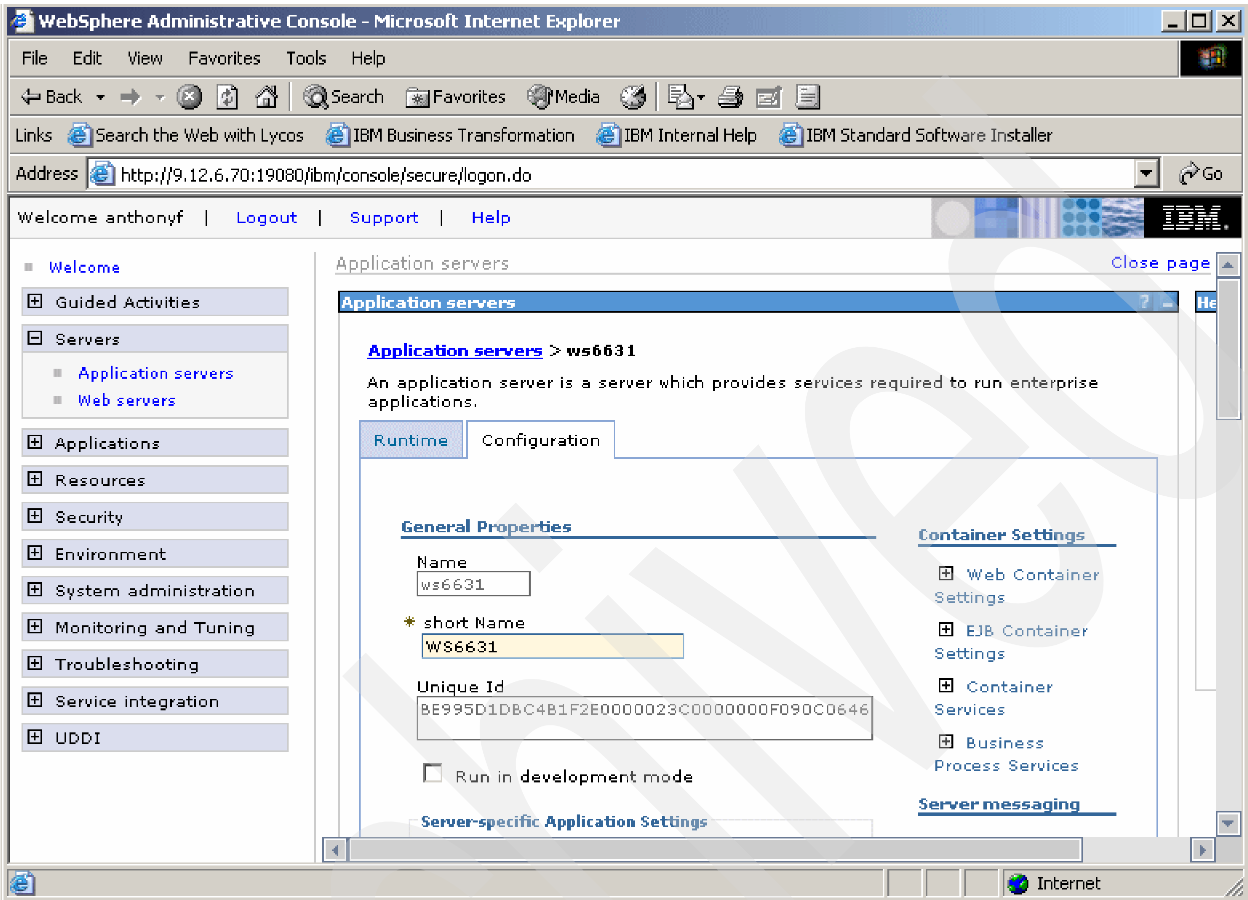


Figure 6-40 WAS showing application server runtime settings

Figure 6-41 shows the Application Server settings panel with Java and Process Management expanded. Select **Process Definition**.

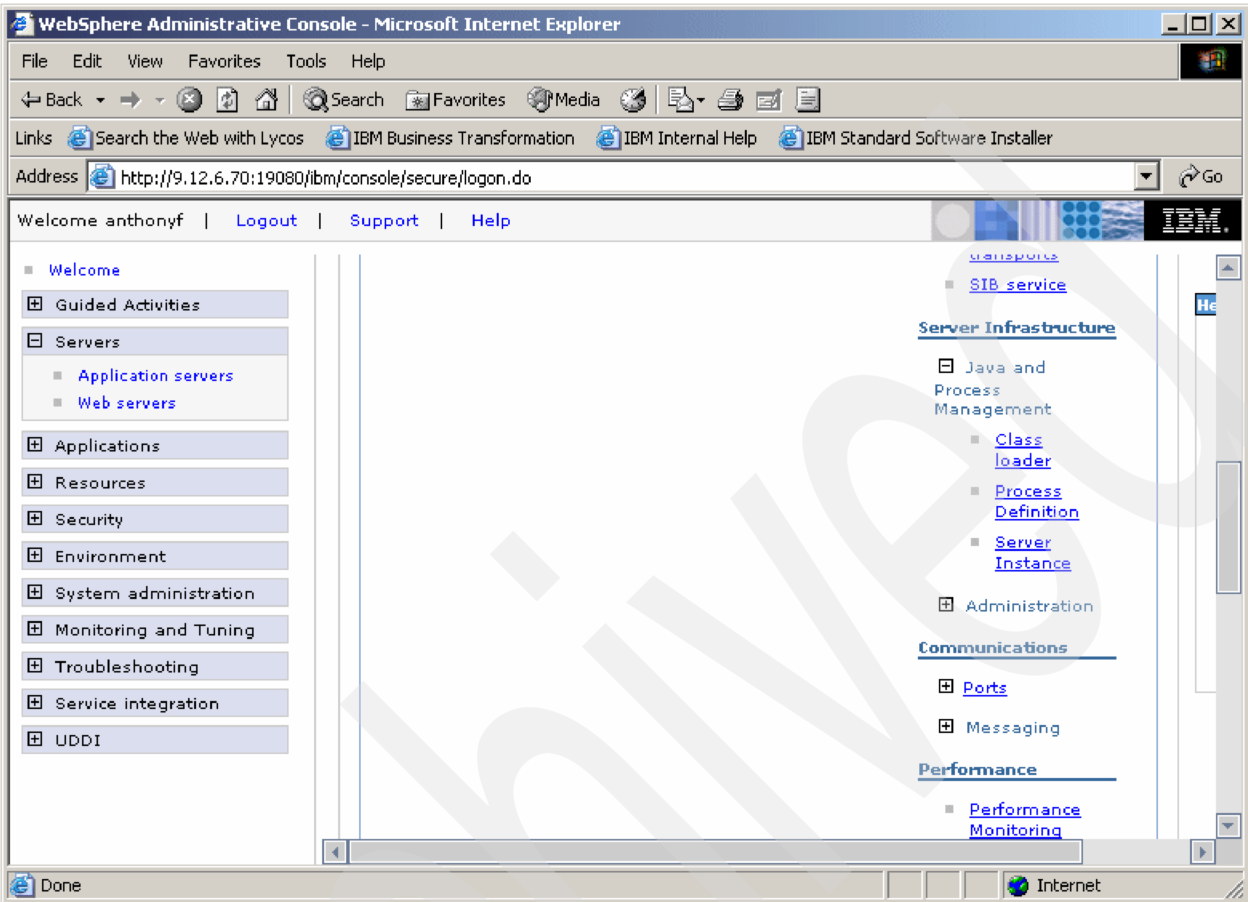


Figure 6-41 WAS applications server settings panel showing expanded Java and Process Management

Figure 6-42 shows the Process Definitions selection panel. Select **Servant**.

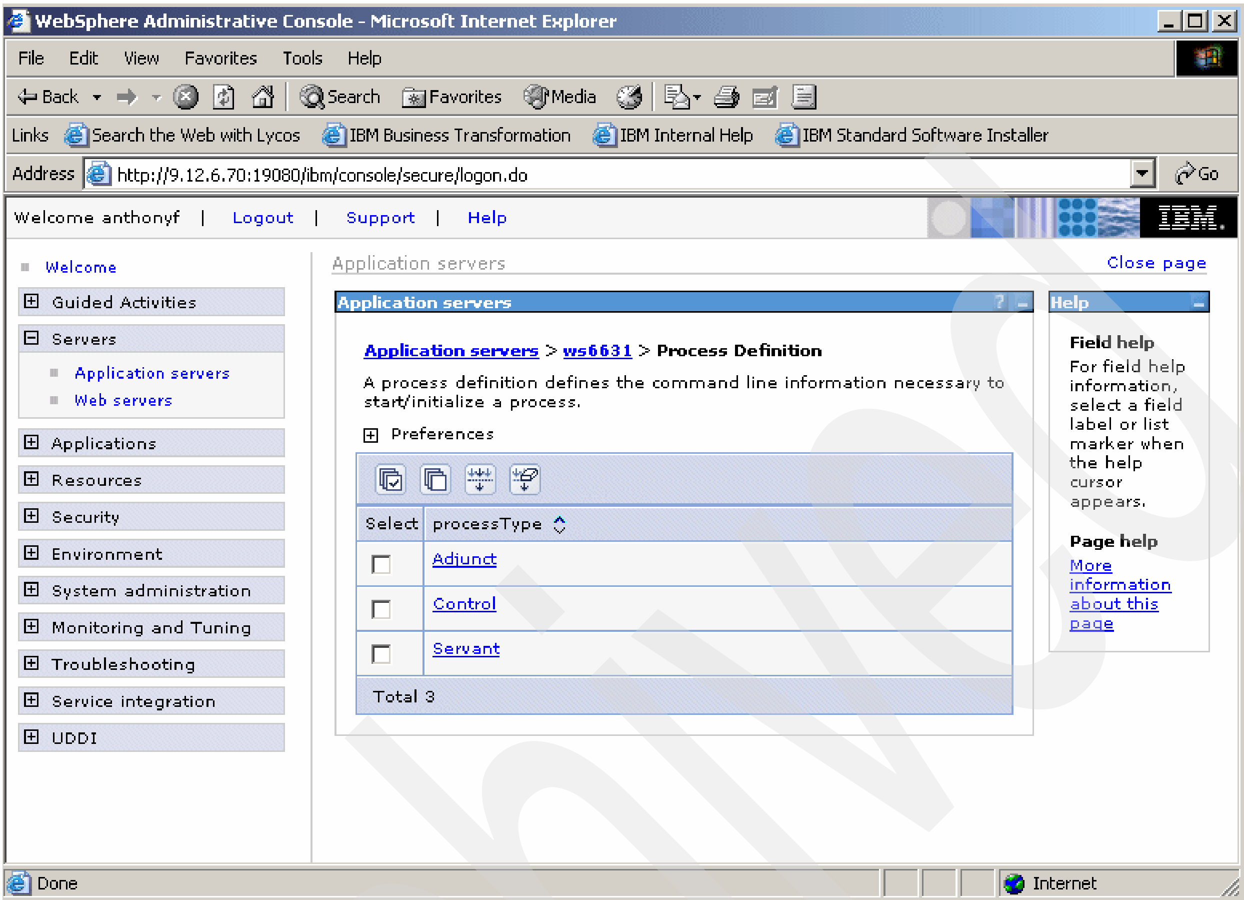


Figure 6-42 WAS application servers settings showing process definition options

Figure 6-43 shows the application server servant attributes specification panel, showing, under Additional Properties, the Java Virtual Machine option. Select **Java Virtual Machine**.

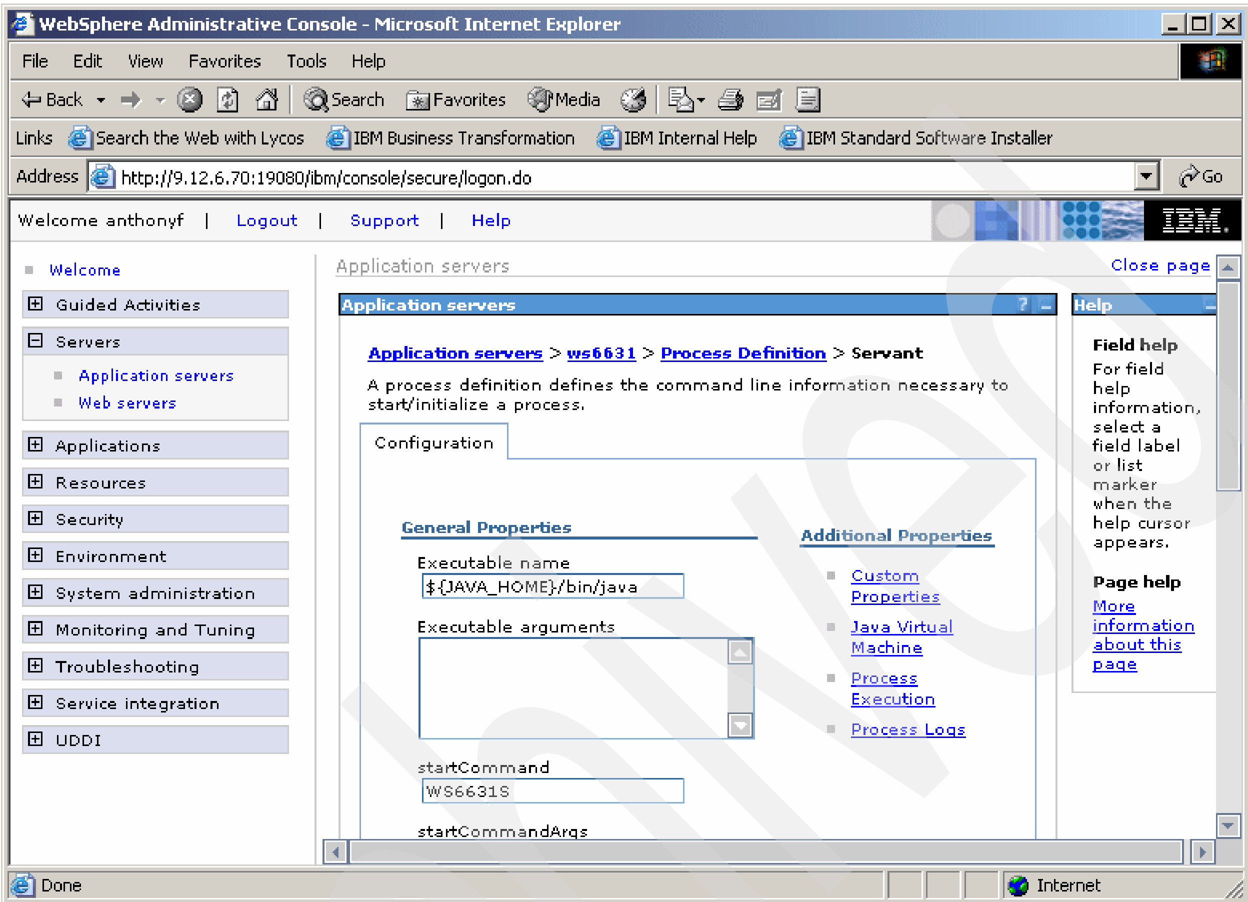


Figure 6-43 WAS application servers showing servant specifications panel

Figure 6-44 shows the panel for setting Java Virtual Machine properties, showing the Custom Properties option. Select **Custom Properties**.

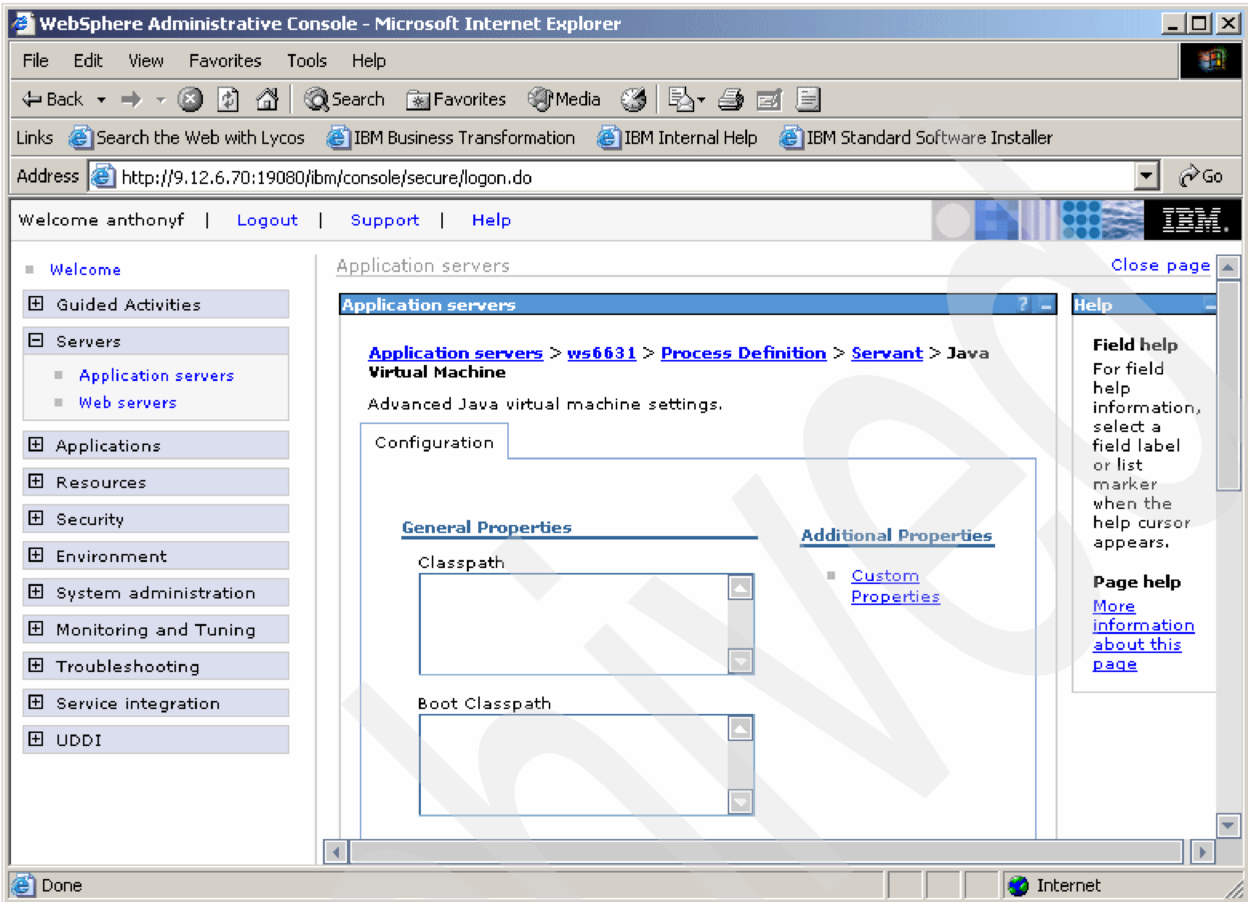


Figure 6-44 WAS showing panel for Java Virtual Machine specifications

Figure 6-45 shows the panel to set custom properties. Select **New** to continue.

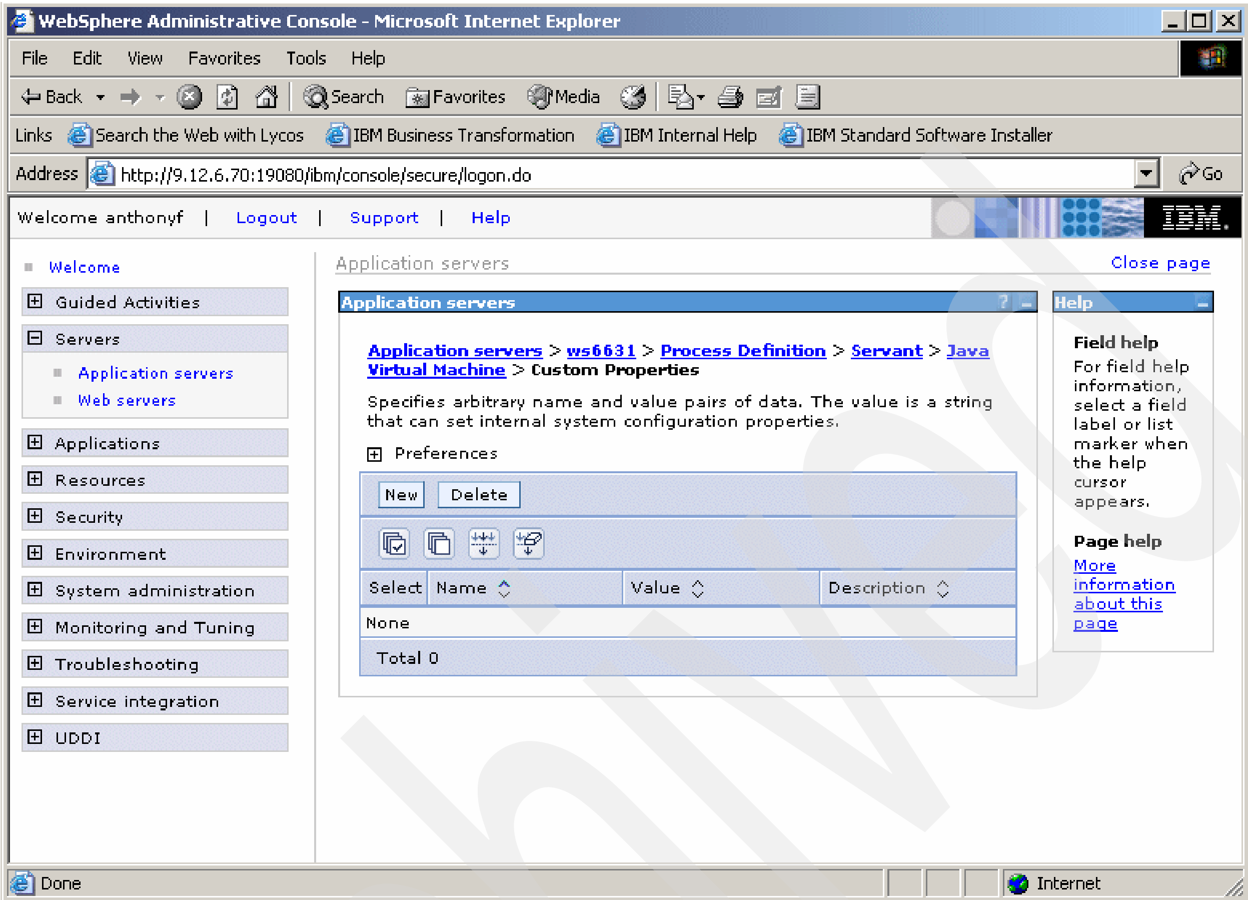


Figure 6-45 WAS panel to set custom properties

Figure 6-46 shows the panel to specify new custom properties. In our scenario we entered RMM_XML_MAX_SIZE for the name and 2000000 for the value. Select **Apply** to continue.

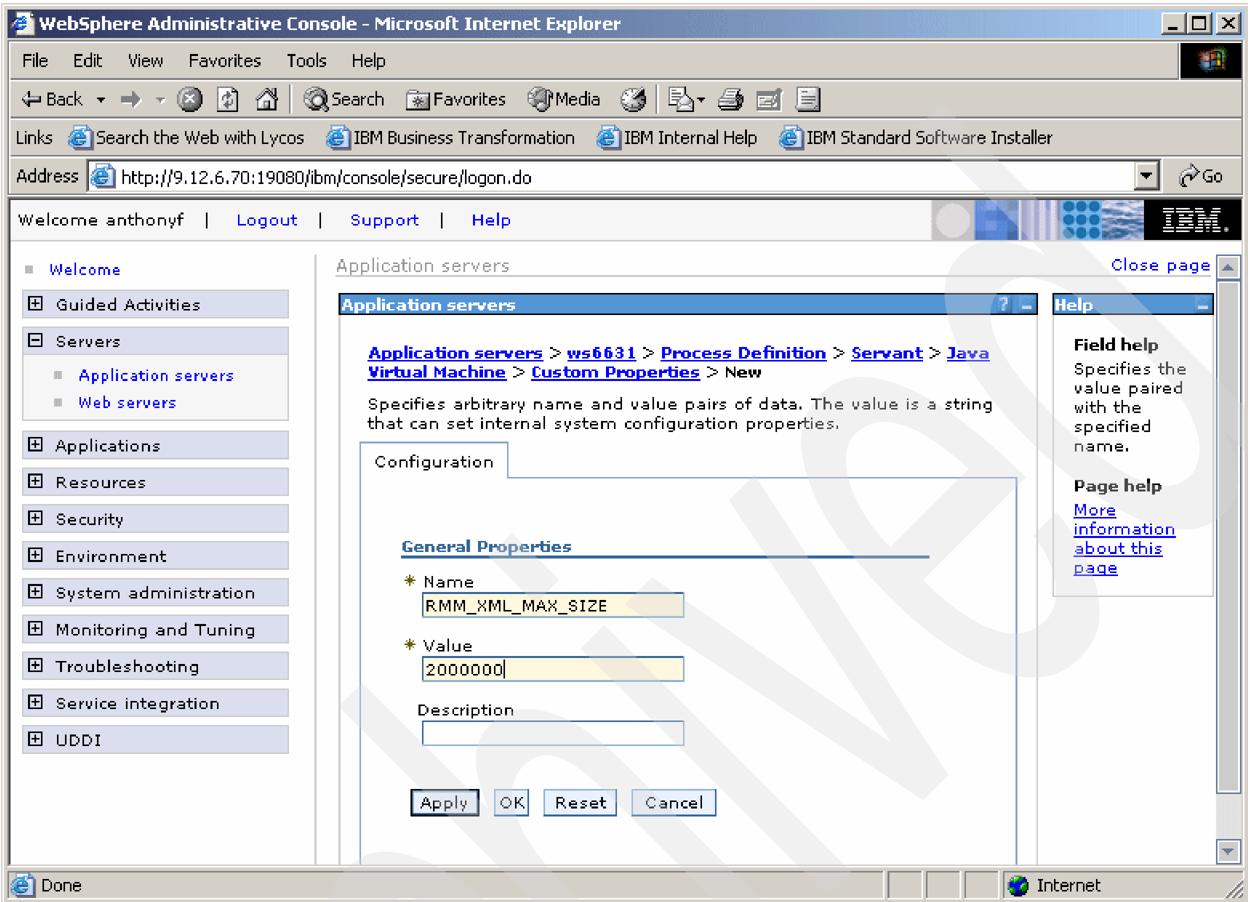


Figure 6-46 WAS panel to set new custom properties for RMM_XML_MAX_SIZE

Figure 6-47 shows the confirmation message issued when a change to the application server properties has been made. Select the highlighted save option.

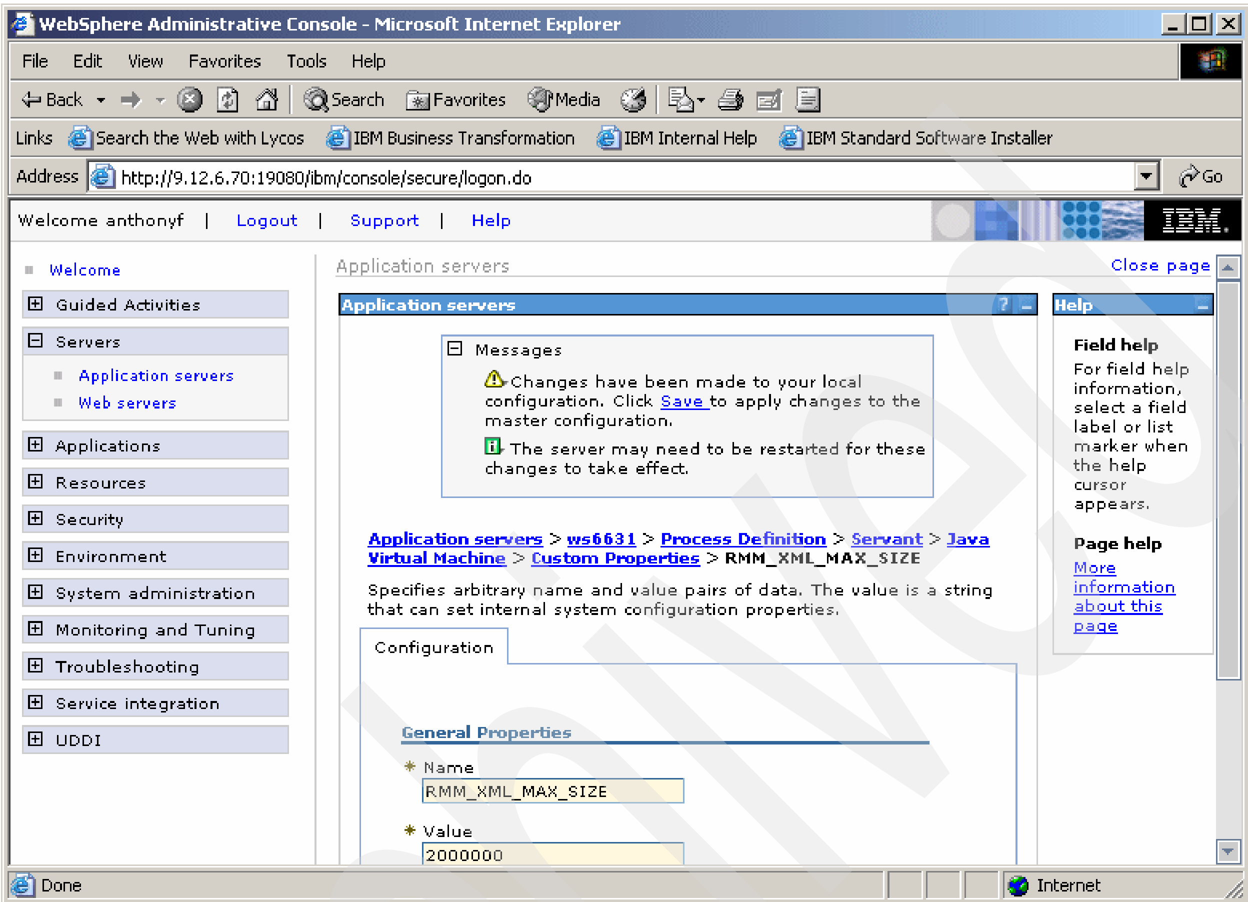


Figure 6-47 WAS application servers messages as a result of applying change

Figure 6-48 shows the save confirmation panel to confirm changes to RMM_XML_MAX_SIZE. Select **Save** to continue.

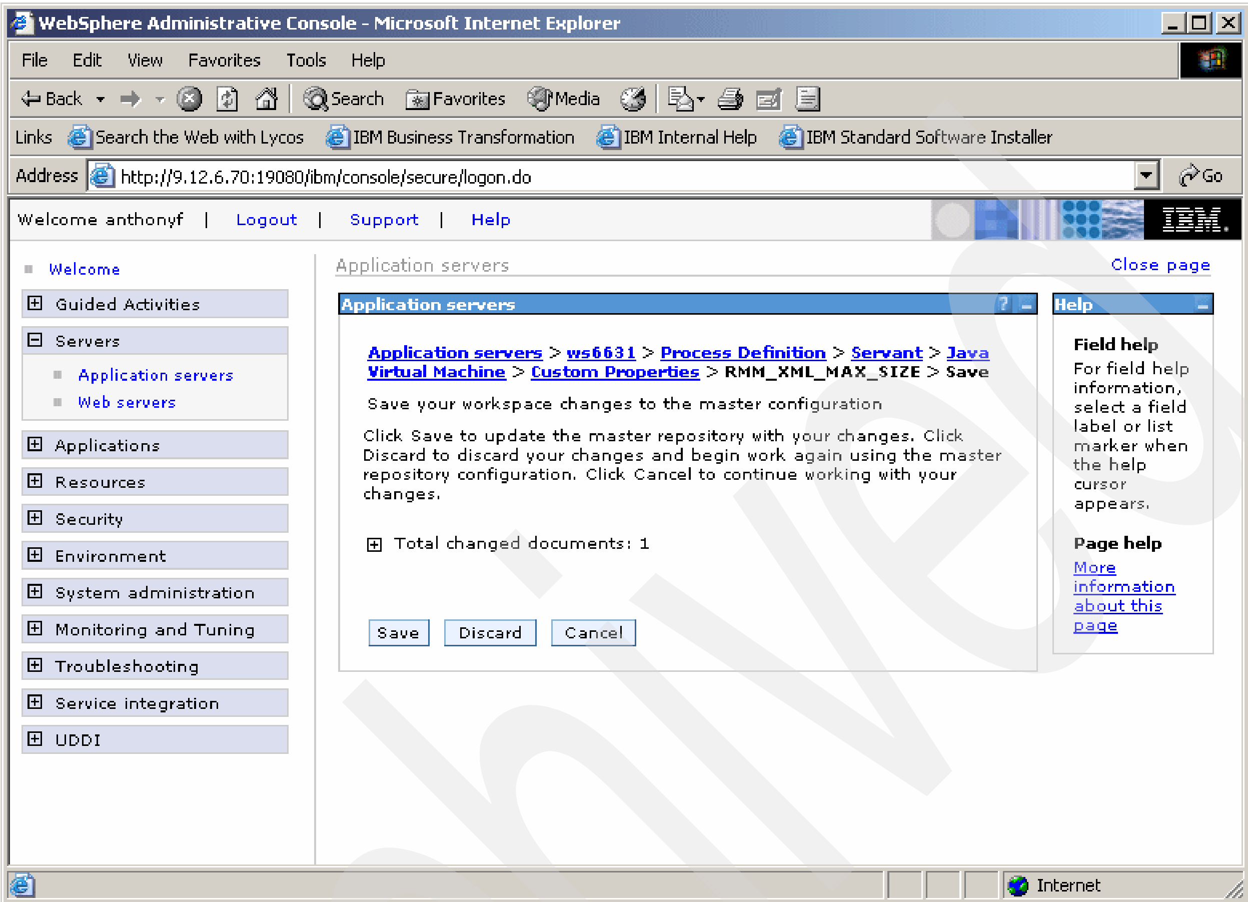


Figure 6-48 WAS application servers custom properties RMM_XML_MAX_SIZE save confirmation panel

Figure 6-49 shows the final confirmation that RMM_XML_MAX_SIZE has been set to 2000000, which is the 2 MB that we specified.

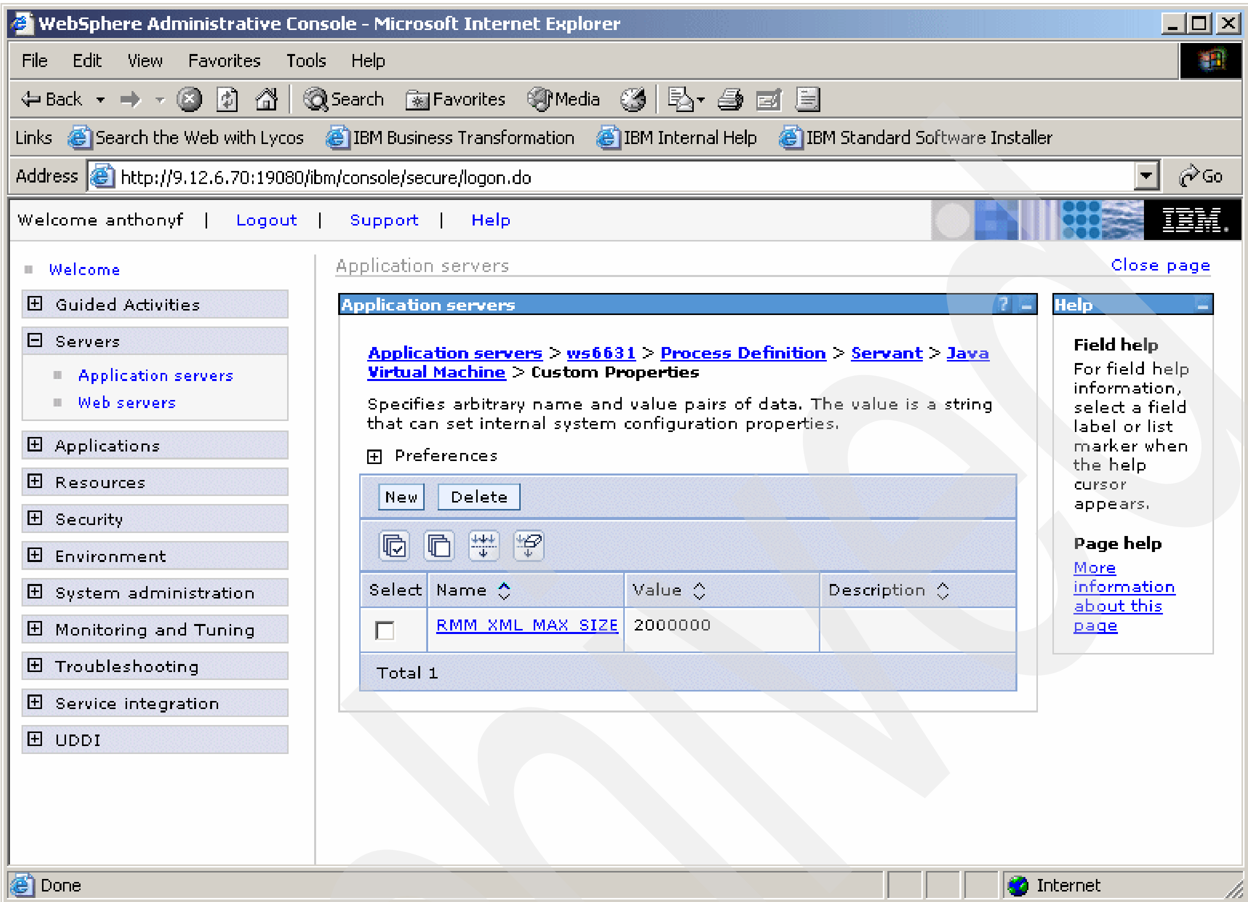


Figure 6-49 WAS Java Virtual Machine custom properties setting confirmation

6.4.8 Starting and stopping DFSMSrmm Web service

After you have successfully installed and saved the DFSMSrmm EAR it is available to be started and stopped as appropriate.

Starting DFSMSrmm Web service

You must log in to the WAS server that you installed DFSMSrmm into. We logged on using the same Web address 9.12.60.70 and port 19080, and to the admin function by specifying the URL as http://9.12.60.70:19080/admin/. As before, the /admin/ part of the address changed to /ibm/console/. The initial screen is the login screen.

Figure 6-50 shows the prompt for the user ID, which is again just for tracking purposes.

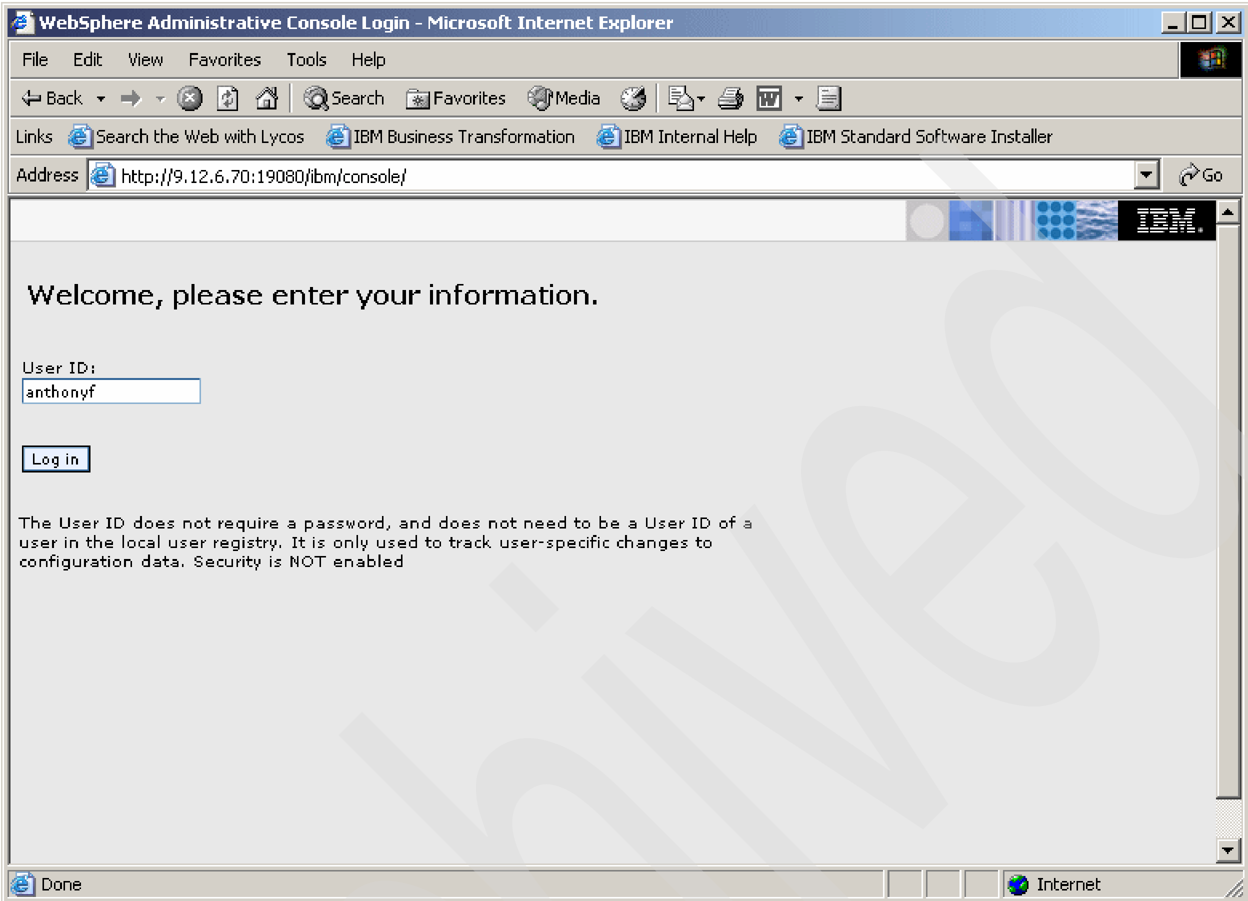


Figure 6-50 WebSphere application initial login screen

Figure 6-51 shows the panel from which activities are selected. The applications section has been expanded. Now that we have an application registered, we will use the enterprise applications path.

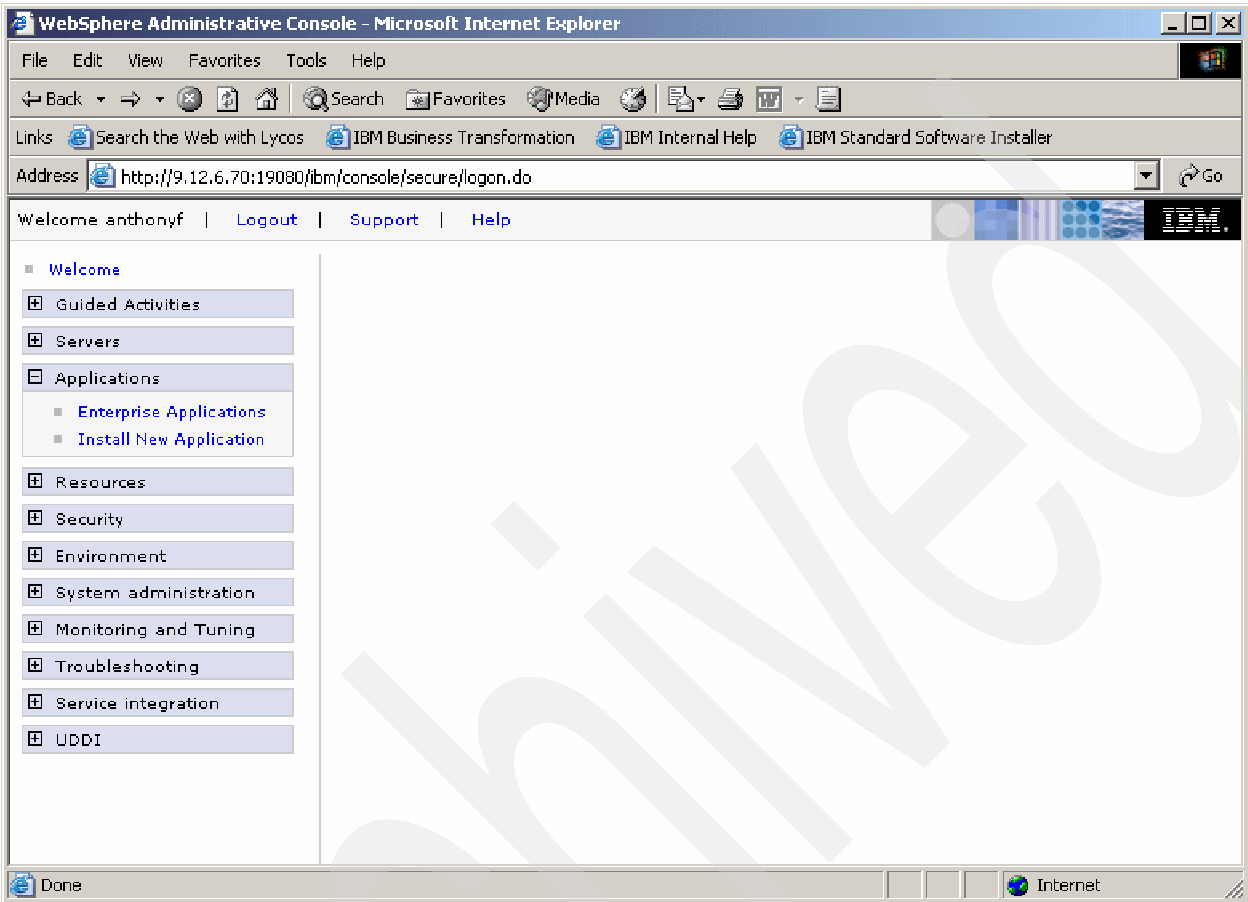


Figure 6-51 WAS activity selection panel with applications expanded

Figure 6-52 shows the first page (of several) listing the available applications. To locate the DFSMSrmm application page down, selecting next page to continue, until you find the one with RmmApiEAR.

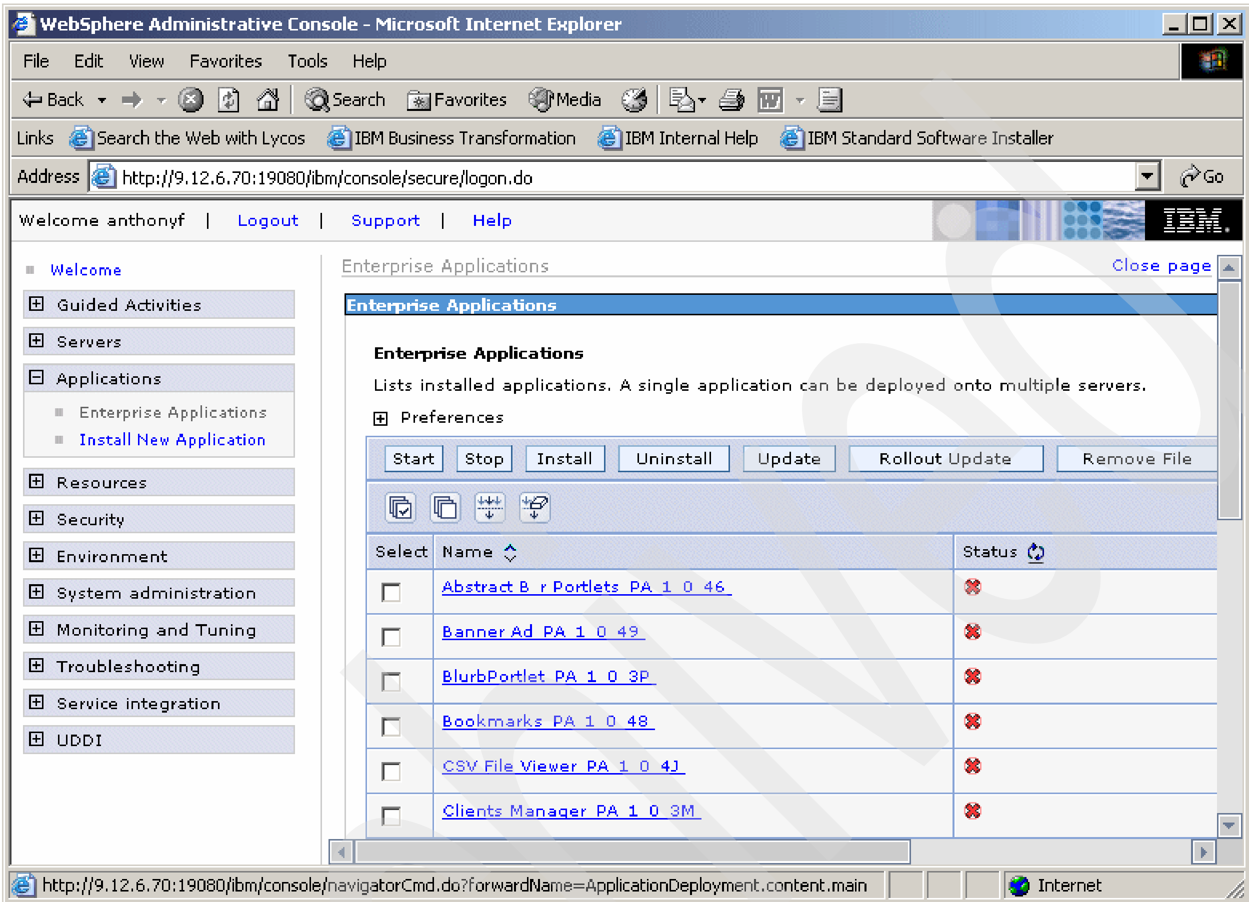


Figure 6-52 WAS Enterprise Applications selection panel page 1

In Figure 6-53 we show the page containing RmmApiEAR with its selection box checked. Complete the process by clicking **Start** on the top bar.

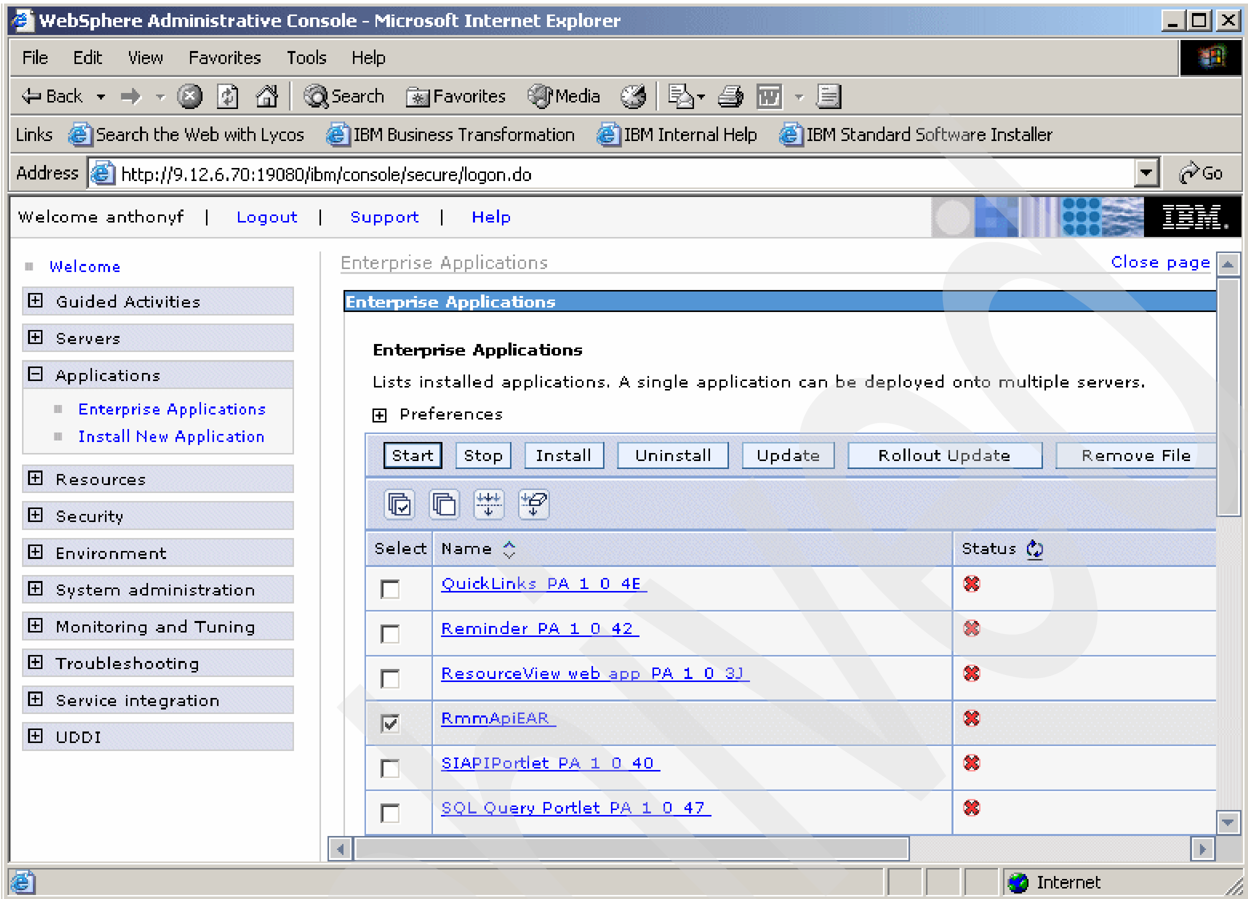


Figure 6-53 WAS Enterprise Applications selection of RmmApiEAR

Figure 6-54 shows the Enterprise Applications selection panel with the message confirming that RmmApiEAR has started.

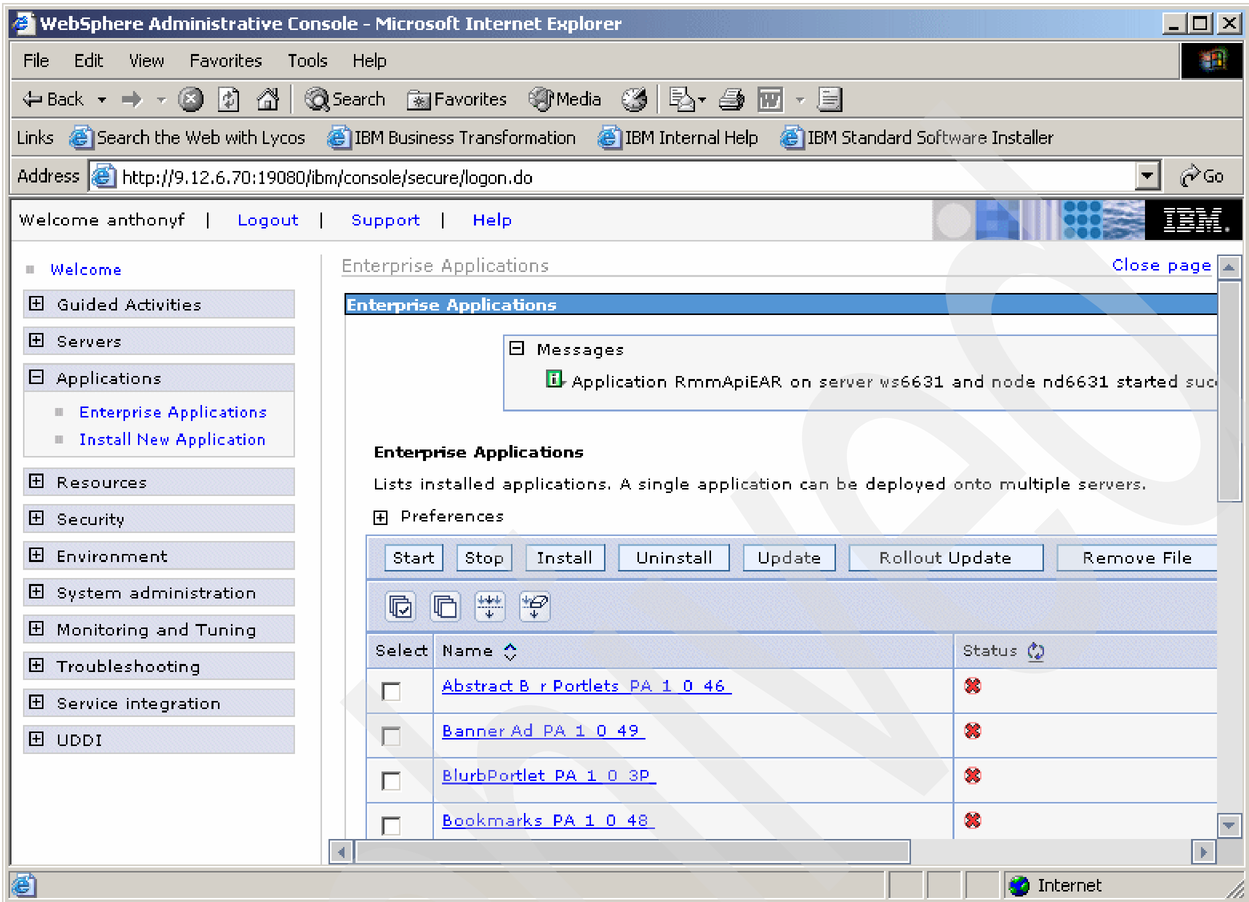


Figure 6-54 WAS application start with confirmation message

To verify that the status of the RmmApiEAR application shows as started, page down, selecting subsequent pages until RmmApiEAR is found.

In Figure 6-55 we show the status of RmmApiEAR with an arrow where the other applications have a cross in their status field.

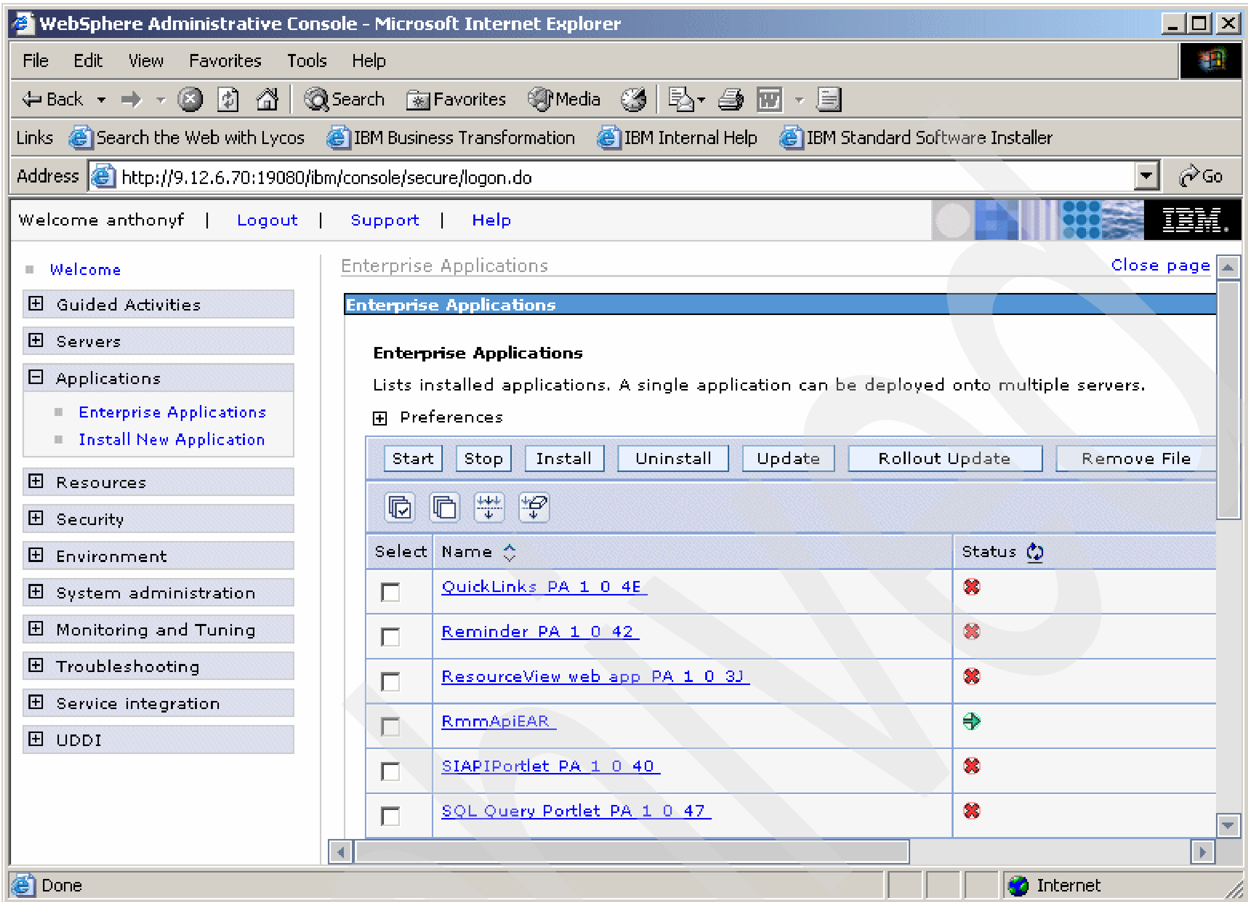


Figure 6-55 WAS Enterprise Applications showing status of RmmApiEAR as started

Figure 6-56 shows the messages that appear on SYSLOG as RmmApiEAR starts.

```
+BBOOO222I: WSVR0200I: Starting application: RmmApiEAR
+BBOOO222I: SRVE0169I: Loading Web Module: RmmJApi.
+BBOOO222I: WSVR0221I: Application started: RmmApiEAR
```

Figure 6-56 WAS SYSLOG messages showing RmmApiEAR starting

Stopping DFSMSrmm Web service

To stop the DFSMSrmm Web service, follow the same process as to start it. Check the selection box for RmmApiEAR, then select the **Stop** option.

In Figure 6-57 we show SYSLOG messages issued as RmmApiEAR is stopped.

```
+BBOOO222I: WSVR0217I: Stopping application: RmmApiEAR
+BBOOO222I: WSVR0220I: Application stopped: RmmApiEAR
```

Figure 6-57 WAS SYSLOG messages showing RmmApiEAR stopping

6.4.9 DFSMSrmm CIM provider

Common Information Model (CIM) is a set of standards that define a conceptual model representing IT resources, as shown in the overview in Figure 6-58. The model presents this in a platform-independent, technology-neutral way. CIM provides the ability to manage a heterogeneous environment from a single management application.

The Distributed Management Task Force (DMTF) is the standards organization that is driving CIM. The mission of the DMTF is to lead the development of management standards for distributed desktop, network, enterprise, and Internet environments. One of the goals of the DMTF is to "Promote interoperability among management solution providers."

For further information about DMTF and CIM see:

http://www.dmtf.org/home

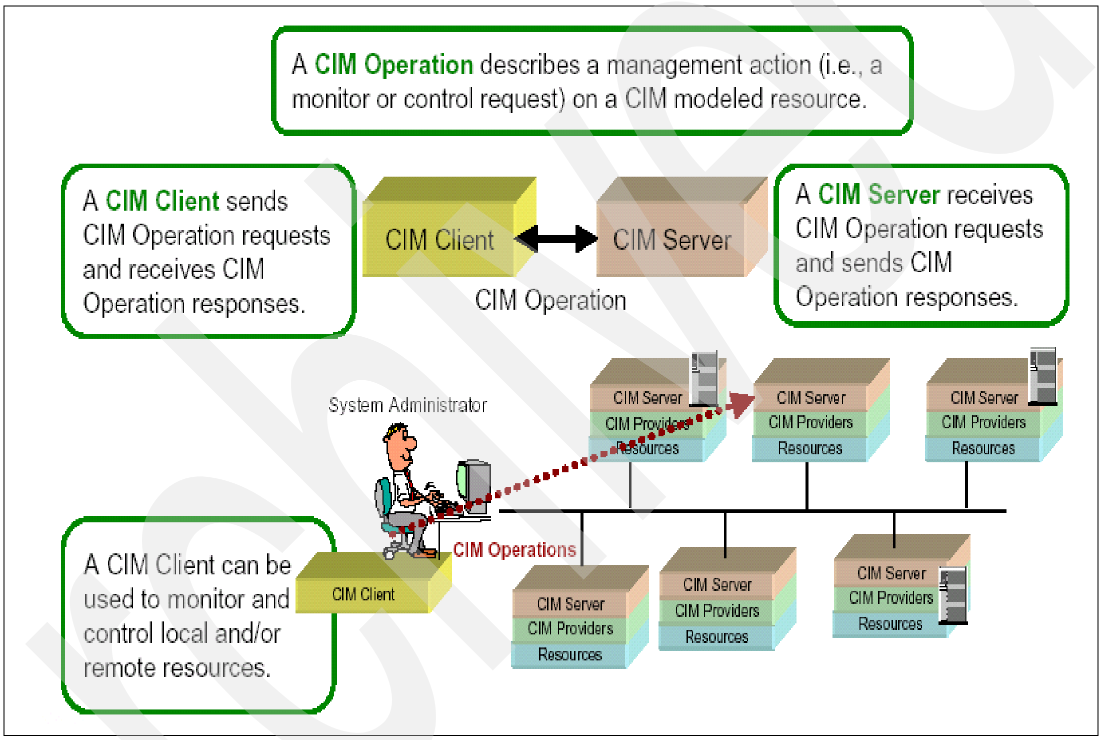


Figure 6-58 Overview of the CIM concept

The z/OS V1R7 DFSMSrmm includes a CIM provider that is a plug-in for the SNIA's Open Source CIM Object Manager (CIMOM). This enables RMM resources to be manageable via CIM.

The DFSMSrmm CIM provider application programming interface is a Java class that implements the CIM-specified methods required of providers. It enables RMM resources to be manageable via CIM.

Figure 6-59 shows a high-level overview of the CIM process.

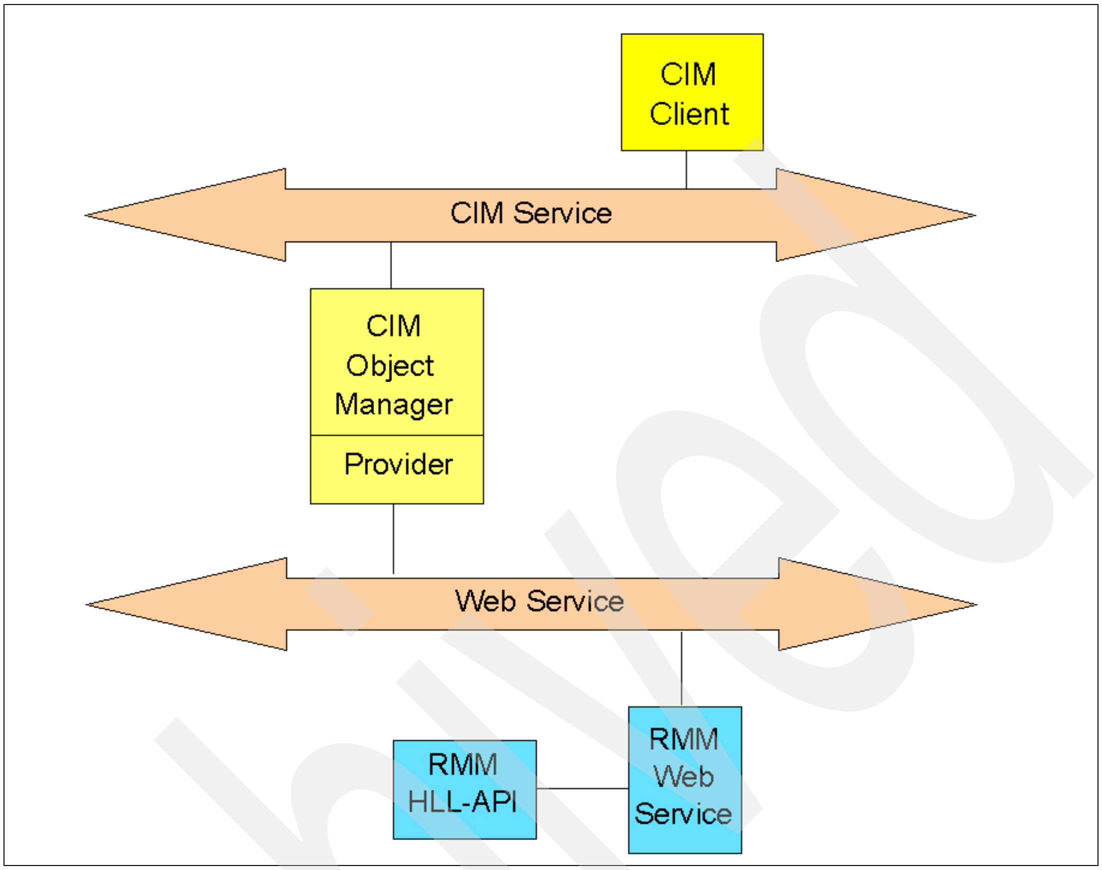


Figure 6-59 Overview of CIM processes

The CIM client (which can be remote) issues general requests to the CIM object manager, via the CIM Service bus.

The CIMOM, if addressed, decides to forward the request to the appropriate CIM provider, which calls the RMM Web service.

The CIM provider consumes the XML data that is returned from RMM, parses it, and returns the requested information back to the CIM client.

Usage

You must register the CIM provider so that the CIMOM knows to go through a provider for RMM resources rather than look in its own static repository, and which provider to use. Registration is done via the managed object format (MOF) file.

The DFSMSrmm CIM provider application programming interface uses the DFSMSrmm Web service via a proxy integrated with the provider. Thus, you must implement the Web service and, optionally, publish it in a local UDDI registry. This enables the provider application programming interface to find the DFSMSrmm system that can return details about volumes, data sets, and their associations, managed by DFSMSrmm.

The manual *DFSMSrmm Implementation and Customization Guide,* SC26-7405, provides detailed information about setting up the CIM provider.

7

NFS enhancements

In this chapter we discuss new functions in NFS 1.5 and NFS 1.7. The following topics are covered:

- NFS overview
- NFS V1R5 line items
  - 64-bit Posix time support
  - CRC32 conversion
  - Multi-volume support
  - Convert Legacy Support to Fastpath
  - NFS Server 878 abend handling improvement
- NFS V1R7 Server
  - Overview
  - Line items
    - Miscellaneous small enhancements
    - Serviceability improvements
    - IPV6 support
    - DHCP (Dynamic IP) support for clients
    - NFS Server locking interoperability
    - NFS Server V4 Protocol enhancements
- NFS V1R7 client
  - Overview
  - Line item
    - RPC to socket conversion
- Migration/coexistence
- NFS V1R8 futures

7.1 Overview

This section provides an overview of Network File System (NFS), and details how data is accessed between the NFS Server and NFS Client.

Network File System provides a flexible option for exchange of data between like and unlike z/OS and non-z/OS systems that support the NFS protocols. The current version of z/OS NFS supports the industry-standard NFS specification for V2, V3, V4, IPv4, and IPv6.

NFS V2 and NFS V3 are described in Request For Comments (RFC) 1094 at the following Web site:

http://www.faqs.org/rfcs/rfc1094.html

NFS V4 is described in RFC 3530 at the following Web site:

http://www.faqs.org/rfcs/rfc3530.hmtl

NFS basically takes the files or data sets on one machine and makes it possible to access them from another machine. It does so transparently so that the user does not know whether the files are on the machine they are working with or on another machine. z/OS NFS also makes it possible to access data on other machines from the z/OS environment through the UNIX System Services (USS).

7.1.1 Client/server relationship

A client is a computer or process that requests services in the network. A server is a computer or process that responds to a request for service from a client. A user accesses a service, which obtains data or other resources.

Figure 7-1 illustrates the z/OS NFS server implementation. z/OS NFS consists of two parts, the NFS client and the NFS server. z/OS data sets and the UNIX System Services (USS) HFS files are accessed from remote clients through the z/OS NFS server. Other systems, shown as terminal screens in the figure, also support NFS as a client or a server. The systems are linked through the Transmission Control Protocol/Internet Protocol (TCP/IP) network used to communicate between the clients and servers. The de-facto standard port to communicate via NFS is 2049.

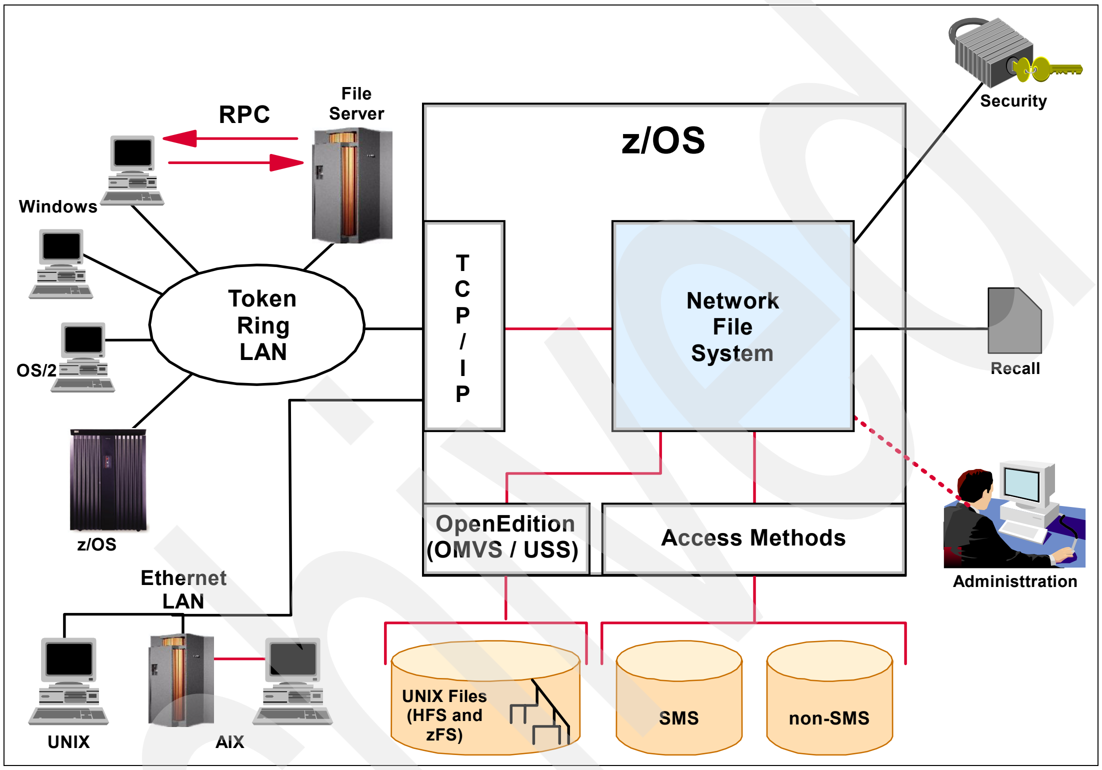


Figure 7-1 Typical NFS implementation

NFS uses the communication services provided by TCP/IP, a suite of protocols that includes the remote procedure call (RPC) and External Data Representation (XDR) protocols. RPC allows a program on one machine to start a procedure on another machine, as though the procedure is local. XDR resolves the differences in data representation of different machines. NFS can be used for file sharing between platforms and file serving (as a data repository). If you use NFS as a file server, the z/OS UNIX file system might be a better choice than using conventional MVS data sets, because of its UNIX-based features.

Note: NFS supports both UNIX files and MVS data sets. When we refer to HFS files, we mean z/OS UNIX files (HFS and zFS files). These are supported by NFS and accessed through UNIX System Services (USS).

7.1.2 NFS version history

The first revision of NFS V2 was published in 1985. It exports basic POSIX 32-bit file systems, but was slow particularly for writing files.

The NFS V3 protocol was published in 1994. It supported 64-bit files and improved write caching. It became the most commonly used protocol for sharing files on *NIX/Linux LANs today.

The available NFS V4 protocol is secure and firewall friendly, and the new operations provide better performance. It is extensible and based on the IETF standard published in 2003.

Table 7-1 is a summary of the z/OS NFS history on the IBM mainframe.

Table 7-1 NFS history on IBM mainframe

Year	NFS release	MVS/OS390/ zOS release	NFS function modification identifiers (Fmids)
1990	R1	MVS/ESA™ 4.01	HDP3FS1
1993	R2	MVS/ESA 4.03	JDZ1170
1995	R3	MVS/ESA 5.02	HDZ11NP/N0/NE
1996	R4	OS390 R1.03	HDZ11SM/HDZ11SE First release having both NFS Server (SM) and NFS Client (SE)
1998	R5	OS390 R2.06	
1999	SPE1 (OW38745)	OS390 R2.08	
2000	SPE2 (OW46949)	z/OS 1.1 (OS/390 2.10)	
2001	SPE3 (OW49104)	z/OS 1.2	
2003	SPE3+ (OW55830)	z/OS 1.3	
2004	V1 R5	z/OS 1.5 -1.6	HDZ11US/HDZ11UC
2005	V1 R7	z/OS 1.7	HDZ11VS/HDZ11VC
2005	SPE (OW11875 / Kerberos Support)	z/OS 1.7	
2006	V1 R8	z/OS 1.8	HDZ118N Both server and client have the same FMID

7.2 NFS V1R5 overview

This section provides information about the NFS V1.5 line items. NFS V1.5 became generally available in September 2004.

In NFS V1.5 (which is part of z/OS 1.5) there are only NFS Server enhancements, no NFS Client enhancements. In NFS V1.5 support is added in USS to exploit the 64-bit time values through NFS.

7.2.1 64-bit Posix time support

This line item adds support for the 64-bit Posix time values, versus just 32-bit Posix time. The effected z/OS NFS server attributes are:

- atime
- ctime
- mtime
- createtime

Purpose

This line item addresses the year 2038 time wrap problem. The seconds/offset will be continuously increased in the integer field. Thus in the year 2038 the 3-bit field will be too small.

Restriction

NFS V2 and NFS V3 protocols only support the 32-bit time values. Only NFS V4 exploits the 64-bit values. With NFS V4 the 64-bit time value will be selected and sent to the NFS server.

Enablement

z/OS UNIX System Services did not activate this function until z/OS V1R6. With the NFS V4 support for the z/OS NFS server and the z/OS NFS client, the 64-bit Posix time support will be fully implemented through NFS.

Implementation

This section provides information about the z/OS NFS server and the z/OS NFS client requests, which implement the 64-bit Posix support.

NFS client

For the NFS client:

- SetAttr accepts new 64-bit time values in Attr Structure.
  - If Value > 2**31-1, Returns EINVAL Error
  - Value > 2**31-1 cannot be sent to Server
- GetAttr, ReadDirPlus sets time values in both 32-bit and 64-bit Time fields in Attr Structure.

NFS Server

No changes.

HFS effects

GetAttr,ReadDirPlus:If Time > 2**31-1.

Returns time value of 0xFFFFFFFF in 32-bit Time field. This means that a local user can set a 64-bit time value for a USS file, but the z/OS NFS Server does not export this 64-bit value to the NFS client without NFS V4 support. Prior to NFS V1.7, the z/OS NFS server gives just the 32-bit value back to the NFS client, which is invalid (0xFFFFFFFF) in this case.

> **Note:** This constraint of only seeing the 32-bit Time value and sending this invalid value back to the NFS client is for NFS V2 and V3 only. With NFS V4 the 64-bit Time value will be selected and sent to the NFS client.

7.2.2 CRC32 conversion

With NFS V1.5 the z/OS NFS server uses the cyclic redundancy check (CRC) algorithm, which is used to create a file ID for MVS data sets.

Overview

A CRC is a type of hash function used to produce a checksum (which is a small, fixed number of bits) against a block of data, such as a packet of network traffic or a block of a computer file. The checksum is used to detect errors after transmission or storage. A CRC is computed and appended before transmission or storage, and verified afterwards by recipient to confirm that no changes occurred on transit. CRCs are popular because they are simple to implement in binary hardware, are easy to analyze mathematically, and are particularly good ar detecting common errors caused by noise in transmission channels.

z/OS NFS server function

This line item converts the MVS data set file ID with the CRC32 instead of the CRC16 algorithm.

Purpose

The purpose is to reduce the potential problem for multiple MVS data sets accessed by NFS to have the same file ID.

The NFS protocol requires that every file has a persistent and unique file ID.

The UNIX files have a unique file ID assigned by the physical file system (PFS) when the file is created. MVS data sets do not have a unique fie ID assigned by MVS. Therefore, the z/OS NFS Server uses the CRC16 algorithm to generate a file ID from the name.

Solution

The z/OS NFS server changed its algorithm to CRC32 Algorithm, which greatly reduces the potential for having a duplicate file ID (1 in 4 Gb verses 1 in 64 K).

7.2.3 Multi-volume support

This section contains information about the new support for multi-volume MVS data sets and its implementation. The purpose is to add the support for multi-volume MVS data sets.

Purpose

To enhance the z/OS NFS MVS data set support to include multi-volume data sets. The supported data set types and restrictions are the same as supported by DFSMS, as shown in Table 7-2.

- Physical sequential (PS)
- VSAM with the following restrictions:
  - Multi-volume striped data sets are not supported.
  - Maximum 59 volumes, 123 extents per volume, 255 extents total.

Table 7-2 Capabilities

PS and VSAM data sets	Non-SMS managed	SMS managed
Create	No	Yes
Read/write	Yes	Yes

Buffer stealing

The internal buffer stealing is disabled for multi-volume data sets. This means that:

- Within a volume, buffers can be reused within a data set.
- Idle multi-volume data sets will be closed to free resources when the buffer size exceeds the value defined for the bufhigh z/OS NFS server attribute.

7.2.4 Convert legacy support to Fastpath

This section provides information about the Fastpath used for MVS data sets.

Purpose

The handling of MVS data sets is redesigned to improve performance. The legacy Fastpath processing is modeled after HFS Fastpath processing and a new historical worker task load module (GFSALEGT) is created.

Legacy Fastpath - request processing assignments

The z/OS NFS server main task will continue to handle the following non-NFS protocol requests:

- Mount
- ShowAttr
- PCNFSD
- HFS and Legacy GetCred
- The z/OS NFS server internal requests
  - Shutdown
  - Operator commands
  - Timer control

7.2.5 NFS Server 878 abend handling improvement

The purpose of this line item is to improve the handling of 878 and 80A Abend failures and avoid z/OS NFS server shutdown for 878-10 or 80A-10 Abend. This is implemented in the NFS Server only.

NFS obtains storage from the system into the NFS storage pool. When MAIN or transport tasks receive the 878 or 80A abend the z/OS NFS Server tries to recover to keep the task. It may be that the task is terminated or the task could suffer a second 878/80A Abend immediately. With the second abend the task is terminated automatically to get rid of the failure. Thus, the total number of tasks is reduced and may fall below the MINTASKS specified by the system programmer for the NFS Server, causing the NFS Server to shut down.

New site attributes

There are new site attributes related to the 878 Abend handling.

MINTASKS(n,m,o)

The variables are:

n	Minimum number of async I/O or short blocking tasks
m	Minimum number of HFS tasks
o	Minimum number of long blocking tasks

The default is half of the corresponding NFSTASKS values (NFSTASKS/2). It can be set to a different value by MINTASKS in the site attributes file (for example, when NFSTASKS(m) = 20, the MINTASKS(n) = 10)).

> **Note:** NFSTASKS is another site attribute of the NFS Server. Refer to Chapter 9, "Initialization attributes for the z/OS NFS server," in the manual *Network File System Guide and Reference*, SG26-7417.

NOREC878/REC878

This attribute turns the 878 Abend handling function on and off.

Default:REC878, meaning that 878 Abend handling is active.

New error messages

There are new error messages related to the 878 Abend handling process:

- 878 Abend Task Terminated

  ```
  GFSA453E (MVSNFS  ) NFS SERVER LOST TASK <tasktype>.
  NOW AVAILABLE: nn1 LEGACY TASKS, mm1 HFS TASKS, oo1 LONG    SERVICE TASKS.
  LEGACY TASKS SHUTDOWN LIMIT IS < nn2. HFS TASKS SHUTDOWN LIMIT IS < mm2.
  LONG SERVICE TASKS SHUTDOWN LIMIT IS < oo2.
  ```

  The tasktype field can be *legacy*, *HFS*, or *Long service*.

  nn1, mm1, oo1, nn2, mm2, oo2 counts of tasks.

  Description: This message means that one task of tasktype was detached due to 878 abend. If available tasks of that type are less than its shutdown limit, the NFS server will start the shutdown procedure.

- 878 Abend Task Recovered

  ```
  GFSA454I (MVSNFS  ) <taskname> TASK HAS BEEN RECOVERED AFTER 878 ABEND.   ALL
  TASKS WILL BE STARTED NOW
  ```

  <taskname> is the name of worker task that had the 878 (80A) abend.

  Description: This message means that one task of task name had a 878 (80A) abend and was recovered after it. The NFS server is now ready to work.

- 878 Abend Handling in Process

  ```
  GFSA472I (MVSNFS  ) 878 ABEND HANDLING FOR TASK <taskname> IN PROCESS. ALL
  TASKS STOPPED. PLEASE WAIT FOR RECOVERY.
  ```

  <taskname> is the name of worker task that had the 878 (80A) abend.

  Description: This message means that worker task had an 878 (80A) abend and a recovery procedure is in progress. The NFS server is not ready to work.

878 Abend handling algorithm

If 878 or 80A abend occurs, the failing worker task will:

- Stop all NFS tasks, except the NFS main task and failing task.
- If the failure is not within five minutes of previous 878/80A for this task the failing worker task will be resumed and be able to process the appropriate requests:
  - If the request was being processed at the time of failure, the fail request is NFSER$R_JUKEBOX.
  - Release the free memory in the sub-pools.

- Restart all NFS tasks.
- Resume request processing with the next request.

- If failure is within 5 minutes of the previous 878/80A abend for this task the worker task will be deleted.
  - If the request was being processed at the time of failure the fail request is NFSER$R_JUKEBOX.
  - For all other requests on the failed task's queues:
    - If physical I/O is in process, fail the request.
    - Otherwise, assign the request to another task.
  - Release the free memory in the sub-pools.
  - Delete the failing task.
  - If the number of tasks remaining is less than the specified minimum number of tasks, shut down the NFS Server.
  - If the number of tasks remaining is greater than the specified minimum number of tasks:
    - Restart all NFS tasks.
    - Resume NFS processing.

7.2.6 Miscellaneous modified NFS function

The restriction of APAR OW55365 that open of unopened data set must be done by same worker task as the previous Open and Close has been removed. Open and Close of the data set can be done by a different existing task.

If the worker task is terminated by 878/80A abend, the data sets opened by that task will be closed. A subsequent read/write by another worker task will reopen the data set.

> **Note:** New attributes MINTASKS(n,m,o), NOREC878, and REC878 are displayed in the showattr output.

7.3 z/OS NFS V1R7

This section provides an overview of the z/OS NFS Server and information about the NFS V1.7 line items for both the z/OS NFS Server and the z/OS NFS client. NFS V1.7 became generally available in September 2005.

7.3.1 Overview of the z/OS NFS Server (NFSS)

With the z/OS NFS Server, you can remotely access z/OS MVS conventional data sets or z/OS UNIX files from workstations, personal computers, and other systems that run client software for the industry standard NFS V2, V3, and V4 protocols, and the WebNFS™ protocol over TCP/IP network. The z/OS NFS server acts as an intermediary to read, write, create, or delete z/OS UNIX files and multiple virtual storage (MVS) data sets that are maintained on a z/OS host system. The remote MVS data sets or z/OS UNIX files must be mounted.

7.3.2 Server control files

These special files are used by the z/OS system administrator to control the z/OS NFS server.

Attributes data set

The attributes data set contains the settings for the z/OS NFS server. There are three types of attributes stored in this data set:

- Data set creation attributes are used to define the structure of MVS data sets when creating a file (for conventional MVS data sets only).
- File processing attributes are used to control how files are accessed by the client.
- Site attributes are used to control z/OS NFS server resources.

The system administrator changes the default settings by editing the attributes data set and restarting the server. Client users can override the data set creation and file processing attributes at the command line. For conventional MVS data sets, the client user can specify the data set creation attributes when mounting, creating, or accessing files. The client user can override the file processing attributes when mounting, creating, or accessing files. However, some file processing attributes can only be overridden on a mount point basis.

The attribute list is in the GFSAPATT member of the SYS1.NFSSAMP data set. It contains the actual attributes, their usage, and defaults.

> **Note:** “Attributes data set” on page 210 provides information about the new and changed NFS server attributes, their values, and the defaults.

Exports data set

The exports data set can control which client users can mount which MVS data sets and z/OS UNIX file systems. The entries in the exports data set specify which MVS high-level qualifiers or HFS directories can be mounted. The system administrator can use this data set to limit mounts to accredited clients only. It also controls which client users can mount all or part of the z/OS UNIX file system, based on the client machine’s specified Internet Protocol (IP) address. To use the exports data set, the security site attribute must be set to either safexp or exports by the MVS system administrator.

Mount handle data sets

The z/OS NFS server maintains a list of the active mount points in a pair of files called the mount handle data sets on MVS. The two data sets are used alternately to automatically reestablish the client mount points when the server is started. If the file system is not available, the mount point is not reestablished and the mount failure is recorded in the log data set. The z/OS NFS server does the cleanup activity during z/OS NFS server shutdown and daily at the cleanup time specified by the *restimeout* site attribute. During cleanup time, the z/OS NFS server reads the list and checks all mount points against the retention period specified in the *restimeout* site attribute. If your mount points are idle longer than the retention period specified in the restimeout site attribute, they are removed. Only the active mount points are reconnected. If a mount handle is removed by the cleanup activity, the client user might receive the `Stale NFS File Handle` message or some other appropriate message. If so, all the client user needs to do is unmount the stale mount point and mount it again.

> **Attention:** Consider using no cleanup of the mountpoints for an SAP® HA solution system by specifying *restimeout(0,0)* and using the attribute *remount* to avoid *NFS stale* problems during reconnect after an IPL or failover of the NFS Server.

Log data sets

The log data sets store the messages for the z/OS NFS start-up procedures. This log can be used to identify the user's correctable errors or the user's problem errors. There are two logs that this information is stored in, the primary log and the secondary log. The primary log is used at start-up until it is filled and then overflows into the secondary log. When the secondary log is full, the primary log will then be overwritten with new error messages.

The z/OS NFS server also records messages and diagnostic information in a z/OS component trace buffer, if one is specified. Component trace buffers can be used in addition to, or instead of, the log data sets. Using a component trace buffer can provide performance improvements over the log data sets. The messages will always be written to the LOG data sets, not just to the Ctrace.

Checklist data set

The checklist data set contains entries for files or directories that are to be exempt from System Authorization Facility (SAF) checking even though saf or safexp are specified as the security options. This file is only used when saf or safexp are specified for the particular data type and the checklist option is specified as a site attribute. The entries specified here must match a subsequent mount point, or be the parent of a subsequent mount point, to allow SAF checking to be bypassed for everything underneath that mount point. If the entry does not match a subsequent mount point and it is not a parent of any subsequent mount point then it has no effect.

Kerberos configuration file (read by the server)

A copy of the sample Kerberos configuration file will be found in z/OS UNIX /usr/lpp/skrb/examples/krb5.conf. This file must be copied to /etc/skrb/krb5.conf. The permission bits of this file should allow only the administrator to modify it but everyone else to be able to read.

7.3.3 Supported clients for the z/OS NFS server

The tested and supported clients for the z/OS NFS server are:

- IBM RS/6000® AIX V5.3
- Sun™ Solaris™ Version 10 v Enterprise Linux 4
- Windows 2000/XP with Hummingbird Maestro™ 9 and Maestro 10

Other client platforms should work as well since NFS V4 is an industry standard protocol, but have not been tested by IBM. Older versions of the above clients are still supported under the NFS V2 and V3 protocols, but all have not been tested by IBM.

7.3.4 Startup of the z/OS NFS Server

This section gives a brief overview of the startup of the z/OS NFS Server.

Figure 7-2 shows an example of the NFS V1.7 server started task job control language (JCL).

```
//NFSMVS   PROC MODULE=GFSAMAIN,
//           SYSNFS=SYS1,
//           NFSPRFX=OS390NFS,
//           TCPIP=TCPIPMVS,
//           TCPDATA=TCPDATA
//GFSAMAIN EXEC PGM=&MODULE,
//             REGION=0M,
//             TIME=1440,
//          PARM=(,
//          'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP")/')
//SYSTCPD  DD   DISP=SHR,DSN=&TCPIP..&SYSNAME..TCPPARMS(&TCPDATA.)
//STEPLIB  DD   DISP=SHR,DSN=&SYSNFS..NFSLIBE
//SYSPRINT DD   SYSOUT=*
//OUTPUT   DD   SYSOUT=*
//SYSERR   DD   SYSOUT=*
//NFSATTR  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.PARMS(ATTRIB)
//NFSLOG1  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.SERVER.LOG1
//NFSLOG2  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.SERVER.LOG2
//FHDBASE  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.FHDBASE1
//FHDBASE2 DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.FHDBASE2
//LDBASE   DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.LDBASE
//LDBASE2  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.LDBASE2
//NFSXLAT  DD   DISP=SHR,DSN=&NFSPRFX..&SYSNAME.MVS.XLAT
```

Figure 7-2 Example of the 1.7 NFS server started task JCL

Note: An example of the NFS Server startup procedure is available in data set SYS1.NFSSAMP as member GFSAPROC.

Figure 7-3 shows a typical sequence when the NFS V1.7 Server starts. The NFS Server is started successfully when the GFSA348I message is shown in the log. The startup message GFSA348I can be used as a good indication for the system automation process to handle the NFS Server.

```
S NFS
IEF695I START NFS      WITH JOBNAME NFS      IS ASSIGNED TO USER STC
   , GROUP SYS1
IEF403I NFS - STARTED - TIME=12.15.17 - ASID=008C - SC65
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
IEC161I 072-053,NFS,NFS,FHDBASE,,,OS390NFS.SC65MVS.FHDBASE1, 837
IEC161I OS390NFS.SC65MVS.FHDBASE1.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,FHDBASE2,,,OS390NFS.SC65MVS.FHDBASE2, 838
IEC161I OS390NFS.SC65MVS.FHDBASE2.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,LDBASE,,,OS390NFS.SC65MVS.LDBASE, 839
IEC161I OS390NFS.SC65MVS.LDBASE.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,LDBASE2,,,OS390NFS.SC65MVS.LDBASE2, 840
IEC161I OS390NFS.SC65MVS.LDBASE2.DATA,UCAT.VSBOX01
GFSA348I (NFS     ) Z/OS NETWORK FILE SYSTEM SERVER (HDZ11VS ,
OA15050) STARTED.
```

Figure 7-3 NFS Server startup sequence

Possible problems that this message does not indicate are that the required services PORTMAPPER, TCPIP, and USS are not available, or when the NFS de facto standard port 2049 is not available.

7.4 z/OS NFS V1.7 line items

This section illustrates the new z/OS NFS 1.7 line items for both the NFS server and the NFS client. Note that most line items are for the z/OS NFS server, but there are z/OS NFS client line items also.

> **Note:** The enhancements for the z/OS NFS client only are described in 7.5.2, “z/OS NFS client line items” on page 279.

7.4.1 Miscellaneous small enhancements

This section provides information about the miscellaneous small enhancements of NFS V1.7.

Large format data set support

Large format data sets are a type of physical sequential data set other than extended format data sets, which can grow beyond a size limit of 65 535 tracks on each volume. That size limit applies to conventional (basic format) sequential data sets. Large format data sets can exploit the increased storage capacity of most hardware storage devices, and reduce the need for very large data sets to span multiple volumes. To create a large format data set with z/OS NFS, specify a data class that has a dsntype value of *large*. The DSORG value must be PS, PSU, or omitted. The DFSMS large data set support is described in 2.1, “Large format data sets” on page 12.

Mixed case password support

This line item provides the mixed case password support for the z/OS NFS server to allow the mixed case passwords send by the NFS client. It is not available prior to z/OS V1.7.

The basic implementation is done by z/OS V1R7 RACF to allow mixed case password support. The field RCVTFLG3 in the common vector table (CVT) indicates whether this support is activated.

When the bit in RCVTFLG3 is active, the z/OS NFS server bypasses the conversion of the password to upper case and passes it to RACF *as is*.

The RACF command SETROPTS PASSWORD(MIXEDCASE) will turn on the bit in the RCVTFLG3 field. More information can be found in the RACF APAR OA08988.

Merged and restructured publication

The new manual *z/OS Network File System Guide and Reference,* SC26-7417, contains information that was previously presented in the following manuals:

- *z/OS Network File System Customization and Operation*, SC26-7417
- *z/OS Network File System User's Guide*, SC26-7419
- *z/OS Network File System Performance Tuning Guide*, SC26-7418

The advantage of the combination of the manuals is the elimination of redundancies for easier reading and to provide easier access to the required information.

Subdivided into logical parts for easy access

The new NFS publication is divided into the following logical parts. Each part contains multiple chapters:

- Introduction
- NFS User's Guide
- Customization and Operations
- Performance Tuning
- Diagnosis and Messages
- Appendixes

Operator commands

This section discusses the new LISTLOCK and the enhanced RELEASE NFS Operator command. This support is for the z/OS NFS Server only.

LISTLOCK operator command

This operand lists client NFS processes that hold locks for a specified file on the z/OS Network File System server. The list of lock holders can be used to diagnose problems with locking conflicts. The LISTLOCK operand lets you specify an MVS data set, PDS or PDSE member, or z/OS UNIX file, and writes a message (GFSA791I or GFSA792I) for each unique user ID and client host pair that holds locks for it.

When APAR OA16306 is applied, the LISTLOCK messages will be written to the console if the list of locks/shares is short (less than 10). When the list of locks/shares is greater than 10, then all LISTLOCK messages will be written only to the log data sets of the z/OS NFS server.

Without APAR OA16306, all LISTLOCK messages will be written to the console.

The general command syntax follows.

```
MODIFY mvsNFS,listlock='file_name'
```

- *file_name*

  This is the name of an MVS data set, z/OS UNIX file, PDS member, or PDSE member. If file_name references a PDS or a PDSE data set (does not include the member name), a syntax error will result. A z/OS UNIX file must be specified using the prefix identified in the NFS site attributes parameter HFS. For example, if the HFS(/hfs) site attribute was specified, LISTLOCK='/hfs/u/user' will indicate the z/OS UNIX file /u/user.

  If the specified file does not have any locks, message GFSA793I is issued to report that no locks exist.

  Since the lock information may only be reported to the server log data set or component trace buffer, and not back to the console, message GFSA794I is issued to indicate that the listlock function completed successfully.

  Response messages are sent to the console if there are 10 or fewer locks to be reported. Otherwise, the response messages are only sent to the NFS log data set.

- Example for a HFS file:

  ```
  /F <mvsNFS>,LISTLOCK='/hfs/u/user'
  ```

  Note: hfs is case sensitive.

- Example for a MVS data set:

  ```
  /F <mvsNFS>,LISTLOCK=user1.testfile
  ```

  Note: <mvsNFS> is the name of the started task (stc) of the NFS Server.

Figure 7-4 on page 216 illustrates a test sequence using OPEN/CLOSE semantics and share reservations. It shows the issued commands to lock the file and the appropriate operator commands to list the status of the HFS files while the tests are done using HFS files.

```
F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'

GFSA791I (NFSJSR56) Owner(SJVM5153 mumble.storage.sanjose.ibm.com 00000836
1ff 000045a5) offset=000000000 len=00000000a lockacc=EX status=G
GFSA792I (NFSJSR56) Owner(SJVM5153 mumble.storage.sanjose.ibm.com
000001ff 00000000) access=R deny=DW
GFSA794I (NFSJSR56) LISTLOCK /hfs/nlm2/NFSOPEN1 command was completed
successfully
   Note:The info 'access=R deny=DW' of the message GFSA791I shows that there is
   a share reservation lock held for Read access and deny the Write access
   shown by the deny mode DW.

F NFSJSR56,RELEASE='/hfs/nlm2/NFSOPEN1'

GFSA914I (NFSJSR56) /hfs/nlm2/NFSOPEN1 Deallocated.

F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'
GFSA793I (NFSJSR56) /hfs/nlm2/NFSOPEN1 does not have Locks.

F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'

GFSA791I (NFSJSR56) Owner(SJVM5153 mumble.storage.sanjose.ibm.com 00000850
1ff 000045a5) offset=0001e0000 len=000000014 lockacc=SH status=G
GFSA792I (NFSJSR56) Owner(SJVM5153 mumble.storage.sanjose.ibm.com
000001ff 00000000) access=W deny=DR
GFSA794I (NFSJSR56) LISTLOCK /hfs/nlm2/NFSOPEN1 command was completed
successfully
   Note:The info 'access=W deny=DR' of the message GFSA791I shows that there is
   a share reservation lock held for Write access and deny the Read access
   shown by the deny mode DR.

F NFSJSR56,RELEASE='/hfs/nlm2/NFSOPEN1'
GFSA914I (NFSJSR56) /hfs/nlm2/NFSOPEN1 Deallocated.

F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'
GFSA793I (NFSJSR56) /hfs/nlm2/NFSOPEN1 does not have Locks.

F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'

GFSA792I (NFSJSR56) Owner(SJVM5153 mumble.storage.sanjose.ibm.com
000001ff 08937652) access=RW deny=DN
GFSA794I (NFSJSR56) LISTLOCK /hfs/nlm2/NFSOPEN1 command was completed
successfully
   Note:The info 'access=RW deny=DN' of the message GFSA791I shows that there
   is a share reservation lock held for Read/Write access and do not deny any
   access shown by the deny mode DN.

F NFSJSR56,RELEASE='/hfs/nlm2/NFSOPEN1'
GFSA914I (NFSJSR56) /hfs/nlm2/NFSOPEN1 Deallocated.

F NFSJSR56,LISTLOCK='/hfs/nlm2/NFSOPEN1'
GFSA793I (NFSJSR56) /hfs/nlm2/NFSOPEN1 does not have Locks.
```

Figure 7-4 Stateful NFS V4 tests using share reservations

z/OS NFS Server V1.7 LISTLOCK messages

This section illustrates the new messages of the LISTLOCK command.

- GFSA791I

  ```
  GFSA791I (procname) Owner (serverid clientid userid processid) offset=hex1
  len=hex2 lockacc=mode status=status
  ```

  – Explanation: This message displays a byte range lock owner for the file requested in the LISTLOCKS operator command. Message GFSA791I will be returned for each lock. The lock owner is identified by serverid, clientid, userid, and processid. The lock is for offset hex1 and length hex2. The lock access mode (share/exclusive) and the current lock status (waiting/granted) are also displayed.

  – System action: The Network File System server continues.

  – Operator response: None.

  – System programmer response: None.

  – Source: z/OS NFS Server.

- GFSA792I

  ```
  GFSA792I (procname) Owner (serverid clientid userid processid) access=access
  deny=deny
  ```

  – Explanation: This message displays a lock share holder for the file requested in the LISTLOCKS operator command. Message GFSA792I will be returned for each share holder. The share owner is identified by serverid, clientid, userid, and processid. The lock access mode access and deny mode deny are displayed.

  – System action: The Network File System server continues.

  – Operator response: None.

  – System programmer response: None.

  – Source: z/OS NFS Server.

- GFSA793I

  ```
  GFSA793I (procname) text does not have Locks.
  ```

  – Explanation: This message is returned from the LISTLOCKS operator command for the specified file text, if the file does not have any locks or shares.

  – System action: The Network File System server continues.

  – Operator response: None.

  – System programmer response: None.

  – Source: z/OS NFS Server.

- GFSA794I

  ```
  GFSA794I (procname) LISTLOCK text command was completed.
  ```

  – Explanation: This message indicates that processing of the LISTLOCKS operator command for the specified file text has been completed. Any existing locks will be listed in message GFSA794I.

  – System action: The Network File System server continues.

  – Operator response: None.

  – System programmer response: None.

  – Source: z/OS NFS Server.

RELEASE operator command

Before z/OS NFS 1.7, the RELEASE command only supported z/OS data sets and members' PDS and PDSEs. It has been enhanced to support USS files and to release any locks that exist for the file at the time the command is issued. Byte range lock requests that are waiting will be purged.

This operand forces the Network File System to close and deallocate the MVS data set or PDS/PDSE member specified. Any locks that exist will also be released and any waiting locks for the data set or member will be cancelled. Any waiting byte range lock requests (for the specified data set) will be purged. File share reservations will also be released. For a z/OS UNIX file, any locks held will be released and any waiting locks will be canceled. z/OS UNIX file names must be specified in single quotation marks (' ') with the hfs prefix (/hfs) followed by the z/OS UNIX path. This command can be entered at any time.

> **Warning:** The NLM protocol does not provide any means for the NFS Server to notify the NFS Client that the locks have been released. Use of the RELEASE command against files with NLM locks can lead to data corruption because the NFS Client will be unaware of the fact that its locks are no longer held.

- An example for an MVS data set member is:

  ```
  MODIFY mvsNFS,release=data_set_name (member)
  ```

  Where:

  data_set_name is the name of an MVS data set or an z/OS UNIX file. If data_set_name references a PDS or a PDSE data set (does not include the member name), a syntax error will result.

  If the object you specify is active, you receive the message:

  ```
  GFSA914I data.set.name(member) DEALLOCATED
  ```

  If the object you specify is not active, you receive the message:

  ```
  GFSA915I data.set.name(member) NOT ALLOCATED
  ```

- An example for z/OS UNIX file names follows. Note that the HFS path is specified in single quotation marks (' '):

  ```
  f mvsNFS,RELEASE='/hfs/u/user'
  ```

z/OS NFS Server V1.7 RELEASE messages

This section illustrates the z/OS NFS Server messages related of the RELEASE command.

- GFSA914I

  ```
  GFSA914I (procname) dsname DEALLOCATED.
  ```

  – Explanation: The data set name dsname appears in response to a release=dsname(member) operand of the modify command after successful deallocation.
  – In the message text *procname* is the name of the started procedure.
  – System action: NFS processing continues.

- GFSA915I

  ```
  GFSA915I (procname) dsname NOT ALLOCATED.
  ```

  – Explanation: The data set name dsname appears in response to the release=dsname(member) operand of the MODIFY command if the data set or member specified to be released was not found.

– In the message text *procname* is the name of the started procedure.
– System action: NFS processing continues.

7.4.2 Serviceability improvements

This section provides information about the serviceability improvements of NFS V1.7. All the serviceability improvements in NFS 1.7 are for both the z/OS NFS server and the z/OS NFS client.

Upgrade compiler level and remove Language Environment modules

NFS V1.7 now exploits the latest compiler improvements.

The Language Environment® modules are removed to minimize the update incompatibilities of the past and to exploit the latest level on the client system. The changes are acceptable because the compiler technology has matured. NFS will now use the client's LE libraries, which Binder will link to on the client's system.

Module-level parameterization

This line item provides information about the appropriate module parameterization and implementation as preparation for the new VERSION commands. It Implements the support for tracking the level of the z/OS NFS server and z/OS NFS client modules.

This support checks the module levels during NFS start-up and provides the following attributes:

- Compatible release levels of modules
- The same level of part in all load modules
- VERSION operator command

VERSION commands

There are new NFS version commands for the z/OS NFS Server and the z/OS NFS client. New commands are available for querying the module levels to display levels of all parts, display level of a specific part, and provides the release, the APAR, and the compile date and time.

The new support will be externalized through the new z/OS NFS Server operator command and the new z/OS NFS Client shell commands. The commands can be used to determine the level of the product or of a single part and to validate installation of an APAR.

VERSION command syntax

This section illustrates the syntax of the new z/OS NFS server VERSION commands and shows examples.

- Specifying the VERSION command with no additional keywords provides an overview of the NFS Server, as shown in Figure 7-5.

```
/F  mvsNFS,VERSION
```

```
GFSA944I (NFS     ) Z/OS NETWORK FILE SYSTEM SERVER RELEASE HDZ11VS ,
LAST APAR OA15050, LAST CHANGED MODULE: GFSASCOM COMPILED AT TUE MAR  7
10:25:00 2006
```

Figure 7-5 Output of the f NFS,version command

- Specifying the VERSION command with the ALL keyword provides a list of all modules and related maintenance information, as shown in Figure 7-6.

  `/F  mvsNFS,VERSION=ALL`

```
GFSA945I (NFS     ) GFSAAACC | OA11955 | WED DEC 21 10:47:18 2005
GFSA945I (NFS     ) GFSAAARN | HDZ11VS | FRI APR 15 22:01:00 2005
GFSA945I (NFS     ) GFSAAATT | OA11955 | FRI JAN  6 16:48:30 2006
GFSA945I (NFS     ) GFSAACRE | OA15114 | THU MAR  2 09:14:36 2006
GFSA945I (NFS     ) GFSAAFAM | HDZ11VS | WED DEC 21 10:47:35 2005
::::::::::::::::::
GFSA945I (NFS     ) GFSA4XDR | OA11875 | WED DEC 21 11:33:11 2005
GFSA946I (NFS     ) TOTAL MODULES: 144
```

Figure 7-6 Maintenance levels of all the NFS Server modules

- Specifying the VERSION command with the MODULE keyword information about a specific ll module and related maintenance information. It will show the base level or the APAR level of the module as appropriate.

  `/F  mvsNFS,VERSION=module`

  Figure 7-7 shows the output of the command **`/f NFS,VERSION=GFSAAARN`**.

  Note that the NFS Server is at the base level, as indicated by the FMID being listed. There has been no maintenance applied.

```
F NFS,VERSION=GFSAAARN
GFSA945I (NFS     ) GFSAAARN | HDZ11VS | FRI APR 15 22:01:00 2005
```

Figure 7-7 Base level installed for a specific module

Figure 7-8 shows the output of the command /F NFS,VERSION=GFSAXTM2. Note that the APAR level of the module is listed.

```
F NFS,VERSION=GFSAXTM2
GFSA945I (NFS     ) GFSAXTM2 | OA14584 | FRI JAN  6 16:55:33 2006
```

Figure 7-8 APAR level installed for a specific module

NFS Client - shell command

This section provides information about the syntax of the new z/OS NFS Client VERSION command and shows some examples.

> **Note:** The **`NFSstat`** command for the NFS Client available at the USS path /usr/lpp/NFS must be issued from there.

- Specifying the VERSION command with no additional keywords provides an overview of the NFS Client, as shown in Figure 7-9 on page 221.

  `# NFSstat -v`

```
z/OS Network File System Client release HDZ11VC,
last apar OA14847,
last changed module GFSCMNTD compiled at Feb 22 2006 10:56:35
```

Figure 7-9 APAR level of the NFS Client

- Specifying the VERSION command with the -a parameter lists all of the client modules, as shown in Figure 7-10.

  `NFSstat -v a`

```
GFSCAMVS | HDZ11VC | Apr 15 2005 22:44:00
GFSCBIOA | HDZ11VC | Jun  9 2005 08:15:09
GFSCBIOB | OA14703 | Jan 26 2006 11:29:05
GFSCBIOC | OA11737 | Sep  8 2005 14:07:23
GFSCBIOD | OA01525 | Feb  1 2006 10:08:25
GFSCCERR | HDZ11VC | Jun  9 2005 08:16:29
:::::::::::::::::::::
GFSCORPC | OA01525 | Feb  1 2006 10:09:34
GFSC1MEM | HDZ11VC | Apr 15 2005 22:45:00
GFSC1RPC | HDZ11VC | Jun  9 2005 08:44:07
Total count: 33
```

Figure 7-10 Level of all NFS Client modules

- Specifying the VERSION command with the t parameter and the name of the task provides the APAR level of the task, as shown Figure 7-11.

  `# NFSstat -v t gfscmain`

```
GFSCVMNT | OA14605 | Feb 15 2006 08:49:42
```

Figure 7-11 APAR level of the GFSCMAIN task

- Specifying the VERSION command with the m parameter and the name of the module provides the APAR level of the specific module and lists all of the client modules, as shown in Figure 7-12.

  `# NFSstat -v m gfsc1mem`

```
GFSC1MEM | HDZ11VC | Apr 15 2006 22:45:00
```

Figure 7-12 APAR level of a special module

Prelink elimination

This section describes the pre-NFS 1.7 maintenance process and the new packaging process in NFS V1.7, which eliminates the prelink step.

Pre-NFS V1.7

Prior to NFS V1.7 the NFS packaging structure was based on the nine z/OS NFS Server load modules and the two z/OS Client load modules.

Each APAR and PTF had to ship the entire prelink for the load module, including all object modules in that load module if required.

The disadvantages of this packaging method are:

- Each APAR/PTF is much larger than necessary because other object modules must be shipped.
- The ability to work on multiple APARs in parallel is minimized. Multiple problem fixes packaged in each APAR/PTF delayed the closure of all APARs.

Figure 7-13 illustrates the pre-NFS V1.7 NFS packaging process.

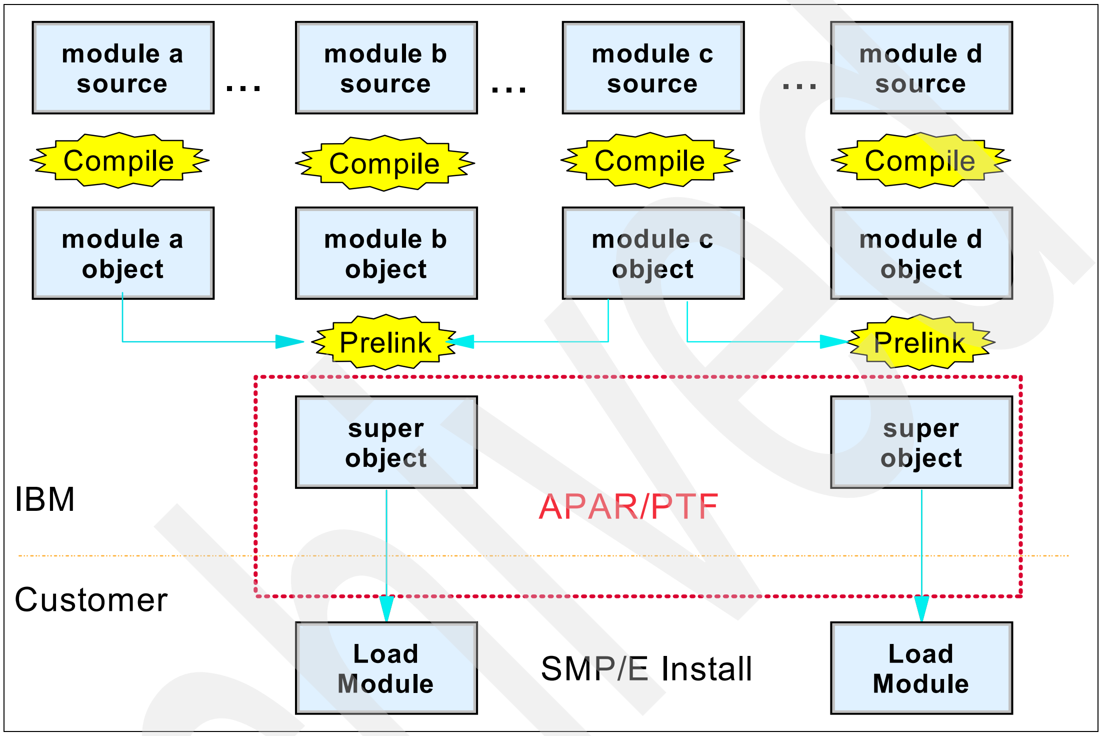


Figure 7-13 Current NFS packaging process - example

Assume that module C is part of two load modules: Then:

- It is prelinked into both super objects.
- Binder builds load modules from those super objects.

Assume that a fix must be made to module C. Then:

- IBM updates the module.
- Rebuilds both super objects, containing all their modules.

The appropriate APAR/PTF must contain both super objects (all modules).

NFS V1.7 packaging restructure

This section describes the new restructured packaging process with z/OS NFS V1.7. The new installation process exploits new binder capabilities. A two-stage bind has been changed to a one-stage bind.

Each APAR only ships modified parts, while the prelink process in between is eliminated. SMP/E will handle the consolidation when the changed part is in multiple load modules.

The library of the NFS load modules SYS1.NFSLIB must be reallocated as a PDSE data set named SYS1.NFSLIBE.

The APARs/PTFs are much smaller because other object modules are not shipped, which improves the potential when having multiple APARs in parallel. This allows for a single fix to be packaged in each APAR or PTF, which results in faster APAR closure and faster availability of its PTF.

> **Note:** Modules are moved from load modules to program objects. Thus clients must use a PDSE now instead of a PDS to exploit the new Binder functions. Packaging rules require the new library name.

Figure 7-14 shows the new NFS packaging process example.

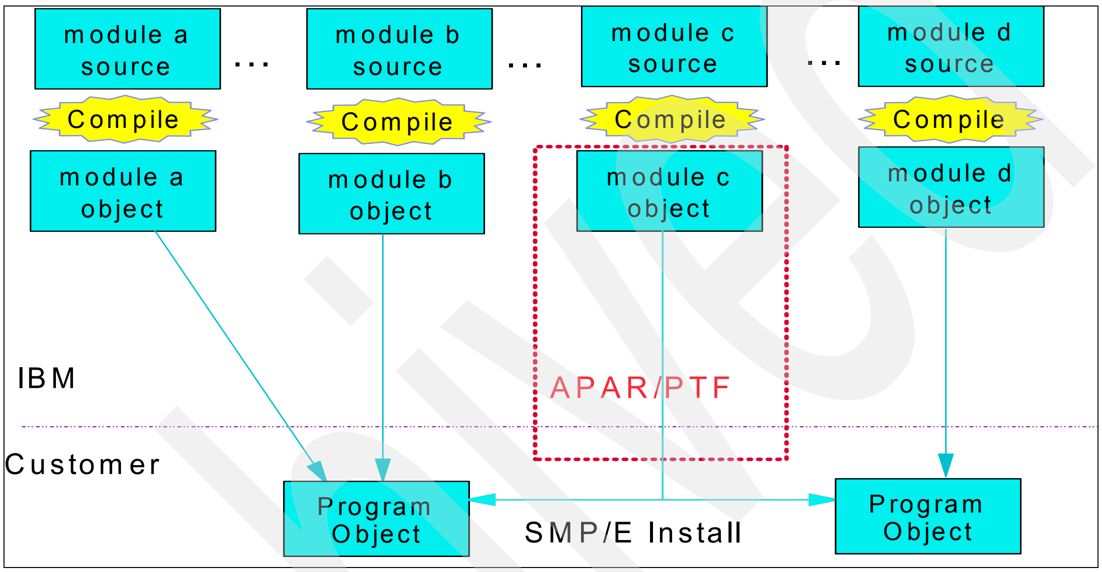


Figure 7-14 The NFS V1.7 packaging process

Assume that module C is part of two load modules. The Binder builds the load modules. Assume that a fix must be made to module C, and then IBM updates the module and the APAR/PTF contains only the modified part. SMP/E installs the modified module correctly.

NFS Server CTRACE exploitation

This section provides information about the new CTRACE function and gives an overview of its functionality and how to set it up.

Note that the CTRACE support is for the z/OS NFS Server only. There is no CTRACE available for the z/OS NFS Client.

> **Note:** The new NFS Server CTRACE is equivalent to the existing CTRACE for other components on z/OS, for example, TCPIP and USS.

CTRACE summary

To convert the external trace file to the component trace means to convert NFS debug logging to use the CTRACE mechanism. The advantages are:

- The compatibility with the debug logging for other components
- The improved performance because of tracing to memory

The disadvantages are:

- The CTRACE may wrap faster when using memory only.
- A dump must be taken to capture the in-memory CTRACE.

The compatibility modes are the following:

- The error/warning/informational messages can still go into the NFS log data sets.
- The CTRACE external writer support can also be used.
- The debug trace data can be directed to the CTRACE or the log data sets.
- Having the trace buffer in the data space means minimal memory impact.

The CTRACE is stored in the data space of the z/OS NFS server address space using circular buffers. The benefit is low impact to the z/OS NFS Server performance, while having a consistent interface as for other z/OS components.

The summary and the attributes of the new CTRACE support are the following:

- The SYS1.PARMLIB member CTINFS00 is the default CTRACE control member.
- The CTRACE is controlled by the TRACE CT operator command.
- The CTRACE data can be written into the data space or on DASD by the external writer.
- CTRACE is controlled by the following new z/OS NFS Server startup parameters:
  - DSPS=xx defines the CTRACE buffer size in MegaBytes.
  - CTRACE=yy specifies different SYS1.PARMLIB(CTINFSyy).
- Messages are still written to z/OS NFS Server Log data sets using the command:

  `/F MVSNFS,LOG=xx`

> **Note:** The CTRACE buffer is a data space that is separate from the address space. The default value of DSPS is 1 with max 600 MB set by the parms='dsps=nn'.

ctrace=nn specifies that diagnostic information for the NFS server be recorded in z/OS component trace buffers, using trace options specified in member CTINFS*nn* of SYS1.PARMLIB. To use the default trace options for the z/OS NFS server, specify *ctrace=00* for the default SYS1.PARMLIB member CTINFS00.

dsps=nn specifies the size of the data space to be allocated for the NFS component trace buffers, where (*nn* * 3) equals the number of megabytes to be allocated. The value range for *nn* is from 1 (the default), which allocates a 3 MB data space, to 600 (which allocates a 1.8 GB data space).

Figure 7-15 shows an extract of the startup procedure of the NFS Server with a combination of the new Ctrace parameter dsps and ctrace.

```
//NFSMVS  PROC MODULE=GFSAMAIN,PARMS='INFO,CTRACE=00,DSPS=600',
//           SYSNFS=SYS1,
//           NFSPRFX=OS390NFS,
//           TCPIP=TCPIPMVS,
//           TCPDATA=TCPDATA
//GFSAMAIN EXEC PGM=&MODULE,
//             REGION=0M,
//             TIME=1440,
//         PARM=(,
//         'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP")/&PARMS')
```

Figure 7-15 Example of the new startup Ctrace parameter

CTRACE - obtain status information

This section shows how the CTRACE can be started, how to provide the appropriate Ctrace information, and how to stop the CTRACE facility.

Starting Ctrace

NFS uses the parameters in the CTINFSxx member of PARMLIB. In the output shown in Figure 7-16 the member CTINFS00 has TRACEOPTS OFF specified. To start the NFS server use the command /S NFS.

```
  IEF403I NFS - STARTED - TIME=14.52.22 - ASID=0088 - SC65
  IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
  GFSA348I (NFS     ) Z/OS NETWORK FILE SYSTEM SERVER (HDZ11VS,OA15050)
  STARTED.
```

Figure 7-16 Output of NFS START command

Displaying TRACE information

The command **/D Trace** is used to display CTRACE, as shown in Figure 7-17.

```
IEE843I 14.57.20  TRACE DISPLAY 393
        SYSTEM STATUS INFORMATION
 ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K)
 COMPONENT MODE  COMPONENT MODE  COMPONENT MODE  COMPONENT MODE
 --------------------------------------------------------------
 CSF       MIN   SYSGRS    MIN   SYSANTOO  MIN   SYSRRS    MIN
 SYSSPI    OFF   SYSJES    SUB   NFSS      ON    SYSJESB   SUB
 SYSHZS    ON    SYSSMS    OFF   SYSDLF    MIN   SYSOPS    ON
 SYSXCF    ON    SYSLLA    MIN   SYSXES    ON    SYSAPPC   OFF
 SYSTTRC   OFF   SYSTCPDA  SUB   SYSRSM    OFF   SYSAOM    OFF
 SYSVLF    MIN   SYSTCPIP  SUB   SYSLOGR   ON    NFS       OFF
 SYSOMVS   ON    SYSWLM    MIN   SYSTCPIS  SUB   SYSTCPRE  SUB
 SYSIOS    MIN   SYSANTMN  MIN   SYSDMO    MIN   SYSIEFAL  ON
```

Figure 7-17 Display Trace command output

Figure 7-18 shows the output from the display command specifying the component NFS.

```
/D TRACE,COMP=NFS

IEE843I 14.59.48  TRACE DISPLAY 395
        SYSTEM STATUS INFORMATION
 ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K)
  COMPONENT     MODE BUFFER HEAD SUBS
 -----------------------------------------------------
NFS           OFF
   ASIDS      *NOT SUPPORTED*
   JOBNAMES   *NOT SUPPORTED*
   OPTIONS    NONE
   WRITER     *NONE*
```

Figure 7-18 D TRACE command output specifying the component NFS

Figure 7-19 shows the command to start the component trace specifying the member of Parmlib that contains the trace options.

```
/TRACE CT,ON,COMP=MVSNFS,PARM=CTINFS01
IEE252I MEMBER CTINFS01 FOUND IN SYS1.PARMLIB
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND WERE
SUCCESSFULLY EXECUTED.
```

Figure 7-19 Explicitly with a new SYS1.PARMLIB(CTINFSyy)

Figure 7-20 shows starting a trace without specifying the Parmlib member containing the trace options. In this case you are prompted for the options.

```
/TRACE CT,ON,COMP=MVSNFS
TRACE CT,ON,COMP=NFS
*374 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 374,OPTIONS=(ENTRY,EXIT),END
IEE600I REPLY TO 374 IS;OPTIONS=(ENTRY,EXIT),END
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
```

Figure 7-20 Starting trace with prompting for options

Figure 7-21 shows the command and output after stopping the z/OS NFS Server CTRACE.

```
/TRACE CT,OFF,COMP=NFS
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,064K) 435
        ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
        ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Figure 7-21 Stopping NFS CTRACE

Capturing CTRACE data

This section provides information about how to capture the CTRACE data. Captured CTRACE data can be written out using:

- The SVC dump
- The summary dump
- The stand-alone dump
- The external CTRACE writer

The most convenient way is to use the SVC dump, which can be obtained through the following operator commands:

- DUMP COMM=(' NFSdump')
- R xx,JOBNAME=(NFS),dspname=('NFS'.NFSsctds),END

There is one data space available for the NFS Server. Its name is *<NFSsctds>*. It is possible to use an asterisk (*) to define the data space of the NFS Server like dspname=('NFS'.*) as for the other CTRACEs on z/OS.

Using IPCS to view the trace

This section illustrates how to look at the CTRACE via the IBM standard Analysis tool Interactice Problem Control System (IPCS).

The following figures show how to set up the new NFS CTRACE.

From the Interactice Problem Control System (IPCS) Primary Option Menu shown in Figure 7-22 select option 2.

```
------------------- z/OS 01.07.00 IPCS PRIMARY OPTION MENU
OPTION  ===>   2

   0  DEFAULTS    - Specify default dump and options
   1  BROWSE      - Browse dump data set
   2  ANALYSIS    - Analyze dump contents
   3  UTILITY     - Perform utility functions
   4  INVENTORY   - Inventory of problem data
   5  SUBMIT      - Submit problem analysis job to batch
   6  COMMAND     - Enter subcommand, CLIST or REXX exec
   T  TUTORIAL    - Learn how to use the IPCS dialog
   X  EXIT        - Terminate using log and list defaults
```

Figure 7-22 IPCS POM

Next select option 7 to format tracing, as shown in Figure 7-23.

```
------------------- IPCS MVS ANALYSIS OF DUMP CONTENTS ---------
OPTION  ===>   7

To display information, specify the corresponding option number.

   1  SYMPTOMS    - Symptoms
   2  STATUS      - System environment summary
   3  WORKSHEET   - System environment worksheet
   4  SUMMARY     - Address spaces and tasks
   5  CONTENTION  - Resource contention
   6  COMPONENT   - MVS component data
   7  TRACES      - Trace formatting
```

Figure 7-23 Trace formatting

Figure 7-24 shows the entry of the IPCS trace processing. Select option 1 for CTRACE.

```
--------------------------- IPCS Trace Processing ------------------
OPTION  ===>    1

To display trace information, enter the corresponding option number.

   1  CTRACE      - Component trace
   2  GTFTRACE    - Generalized trace facility
   3  MTRACE      - Master trace
   4  SYSTRACE    - System traces
   5  MERGE       - Merge multiple traces
   T  TUTORIAL    - Details on these traces
```

Figure 7-24 IPCS trace processing

Figure 7-25 shows how to query the actual NFS CTRACE settings.

```
-------------------------- CTRACE OPTION MENU  ------------
OPTION  ===> q

   To display CTRACE information, enter the corresponding o

   Q  QUERY       - Specify parameters for QUERY report
   D  DISPLAY     - Specify parameters to display CTRACE en
   S  START       - Start CTRACE subcommand as specified be
   R  RESET       - Reset the CTRACE parameters

Pending CTRACE subcommand shown below:
```

Figure 7-25 Specify parameters for QUERY report

Figure 7-26 shows how to query the actual NFS CTRACE settings.

```
--------------------- CTRACE QUERY PARAMETERS  ----------------
COMMAND ===> s

  Enter/verify CTRACE QUERY parameters below:

  System      ===>          (System name or blank)
  Component   ===> NFS      (Blank for all active components)
  Subnames    ===>

  Report type ===> FULL     (Short or Full, short is default)

  GMT/LOCAL   ===> G        (G or L, GMT is default)
  Subname
  entry panel ===>
  Override
    source    ===>
```

Figure 7-26 NFS CTRACE parameters

Figure 7-27 shows the options and start and end times of how long the NFS CTRACE was running.

```
COMPONENT TRACE QUERY SUMMARY

 COMP(NFS)

               START = 04/27/2006 18:52:23.959448   GMT
               STOP  = 04/27/2006 19:39:55.91817

             OPTIONS: ENTRY,EXIT
```

Figure 7-27 Actual NFS CTRACE settings

Figure 7-28 shows how to display the CTRACE data instead of just query the CTRACE information.

```
-------------------------- CTRACE OPTION MENU  ------------------
OPTION  ===> d

   To display CTRACE information, enter the corresponding option.

   Q  QUERY       - Specify parameters for QUERY report
   D  DISPLAY     - Specify parameters to display CTRACE entries
   S  START       - Start CTRACE subcommand as specified below
   R  RESET       - Reset the CTRACE parameters
```

Figure 7-28 Display the NFS CTRACE data

Figure 7-29 illustrates how to get NFS CTRACE data.

```
-------------------- CTRACE DISPLAY PARAMETERS  ------------------------
COMMAND ===> s

  System      ===>          (System name or blank)
  Component   ===> NFS      (Component name (required))
  Subnames    ===>

  GMT/LOCAL   ===> G                       (G or L, GMT is default)
  Start time  ===>                         (mm/dd/yy,hh:mm:ss.dddddd or
  Stop time   ===>                          mm/dd/yy,hh.mm.ss.dddddd)
  Limit       ===> 0        Exception ===>
  Report type ===> FULL     (SHort, SUmmary, Full, Tally)
  User exit   ===>          (Exit program name)
  Override source ===>
  Options         ===>
```

Figure 7-29 Define how to display the NFS Ctrace data

Figure 7-30 shows the NFS CTRACE data based on the selected options ENTRY and EXIT.

```
COMPONENT TRACE FULL FORMAT
COMP(NFS)
**** 04/27/2006

SYSNAME   MNEMONIC  ENTRY ID    TIME STAMP     DESCRIPTION
-------   --------  --------  ---------------  -------------

SC65      VERS_1.1  00000001  18:52:23.959448  NFSS_TRACE

C o m m o n   f i e l d s:
TaskName. GFSAMAIN  ModName.. GFSANMAI  RecType.. INFO
LinNm(d). 01577     Function. main
pReq..... 00000000  ReqID.... 00000000  Operatn.. NULL
DataCnt.. 00000001
T r i a d s:
Message Str
Thu Apr 27 14:52:23 2006
```

Figure 7-30 NFS CTRACE data FULL format

Note: The CTRACE data can vary based on the options defined for the NFS CTRACE.

7.4.3 IPV6 support

This section provides an overview of the new support for IPV6 for the z/OS NFS server and the z/OS NFS client and information about the changes in the SMF record.

The industry is running out of IPV4 addresses. Most of today's Internet uses IPV4, which is approximately 20 years old and is approaching the end of its physical limits. The most significant issue surrounding IPV4 is the growing shortage of IPV4 addresses. In theory, 32 bits allow over 4 billion nodes, each with a globally unique address. In practice, the interaction between routing and addressing makes it impossible to exploit more than a small fraction of that number of nodes. Consequently, there is a growing concern that the continued growth of the Internet will lead to the exhaustion of IPV4 addresses early in the 21st century.

As the solution, a new IPV6 protocol has been defined with the following attributes:

- Data length of 16 bytes versus 4 bytes to store the IPv6 IP address.
- It provides virtually unlimited addresses.
- IPv6 supports imbedded IPv4 address.
- z/OS NFS 1.7 implements the IPv6 support.
- The z/OS NFS server will respond to IPv6 Client requests via IPv6.
- The z/OS NFS server will respond to IPv4 Client requests via IPv4.
- The NFS client will attempt to contact a server via IPv6 first. If there is no response, it will attempt IPv4.

The requirements are:

- The network must be configured and customized for IPV6.
- The z/OS NFS server must be configured and customized for IPV6.

The IPV6 protocol advantages include having expanded addressing capabilities. IPV6 increases the IP address size from 32 bits to 128 bits to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a scope field to multicast addresses. A new type of address called an anycast address is defined, used to send a packet to any one of a group of nodes.

Shown header format simplification

Some IPV4 header fields have been dropped or made optional to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPV6 header:

- Authentication and privacy capabilities

  Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPV6.

- Provides improved support for extensions and options

  Changes in the way IP header options are encoded. Allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- Flow labeling capability

  A new capability is added to enable the labeling of packets belonging to particular traffic *flows* for which the sender requests special handling, such as non-default quality of service or *real-time* service.

RPCBIND verses PORTMAPPER

The RPCBIND protocol is used on the IPV6 network instead of PORTMAPPER protocol. RPCBIND protocol is not supported in z/OS Release 1.7.

Other NFS Clients will not be able to use IPV6 protocol to communicate with the z/OS NFS Server V2 and V3 only. With a NFS V4 this will be possible because IPV6 uses the portmap protocol of V2 or V3, which corresponds only with RPCBIND, not with the portmapper. The NFS V4 IPv6 request will not get a response back because RPCBIND it is not supported with z/OS 1.7. But, since no mount is done using NFS V4 and it is not necessary to have a port, the IPV6 request works using the de-facto standard NFS port 2049 as a default.

User exit impact

There is no user exit impact. The NFS Server still provides unique identifier (terminal ID) of the host with the same length. It does not depend on the version of the network protocol.

Contents of the login exit parameter list

The parameter list is mapped by macro and DSECT GFSAULOG. For configurations that use Internet Protocol Version 4, the client IP address is in field LEDSIA. If your configuration uses IPV6, refer to field LEDSIA6 for the client IP address. When an IPV6 address is provided in LEDSIA6, LEDSIA takes a value of -1. If the DHCP site attribute is specified for the NFS server to use dynamic client IP addresses, the contents of the client IP address field will be correct when the exit parameter list request is processed. However, the exit must not have any dependency on the persistence of the IP address value beyond the duration of the individual request. That IP address can change between requests. Table 7-3 shows an extract of the login installation-wide exit routine parameter list.

Table 7-3 Login installation wide exit routine parameter list

Field name	Description	Contents
LEDSRC	Return code	Codes generated and returned by the calls.
LEDSM	Client machine name	Character string ended by single byte containing X'00'.
LEDSIA	Client IP address	Number (32-bit Internet address). Contains -1 (X'0xFFFFFFFF) if an IPV6 address is provided in the LEDSIA6 field.
LEDSIA6	Client IP address	Number (128-bit Internet address, for Internet Protocol V6 users).

Contents of the file security exit parameter list

The file security installation-wide exit parameter list is mapped by macro and DSECT GFSAUSEC. For configurations that use Internet Protocol V4, the client IP address is in field FEDSIA. If your configuration uses IPV6, refer to field FEDSIA6 for the client IP address. When an IPV6 address is provided in FEDSIA6, FEDSIA takes a value of -1. If the DHCP site attribute is specified for the NFS server to use dynamic client IP addresses, the contents of the client IP address field will be correct when the exit parameter list request is processed. However, the exit must not have any dependency on the persistence of the IP address value beyond the duration of the individual request. That IP address can change between requests.

IPV6 support - z/OS IPV4/IPV6 quick primer

This section gives an overview of the different IPV6 modes.

The IPV6 protocol is described in RFC 2460 (128-bit IP address). More information can be found at the following site:

http://www.faqs.org/rfcs/rfc2460.html

For IPV4-only mode, only IPV4 stack is enabled.

In dual IPV4 and IPV6 mode:

- Both IPV4 AND IPV6 stacks are enabled.
- The application can operate with both types of IP addresses.
- The IPv6 address format is x:x:x:x:x:x:x:x.
- FEDC:BA98:7654:3210:FEDC:BA98:7654:3210.

The IPV4-mapped IPV6 address format is x:x:x:x:x:x:d.d.d.d:

```
0:0:0:0:0:FFFF:129.144.52.38 (or ::FFFF:129.144.52.38 in compressed format)
```

The IPV6-only mode is currently not supported by the z/OS Communication Services (CS).

Additional information can be found in the manual *z/OS CS, IPv6 Network and Application Design Guide*, SC31-8885.

Figure 7-31 shows an overview of a dual IPV4/IPV6 environment.

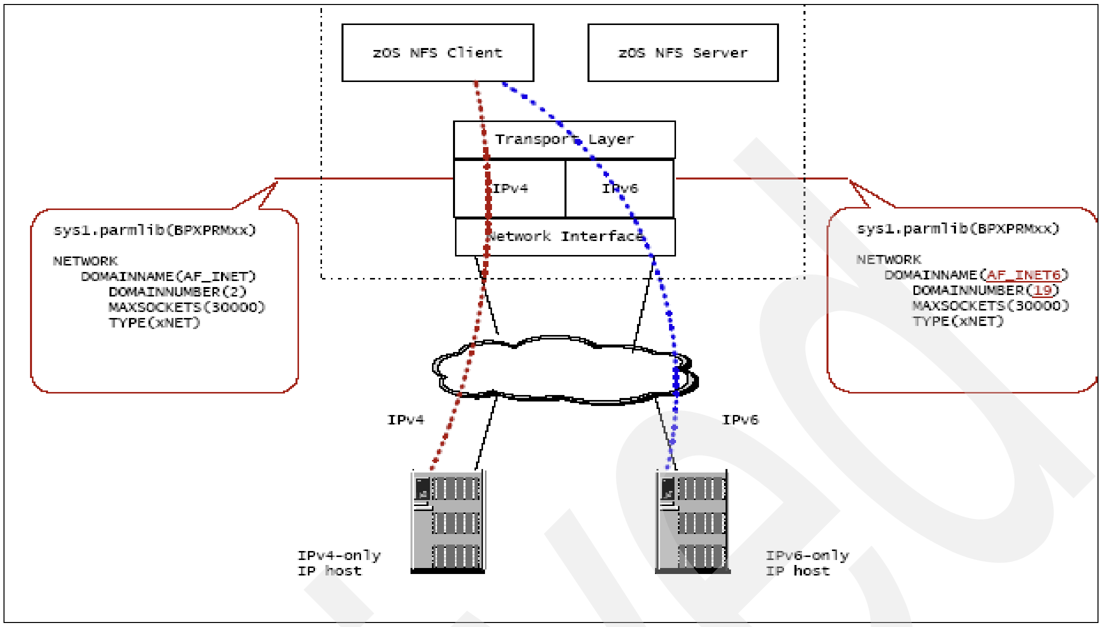


Figure 7-31 Dual mode IPV4/IPV6 environment

In Figure 7-31, the left-hand dotted line shows a IPV4 connection, while the dotted line on the right side represents a IPV6 connection defined by the NETWORK statement of the SYS1.PARMLIB BPX member of the z/OS NFS client system.

The TCP/IP version is defined via the AF_INET DOMAINNAME statement in the member BPXPRMxx of the SYS1.PARMLIB.

Dual mode z/OS IP server host means:

- If a partner is IPV6, then all communication will use IPV6 packets.
- If a partner is IPv4, the following will occur: both source and destination will be IPV4-mapped IPV6 addresses.

On inbound, the transport protocol layer will map the IPV4 address to its corresponding IPV4-mapped IPV6 address before returning to the application with AF_INET6 addresses.

On outbound the transport protocol layer will convert the IPV4-mapped addresses to native IPV4 addresses and send IPV4 packets.

SMF record changes - z/OS NFS Server only

This section provides information about the changes in the SMF records for the IPV6 support.

The z/OS NFS client does not produce any system management facilities (SMF) records. However, it does provide the accounting information to z/OS UNIX for SMF recording. z/OS in turn provides the SMF recording services for all physical file systems (PFSs). The z/OS NFS server does not produce z/OS UNIX SMF records. However, z/OS UNIX provides the SMF recording services for all physical file systems (PFSs). You can use the SMF records that the z/OS NFS server produces to keep track of how MVS conventional data sets are accessed, and how long each Network File System user session lasts. The z/OS NFS server writes the following SMF records:

- Record type-42 subtype 7 - This record, written when a file times out, provides the Network File System file usage statistics.
- Record type-42 subtype 8 - This record, written when a client user logs out of NFS, provides the Network File System user session statistics.

The SMF record also provides the product section and the client identification section.

SMF reports z/OS NFS server only, there are no SMF records for the z/OS NFS client. SMF records are just for Legacy MVS Data Sets only (not for UNIX files).

The main purpose of SMF change is to support IPV6. To provide this support, there was a version field added to each SMF record. This field will be filled by the NFS 1VS version and will remain zero for 1US and 1TS. This version field allows recognizing SMF record changes, now and in the future.

For records containing Internet Protocol (IP) Version 6 addresses, the z/OS NFS server writes a specific type of SMF record. This record type is indicated by a version number of 2 in the smf42psv record field. In these records, the IP address field named smf42cip is expanded to hold the larger IP V6 address values.

SMF reports

This section provides information about the changes to the SMF records regarding the new IPV6 support.

The utilization of the SMF IPV6 logging features is that when started with IPV6, the z/OS NFS server V1R7 will create new type 42 subtype 7 and subtype 8 SMF records.

The z/OS NFS server report tool will tolerate old and new versions of SMF records. The report tool in V1R7 will read and understand old and new versions of the SMF records. The report tool in pre-V1R7 will ignore records created by z/OS NFS server V1R7 for IPV6 connections.

Note: The complete layout of the SMF 42 record is in Appendix K of the *Network File System and Reference*, SC26-7417.

IPV6 support from the z/OS NFS client perspective

This section discusses the support for IPv6 from the z/OS point of view.

RPCBIND(Y|N) NFS Client parameter

A new z/OS NFS Client startup parameter has been defined to bypass the time spent issuing RPCBIND requests to servers that do not support RPCBIND as z/OS.

The new RPCBIND(N | Y) provides a performance improvement by not having to wait for the RPCBIND request to time out.

RPCBIND is used to determine port IDs for IPV6 and portmap is used to determine port IDs for IPV4. This parameter is introduced to support IPV6 protocol with systems that do not support RPCBIND protocol.

The NFS z/OS NFS Client is able to use IPV6 protocol to communicate with NFS servers without using the RPCBIND protocol.

The new input parameter -norpcbind was introduced for NFS Client utilities (SHOWMOUNT,...).

The new NFS Client parameter RPCBIND (Y| N) specifies whether the target NFS server platform supports the RPCBIND protocol, so the NFS client will not have to attempt to use the RPCBIND protocol if that protocol is not supported. The default is rpcbind(Y) to indicate that RPCBIND is supported. If N is specified, the z/OS NFS Client will immediately use the PORTMAPPER protocol instead. This keyword has no effect if the client system is not enabled for TCP/IP V6.

Summary and restrictions for TCP/IP Version 6

TCP/IP V6 expands the range of addresses that are available for Internet communications. TCP/IP V6 extends address sizes from a 32-bit value to a 128-bit value, vastly expanding the number of globally unique addresses that can be assigned. z/OS V1R7 NFS allows both the z/OS NFS client and the z/OS NFS server to support the longer addresses of TCP/IP V6, as well as the 32-bit addresses of TCP/IP V4 and earlier.

The network infrastructure must be enabled to use TCP/IP V6. If the network does not support IP V6, z/OS NFS falls back to use IP V4. Use of TCP/IP V6 depends on the NFS protocol version level. The z/OS NFS server can use TCP/IP V6 for all NFS functions when communicating with the NFS V4 protocol. With the NFS V2 or V3 protocols, the z/OS NFS client is restricted to TCP/IP V4, which supports the portmapper program that clients must use to identify communication ports. For NFS V4 the NFS client does not use the portmapper, so the client can communicate with the NFS server using NFS V4 and TCP/IP V6. However, for the NFS client to use the client enabling utilities, such as mvslogin, mvslogout, and showattr, the client reverts to TCP/IP v4 and attempts to identify the ports via the server portmapper. Once the ports are identified, utility communication with the NFS server is done using TCP/IP V4. SMF records for the z/OS NFS server report client IP addresses for both TCP/IP V4 and V6, with separate address fields for each.

With NFS V2/V3 other NFS clients are restricted to use IPV4 because z/OS only supports portmapper, which does not support IPV6. For IPV6, clients query RPCBIND for NFS ports, but z/OS CS does not support RPCBIND yet.

7.4.4 Dynamic IP (DHCP) support for clients

Before z/OS V1R7, the z/OS NFS client and server were based on the static IP address model to handle all communications with other systems. However, many systems have migrated from the use of static IP addresses to the dynamic host configuration protocol (DHCP). Now the z/OS NFS server accepts dynamic NFS client IP address changes and properly understands the source of the communication even if the sender's IP address has changed. Since not all clients' environments use dynamic IP addresses, NFS server site attributes have been added to specify whether the NFS server should use the dynamic IP algorithm (DHCP) or the current static IP algorithm (NODHCP). The default is NODHCP, to use the static IP algorithm.

The DHCP protocol is described in RFC 2131 (replaced RFC 1541):

http://www.faqs.org/rfcs/rfc2131.html

To use dynamic IP addressing, the client must have a constant host name that the NFS server can identify it by. You can then dynamically update the authentication dynamic name server (DNS) with new IP addresses whenever they change.

Maintain the time to live (TTL) value that the authentication DNS server specifies to any caching DNS server, based on the frequency with which system IP addresses might change.

The larger the TTL value, the greater the possibility that the caching DNS server will have obsolete information. If dynamic addressing is used, the TTL value should be small, ideally zero, but a small value defeats the benefit of caching, so a compromise must be set with the understanding that cached values can become obsolete during the TTL interval and report incorrect information to querying systems like the NFS server.

Note: Regardless of the DHCP/NODHCP attribute value, the z/OS NFS server itself continues to have a static IP address.

z/OS NFS Server site attributes for DHCP

This section gives an overview of the new attributes for DHCP.

The attributes are DHCP and NODHCP (the default is NODHCP). We recommend that start the NFS server with the DHCP site attribute set to nodhcp. If DHCP is desired, shut down the server and restart it with the dhcp site attribute.

The DHCP site attribute effects are:

- The client hosts can change their IP addresses but must not change their host names.
- All NFS client hosts must register with DNS and the DNS service must be available.
- The z/OS NFS server will ignore all MHDB records with empty host names.
- The z/OS NFS server will perform reverse IP lookup (nslookup a.b.c.d) to ensure that the new TCPIP connection is coming from the same host name.

In comparison, the NODHCP site attribute effects are:

- The NFS client hosts should never change their IP addresses.
- The NFS client hosts do not need to be registered with DNS.
- The z/OS NFS Server still performs reverse IP lookup of a new TCPIP connection, but it is done asynchronously.

Impacts

Important: When installing the new release, MHDB must be reallocated due to the format change — longer record size.

The SMF record IP address is not valid in DHCP mode since the IP address may change. The host name must be used instead when compiling statistics.

For user exits the IP address is only valid for the duration of the RPC call. The next RPC call from the same client/user may have a different IP address.

Note: The NFS Server identifies the client by its host name through the export file while DNS resolves the IP address.

Challenge for z/OS NFS server

The NFS Client's IP address can be changed implicitly. After issuing MOUNT or MVSLOGIN requests, clients can change IP addresses. The NFS clients expect their old credentials to be used with the new IP address.

Effects for NFS Clients

Mount points suddenly disappear. Because the mount handle database and mount points are currently attached to the IP address, the changed IP address can cause mount points to disappear. Therefore, the related host name of the IP address will be used instead.

Non-NFS TCP process flow (DHCP option set)

When accepting incoming connections the z/OS NFS server starts reversed name resolution. Using DHCP, there will be a reverse lookup call to the DNS based on the IP address to get the related host name. Then the z/OS NFS server will use the host name only to process the request.

No other remote procedure call (RPC) packets can be processed from this connection until the name is resolved. Follow-on RPC packets will be handled without new call to DNS.

MHDB records are written with IP and host name information.

Non-NFS UDP process flow (DHCP option set)

The UDP protocol, as it is connectionless, does not have connection to the internal host cache block (HCB) of the z/OS NFS server.

Each incoming UDP request will cause the z/OS NFS server to resolve its IP address to its host name, which will increase the performance cost.

Non-NFS TCP and UDP flow (NODHCP)

The client's names are resolved asynchronously for informational purposes (showmount) based on the IP address of the request.

The MHDB records are written with IP addresses and the related host names when they are available.

EXPORTS file and EXPORTFS command

The host names that are not registered in DNS will be ignored because there is no other way to obtain the host name.

With the DHCP option set, the IP address in HCBs will be updated on receiving first RPC packet from given host.

At NFSS startup

With the new attribute NODHCP, the z/OS NFS server will restore all mount points from MHDB, as in prior releases.

With the new attribute DHCP, the NFS server will ignore MHDB records with empty host names.

At NFSS shutdown

When working with the new attribute NODHCP, the NFS server will update host name information in MHDB to help the possible startup with the DHCP attribute.

7.4.5 NFS Server locking interoperability

This section provides an overview of the z/OS NFS locking and access control functions provided by the Network Lock Manager (z/OS NFS NLM) and the z/OS NFS Network Status Monitor (z/OS NFS NSM). It explains how they work together to provide file locking and access control capability over z/OS NFS. In addition, this section also explains the following features:

- Monitored lock
- Non-monitored locks versus locking files
- Locking records

In z/OS V1R7, NLM and NSM are integrated into the z/OS NFS server to facilitate the expanded locking and serialization functions. Separate procedures for starting and stopping NLM and NSM are replaced by the server site attributes nlm and nonlm, which specify their startup along with the NFS server. This integration also coordinates the locking function with stale file handle processing. When a file handle becomes stale, not only will the code clean up the file-related blocks as it does in prior releases, but it will also release any locks that remain held for that file.

Network Lock Manager (NLM)

The Lock Manager Protocol contains NLM (lockd) protocol and NSM (statd) protocol. In z/OS 1.6 and earlier, the NLM is in an address space separate from the NFS server address space.

With z/OS 1.7, the LM is moved into NFS server address space to improve the cooperative nature of the NFS server and NLM.

The stale file handle - locks are released (before, the locks would stay in place until the lock was cancelled).

The z/OS NFS server is using multi-tasking for performance improvement (was single-task).

Using Network Lock Manager in NFS V2 and V3

In NFS V2 and V3, the z/OS NFS NLM allows a client on the host to lock a record or a file on the z/OS NFS server. A client user can either choose to lock the entire file or a record section of a file. The two types of locks that the client host uses are monitored locks and non-monitored locks.

The z/OS NFS NLM only supports advisory locking. Advisory locking is when the operating system keeps track of which files have been locked by which process, but does not prevent a process from writing to a file that is locked by another process. This means that a process can ignore an advisory lock.

Monitored locks provide the client user with reliability. If the server host on which the monitored locks are established fails, the locks are reinstated when the server host recovers. The locks that are held by the client host are discarded by the z/OS NFS NLM on the server host if the client host fails before the locks are released.

> **Note:** Monitored locks will only work correctly if both the server host and the client host are running NSM.

Non-monitored locks are used on personal computer operating systems. Non-monitored locks provide the same functionality as the monitored locks with one exception. If the server host on which the locks are established fails and recovers, the locks will not be re-established. The client host is responsible for detecting a server host failure and re-establishing the locks. In addition, the client host informs the z/OS NFS NLM when it has rebooted so that the server host can discard all of the locks and file shares held for the client.

You can specify a time limit, or grace period, for clients to reclaim NFS V4 or NLM locks and share reservations when the z/OS NFS server restarts after a failure. To set this time limit, use the leasetime site attribute. During the reclaim grace period after a z/OS NFS server restart, the grace period may be extended after an open or lock reclaim event. If the leasetime site attribute is greater than 1200 seconds (20 minutes), the grace period will not be extended at a reclaim event. If the leasetime site attribute is less than 20 minutes, the grace period will be extended to one lease time after the reclaim event, up to but not exceeding 20 minutes after the z/OS NFS server completed its restart.

To diagnose possible problems with conflicting locks, z/OS operators can issue a `listlock` command that displays all client programs and users that hold a lock on a file. The output messages include client and user ID, the lock ranges held, and lock status. The listlock command can be used for MVS data sets, PDS or PDSE members, or z/OS UNIX files. For more information see "LISTLOCK operator command" on page 214.

You can use the `listlock` command to find locking information in cases where a lock is unavailable and the blocker is managed by another NFS server address space running on the z/OS system. To determine the identity of the blocker in this case, the listlock command should be issued on the system that owns the lock. You can also detect some situations in which a deadlock occurs.

To release all locks for a file, z/OS operators can issue a release command that releases locks for z/OS UNIX files and MVS data sets or members. For z/OS MVS data sets, the command also forces the z/OS NFS server to release the file, and if the file is active, to close and deallocate it.

Note: The NLM protocol does not provide any means for NFS server to notify the NFS client that the locks were released. Thus, the client may proceed under the false assumption that it still has the locks. Therefore, the release command should only be used in extreme circumstances.

In NFS V2 and V3, the z/OS NFS NSM is a service that provides applications with information about the status of the network host. Each z/OS NFS NSM keeps track of its own state and notifies any interested parties of a change in its state.

For correct operation of the z/OS NFS NSM, the client and the server hosts are required to monitor each other. When a lock request is issued by a process running on the client host, the NLM on the client host requests the NSM on the client host to monitor the server host. The client NLM then transmits the lock request to the z/OS NFS NLM on the server. On reception of the lock request the z/OS NFS NLM on the server host will request the z/OS NFS NSM on the server host to monitor the client host. In this way, each host is monitored by the NSM on the other host.

NLM and NSM functions moved into the NFS server address space

Before z/OS 1.7 the NLM (lockd) and NSM (statd) functions are implemented in an address space separate from the NFS server address space

With z/OS 1.7 the cooperative nature of the z/OS NFS server and NLM is improved by moving NLM and NSM into its address space. This provides coordination of stale file handle processing with the locking function — that is, when a file handle becomes stale, not only will the code clean up the file-related blocks, as it does currently, but it will also release any locks that remain held for that file.

Advantages

The z/OS NFS Server will now distribute the lock requests to the NFS historical and HFS sub tasks.

- Before z/OS NFS 1.7, the z/OS NLM server performs all the locking functions under a single task, which could cause performance problems.
- Locking will now exploit the multi-worker task structure of the PFS server, allowing multiple lock requests to be processed in parallel, assuming they do not conflict.
- Improved recording of client host names.
- Externalize the grace period with the new LEASETIME site attribute.
- Provide lock deadlock detection.
- A new operator command to display a list of locks for a specified file.
- An enhanced RELEASE operator command for HFS files to enable releasing all locks for an HFS file and freeing that vnode.

The z/OS NLM support provides the following:

- The support for TCP protocol.
- The duplicate XID checking.
- Dedicated tasks for socket sending and receiving services.
- Advanced tracing facility and memory management.

The new Lock DB is now a set of two VSAM data sets. Additional information about the pair of VSAM data sets where the locks are recorded can be found in “Lock data sets” on page 241.

New NFS Server site attributes

This section provides information about the new z/OS NFS server NLM attributes.

The nlm attribute specifies that the initialization of the z/OS NFS server should include starting the NLM and NSM daemons.

When the z/OS NFS Server initialization will start NLM and NSM daemons, the rpcinfo would show:

- program=100021 vers=1/3/4 (NLM)
- program=100024 vers=1 (NSM)

> **Restriction:** z/OS NFS 1.7 does not support NFS V4 Byte Range locking. It does support *advisory* locking and share reservation only. Byte range locking is supported for NFS V2 and V3.

The nonlm attribute specifies that the initialization of the z/OS NFS server should not include starting the NLM and NSM daemons. The system will run without lockd and statd.

Specifying nonlm does not affect the availability of byte range locking and share reservation support for NFS Version 4 protocol access. If nonlm is specified, the NLM may not be started after NFS has initialized. If NLM is desired, you must stop and restart NFS after specifying the nlm site attribute. The only way to stop NLM is to shut down the NFS server. It is no longer necessary to define the NLM and NSM startup procedures to a z/OS UNIX segment as UID(0) to RACF because the NLM and NSM startup procedures are no longer supported.

The lock data sets must always be allocated, even if nonlm is specified in the site attributes. The old startup procedures for NLM and NSM are not shipped in z/OS V1R7. These procedures are obsolete, and old copies from previous releases should not be used on V1R7 or later releases.

LEASETIME attribute

The leasetime attribute specifies the length of time (the lease interval) in seconds that the z/OS NFS server allows NFS clients to:

- Reclaim locks and share reservations following an NFS server restart. During this grace period, clients can reclaim locks on behalf of their users.
- Remain active without communicating with the NFS server. If an NFS V4 client does not communicate with the z/OS NFS server for the length of the lease interval, its client ID will expire. The value of n can range from 5 to 3600. The specified value must be smaller than the value of the logout attribute, if logout is not set to zero. The default value is 120.
- The grace period (the amount of time during which a client can reclaim locks following a crash recovery of a z/OS NFS Server) can now be externally specified via the new leasetime site attribute.
- When a z/OS NFS server fails and restarts, it notifies each of the clients that held NLM locks that it has restarted.
- After each of the clients has been notified (or at least one attempt has been made to each), a grace period begins during which clients can reclaim locks on behalf of their users. The duration of this grace period will be defined by the lasting value.
- Similarly, the NFS V4 protocol defines a grace period for clients to reclaim locks following a server crash. The duration of this grace period will also be defined by the leasetime value.
- The leasetime will also be used when the z/OS NFS server will look for clients that have not renewed lock leases.
- If the server finds a client whose lock lease has expired, all NFS V4 locks and share reservations will be released and V4 opens will be closed.
- This processing will occur every n seconds as specified by LEASETIME(n).

denyrw attribute

With the denyrw attribute, the server honors deny requests for file share reservations (the Windows Share_Deny value) from the NFS client. The deny requests may be specified on an NFS V4 Open operation or an NLM_share RPC.

nodenyrw attribute

With the nodenyrw attribute, the server ignores deny requests from NFS clients (the Windows Share_Deny value) and treats the requests as though deny_none were specified.

Lock data sets

This section provides details on the enhanced mechanism to record the names of client hosts accessing the server and gives an overview of the NLM integration and its purpose, properties, and state. Upon NFS server startup, LDB provides a means to indicate what NLM hosts and NFS V4 clients can reclaim state held in a previous instance.

The allocation attributes are the same as for the Mount Handle DB(MHDB) of the NFS Server. Ensure that RECORDSIZE is changed to RECSZ(1700 2000) and the keysize is different.

The sample JCL for the lock data sets is shown in Figure 7-32.

```
DEFINE CLUSTER (NAME(lock_data_set_name) -
      VOL(vsam_volume_name) -
      CYL(1 1) -
      INDEXED -
      REUSE -
      KEYS(8 0) -
      SHAREOPTIONS(3 3) -
      RECSZ(1700 2000))
```

Figure 7-32 Lock data set JCL

NFS V4 state

The NFS V4 protocol introduces state information that allows clients and servers to keep track of certain resources.

Note: The NFS V4 state is part of the NFS protocol internals and does not affect how applications are written.

NFS Version 4 uses a value of clientid or stateid to represent the current state (instance) of client-held resources such as locks, opens, and host restarts. The client and server pass this state information between them on certain operations, allowing both to agree on the current instance of resources held by the client.

NFS Version 4 includes new states for the following:

- Client/server restart instance
- Open share/deny instance
- Byte range locks instance
- Client delegation instance.

The NFS V4 state that is passed between the client and the server represents a single instance in a dynamically changing environment. It is incremented when a state is changed within a group of held resources (restart, open, or lock). Once state is established on the server, the client returns what it believes is the current state. The server compares the client state to the server state to detect stale and out-of-order requests.

The client uses the setclientid operation to notify the server of its intention to use a particular client identifier for subsequent requests that entail creating lock, share reservation, and delegation state on the server. Upon successful completion the server returns a shorthand client ID that, if confirmed via a separate step, will be used in subsequent file locking and file open requests. Confirmation of the client ID must be done using the setclientid_confirm operation to return the client ID and setclientid_confirm values, as verifiers, to the server.

NFS V2 and V3 used the Network Status Monitor (NSM) protocol to determine whether resources such as file open share or byte range locks were still in use by a remote client. NFS V4 no longer uses NSM to communicate a client or server restart. NFS V4 instead uses a current state on both the client and server, where the state is established and passed in subsequent NFS V4 operations.

In NFS V2 and V3, a client or server issues an NSM sm_notify RPC procedure to notify the remote host of a restart. Server resources such as an exclusive byte range lock on a file might remain held until explicitly released by the client. If a client that holds a server resource is removed from the network for a long period without the server being notified, the server resource would be unavailable to other clients until timed out by the server.

NFS V4 provides a protocol for the client to establish or reestablish state, and associates ownership of subsequent server stateful operations to previously established states. To resolve the absent client problem mentioned above, the NFS V4 client must routinely refresh the state within the server-specified lease time. Upon lease time-out, the server may release resources for the client and make them available to other applications.

A client obtains the server-specified lease time-out attribute by issuing a getattr operation. The getattr is not a stateful operation, thus it does not require prior state to be established. A getattr operation may precede a setclientid or setclientid_confirm operation.

NFS lock database messages

This section illustrates the new NLM messages of the z/OS NFS Server.

- GFSA350E

  ```
  GFSA350E Locking Database ddname cannot be opened, VSAM rc=d_digits1
  rsnc=d_digits2 LastOp=d_digits3
  ```

  - Explanation: The NFS server tried to open a locking data set but could not do so. This message identifies the data set, the return and reason codes, and the code of the last operation. In the message text:

ddname	The ddname of the failing data set
d_digits1	The failing information in fopen for the VSAM KSDS data set — the decimal return code
d_digits2	The decimal error code or reason code
d_digits3	The code for the last operation

  - System action: If this is the first of the two lock data sets to fail, NFS continues using the other lock data set. If this is the second of the two lock data sets to fail, lock information will not be recorded, and reclaim permissions on the next NFS restart may be inaccurate.
  - Operator response: Notify the system programmer.
  - System programmer response: The lock data set has probably been corrupted. Plan to delete the data set and reallocate it. You may also wish to verify that the ddname in the NFS startup job refers to the same data set as was created for it.
  - Source: z/OS NFS Server.

- GFSA351E

  GFSA351E Locking Database *ddname* is not proper VSAM format: VSAM=*d_digits1* type=*d_digits2* (expect *d_digits3*) KeyOff=*d_digits4* (expect 0) KeySize=*d_digits5* (expect *d_digits6*)

  - Explanation: The Network File System server detected that the lock data set does not have the proper characteristics for a VSAM key sequenced data set (KSDS). This message identifies the incorrect characteristics and their expected values. In the message text:

ddname	The ddname of the failing data set.
d_digits1	Indicates VSAM data set or not. Value of 1 indicates VSAM. Value of 0 indicates not VSAM.
d_digits2	The detected type of the data set.
d_digits3	The expected type of the data set (KSDS).

d_digits4 The detected offset of the VSAM key within the record, which was expected to be 0.

d_digits5 The detected size of the VSAM key within the record.

d_digits6 The expected size of the VSAM key within the record.

- System action: This lock data set will not be used. NFS will continue using the other locking data set. If both lock data sets have failed, lock information will not be recorded, and reclaim permissions on the next NFS restart may be inaccurate.
- Operator response: Notify the system programmer.
- System programmer response: The lock data set has been created incorrectly. Reallocate the lock data set with the correct parameters, as described in “Lock data sets” on page 241.
- Source: z/OS NFS Server.

- GFSA352E

```
GFSA352E Locking database ddname failed op opname with rc=d_digits1
```

- Explanation: An attempt to read, write, or modify the lock data set has failed. This messages identify the data set, the operation, and the failure code. In the message text:

  ddname The ddname of the failing data set

  opname The name of the operation that failed

  d_digits1 The failure code from the operation

- System action: This lock data set will not be used. NFS continues, using the other lock data set. If both lock data sets have failed, lock information will not be recorded, and reclaim permissions on the next NFS restart may be inaccurate.
- Operator response: Notify the system programmer.
- System programmer response: The lock data set has probably been corrupted. Plan to delete the data set and reallocate it.
- Source: z/OS NFS Server.

Lock DB during the startup of the z/OS NFS server

This section provides information about the use of the LDB during the start of the z/OS NFS server.

> **Restriction:** The new lock DB (LDB) is required to start up the NFS 1.7 Server. An error message will be issued when the LDB is not allocated and defined in the started task (STC) of the NFS Server.

Figure 7-33 shows the SYSLOG of the NFSS STC without a LDB defined as the DD statement.

```
S NFS
IAT6100 ( DEMSEL ) JOB NFS      (JOB03869), PRTY=15, ID=STC
ICH70001I STC      LAST ACCESS AT 12:26:46 ON MONDAY, MAY 1, 2006
IEF695I START NFS      WITH JOBNAME NFS      IS ASSIGNED TO USER STC
   , GROUP SYS1
IEF403I NFS - STARTED - TIME=12.26.46 - ASID=0088 - SC65
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
IEC161I 072-053,NFS,NFS,FHDBASE,,,OS390NFS.SC65MVS.FHDBASE1, 145
IEC161I OS390NFS.SC65MVS.FHDBASE1.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,FHDBASE2,,,OS390NFS.SC65MVS.FHDBASE2, 146
IEC161I OS390NFS.SC65MVS.FHDBASE2.DATA,UCAT.VSBOX01
GFSA351E (NFS     ) LOCK DATA SET DD:LDBASE (DD:LDBASE) IS NOT PROPER
VSAM FORMAT: VSAM=0 TYPE=0 (EXPECT 2) KEYOFF=0 (EXPECT 0) KEYSIZE=0
 (EXPECT 8) RECSIZE=1024 (EXPECT 1416)
GFSA351E (NFS     ) LOCK DATA SET DD:LDBASE2 (DD:LDBASE2) IS NOT
 PROPER VSAM FORMAT: VSAM=0 TYPE=0 (EXPECT 2) KEYOFF=0 (EXPECT 0) KEYSIZE=0
 (EXPECT 8) RECSIZE=1024 (EXPECT 1416)
 GFSA330I (NFS     ) SERVER SHUTDOWN COMPLETE.
```

Figure 7-33 Output of attempt to start up NFSS without a lock DB

> **Note:** Running a test, the message GFSA351E was issued when the LDB is not defined. The message GFA737I is just a warning and can be ignored when Kerberos is not used.
>
> The messages IEC161I 072-053 are also warnings that inform that the MHDB VSAM data sets are empty and do not contain a mountpoint yet to reestablish. See APAR OA12684 and OA14129.

Figure 7-34 shows the flow of the NFS Server startup when the LDB is allocated and defined in the related started task.

```
S NFS
IEF695I START NFS      WITH JOBNAME NFS      IS ASSIGNED TO USER STC
   , GROUP SYS1
IEF403I NFS - STARTED - TIME=12.15.17 - ASID=008C - SC65
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
IEC161I 072-053,NFS,NFS,FHDBASE,,,OS390NFS.SC65MVS.FHDBASE1, 837
IEC161I OS390NFS.SC65MVS.FHDBASE1.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,FHDBASE2,,,OS390NFS.SC65MVS.FHDBASE2, 838
IEC161I OS390NFS.SC65MVS.FHDBASE2.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,LDBASE,,,OS390NFS.SC65MVS.LDBASE, 839
IEC161I OS390NFS.SC65MVS.LDBASE.DATA,UCAT.VSBOX01
IEC161I 072-053,NFS,NFS,LDBASE2,,,OS390NFS.SC65MVS.LDBASE2, 840
IEC161I OS390NFS.SC65MVS.LDBASE2.DATA,UCAT.VSBOX01
GFSA348I (NFS     ) Z/OS NETWORK FILE SYSTEM SERVER (HDZ11VS ,
OA15050) STARTED.
```

Figure 7-34 Startup with the LDB allocated and defined

7.4.6 z/OS NFS Server Version 4 protocol support

All the NFS V4 enhancements in NFS 1.7 are for the z/OS NFS server only. The z/OS NFS client does not support NFS V4.

Protocol support - V1R7 restrictions

The following functions and operations are not supported in z/OS NFS V1R7:

- The z/OS NFS Client does not support NFS V4.
- The unsupported operations are:
  - Byte range locking (Lock,LockT,LockU)
  - Delegation (DelegPurge, DelegReturn)
  - Named attributes

Note: See the NFS V4 operations in “NFS V4 operations” on page 265.

The NFS V4 protocol is described in RFC 3530:

`http://www.faqs.org/rfcs/rfc3530.html`

NFS Version 4 protocol

The NFS V4 is a new industry protocol and it is defined to provide:

- Improved security (RPCSEC_GSS)
- Reduce message traffic (compound RPC)
- Integrated locking functions
- Better firewall compatibility

NFS V4 protocol items

This section summarizes the NFS V4 highlights.

- It provides stronger network authentication and security because:
  - The transmitted network data could be encrypted.
  - It is resistant to spoofing and masquerading.
- Better firewall access is given because only port=2049 is used.
  - No MOUNT program and MOUNT/UNMOUNT RPC.
- It is broken down into many *operations* to allow building of complex compound procedures with many operations.
- A hierarchical view of server file system (start from root), having a pseudo file system and cross-mount to local file systems.
- The stateful OPEN/READ/WRITE/LOCK/LOCKU/CLOSE versus Stateless NFS V3 READ/WRITE and NLM/NSM because of:
  - The lease concept
  - The states maintained by both the z/OS NFS Client and the z/OS NFS Server
  - Possible integrated file locking (share reservation) and POSIX byte range locking
- The availability of a volatile FileHandle to implicitly recover by the NFS Client upon NFS4ERR_FHEXPIRED (server restart):
  - Hopefully no more *Stale FileHandle* upon server restart.

Security overview

This section details the existing security of the NFS Server and the enhancements to support the NFS Version 4.

- The current data access for z/OS NFS are classified into:
  - Conventional z/OS data
  - UNIX data
  - Public data
  - Data accessed using the public file handle
- The current levels of security were classified into the following categories:
  - None/unrestricted data access
  - Exports list checking
  - Security Authorization Facility (SAF) checking
  - Both SAF and exports list checking
  - SAF with checklist processing

Summary of existing security support

For support:

- The security of z/OS NFS before V1.7 provides good system access security with minimal administrative overhead.
- The pre-NFS V1.7 security shows some lack of data protection across network transmissions, as the data cannot be encrypted and not be locked.
- z/OS NFS V4 continues to support existing security flavors.

NFS V4 security (Kerberos)

This section discusses the security enhancements of Kerberos and gives an overview of the Kerberos security to set up the system from the Z/OS NFS Server point of view.

Additional security flavor: RPCSEC_GSS only with V4 protocol

RPCSEC_GSS provides for out-of-sequence and replay detection of RPC messages while the NFS server and NFS client must negotiate the specific security that will be used. It uses the Kerberos V5 Security Mechanism.

The following RFCs provide additional information about Kerberos V5:

- RFC2624 NFS V 4 design considerations

  `http://www.faqs.org/rfcs/rfc2624.html`
- RFC4121 The Kerberos V5 GSS API

  `http://www.faqs.org/rfcs/rfc4121.html`
- RFC4120 The Kerberos Network Authentication Service V5

  `http://www.faqs.org/rfcs/rfc4120.html`
- RFC2203 The RPCSEC_GSS Authentication flavor

  `http://www.faqs.org/rfcs/rfc2203.html`

RPCSEC_GSS provides for out-of-sequence and replay detection of RPC messages while the NFS server and NFS client must negotiate the specific security that will be used.

> **Restriction:** When the z/OS NFS server is configured talk to multiple TCP/IP stacks as part of a common inet (CINET) configuration then the RPCSEC_GSS workload is only allowed for the TCP/IP stack defined as the local host. However, the AUTH_sys workloads are allowed for all TCPIP stacks for all NFS versions.

RCSEC_GSS has the following service types and corresponding Kerberos pseudo flavors, depending on the service type:

- None/krb5

  Provides Kerberos V5 based integrity on the RPC credentials (but not data)
- Integrity/krb5i

  Provides Kerberos V5 based integrity on both RPC credentials and data
- Privacy/krb5p
  - Provides Kerberos V5 based integrity on RPC credentials
  - Provides Kerberos V5 based encryption on RPC data (and not credential, because encryption causes a change in size, and the RPC designers decided not to accommodate this behavior in the credential size)

> **Note:** The new NFS V4 Security provided by Kerberos V5 was implemented by a SPE via the new function APAR OA11875, as discussed in "OA11875" on page 350.

Protecting the file system on z/OS with the NFS V4 protocol

The NFS Version 4 protocol improves on the NFS V2 and V3 protocols with stronger authentication and network transmission protection for NFS data. The NFS V4 protections include encryption algorithms for data privacy, multiple protections per file, and the means to negotiate security as NFS clients explore the file system. The NFS V4 protocol provides these protections through the required RPCSEC_GSS security authentication flavor and the SECINFO operation.

The z/OS NFS server enforces these protections, for client RPC requests that use RPCSEC_GSS, through the site attributes mvssec, hfssec, and pubsec. These attributes provide the Kerberos V5 Security Mechanism (RFC1964) subset of the V4 protocol, at the file-system level. The z/OS NFS server also continues to support NFS V2 and V3, as well as V4 protocol requests with the protections provided by the security site attribute.

The export data set also contains a new security keyword, sec, that specifies the security authentication level that clients must have to access individual files and data sets on the z/OS NFS server. That is, specific export entries can be further constrained with different authentication flavors by using this security keyword. For example, an important export entry can be protected with krb5p level set by the security keyword, while other exported entries in the file system can be accessed by all authentication levels that are specified by the mvssec, hfssec, and pubsec site attributes. The authentication flavors specified by the sec keyword in the export entries should be a subset of the authentication flavors of site attributes mvssec, hfssec, or pubsec. In other words, an authentication level is effective only if it is specified by the site attribute logically AND'ed with the security keyword. For this reason, if the sec keyword is not specified, meaning all flavors are on, the authentication level is defaulted to the site attribute mvssec, hfssec, or pubsec.

The mvssec, hfssec, and pubsec attributes let you specify the default network security flavors, and order, that can be used by requests accessing MVS, z/OS UNIX, and public file systems, respectively. These site attributes apply to all NFS versions. If you protect a data set with any one of these transmission attributes, NFS V2/V3 requests will get responses of AUTH TOOWEAK unless 'sys' is listed as a valid authentication flavor. NFS V4 requests that do not comply with these protections will get WRONG SEC.

Note: Since z/OS NFS only supports RPCSEC_GSS security for NFS Version 4, if one of the site attributes is set to require RPCSEC_GSS, then clients using NFS V2 and V3, which only support AUTH_SYS security, cannot access those file systems. On the other hand, if the attribute is specified at a mount point, then only that mount point is affected.

Required changes to use the NFS V4 RPCSEC_GSS security flavors

To use the NFS V4 RPCSEC_GSS security flavors, the following changes to the security infrastructure are required:

- Kerberos services must be activated on the z/OS system where the NFS server is running. This activation includes the definition of realms, inter-realm relationships, and the Kerberos Principal for the z/OS NFS server. For details on these definitions, see the z/OS Integrated Security Services Network Authentication Service Administration.
- The Kerberos principals on NFS clients must be defined to RACF and assigned a RACF identity. In addition, for Linux clients a principal NFS/hostname.domain should be defined to RACF. This is because Linux Clients use this principal for mounts and some state operations. For further details on defining principals, see the z/OS Security Server RACF Security Administrator's Guide.
- The z/OS NFS server must have READ access to the IRR.RUSERMAP resource in the FACILITY class.

The Kerberos configuration steps for NFS are reviewed in "Additional information for Kerberos configuration steps for NFS" on page 258.

Control flow in NFS

This section describes the server startup, security context management, and data exchange process.

Figure 7-35 shows the flow during the start of the NFS Server.

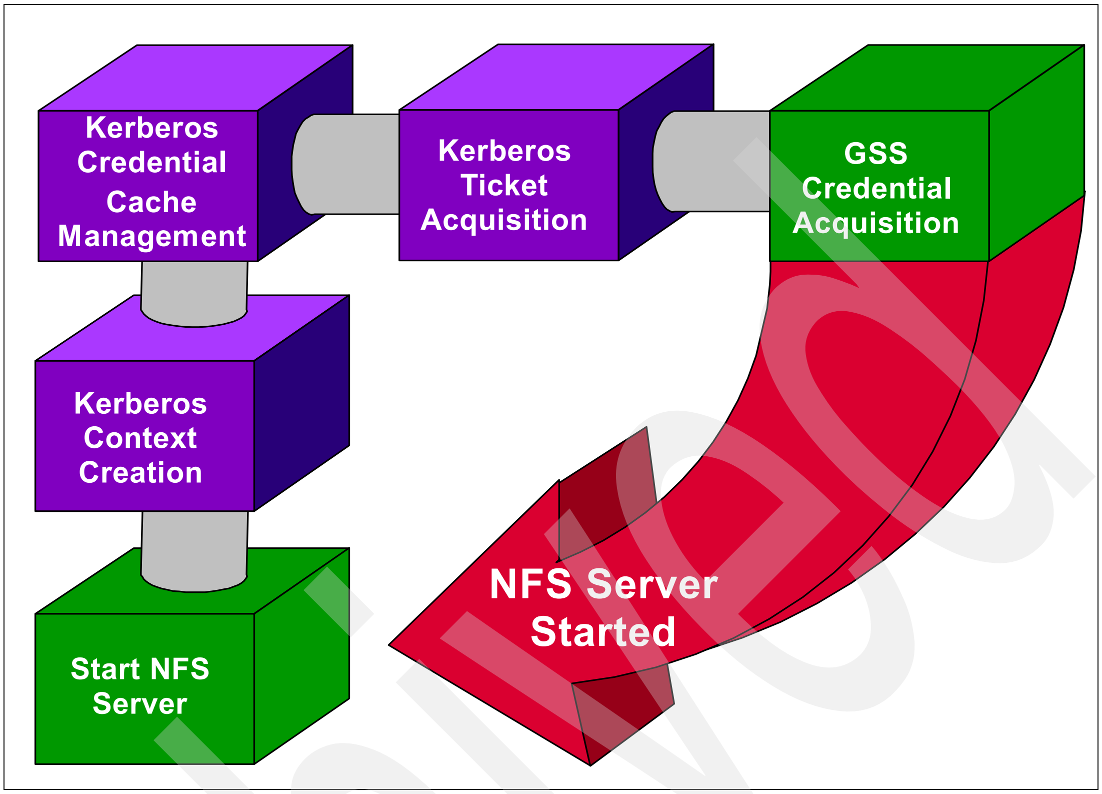


Figure 7-35 Kerberos during z/OS NFS Server startup phase

Kerberos context creation

During server startup a Kerberos context is created. Kerberos Context is a data structure that contains configuration data obtained from the Kerberos configuration file and is required for using Kerberos services.

This Kerberos context lasts through the lifetime of the NFS server.

Kerberos credential cache management

For this:

- The z/OS NFS server then initializes the credential cache.
- The principal name for the credential cache is set to the NFS server's principal.
- This principal name will also be the default name for tickets that the NFS server acquires and are placed in this credential cache.
- The z/OS NFS server then tells the GSS API to set this as the default credential cache for use by the Kerberos security mechanism.

Kerberos ticket acquisition

For this:

- The z/OS NFS server uses the keytable to get an initial ticket.
- The z/OS NFS server will only allow the local host system on which it is running to generate requests by using this ticket.
- The credential cache specified for this ticket acquisition will be the one generated by the NFS server.

Supported encryption types

The z/OS NFS server supports the following encryption types:

- MD5 checksum with DES encryption
- 32-bit CRC checksum with DES encryption
- MD4 checksum with DES encryption
- SHA1 checksum with DES encryption and key derivation
- SHA1 checksum with DES3 encryption and key derivation

GSS credential acquisition

The NFS server acquires two GSS credentials, one for accepting requests (accept type) and the other for initiating requests (initiate type).

The accept type credentials of the NFS server never expire.

The initiate type credentials may expire and will be periodically renewed by the NFS server.

The credentials will be acquired to be valid for the maximum lifetime. This depends on the lifetime of the underlying TGT and is not controlled or governed by the NFS server in anyway.

GSS credentials enable the communicating applications to establish security contexts with each other. They can contain multiple cryptographic keys that are required for authentication and message encryption to be performed with different algorithms. The z/OS NFS server uses Kerberos V5 as its security mechanism for acquiring the GSS credentials. The z/OS NFS server initially acquires these credentials during server startup. The z/OS NFS server uses the credentials for accepting the security context requests from NFS clients. The z/OS NFS server will attempt to acquire the GSS credentials for the maximum credential lifetime, but the actual lifetime of credentials will depend on the lifetime of the underlying ticket granting ticket of the Kerberos security server, and is not controlled or governed by the z/OS NFS server. On expiration of the server's GSS credentials, client requests will receive the RPCSEC_GSS documented errors and the client is expected to refresh the contexts and retry the requests.

Security negotiation

The NFS Version 4 protocol facilitates the use of multiple RPC authentication flavors. The z/OS NFS server supports the Kerberos V5 security mechanism and all the pseudo flavors of the Kerberos security mechanism using the cryptographic algorithms referred to in NFS V4 (RFC3530). To facilitate the selection of a particular pseudo flavor, the z/OS NFS server supports security negotiation using the NFS V4 protocol's SECINFO operation.

> **Note:** IBM recommends that security negotiation be done by the NFS clients using the SECINFO operation with an RPC authentication flavor of RPCSEC_GSS with the krb5i or krb5p pseudo security flavors.

When responding to SECINFO for security negotiation (when multiple security flavors are present for a file system or file), the z/OS NFS server uses an order of preference that has RPCSEC_GSS as the most favored flavor followed by AUTH_SYS. For the authentication flavor of RPCSEC_GSS, the z/OS NFS server has krb5, krb5i, and krb5p as its listed pseudo flavors in descending order of preference. NFS clients are, however, free to choose from any one of the z/OS NFS server-supported security flavors for their NFS V4 requests.

Security context acceptance

A security context is a data structure that contains information about the cryptographic state of a program on the client communicating with the server, and is required for RPC message security services. NFS clients create security contexts with the z/OS NFS server as part of the RPCSEC_GSS protocol of data flow. The z/OS NFS server accepts security context requests subject to the following restrictions and recommendations:

- The z/OS NFS server does not support channel bindings.
- The z/OS NFS server never initiates any requests as an agent of NFS clients and therefore recommends that clients do not use credential delegation services while creating security contexts.
- The z/OS NFS server does not support the out-of-sequence detection services of GSS API. It expects NFS clients to have the seq_req_flag parameter turned off on their calls to GSS API gss_init_sec_context.
- The z/OS NFS server recommends that the clients do not use the message replay services of the GSS API. It expects NFS clients to have the replay_det_req_flag turned off on their calls to the GSS API gss_init_sec_context. Note that the z/OS NFS server's implementation of the RPCSEC protocol provides for the protection against replay attacks.
- The z/OS NFS server does not allow clients to authenticate as anonymous principals.
- The z/OS NFS server recommends that NFS clients use mutual authentication services during context creation. The z/OS NFS server will still honor context creation requests from NFS clients that are unable to, or choose not to, use mutual authentication services in the GSS-API. However, clients that would require RPC callbacks from the z/OS NFS server have to support accepting security contexts with mutual authentication, because the z/OS NFS server always initiates security contexts with mutual authentication services.

Restriction: NFS V2 and NFS V3 support the security flavor AUTH_SYS(sys) only.

RPCSEC data exchange flow

Figure 7-36 shows a typical RPCSEC data exchange flow.

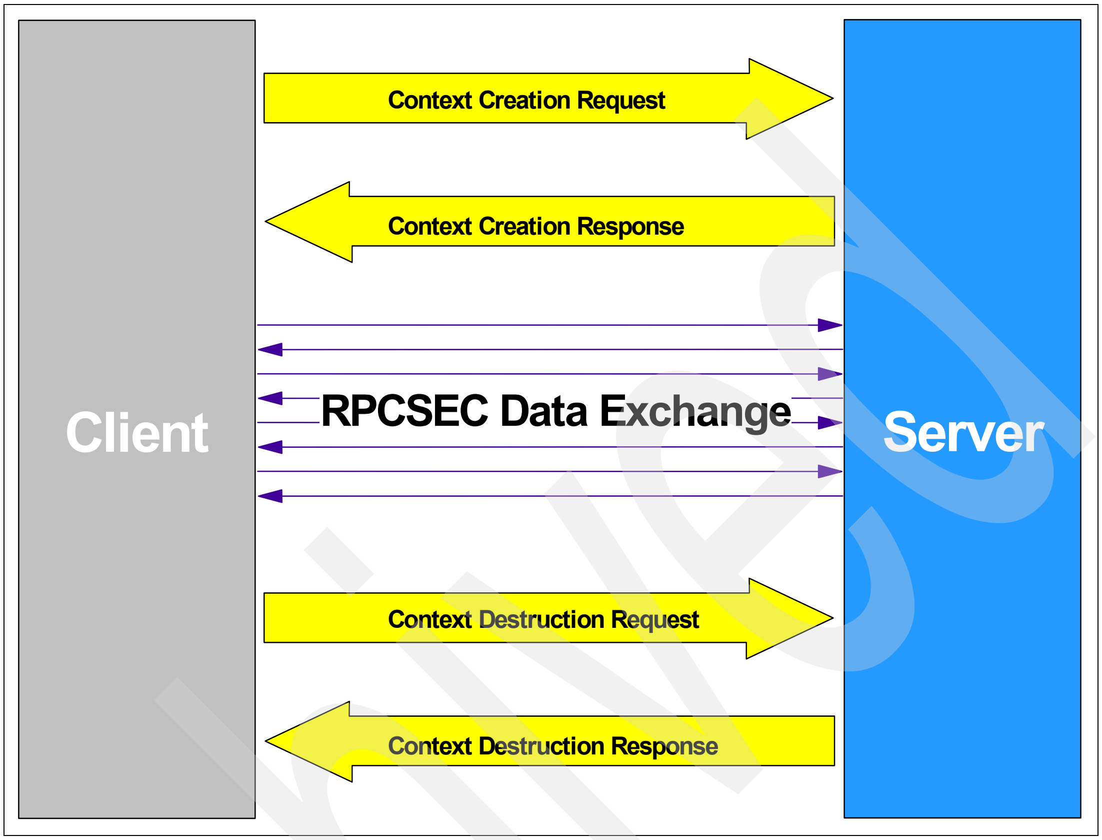


Figure 7-36 RPCSEC context and data exchange

The PRCSEC data flow paradigm has three phases:

- Context creation (establishment)

  Single RPC exchange (client trusts server, and would not require the server to authenticate to the client. The client only authenticates to the server.)

  Multiple RPC exchange or mutual authentication (client tells the server who it is and requires that the server also authenticates to the client).

- Data exchange

  RPC messages are exchanged between the client and server using the context. Real NFS data is exchanged here.

- Context destruction

  After the client is done with its RPCSEC session, the context is destroyed.

Note: In mutual authentication, you will see two exchanges between the client and the server to acknowledge the creation of the environment and receipt of credentials.

Clients create a security context that contains information about the cryptographic state of a principal on the client talking to the server.

NFS data is exchanged in the data exchange phase of RPCSEC_GSS.

Interlocking of RPCSEC_GSS - GSS API - KERBEROS with NFS

This section illustrates the interrelationship of RPCSEC_GSS - GSS API - KERBEROS - NFS and the protocol layering.

- NFS employs the RPCSEC_GSS authentication flavor of the RPC protocol to communicate between the NFS Client and the NFS Server.
- RPCSEC_GSS employs the GSS API to provide security services between the client and the server.
- The GSS API uses Kerberos security mechanism to provide integrity and confidentiality services.

Figure 7-37 show the Kerberos protocol layering.

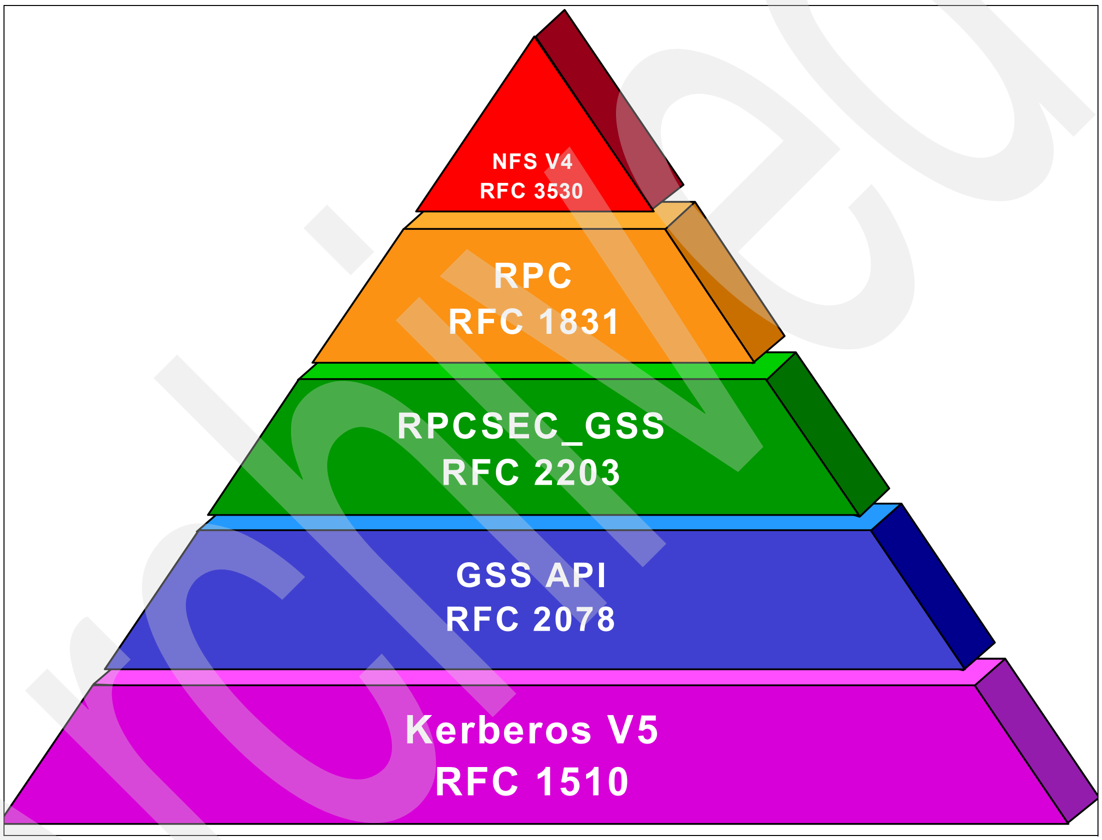


Figure 7-37 Kerberos protocol layering

Figure 7-38 shows an overview of an RPCSEC_SEC security environment.

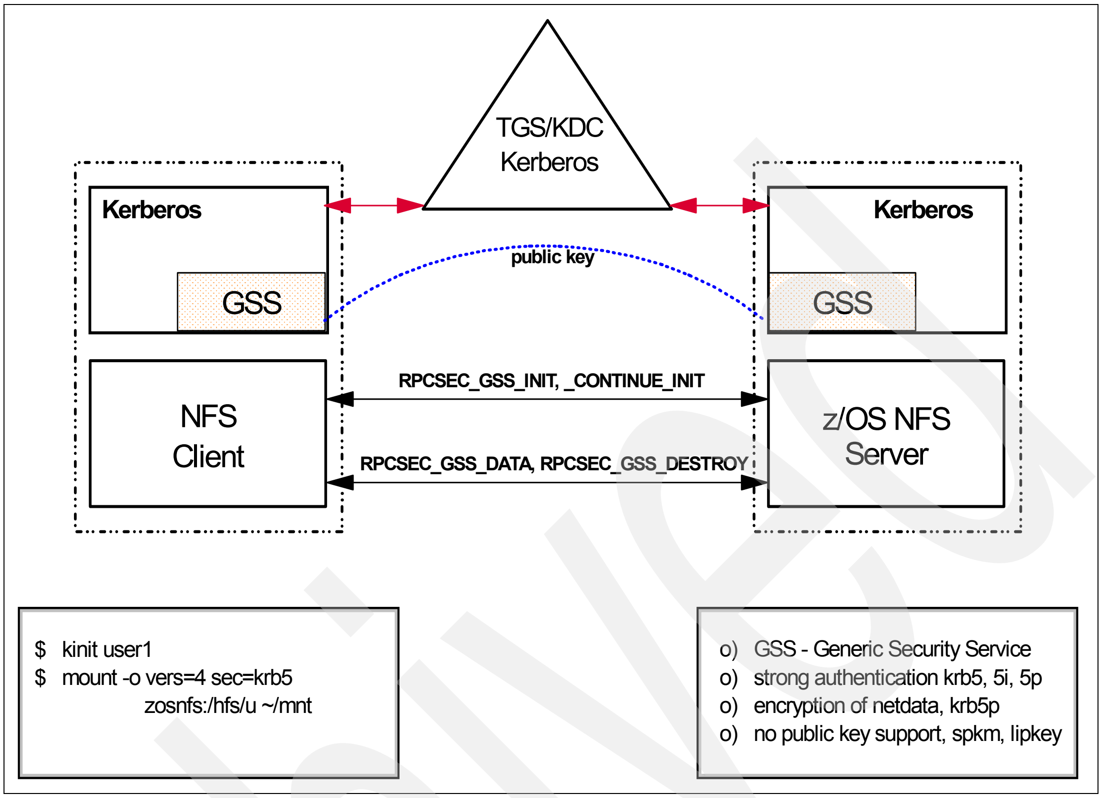


Figure 7-38 RPCSEC_GSS environment

Public key and Kerberos are two different security mechanisms of RPCSEC_GSS. Note that z/OS NFS 1.7 does not support public key encryption yet.

Configuration of a secure z/OS NFS Server

In order for the z/OS NFS Version 4 server to be able to provide RPCSEC_GSS security authentication flavors such as krb5, krb5i, and krb5p, the z/OS NFS server must be configured to communicate with the Kerberos facilities.

The following steps assume that the Resource Access Control Facility (RACF) is available in the system. If you have a different but equivalent external security manager, refer to the product documentation for instructions.

1. The Kerberos key distribution center (KDC) must be running, and must contain the z/OS NFS server's principal before the z/OS NFS server starts. Otherwise, the following message will appear when the z/OS NFS server starts:

   ```
   GFSA737I   NETWORK FILE SYSTEM SERVER COULD NOT GET KERBEROS TICKET IN ROUTINE
   procName(),KERBEROS RETURN CODE(krbRc)
   ```

   If only AUTH_SYS authentication is specified as a site attribute (sys) then the message GFSA737I is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any Kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then message GFSA737I is issued to the console as well as the NFS server log.

   In this case the z/OS NFS server will be functional and only allow authsys, but not the other Kerberos authentication levels such as krb5, krb5i, and krb5p.

The KDC can be running on z/OS, either on the same host as the z/OS NFS server itself or remotely from the z/OS NFS server. It can also be a KDC running on other platforms, for example, a SUN Solaris system or any other platform.

For a description of how to set up a z/OS KDC, refer to *z/OS Integrated Security Services Network Authentication Service Administration,* SC24-5926, for more advanced details. For setting up other platforms KDCs, refer to the specific platform's documentation.

2. Define local realm and default policy. For example, issue the following Ishell command:

```
RDEFINE REALM KERBDFLT KERB(KERBNAME(REALM_NAME) PASSWORD(password)
```

3. Define IRR.RUSERMAP and grant READ authority to all system users, issuing Ishell commands:

```
RDEFINE FACILITY IRR.RUSERMAP UACC(READ)
SETROPTS RACLIST (FACILITY) REFRESH
PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(MVSNFS) ACCESS(READ) SETROPTS CLASSACT
(FACILITY)
```

4. Create RACF user IDs with Kerberos segments on the z/OS NFS server. For example, if a user is using a Kerberos principal user1 on the NFS client, the Kerberos segment user1 must be defined to a z/OS RACF user on the z/OS NFS server.

To add a RACF ID, issue the Ishell command:

```
AU (RACFID1) OWNER (IBMUSER) OMVS(UID(101))
```

To add a Kerberos segment to the above user definition, issue the Ishell command:

```
ALTUSER RACFID1 PASSWORD(password) NOEXPIRED KERB (KERBNAME(user1))
```

If the Kerberos segment is not defined correctly to RACF, the following error message appears on the server when a NFS client tries to mount to the z/OS NFS server with Kerberos:

```
GFSA728E SAF APPLICATION USER MAPPING FAILED WITH SAF RETURN CODE 8, RACF
RETURN CODE 8, AND RACF REASON CODE 16.
```

The ALTUSER command converts the password to upper case if the MIXEDCASE SETROPTS option is not set. If MIXEDCASE is not set, you must ensure that the uppercase value is used when you request an initial ticket. The principal name is not converted to upper case, and the realm name is not included. You must change the password for the user in order to create the Kerberos secret key.

5. Add the PATH=/usr/lpp/skrb/bin:$PATH path in the z/OS UNIX /.profile and export the PATH.

6. Create or edit the krb5.conf file to contain the correct realm information for the z/OS NFS server. By default, the location of the krb5.conf file is in directory /etc/skrb. The sample /etc/skrb/krb5.conf file to be put on the z/OS NFS server host is:

[libdefaults]

```
default_realm = KRB390.IBM.COM
kdc_default_options = 0x40000010
use_dns_lookup = 0
default_tkt_enctypes = des-cbc-crc
default_tgs_enctypes = des-cbc-crc
```

[realms]

```
KRB390.IBM.COM = {
kdc = dcesec4.krb390.ibm.com:88
kpasswd_server = dcesec4.krb390.ibm.com:464
admin_server = dcesec4.krb390.ibm.com:749
```

```
}
KRB2000.IBM.COM = { kdc = sstone1.krb2000.ibm.com:88 admin_server =
sstone1.krb2000.ibm.com:749
}
```

[domain_realm]

```
.krb390.ibm.com = KRB390.IBM.COM
.krb2000.ibm.com = KRB2000.IBM.COM
```

[capaths]

```
KRB390.IBM.COM = {
KRB2000.IBM.COM = .
}
```

7. Generate the keytab from the KDC and put it in /etc/skrb of the z/OS NFS server unless otherwise defined. Examples of generating keytabs can be found in *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926.
8. If there is any multi-realm setup in the environment, the z/OS NFS server must have the foreign principals mapped to RACF. For example, to map a foreign principal *princ* in realm 2 to RACF, issue the Ishell command:

   ```
   RDEFINE KERBLINK /.../REMOTE_REALM/princ APPLDATA('RACF_ID')
   ```

   To map the entire foreign realm (every principal in the trusted foreign realm) to a RACF user, issue the Ishell command:

   ```
   RDEFINE KERBLINK /.../REMOTE_REALM/ APPLDATA('RACF_ID')
   ```

   > **Note:** The /.../ and trailing slash (/) are required.

9. Start the z/OS NFS server. If setup is correct, the following message should be shown:

   ```
   GFSA730I NETWORK FILE SYSTEM SERVER KERBEROS INITIALIZATION SUCCESSFUL
   ```

Additional configuration considerations

Additional considerations are:

- The above are the minimal requirements to set up a secure z/OS NFS server in order for it to communicate with Kerberos facilities. For more advanced configurations see *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926.
- If z/OS NFS server is configured to use a KDC that resides on a remote host, the local KDC procedure (for example, skrbkdc) on the same host as the z/OS NFS server should not be started.
- If the security site attribute is set to SAF or SAFEXP, in order to communicate with the z/OS NFS server using Kerberos, the MVS ID used by the mvslogin tool should be the RACF ID of the Kerberos segment that is used in the *kinit* client command. For instance, to access a SAF z/OS NFS server using Kerberos, and RACF has an user ID RACFID1, which has a Kerberos segment USER1, a user will need to perform *kinit USER1* on a NFS client to acquire the principal, and also issue the command **`mvslogin RACFID1`** to get authorized by RACF, in order to be able to mount and access the SAF or SAFEXP-enabled z/OS NFS server. In another example for the Kerberos multi-realm environment, if an entire trusted remote realm is mapped to RACFID2 with Kerberos segment USER2, then all users in the remote realm will have to perform **`mvslogin RACFID2`** and **`kinit USER2`** in order to access SAF or SAFEXP-enabled z/OS NFS server.
- For Linux (Enterprise Linux 4) users, as of July 2005, Linux NFS Version 4 is still considered experimental. Although base NFS Version 4 function is considered complete,

RPCSEC_GSS is still under development. Linux Kerberos works differently from other UNIX platforms.

In order to perform a secure Kerberos mount, Linux requires a keytab generated by the Kerberos KDC to be put in the Linux machine's /etc. This behavior will change once kernel keyring support is completed.

Because Linux has the keytab, the system is able to perform secure NFS mount without having the credentials acquired by kinit command. This behavior will change once kernel keyring support is completed.

Root user (uid = 0) uses Linux machine credentials but not the regular user credentials obtained via kinit. Thus, the root user will be able to browse the NFS mount point without performing kinit. Regular users must kinit to access the mount points. This behavior will change once kernel keyring support is completed.

Kdestroy will not destroy the context in the Linux kernel. This behavior will change once kernel keyring support is completed.

For Linux, multiple mounts to the same server with different security flavors are not supported yet.

Extra configurations are needed for Linux remote realm setup since Linux sends NFS/host.domain instead of user principal during mount time. If the Linux's NFS principal is not defined to RACF, the z/OS NFS server will reject mount requests. A simple way to solve this problem is to map the entire remote realm to RACF. Another more secure way to work around this is to map individual Linux machines to a special realm in the [domain_realm] section in the /etc/skrb/krb5.conf, and map that realm to a special RACF on the z/OS NFS server, thus leaving all other machines in the remote realm intact.

- For Windows users, we recommend that you use the MIT Leash Kerberos Ticket Manager with Hummingbird Maestro NFS client, especially when using Windows XP. However, to use Hummingbird NFS Maestro Version 9 or earlier with Windows XP, a user might need to use Microsoft® SSPI instead of the MIT Leash Kerberos Ticket Manager. For more information about the Hummingbird NFS Maestro client and the configurations of the Kerberos on a Windows platform, consult the Hummingbird NFS Maestro User's Guide.

  If the z/OS NFS server security site attribute is set to SAF or SAFEXP, the only supported authentication level on the Windows platform is authsys. RPCSEC_GSS authentication levels such as krb5, krb5i, and krb5p are not supported because Hummingbird Maestro 9 and 10 require NIS/NIS+ or LDAP with Kerberos instead of PCNFSD, which serves as the mvslogin tool for the Windows platform. Thus, if the z/OS NFS server is SAF enabled and the network transmission protection is mandatory, the Windows platform is not recommended.

Additional information for Kerberos configuration steps for NFS

This section describes the steps and commands to set up the Kerberos for NFS in more detail.

1. Allow the NFS server to have read access to IRR.RUSERMAP using the following RACF commands:

```
RDEFINE FACILITY IRR.RUSERMAP UACC(NONE)
PERMIT IRR.RUSEMAP CLASS(FACILITY) ID(MVSNFS) ACCESS(READ)
SETROPTS CLASSACT(FACILITY)
```

2. Set up your Kerberos configuration file and the NFS server's keytab file by defining the local realms:

```
RDEFINE REALM KERBDFLT KERB(KERBNAME(<realm>)
PASSWORD<realmpass>))
```

3. For a multi realm, define inter-realm trust relationships by:

```
RDEFINE REALM realm1/krbtgt/realm2 KERB(PASSWORD(<TrustP1>))
RDEFINE REALM realm2/krbtgt/realm1KERB (PASSWORD(<TrustP2>))
```

4. Define the NFS server's Kerberos principal:

```
ALTUSER <MVSNFS> PASSWORD<userpass> NOEXPIRED KERB(KERBNAME(NFS/hostname))
```

5. Add your NFS client users:

```
AU(<USER1>) OWNER(<IBMUSER>) OMVS(UID(<n>))
```

6. Define local principals for your NFS Client users:

```
ALTUSER <USER1> PASSWORD(<ibm>) NOEXPIRED KERB(KERBNAME(<user1>))
```

7. Define foreign principals:

   – Map a single NFS Client principal to a RACF user:

   ```
   RDEFINE KERBLINK /.../foreign_realm/foreign_principal APPLDATA(<racf_user>)
   ```

   – Map all NFS Client principals for the realm to a RACF user:

   ```
   RDEFINE KERBLINK /.../foreign_realm/APPLDATA(<racf_user>)
   ```

Kerberos tests

This section illustrates an overview of the Kerberos interaction shown by a test sequence done on AIX® as NFS V4 Client and z/OS as the NFS server for HFS files.

Figure 7-39 shows a mount without Kerberos V5 ticket that fails.

```
-bash-2.05b$ klist
Unable to get cache name (ticket cache: /var/krb5/security/creds/krb5cc_504).
        Status 0x96c73ac3 - No credentials cache found.


-bash-2.05b$ mount -o vers=4,sec=krb5 sjvm5154:/hfs/u /home/keithc/mnt1
mount: 1831-008 giving up on:
sjvm5154:/hfs/u
vmount: The file access permissions do not allow the specified action.
```

Figure 7-39 NFS V4 mount fails missing the Kerberos ticket

Figure 7-40 shows the appropriate Kerberos commands and the successful mount.

```
-bash-2.05b$ kinit user2
Password for user2@STORAGE.SANJOSE.IBM.COM:


-bash-2.05b$ klist
Ticket cache:  FILE:/var/krb5/security/creds/krb5cc_504
Default principal:  user2@STORAGE.SANJOSE.IBM.COM
Valid starting     Expires           Service principal
05/10/06 11:39:39  05/11/06 07:39:39
krbtgt/STORAGE.SANJOSE.IBM.COM@STORAGE.SANJOSE.IBM.COM
        Renew until 05/11/06 09:41:03
-bash-2.05b$ mount -o vers=4,sec=krb5 sjvm5154:/hfs/u /home/keithc/mnt1
-bash-2.05b$ mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs    Apr 13 11:52 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs    Apr 13 11:52 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs    Apr 13 11:52 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs    Apr 13 11:52 rw,log=/dev/hd8
         /dev/hd1         /home            jfs    Apr 13 11:52 rw,log=/dev/hd8
         /proc            /proc            procfs Apr 13 11:52 rw
         /dev/hd10opt     /opt             jfs    Apr 13 11:52 rw,log=/dev/hd8
         /dev/lv00        /rofs            jfs    Apr 13 11:52 ro
         /dev/lv01        /smallfs         jfs    Apr 13 11:52 rw,log=/dev/hd8
sjvm5154 /hfs/u           /home/keithc/mnt1 NFS4   May 10 09:45 vers=4,sec=krb5
```

Figure 7-40 Sequence of Kerberos commands and the successful NFS V4 mount

New NFS site attribute settings

This section informs about the new NFS Server V1.7 Kerberos site attributes.

mvssec([{krb5, krbri, krb5p} | sys])

This site attribute specifies the acceptable network transmission levels of security, which can be used as the authentication flavor on NFS V4 requests for accesses to MVS data sets. This attribute is only used when not overridden by authentication specifications in the exports file. Multiple values for this attribute can be specified using the comma as the delimiter. The following are the supported values:

- *krb5* provides Kerberos V5-based integrity on RPC credentials (not data) when the RPC authentication flavor is RPCSEC_GSS.
- *krb5i* provides Kerberos V5-based integrity on RPC credentials and data when the RPC authentication flavor is RPCSEC_GSS.
- *krb5p* provides Kerberos V5-based integrity and privacy on both RPC credentials and data when the RPC authentication flavor is RPCSEC_GSS.
- *sys* indicates that the AUTH_SYS authentication flavor can also be used to access this data set. It does not provide any integrity or privacy protection.
- *hfssec([{krb5, krb5i, krb5p} | sys])* is the same as mvssec with the exception that it will govern the acceptable network transmission levels of security for access to UNIX data, which can be specified as the authentication flavor of the RPC request.

- *pubsec ([{krb5, krbri, krb5p} | sys]):* This attribute will govern the acceptable network transmission levels of security for accesses to public file systems, which can be specified as the authentication flavor of the RPC request.

Security - changes to exports data set

This section discusses the changes to the export data set related to Kerberos.

An export data set is a file on the server containing entries for directories that can be exported to NFS Clients and is used by the system administrator to provide controlled access to accredited clients.

The sec keyword will define the acceptable network transmission levels of security access based on the authentication flavor in the client's RPC request. This overrides any network transmission level protection specified in the site attribute and must be equal to or more restrictive then the appropriate site attribute.

NFS Clients are required to access this data set and its descendants with the security flavors specified by the settings of this keyword:

```
sec=krb5:krb5i:krb5p:sys
```

This specifies the acceptable level of network transmission security access that a client's RPC request must provide. If a client attempts to access an object in the directory using a network security level that is not specified on the sec parameter, the access is denied.

- krb5

  This provides Kerberos V5-based integrity on the RPC credentials (but not data) when the RPC authentication flavor is RPCSEC_GSS. It uses the DES_MAC_MD5 integrity algorithm and the RPCSEC_GSS service of rpc_gss_svc_none.

- krb5i

  This provides Kerberos V5-based integrity on both the RPC credentials and data when the RPC authentication flavor is RPCSEC_GSS. It uses the DES_MAC_MD5 integrity algorithm and the RPCSEC_GSS service of rpc_gss_svc_integrity.

- krb5p

  This provides Kerberos V5-based integrity and privacy on both the RPC credentials and data, when the RPC authentication flavor is RPCSEC_GSS. It uses the DES_MAC_MD5 algorithm for integrity and 56-bit DES for privacy. The RPCSEC_GSS service used here is rpc_gss_svc_privacy.

- sys

  This specifies that the AUTH_SYS authentication flavor can also be used to access this file system. Note that the AUTH_SYS authentication flavor does not provide any integrity or privacy protection.

Use a colon (:) to separate security levels.

If the sec parameter is provided, it further restricts the network transmission protection of specific export entries in the domain of the file-system-wide network transmission protection, which is controlled by the mvssec, hfssec, or pubsec site attributes. When this parameter is not specified, the allowed security levels are governed by the mvssec, hfssec, and pubsec site attributes.

Example of a related entry in the export data set

```
/hfs/u -sec=krb5:krb5i:krb5p
   # client must use krb5, krb5i or
   # krb5p authentication levels to
   # access server, provided that
   # hfssec also allows these
   # authentication levels.
```

Restriction: The appropriate security flavor in the export data set or in the mount command given via the sec= keyword is only valid when the same flavor is defined in the related site attribute for the appropriate kind of data set (logical ADD) (for example, hfssec for UNIX data sets (/hfs/u)).

New Kerberos security messages

This section illustrates the new Kerberos messages.

Note: The APAR OA15389 provides information about the Kerberos conditions and lists the available Kerberos messages and their new message text.

- GFSA730I

  ```
  GFSA730I (procname) NETWORK FILE SYSTEM SERVER KERBEROS INITIALIZATION
  SUCCESSFUL
  ```

  – Explanation: The NFS server's Kerberos initialization was successful.In the message text procname is the name of the start procedure.

  – System action: NFS server processing continues.

- GFSA731I

  ```
  GFSA731I NETWORK FILE SYSTEM SERVER COULD NOT LOAD KERBEROS DLL: d_digits
  ```

  – Explanation: Dynamic linked library loading of the Kerberos runtime environment could not be completed during the NFS server startup. This message is issued when NFS is configured to also allow system authentication (AUTH_SYS).

  – If only AUTH_SYS authentication is specified as a site attribute (sys) then this message is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any Kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then this message is issued to the console as well as the NFS server log.

  – System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information.

- GFSA733I

  ```
  GFSA733I NETWORK FILE SYSTEM SERVER COULD NOT CREATE KERBEROS CONTEXT: d_digits
  ```

  – Explanation: The creation of a Kerberos context and its initialization with default values obtained from the Kerberos configuration file during the server startup could not be completed.

  – If only AUTH_SYS authentication is specified as a site attribute (sys) then this message is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any Kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then this message is issued to the console as well as the NFS server log.

– System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information.

- GFSA734E

```
GFSA734E   NETWORK FILE SYSTEM SERVER CREDENTIAL CACHE MANAGEMENT FAILED IN
ROUTINE procName(), KERBEROS RETURN CODE(krbRc) GSS MAJOR STATUS(majorStat) GSS
MINOR STATUS(minorStat)
```

– Explanation: There was a failure in the NFS server's Kerberos credential cache management. This message is issued when NFS is configured to require Kerberos (that is, system authentication (authsys) is not allowed).

In the message text:

- procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
- krbRc is the error code returned by the Kerberos security mechanism. For details, refer to the z/OS Integrated Security Services Network Authentication Service Programming.
- majorStat is the major status returned by the generic security services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
- minorStat is the minor status returned by the Generic SecurityServices. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.

– System action: NFS server processing ends.

– Operator response: Contact the system programmer.

– System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information.

- GFSA735I

```
GFSA735I   NETWORK FILE SYSTEM SERVER COULD NOT SETUP CREDENTIAL CACHE
MANAGEMENT IN ROUTINE procName(), KERBEROS RETURN CODE(krbRc) GSS MAJOR
STATUS(majorStat)
GSS MINOR STATUS(minorStat)
```

– Explanation: There was a failure in the NFS server's Kerberos credential cache management. This message is issued when NFS is configured to also allow system authentication (AUTH_SYS)

– If only AUTH_SYS authentication is specified as a site attribute (sys) then this message is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then this message is issued to the console as well as the NFS server log. In the message text:

- procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
- krbRc is the error code returned by the Kerberos security mechanism. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
- majorStat is the major status returned by the generic security services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.

- minorStat is the minor status returned by the generic security services. For details see Security Services Network Authentication Service Programming.

– System action: NFS server processing continues such that only requests with system authentication (sys) will be supported.

– Operator response: Contact the system programmer.

– System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information about this failure.

- GFSA736E

`GFSA736E   NETWORK FILE SYSTEM SERVER KERBEROS TICKET ACQUISITION FAILED IN ROUTINE procName(), KERBEROS RETURN CODE(krbRc)`

– Explanation: There was a failure in the NFS server's acquisition of Kerberos ticket. This message is issued when NFS is configured to require Kerberos (that is, system authentication (AUTH_SYS) is not allowed). In the message text:

  - procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
  - krbRc is the error code returned by the Kerberos security mechanism. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.

– System action: NFS server processing ends.

– Operator response: Contact the system programmer.

– System programmer response: Refer to z/OS Integrated SecurityServices Network Authentication Service Programming for more information about this failure.

- GFSA737I

`GFSA737I   NETWORK FILE SYSTEM SERVER COULD NOT GET KERBEROS TICKET IN ROUTINE procName(), KERBEROS RETURN CODE(krbRc)`

– Explanation: There was a failure in the NFS server's acquisition of the Kerberos ticket. This message is issued when NFS is configured to also allow system authentication (AUTH_SYS).

– If only AUTH_SYS authentication is specified as a site attribute (sys) then this message is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any Kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then this message is issued to the console as well as the NFS server log. In the message text:

  - procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
  - krbRc is the error code returned by the Kerberos Security Mechanism. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.

– System action: NFS server processing continues such that only requests with system authentication (sys) will be supported.

– Operator response: Contact the system programmer.

– System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information about this failure.

- GFSA738E

`GFSA738E NETWORK FILE SYSTEM SERVER GSS CREDENTIAL ACQUISITION FAILED IN ROUTINE procName(), GSS MAJOR STATUS(majorStat), GSS MINOR STATUS(minorStat)`

– Explanation: There was a failure in the NFS server's acquisition of GSS credentials. This message is issued when NFS is configured to require Kerberos (that is, system authentication (AUTH_SYS) is not allowed). In the message text:
  - procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
  - majorStat is the major status returned by the generic security services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
  - minorStat is the minor status returned by the generic security services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
– System action: NFS server processing ends.
– System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information about this failure.

- GFSA739I

  ```
  GFSA739I NETWORK FILE SYSTEM SERVER COULD NOT ACQUIRE GSS CREDENTIALS IN
  ROUTINE procName(), GSS MAJOR STATUS(majorStat), GSS MINOR STATUS(minorStat)
  ```

  – Explanation: There was a failure in the NFS server's acquisition of GSS credentials. This message is issued when NFS is configured to also allow system authentication (AUTH_SYS). If only AUTH_SYS authentication is specified as a site attribute (sys), then this message is issued to the NFS server log only (and not to the console). If AUTH_SYS authentication (sys) and any Kerberos flavors (KRB5, KRB5I, or KRB5P) are specified as site attributes then this message is issued to the console as well as the NFS server log. In the message text:
    - procName is the name of the failing API. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
    - majorStat is the major status returned by the Generic Security Services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
    - minorStat is the minor status returned by the Generic Security Services. For details refer to the z/OS Integrated Security Services Network Authentication Service Programming.
  – System action: NFS server processing continues such that only requests with system authentication (sys) will be supported.
  – Operator response: Contact the system programmer.
  – System programmer response: Refer to z/OS Integrated Security Services Network Authentication Service Programming for more information about this failure.

NFS V4 operations

This section gives an overview of the NFS V4 operations and provides information about the NFS V4 operations that are supported by the z/OS 1.7 NFS Server.

This is an operation comparison of the existing NFS versions:

- NFS V2 - 18 operations
- NFS V3 - 22 operations

  V2/V3 use traditional RPC.
- NFS V4 - 38 operations

  V4 uses the COMPOUND procedure to build operation sequences.

Note: Technically, these operations are procedures. NFS V4 has two procedures (NULL and compound) and 38 operations under the compound procedure.

Table 7-4 illustrates the new NFS V4 operations that are supported or not by z/OS NFS 1.7 and the similar V2/V3 operations.

Table 7-4 NFS V4 operations

Supported new operation	Unsupported new operation	Similar to V2 and V3 operations
Close	DelegPurge	Access
GetFH	DelegReturn	Commit
LookUpP	Lock	Create
NVerify	LockT	GetAttr
Open	LockU	Link
Open_Confirm	OpenAttr	LookUp
Open_Downgrade		Read
PutFH		ReadDir
PutPubFH		ReadLink
PutRootFH		Remove
Renew		Rename
RestoreFH		SetAttr
SaveFH		Write
SecInfo		
SetClientId		
SetClientId_Confirm		
Verify		
Release_LockOwner		

NFS V4 - mount elimination and name space solution

This section gives an overview of the NFS V4 components mount elimination, compound, and pseudo namespace.

Because the NFS V4 protocol does not use the mount procedure, a mount command from an NFS V4 client will be translated into a PUTROOTFH operation and a series of lookup operations for each component of the mount path name (compound request). A z/OS NFS server cannot accurately ascertain where the mount command will be completed, so both SAF and EXPORT checking are disabled for the z/OS NFS Version 4 lookup operation, making namespace navigation available without mvslogin until an export entry was matched (as SAF would otherwise require, and EXPORT would otherwise prematurely prevent). When non-navigation operations are received by the z/OS NFS server, the proper SAF and EXPORT checking will assure that proper access is given.

Attention: We highly recommend that the attribute mvsmnt be used with SAF or SAFEXP enabled for a z/OS NFS V4 server due to the above limitations.

Also, if processing attribute mvsmnt is *not* used for a SAF or SAFEXP z/OS NFS Version 4 server, security verification issues may arise when a z/OS UNIX file system is being accessed because the z/OS NFS server may use the UNIX permission bits instead of allowing RACF to perform its job as desired.

With NFS V4, the new processing mvsmnt attribute should be specified on user mount commands. In NFS V4, mount requests are passed to the server in the form of a PUTROOTFH operation followed by a sequence of lookup operations. mvsmnt indicates to the z/OS NFS server that the associated lookup operation is emulating a mount procedure and causes the z/OS NFS server to write the mount point to the mount handle database (MHDB), so the server can automatically recover the mount point during a server restart.

MVSMNT specifies that in NFS Version 4, a lookup will be treated as though a mount procedure were given for the LOOKUP result object. For any LOOKUPs that do not specify MVSMNT, any processing attributes that may have been provided will be merged with any that were in effect for the LOOKUP parent directory.

For LOOKUPs that *do* specify MVSMNT, any other processing attributes provided will be merged with the site defaults. MVSMNT cannot be specified for any LOOKUP where the parent directory was navigated to by an mount procedure or a result of an object that was already LOOKUPed with MVSMNT. This is to ensure that only a client system mount specifies MVSMNT. With MVSMNT specified, z/OS NFS server site security attributessaf and safexp are used to control access to z/OS UNIX file systems. MVSMNT is not to be specified as a site default attribute.

Summary of the mount protocol elimination

Using NFS V4, there no mount protocol is given (firewall enhancement). The navigation begins at the server root and the NFS client must navigate to the mount point.

The NFS V4 protocol provides *pseudo* name space. z/OS NFS will not change the name space when using NFS V4. Non-exported z/OS UNIX objects are given limited access to allow navigation to exported objects.

NFS V4 provides the following operations for FileHandle manipulation:

- PUTROOTFH: starting point
  - Puts NFS root in Current FileHandle (CFH)
- PUTPUBFH: starting point for public (Web)
  - Puts NFS public root in CFH.
  - GETFH: Return CFH in result to client.
  - PUTFH: Put argument FH into CFH.
  - SAVEFH: Place CFH into Saved FH (SFH).
  - RESTOREFH: Place SFH into CFH.

Note: The two operations SAVEFH and RESTOREFH are only valid for the duration of the compound request.

Figure 7-412 illustrates the mount elimination by using a compound request through a firewall.

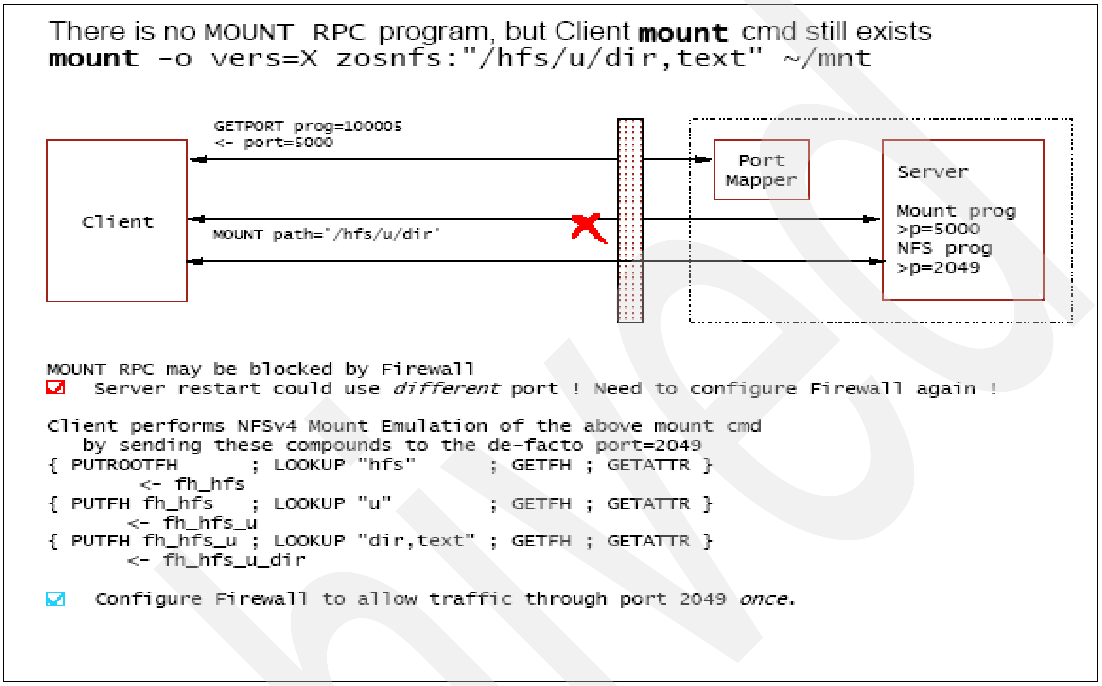


Figure 7-41 Mount protocol elimination

The mount itself must still be issued on the application side as with NFS V2 and NFS V3, but the NFS client does not send the mount itself over the wire to the z/OS NFS Server because there is no MOUNT request known in NFS V4 protocol. Instead, several LOOOKUP requests are sent to the z/OS NFS Server as compound procedure based on the mount cmd syntax. The NFS root will be found by the PUTROOTFH request. The GETFH and GETATTR requests belong to the appropriate compound procedure.

Name space role of NFS root

The following sections explain the new NFS V4 mount process, the name space, and pseudo namespace in more detail based on an example. Figure 7-42 shows the NFS root from the name space point of view.

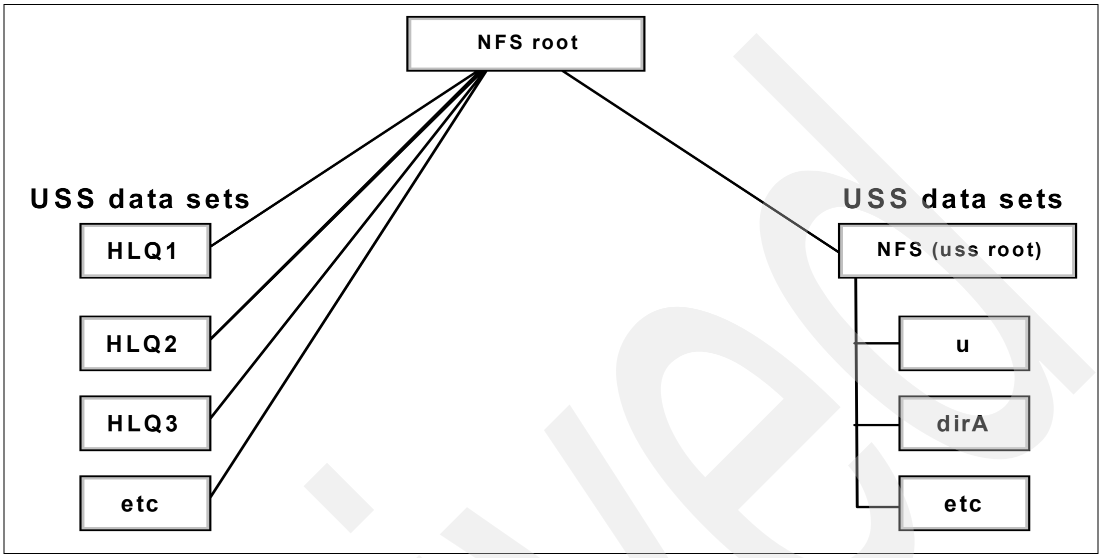


Figure 7-42 Name space with the NFS root as the main anchor

Example of namespace in NFS V4

The NFS root is built internal to the z/OS NFS Server as an anchor point to navigate down the path either for HFS files or MVS data sets based on the given information from the initial mount syntax. This NFS root is found by the NFS V4 request PUTROOTFH.

Assume that the site attribute HFS prefix is /hfs (which means that it is a HFS data set). The application or user issues the following mount:

```
mount -o vers=4 machine:/hfs/u/user3 /localdir
```

The NFS client builds the following RPC based on this mount syntax:

```
COMPOUND:
   PUTROOTFH
   LOOKUP name=hfs
   LOOKUP name=u
   LOOKUP name=user3
   GETFH
```

LOOKUPS will be interspersed with GETATTR, SECINFO, and GETFH, and may even arrive in separate COMPOUNDs with GETFH and PUTFH operations managing progress.

Example of pseudo namespace access

Figure 7-43 shows the path within the NFS Server namespace based on the above given mount example and information about the pseudo namespace.

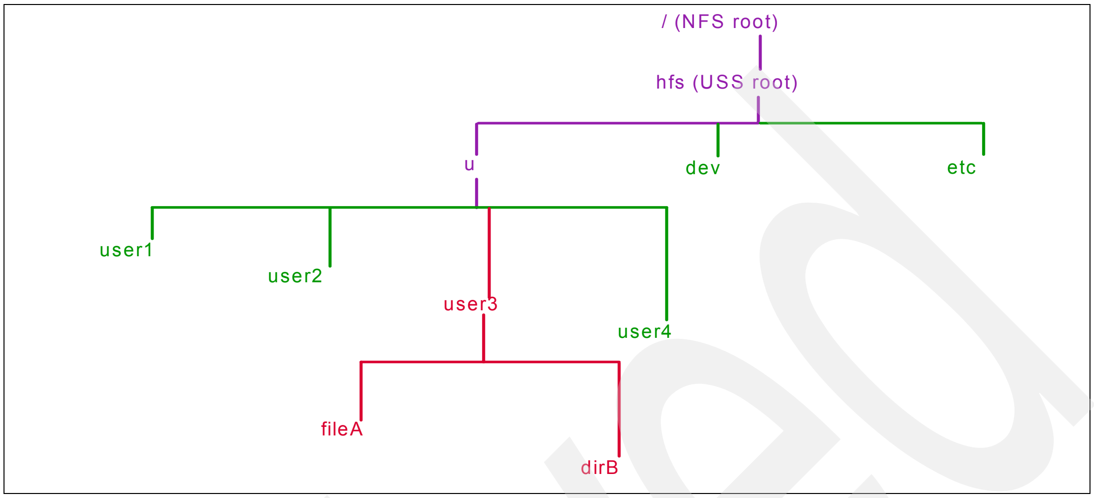


Figure 7-43 NFS Server name space path

The path from the NFS root over the USS root to the subdirectory /u is named *pseudo* namespace because /hfs/u is just built from the lookup requests done based on the related mountpoint /hfs/u/user3, on the mount: **`mount -o vers=4 machine:/hfs/u/user3 /localdir`** `and the appropriate export entry` **`/hfs/u/user3`**. For this pseudo path /hfs/u the user has limited access. The user has no access (hidden files) to /dev, /etc and /u/user1, /u/user2 and /u/user4. The user has full access to /u/user3 and all the underlaying files and directories fileA and dirB because of the mount syntax /hfs/u/user3. The user can also delete and create new files and directories under /hfs/u/user3 because of this full access.

> **Restriction:** Using NFS V4, the z/OS NFS Server cannot determine the initial mountpoint based on the mount syntax as with NFS V2 and NFS V3 because the MOUNT is not sent, only the related lookup requests. Since the z/OS NFS Server does not know any mount, the appropriate mountpoint cannot be stored in the MHDB as before. Therefore, the mountpoint cannot be build during the restart of the z/OS NFS Server.
>
> To store, rebuild, and handle the initial mountpoint from the mount cmd as before, the new processing and mount attribute mvsmnt is required.

Reduced message traffic - compound RPC

This section is related to “NFS V4 - mount elimination and name space solution” on page 266. Many operations are sent in a single request. The advantages are:

- Reduces message traffic
- Improves overall communication throughput

An example of a compound request is:

```
PutRootFH
Lookup
GetFH
GetAttr
Open
Read
```

Compound overview

Using the COMPOUND, the client can:

- Send a list of NFS V4 operations (analogous to V2/V3 procedures) in one COMPOUND procedure request.
- Operation from user interface on client should look no different (for example, ls, dir, cd).
- Should be able to see faster performance due to lower overhead than previous single procedures.

The advantages are the following:

- Decrease user end response time by reducing client/server exchanges.
- Decreases server execution time by reducing z/OS task-to-task communication.
- Decrease server execution time from operation-to-operation within COMPOUND procedure and make more intelligent decisions for managing resources (interoperation).

Summary

The NFS V4 protocol provides design for a pseudo file system, which may allow for providing an alternate name space that can provide shortcuts to exported objects.

- z/OS NFS V4 design point for USS file systems

  The name space does not change. Assume that the HFS prefix site attribute is set to /hfs. A client who wishes to use the older mount protocol to mount /hfs/u/mountpt will use the same name as NFS V4. In this fashion clients would not have to change their mount path names when changing between NFS V3 and V4.

- z/OS NFSv4 design point for MVS file system

  > **Restriction:** The pseudo name space does not apply to MVS data sets.

  A client that would normally mount to an MVS High-Level Qualifier (HLQ) would send:

  ```
  Client: mount -o vers=4 machine:HLQ
     RPC: COMPOUND
     PUTROOTFH
     LOOKUP name=HLQ
     GETFH
  ```

The z/OS NFSv4 server will basically treat any lookup from the root as a mount command.

> **Note:** This statement does not mean that the z/OS NFS server does not need the mvsmnt attribute to store the mount information into the MHDB.

NFS V4 cross-mountpoint local file system

NFS V4 lifts the restriction of not being able to cross existing mountpoints. To allow navigation across mountpoints the client navigation must be able to traverse USS file system boundaries.

Restriction: NFS V2/V3 does not traverse file systems, so these do not use cross mount point.

When crossing a USS mount point, the z/OS 1.7 NFS V4 Server will create an FSYS block. If the server goes down and comes up, this FSYS will not be built at initialization time unless it is recorded.

Any filehandle referring to a crossed mount will fail with NFSERR_FHEXPIRED if the FSYS cannot be found.

- AIX and Solaris recover by re-navigating from the parent FS into the child FS.
- Linux and Windows do not recover well, so to ensure that the FSYS is built at NFSS initialization, an object from this FS should be recorded in the MHDB.

Thus for these operating systems, the clients should be set up to mount these file systems with the mvsmnt attribute.

Figure 7-44 shows an example of a cross-mountpoint with NFS V4.

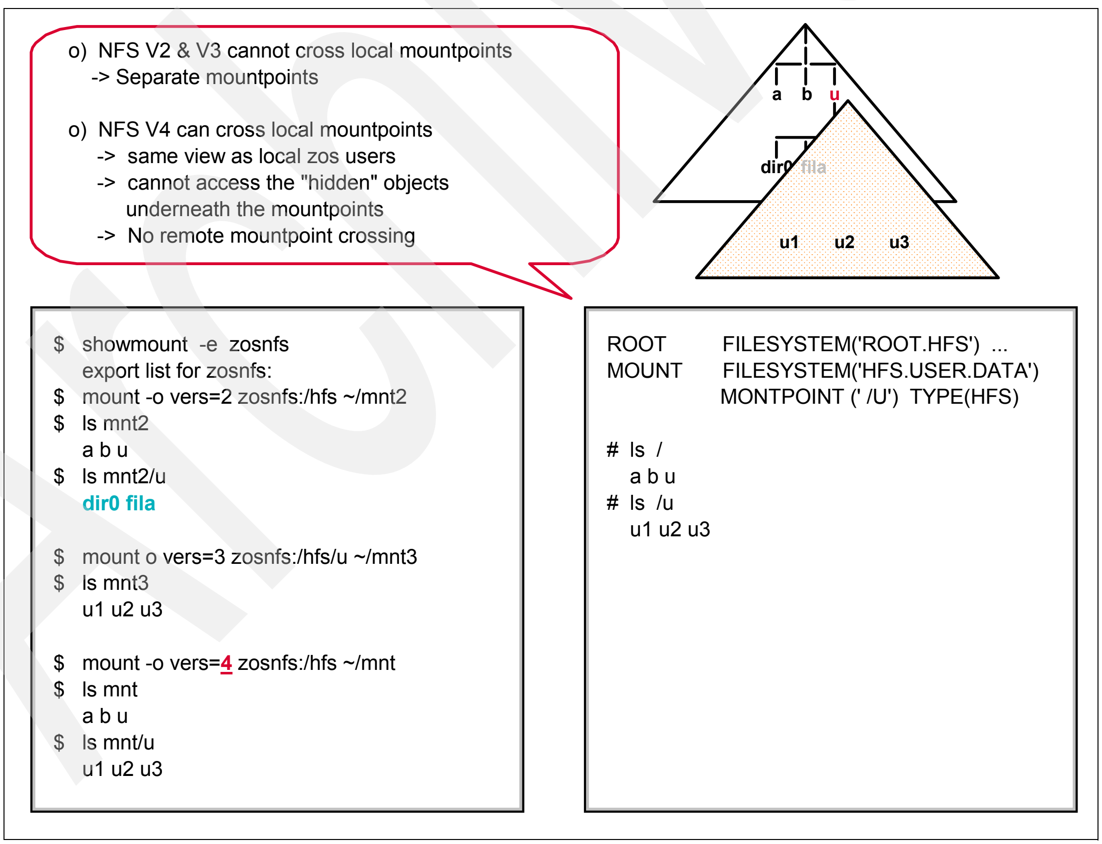


Figure 7-44 NFSV4 cross-mountpoint local file system

This example is similar to the path in Figure 7-43 on page 270 and shows the different access to a HFS file between NFSV2/V3 and NFS V4. The user navigates from the root based on the appropriate mounts shown in the left-hand box, while the USS mountpoints are done based on the USS mounts in the right-hand box. The two triangles on the upper right-hand side reflect the local USS path, while the lower triangle shows just the content of the /u directory.

The right-hand box shows two USS mounts and the output of the related ls command as the content of the files and directories represented by the two triangles above as the local USS view.

The left-hand lower box shows the NFS mounts and the output of the ls commands as the content of the local mountpoints based on the path of the NFS mount syntax and based on the underlaying USS mounts as the remote NFS view.

The first NFS mount in the left-hand lower box is in principle the same mount as shown in the right-hand box. The first ls command shows the directories a, b, and u as the local view.

The second is the command for /u showing its underlaying content diro and fila, *but* does not show the directories u1, u2, and u3 as the local view based on the restriction of not being able to cross mountpoints, while /u is the second mountpoint based on the information in the right-hand box.

The third NFS mount in the left-hand lower box is a NFS V4 mount done on the USS root the same as the USS mount in the right-hand box. For this NFS mount the ls command now shows the same content as the local view in the right-hand box, which demonstrates that a NFS V4 mount is able to cross mountpoints.

Summary: crossing between file systems

When a client does a lookup on a directory on which the server has mounted a file system, the client sees the underlying directory instead of the mounted directory. The NFS server does not support crossing file systems in NFS protocol V2 and V3. For example, if a server has a file system called /usr and mounts another file system on /usr/src, a client can also mount /usr, but the server will not see the mounted version of /usr/src. A client could perform remote mounts that match the server's mount points to maintain the server's view. In this example, the client would also have to mount /usr/src in addition to /usr, even if the mounts are from the same server.

For NFS protocol V4, the NFS server does support crossing file systems. This change is due to the mount protocol being obsolete in NFS V4. Using the example of the previous paragraph, the server would see the mounted version of /usr/src and any objects within that file system.

NFS server crossing file systems means that the NFS client can also potentially be a server, and remote and local mounted file systems can be freely mixed. This leads to some problems when a client travels down the directory tree of a remote file system and reaches the mount point on the server for another remote file system. Allowing the server to follow the second remote mount would require loop detection, server lookup, and user revalidation.

> **Restriction:** The z/OS NFS Server can only cross between file systems that are on the same z/OS system. It cannot cross mount points to remotely mounted file systems because of the problem mentioned in the previous paragraph.

Stateful NFS V4

This section discusses the stateful NFS V4 process.

Overview

NFS V2 and NFS V3 are stateless. The NFS client perceives seamless read/write even if the NFS server crashes and restarts.

Stateful NFS V4 supports:

- OPEN/CLOSE semantics and share reservations
- Byte range locking
- File creation

z/OS NFS 1.7 does not support byte range locking but OPEN/CLOSE semantics and share reservations.

The most UNIX systems, as with the NFS client, use byte range locking to lock the file, while Windows uses share reservations.

NFS V4 states represent the *open* states or *locking* states.

- The *open* state (open_stateid4) represents the current file share reservation of a client-server association instance (lease/clientid4).
- The *locking* state (lock_stateid4) represents the current byte range locking of an *opened* file.

Lease concept:

- The NFS client is established via SETCLIENTID, SETCLIENTID_CONFIRM and must explicit/implicit RENEW the lease if having states on the server.
- When the lease is lost, the z/OS NFS server thinks that the NFS Client crashes and it will release the resources.

Using the LEASETIME site attribute:

- Stateful OPEN/READ/WRITE/LOCK/LOCKU/CLOSE operations provide lock step stateid. Handshake and states are maintained between client and server.
- A server restart will allow an OPEN_RECLAIM.
- A tremendous benefit is in the MVS/legacy processing because previously the z/OS NFS Server would CLOSE a data set or member based on writetimeout or readtimeout. Now the CLOSE operation will tell the z/OS NFS server to CLOSE and to de-allocate the data set or member.
- The READTIMEOUT and WRITETIMEOUT processing attributes are ignored.
- If the NFS client crashes, the z/OS NFS server will CLOSE the data set upon lease expiration.

Figure 7-45 on page 275 shows a flow of a open/read/write/close request based on the operations of the application mentioned on the left-hand side while using a mount as given in the upper left corner. Before NFS V4 the HFS file or MVS data set was closed when the appropriate time out was reached, not when the close occurred. The example shows the NFSV3 requests in the middle and the NFSV4 compounds on the right. The figure also shows a flow comparison of a basic fopen/write/sleep/fwrite/fclose request sequence when executed with NFS Version 3 or 4.

- NFS V3 notes

  Since NFS v3 does not have explicit open/close procedures, the file is not opened until the first write request. Due to lack of file activity during the sleep (90), the file is timed out and closed. The subsequent fwrite fails because the file has already been closed.

- NFS V4 notes

  The file is opened in response to the fopen request. It does not wait until the first fwrite. During the sleep (90), the NFS Client automatically sends an explicit RENEW (heartbeat) to the server due to the outstanding open state with no other stateful activity in process. NFSV4 users are not constrained by xxxtimeout due to the stateful NFS V4 open/close semantic. The second write request completes successfully because the file is still in the open state.

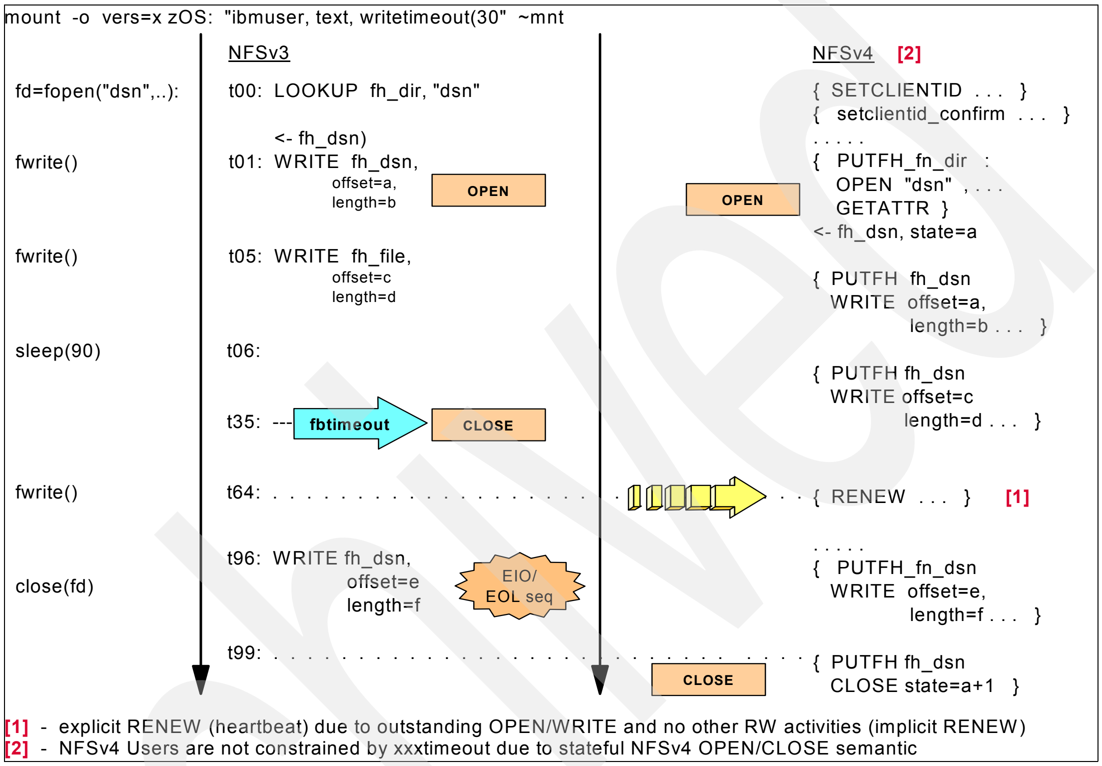


Figure 7-45 Stateful NFSV4 request

7.5 NFS V1R7 Client

This chapter gives an overview of the z/OS NFS Client and provides information about the NFS V1.7 line item for the client.

7.5.1 Overview of the z/OS NFS Client (NFSC)

With the z/OS NFS client you can allow basic sequential access method (BSAM), queued sequential access method (QSAM), virtual storage access method (VSAM), and z/OS UNIX users and applications transparent access to data on systems that support the NFS Version 2 protocols and the NFS Version 3 protocols. The remote NFS Server can be a z/OS, UNIX, AIX, or other system. The z/OS NFS client is implemented on z/OS UNIX and implements the client portion of the NFS Version 2 protocols and the NFS Version 3 protocols.

Supported servers for the z/OS NFS client

The z/OS NFS client supports all servers that implement the server portion of the NFS V2 and V3 protocols.

Attention: The z/OS NFS Client does not support NFS V4 at this time.

A mount parameter vers(*x*), where *x* is either 2 or 3 and is provided to make the z/OS NFS client communicate with the server at the specified protocol level. The z/OS NFS client also communicates at the highest protocol level that is supported by the server if no level is specified.

When no version is specified and the NFS server only supports the NFS Version 2 protocol, then the z/OS NFS client will use the NFS V2 protocol to communicate.

When no version is specified and the NFS server supports both the NFS V2 and V3 protocols, then the z/OS NFS client will use the NFS Version 3 protocol to communicate.

When no version is specified and the NFS server supports the NFS V2, V3, and V4 protocols, then the z/OS NFS client will use the NFS Version 3 protocol to communicate.

When the vers(2) is specified, then use NFS V2 protocol to communicate with the server and the z/OS NFS client fails the mount command if the server does not support NFS V2 protocol.

When the vers(3) is specified, then use NFS V3 protocol to communicate with the server and the z/OS NFS client fails the mount command if the server does not support NFS V3 protocol.

Configuring the z/OS NFS client

This section describes the tasks that you can perform to configure the z/OS NFS client. These tasks include creating the PARMLIB statement and updating z/OS system data sets for the client. This section also includes information about allocating client log data sets and mounting remote file systems.

During z/OS UNIX file system initialization, the z/OS NFS client is started and run in the logical file system (LFS) colony address space. The filesystype parmlib statement for the z/OS NFS Client must be present in the SYS1.PARMLIB(BPXPRMxx) parmlib member in order to start the z/OS NFS client. For more information about the z/OS UNIX file system see *z/OS UNIX System Services File System Interface Reference*.

To accommodate the z/OS NFS client you must update z/OS system data sets PARMLIB, PROCLIB, and the DD statement. PARMLIB updates the data set defined in the GFSCPROC STEPLIB containing the z/OS NFS client library to the system's APF authorization list (IEAAPFxx). A sample cataloged procedure named GFSCPROC is provided as a member of the sample library NFSSAMP. See "Sample z/OS NFS client startup procedures" on page 356.

Add the filesystype parmlib statement shown in Figure 7-46 to the z/OS UNIX parmlib member (BPXPRMxx.

```
FILESYSTYPE TYPE(NFS)
ENTRYPOINT(GFSCINIT)
PARM('installation parms')
ASNAME(proc_name)
```

Figure 7-46 Sample filesystype parmlib statement

The name in the TYPE operand must be NFS. Otherwise the utility program NFSstat fails.

The operand ENTRYPOINT(GFSCINIT) specifies the entry point for the z/OS NFS client initialization.

The operand PARM('installation parms') specifies the installation parameters for the z/OS NFS client.

The operand ASNAME(proc_name) specifies the procedure name in SYS1.PROCLIB that is used by z/OS UNIX to start the address space in which the z/OS NFS client is initialized. Note that the proc_name is also used for the name of the address space.

For data integrity and data isolation among different PFSs, the z/OS NFS client is required to start in a separate and standalone colony address space. To start the NFS client in a separate and standalone colony address space, a unique proc_name must be used. As PROCLIB updates Add the procedure name, proc_name, specified in the ASNAME(proc_name) operand to the system PROCLIB.

PROCLIB updates

Add the procedure name, proc_name, specified in the ASNAME(proc_name) operand to the system PROCLIB. FILESYSTYPE TYPE(NFS) ENTRYPOINT(GFSCINIT) PARM('installation parms') ASNAME(proc_name), shown in Figure 7-46 on page 276.

A sample cataloged procedure named GFSCPROC is provided as a member of the sample library NFSSAMP. See "Sample z/OS NFS client startup procedures" on page 356. Modify the MVSNFSC procedure and place it in your system PROCLIB. Add the DD statements:

```
NFSCMSG1 as the DD for the primary log data set
NFSCMSG2 as the DD for the secondary log data set
SYSxDUMP as the DD for the SYSxDUMP data set ('x' = U or M)
```

Mounting remote file systems: z/OS UNIX does not support z/OS NFS mounts in the SYS1.PARMLIB member statement. You can use the z/OS UNIX automount facility (/etc/rc shell scripts support) or the TSO MOUNT command to make a connection between a mount point on your local z/OS UNIX file system and one or more files on a remote MVS, AIX, UNIX, z/OS, or other file system. The remote file system can be mounted using the TSO MOUNT command only after the z/OS UNIX file system initialization is complete. The TSO MOUNT command can only be used by a z/OS UNIX superuser. For more information about the TSO MOUNT command when used with the z/OS NFS client, see Figure 7-47.

```
MOUNT FILESYSTEM(file_system_name)
TYPE(NFS) MOUNTPOINT(local_mountpoint)
MODE(RDWR|READ)
PARM('hostname:"path_name,server_attributes", client_options')
SETUID|NOSETUID
TAG(TEXT,CCSID)
WAIT|NOWAIT
```

Figure 7-47 TSO MOUNT command syntax operands

An explanation of the TSO MOUNT operands mentioned in Figure 7-47 follows.

- FILESYSTEM(file_system_name)

  This specifies the name of the file system to be added to the file system hierarchy. This operand is required. The file system name specified must be unique among previously mounted file systems. It may be an arbitrary name up to 44 characters in length of a file system. You can enclose file_system_name in single quotation marks (' '), but they are not required.

- TYPE(NFS)

  This specifies the type of file system that performs the logical mount request. The NFS parameter must be used.

- MOUNTPOINT(local_mountpoint)

  This specifies the path name of the mount point directory, the place within the file hierarchy where the file system is to be mounted. The local_mountpoint specifies the mount point path name. The name can be a relative path name or an absolute path name. The relative path name is relative to the current working directory of the TSO/E session (usually the home directory). Therefore, you should usually specify an absolute path name. A path name is case-sensitive, so enter it exactly as it is to appear. Enclose the path name in single quotation marks. Note that the mount point must be a directory. Any files in that directory are inaccessible while the file system is mounted. Only one file system can be mounted to a mount point at any time.

- MODE(RDWR|READ)

  This specifies the type of access for which the file system is to be opened. RDWR specifies that the file system is to be mounted for read and write access. RDWR is the default option. READ specifies that the file system is to be mounted for read-only access.

- PARM('hostname:."path_name,server_attributes"., client_options')

  This specifies the host name of the remote Network File System server, the server attributes, and the options. The double quotation marks (" ") can be omitted if no server attributes are specified. Enclose the entire string in single quotation marks. If the automount facility is being used, the single quotation marks may be specified but are not required. You can specify lowercase or uppercase characters. A blank space must precede any client_options specified.

- SETUID|NOSETUID

  See the *z/OS UNIX System Services Command Reference* for details about SETUID and NOSETUID.

- TAG(TEXT,CCSID)

  See the *z/OS UNIX System Services Command Reference* for details about the TAG keyword.

  Note: When TAG is specified, xlat(Y) must not be specified. Otherwise, mount will fail.

- WAIT|NOWAIT

  See the *z/OS UNIX System Services Command Reference* for details on WAIT and NOWAIT.

When using the automount facility of z/OS UNIX System Services, the remote file system is mounted on its first data access attempt if it is not already mounted. When the automount facility is used to manage remote NFS mount points, the z/OS NFS user could experience ESTALE/EIO errors if the automounter unmounts the accessed mount point when the time limits specified by the automount duration and delay parameters have been exceeded. The automount default, DURATION=NOLIMIT, disables the DURATION time-out period. The DELAY for unmounting file systems should be longer than the time z/OS NFS clients are likely to keep z/OS NFS mounts to the files and directories active. For more information about the z/OS UNIX automount facility (/etc/rc shell scripts support) see the *z/OS UNIX System Services Planning and z/OS UNIX System Services File System Interface Reference*.

The remote file system must be mounted on the z/OS UNIX file system prior to any reference being made to the remote data. Once mounted, the remote file system can be treated as an extension of the local z/OS UNIX file system.

How to start, stop, and handle the z/OS NFS Client

Since the z/OS NFS client is started and run in the logical file system (LFS) colony address space, as mentioned in "Configuring the z/OS NFS client" on page 276, the filesystype parmlib statement for the z/OS NFS client must be present in the SYS1.PARMLIB(BPXPRMxx) parmlib member in order to start the z/OS NFS client.

A sample JCL of the started task for the z/OS NFS Client is delivered in the SYS1.NFSSAMP as member GFSCPROC.

> **Tip:** It is possible to have a single separate parmlib member just with the filesystype statement for the NFS, as shown in Figure 7-46 on page 276. This parmlib member can be used to start the z/OS NFS Client after it was canceled, separate of USS and while USS is active. The appropriate cmd is **`/setomvs reset=(NF)`**, while the parmlib member BPXPRMNF is in SYS1.PARMLIB and refers with ASNAME(NFSC) to the started task of the z/OS NFS Client in the appropriate proclib.

When the z/OS NFS Client is canceled or stopped the following WTO is shown in the syslog:

```
*20 BPXF032D FILESYSTYPE NFS TERMINATED.  REPLY 'R' WHEN READY TO
 RESTART. REPLY 'I' TO IGNORE.
```

With '/20,I' the client will finally be shut down. It must to be started through USS or via a separate parmlib member, as mentioned in the tip above.

With '/20,R' the client will restart, as shown by the startup msg:

```
GFSC700I OS/390 NETWORK FILE SYSTEM CLIENT (HDZ11TC) STARTED
```

Message GFSC700I confirms that the z/OS NFS Client is successfully started.

7.5.2 z/OS NFS client line items

This section contains the z/OS NFS Client line enhancements.

z/OS NFS RPC-to-socket conversion

This section informs about the new RPC-to-socket conversion enhancement.

The changes are hidden from the client, but they will represent a performance enhancement.

Figure 7-48 shows a synchronous RPC versus asynchronous RPC flow.

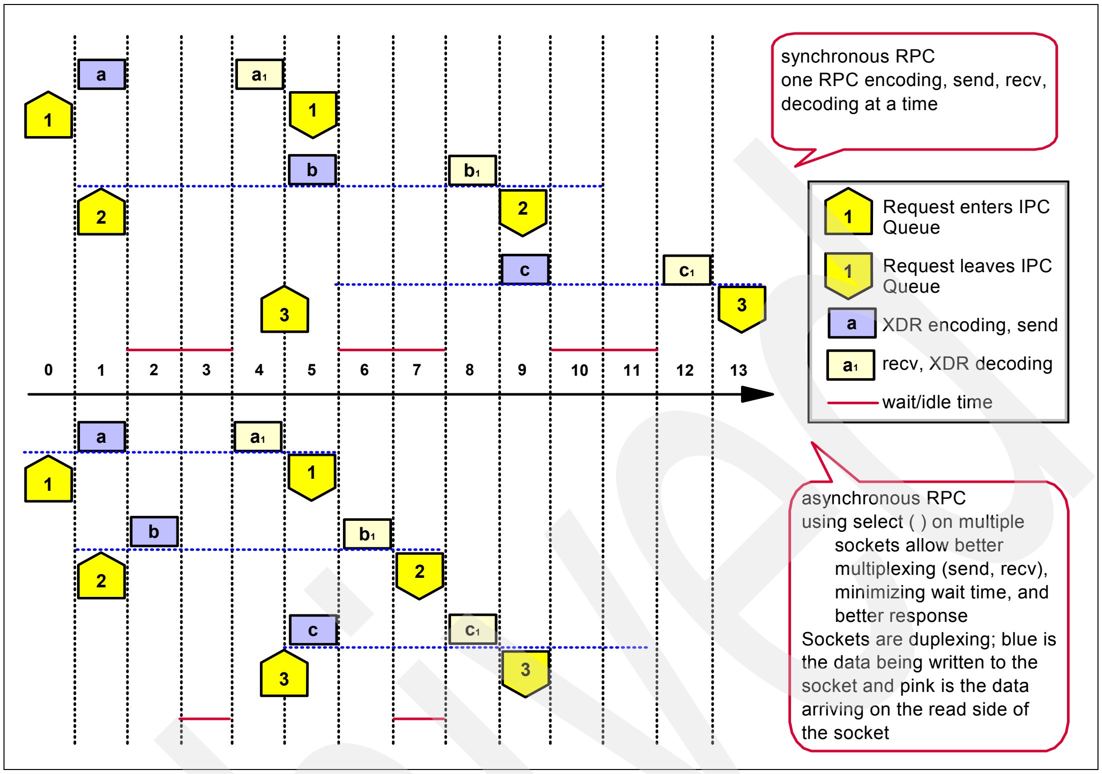


Figure 7-48 Synchronous RPC versus asynchronous RPC

The symbol 1 with the head up indicates that a RPC request enters the internal inter process communication (IPC) queue and when the request leaves the IPC queue shown by symbol 1 with the head down. The symbols 2 and 3 show the same for two additional requests coming in on the IPC queue in parallel.

The flow above the time line shows the existing synchronous RPC process doing one RPC encoding, send, recv, and decoding at a time while the flow below the time line show the new asynchronous RPC process using select() on multiple sockets allow better multiplexing(send,recv) minimizing wait time and better response while the sockets are duplexing.

The stroke on the bottom of each flow indicates its individual wait/idle time show the difference as the performance improvement for the asynchronous process in the lower half doing encoding and decoding in parallel.

7.6 Migration and coexistence

This section documents the migration steps and coexistencies when installing z/OS NFS 1.7:

- The installation of z/OS NFS V1R7 will automatically replace any older release of z/OS NFS.
- All of the NFS servers active on a single z/OS system must be at the same NFS release level.

  If a client installation contains a multi-system z/OS sysplex, and has NFS Servers activated on multiple systems of the sysplex, there is no requirement for all of the systems in the sysplex to be at the same NFS release level. However, sysplex-wide file share and byte range locking is only supported if all of the systems are using the z/OS NFS V1R7 server.

- The NFS server startup procedure must be updated with DD statements for the VSAM data sets that are now used to record the names of client hosts that are accessing the server.

7.6.1 Migration consideration about the mount handle database (MHDB)

This section gives an overview of the new MHDB from the migration point of view and its new format.

APARs OA13899 and OA11612 document the migration/coexistence of the MHDB.

The SYS1.NFSSAMP data set contains a sample job to create the new MHDB and LDB. In addition, there is a job to migrate the existing MHDB to the new V1.7 format.

The MHDB record structure has changed. Before z/OS NFS V1.7 the design was to require a fresh MHDB, meaning clients had to unmount and remount all of their NFS mount points.

With a special procedure and the logic in z/OS NFS V1.7, the MHDB from previous releases can be used.

Comparison of the HDZ11TS/HDZ11US/HDZVS MHDB structure

Before z/OS NFS 1.7 US does not tolerate TS records, even though they are identical. The MHDB used the 4-byte IP address.

The changed elements of the HDZ11VS MHDB structure are the IPV6 IP address, the added term ID, and the added host name (DHCP resolved).

MHDB client actions

The new MHDB structure exceeds record definition size in GFSAMHDJ of previous releases.

The client-required actions are to create new MHDB data sets with new attributes and migrate the records from the old to the new VSAM data sets. The restrictions are:

- Using DHCP, the MHDB record will be ignored when there is no host name.

  It is a design point in DHCP that if an MHDB record does not have the host name field, the record will be ignored. The previous NFS server instance was running in non-DHCP mode and the host name was never resolved. For translated records, there is no host name in the previous versions.

- Mounts into the USS root file system will be unusable.

  The USS root file system (typically OMVS.ROOT) is known to change the fileID attribute for its objects between V1R6 and V1R7. Thus any mount handles from V1R6 in this file system will be unusable. The client must re-mount any mount points in the USS root file system.

7.7 z/OS DFSMSnfs V1R8 preview

In this section we cover the items that are changed in the DFSMSnfs V1R8.

Exports file net group support

The support is:

- NFS server checklist function enhancements
- Exports file format changes
- Dynamic refresh of EXPORTS file information
- SHOWMOUNT command

NFS Server V4 protocol support continuation

Continue implementation of the z/OS NFS Server portion of the NFS V4 protocol:

- Internationalization - UTF-8 and stringprep support
- RPCBIND-based registration

  Change NFS Server registration from Portmapper™ to RPCBIND for IPv6 support.

Serviceability improvements

In the continuing effort to improve the quality and usability of z/OS NFS, several serviceability improvements have been identified. These improvements are grouped under the serviceability improvements line item category.

- Convert NFS Client external trace file to CTRACE.

  Conversion of the z/OS NFS Client from use of its own proprietary debug trace mechanism to exploiting the system component trace (CTrace) facility.

- List mounts operator command host name support.

  Modify the LIST MOUNTS operator command to return the client host name.

- NFS Client stop pfs operator command support.

  Implement support for proper function vector cleanup when the NFS Client is terminated.

> **Note:** Starting with NFS V1.8, both the z/OS NFS Server and z/OS NFS Client have the same FMID of HDZ118N.

7.7.1 Exports file net group support

z/OS NFS provides multiple mechanisms for specifying the authentication to be used when validating an NFS Client's authority to access a z/OS file system. One of these mechanisms (selected via the SECURITY(EXPORTS | SAFEXP) site attribute) uses an exports file to specify access rules. When this mechanism is used, only the file systems listed in this file may be accessed via NFS. Further restrictions may be placed on the access when desired. If only some NFS Clients are to be allowed access to a given file system, those NFS clients may be listed on parameters (that is, Access, ro, rw) to indicate that only those clients may access the specified file system.

Pre-z/OSV 1.8, the format of the NFS Client specification is restricted to lists of single host names. Other platforms support multiple definition formats:

- Single host

  This is the most common format. You may specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

- Netgroups

  NIS netgroups may be given as @group. Only the host part of each netgroup member is considered in checking for membership. Empty host parts or those containing a single dash (-) are ignored.

- Wildcards

  Machine names may contain the wildcard characters of an asterisk (*) and a question mark (?). These can be used to make the exports file more compact. For instance, '*.cs.foo.edu' matches all hosts in the domain 'cs.foo.edu'. However, these wildcard characters do not match the dots in a domain name, so the above pattern does not include hosts such as 'a.b.cs.foo.edu'.

- IP networks

  You can also exports directories to all hosts on an IP (sub-) network simultaneously. This is done by specifying an IP address and netmask pair as address/netmask where the netmask can be specified in dotted-decimal format, or as a contiguous mask length (for example, either '/255.255.252.0' or '/22' appended to the network base address result in identical subnetworks with 10 bits of host).

- To extend z/OS NFS usability and compatibility with other platforms, this line item extends the z/OS NFS support to include the above specification formats with the following restriction: The z/OS platform does not support the NIS environment. z/OS UNIX does support the /etc/netgroup file. Therefore, z/OS NFS is restricted to only support use of the local /etc/netgroup file for definition of net groups. Otherwise, the design will follow UNIX netgroup concepts and other UNIX server's implementations of netgroup support.

  A new *root* keyword will be added to the syntax to support the root authorization specification cited above.

NFS server checklist function enhancements

The NFS Server also supports the use of a checklist data set with which MVS data sets can be specified for which SAF authorization checking is to be bypassed, implying that all NFS users have read/write access to the data set, regardless of what is specified in SAF.

The checklist data set also uses NFS Client specifications, like the exports file, and has the same current specification restrictions. The client specification enhancements being implemented for the exports file would also be of value for the checklist file.

Further, pre-z/OS 1.8, the checklist file can only be refreshed by restarting the NFS Server. This is a severe restriction on the 7x24 high-availability environments becoming common on client configurations. The checklist support should be enhanced to provide refresh capability without it being necessary to restart the NFS Server.

Client configurations are moving to environments where the system administrator is managing the system from designated remote hosts. These administrators manage the systems via special root-level authorizations. Currently, the z/OS NFS server does not allow root-level access via NFS. However, with these remote administration environments, it would be very helpful to the system administrators to allow specific hosts to be identified as having root-level authority to some file systems.

In order to accommodate the above checklist requirements and provide common extensibility to the checklist and exports file data sets in the future, this line item will merge the checklist and exports file data sets.

Restriction: Not all functions of the export list are supported for the checklist.

Dynamic refresh of EXPORTS file information

The z/OS NFS server's exports file information can be refreshed from the exports data set with the EXPORTFS keyword on the **`Modify mvsnfs`** operator command. This line item does not change this function. However, with the merge of the checklist function into the exports file, the checklist information can now also be dynamically refreshed without restarting the NFS Server.

This line item does not change the exports file refresh mechanism. Therefore, the refreshed values will not affect directories that are already mounted. It will be necessary to unmount and remount the directories for changed values to take effect.

SHOWMOUNT command

The SHOWMOUNT command with operand -e displays the list of exported directories. With this line Item, the command will support displaying directory names without their suffixes (because directory suffixes correspond to checklist data). It will also support displaying expanded format host names (that is, single host name, netgroup name, single IP address, IP network, wildcards) with their suffixes (client suffixes). Host specifications will be shown as they are in the EXPORTS data set.

7.7.2 NFS Server V4 Protocol support continuation

The Network File System is a distributed file system that provides transparent processing capability for data and information about worldwide and heterogeneous networks. z/OS NFS provides the implementation that allows the z/OS platform to participate in these networks. A new industry-wide version of the communication protocol, V4, has been defined. The project described by this functional specification is intended to extend the z/OS NFS support to incorporate the new protocol. In z/OS V1R7 most of the NFSV4 protocol functions were implemented in the NFS server. However, some functions had to be deferred.

Note: The UTF-8 support was done in 1.7. Only stringprep is being added in 1.8.

NFS V2 and V3 protocols are limited in the character encoding of strings. In the NFS protocols, strings are used for file and directory names, and symbolic link contents. Although the XDR definition [RFC1832] limits strings in the NFS protocol to 7-bit US-ASCII, common usage is to encode file names in 8-bit ISO-Latin-1. However, there is no mechanism available to tag XDR character strings to indicate the character encoding used by the client or server. Obviously, this limits NFS's usefulness in an environment with clients that may operate with various character sets.

The primary issue in which NFS V4 needs to deal with internationalization is with respect to file names and other strings as used within the protocol. The choice of string representation must allow reasonable name/string access to clients that use various languages. The UTF-8 encoding of the UCS as defined by [ISO10646] allows for this type of access and follows the policy described in "IETF Policy on Character Sets and Languages" [RFC2277].

[RFC3454-Preparation of Internationalized Strings ("stringprep")], otherwise known as "stringprep", documents a framework for using Unicode/UTF-8 in networking protocols, so as "to increase the likelihood that string input and string comparison work in ways that make sense for typical users throughout the world."

UTF-8 encoding and stringprep introduced in the NFS V4 protocol will be fully supported by the z/OS NFS server. Refer to NFS V4 protocol RFC3530 for additional information about UTF-8 and stringprep usage.

NFS conversion of UTF-8 character strings requires adding support of UTF-8 [CCSID 1208] conversion tables to the z/OS Unicode Services.

RPCDIND-base registration

In order for NFS Client systems to access the NFS Server, the server must register its supported program versions with TCPIP. For the IPv4 Internet protocol, NFS Clients query the NFS Server presence via the Portmapper service (Version 2 of the port mapping protocol). However, for the expanded IPv6 Internet protocol, this query is done via the RPCBIND service (using Version 3 or 4 of the port mapping protocol). The RPCBIND service is being implemented by z/OS Communication Services in z/OS V1R8.

Prior to NFS V1.8, the z/OS NFS Server has registered with TCP/IP via the portmapper service. This will not be adequate for the NFS Clients to determine whether the NFS Server supports IPv6 communications. Therefore, it will be necessary to modify the z/OS NFS Server registration function to register via RPCBIND once the RPCBIND service is available on the system.

To avoid a total dependency on the existence of RPCBIND, the z/OS NFS Server must be modified to first query the presence of the RPCBIND service. If it is available, then that will be used for server registration. Otherwise, the existing portmapper service will be used instead to allow continued use of existing functionality.

7.7.3 Serviceability improvements

In this section we discuss serviceability improvements.

Convert NFS client external trace file to CTrace

Prior to NFS V1.8, the z/OS NFS Client stores messages and diagnostic debug information in the z/OS NFS client log data sets specified in the NFSCMSG1 and NFSCMSG2 DD statements of the client's startup procedure.

NFSCMSG1 is associated with the primary log, while NFSCMSG2 is associated with the secondary log. When the client is started, the primary log is used first. When the primary log is full, the logging is automatically switched to the secondary log and a console message is displayed. The last log buffer is lost when the switch takes place. When the secondary log is also full, the logging is switched back to the primary log and the original primary log content is overwritten.

In this release the NFS client will now add support to record diagnostic information in a z/OS component trace buffer. Depending on which debug controls are turned on will determine whether the diagnostics are written to the existing log data sets or to the new trace buffer. If both are turned on, it will be recorded to both. For performance reasons, we highly recommend that the component trace buffer be used, not the log data sets. Error, warning, and informational messages will continue to be recorded to the log data sets. They will also be written to the component trace if selected.

List mounts operator command host name support

The z/OS NFS "List=Mounts" operand of the MODIFY operator command displays all of the mount points that are currently active on the z/OS NFS Server. However, it does not give any information about which mount points are being accessed by which NFS Clients. Therefore, If a mount point problem occurs, the client cannot determine which clients are using that mount point.

This line item will expand the console output from this operator command to follow each mount point with the list of clients that currently have active mounts to this mount point. If available, the client's host name will be displayed. If not, then the IP address will be given. The number of clients in the list will be less than or equal to the number reported on the ACTIVE parameter of the GFSA910I, or GFSA911I, message, which displays the mount point. The list of clients may be less than the active count because a client can have more than one active mount to a given mount point.

NFS Client stop PFS operator command support

Currently, the z/OS NFS client can be stopped by the **`p mvsnfsc`** operator command. However, if the client has active mount points at the time, use of this operator command can cause the user's shells to terminate abnormally. This may occur because the NFS client cannot inform z/OS UNIX LFS that the PFS (that is, NFS) is terminating, and LFS may subsequently try to call one of the NFS client functions, which will then no longer be present.

In z/OS V1R8, z/OS UNIX is implementing a new operator command **`MODIFY OMVS STOPPFS=pfsname`** (that is, pfsname = "NFS" for the NFS Client). This new operator command allows z/OS UNIX LFS to control the shutdown of PFSs. This z/OS NFS line item implements the NFS Client changes necessary to support this new function.

With the implementation of this improved function for stopping PFSs, like the NFS Client, z/OS NFS client support for the **`STOP mvsnfsc`** operator command will be removed.

Note that the new operator command is a new z/OS UNIX operator command, not a new NFS Client operator command. Availability of the new operator command is dependent on its implementation by z/OS UNIX.

8

DFSMS futures

In this chapter we introduce features and functions of DFSMS V1R8.

The chapter covers the DFSMS components in the following sections:

- DFSMSdfp - see “DFSMSdfp V1R8 preview” on page 288.
- DFSMSdss - see “DFSMSdss V1R8 preview” on page 297.
- DFSMShsm - see “DFSMShsm V1R8 preview” on page 298.
- DFSMSrmm - see “DFSMSrmm V1R8 preview” on page 308.
- OAM - see “OAM V1R8 preview” on page 319.
- NFS - see “z/OS DFSMSnfs V1R8 preview” on page 323.

8.1 DFSMSdfp V1R8 preview

In this section we cover the items that are changed in the DFSMSdfp V1R8 components. The following is a summary of these enhancements:

- AMS LISTCAT performance improvement - see “AMS LISTCAT performance improvement” on page 288.
- Catalog Reliability and Service (RAS) - large page data set support - see “Catalog Reliability and Service (RAS) - large page data set support” on page 289.
- Catalog Reliability and Service - dynamic service task count - see “Catalog Reliability and Service - dynamic service task count” on page 289.
- SMS volume selection performance improvement - see “SMS volume selection performance improvement” on page 290.
- SMS ease of use - copy SCDS as ACDS - see “SMS ease of use - copy SCDS as ACDS” on page 291.
- SMS serviceability - TRACE DEBUG parameter - see “SMS serviceability - TRACE DEBUG parameter” on page 292.
- VSAM serviceability - dump on return/reason code and automatic dump on certain errors - see “VSAM serviceability - dump management” on page 292.
- VSAM RLS - first time data capture - see “VSAM RLS - first time data capture” on page 293.
- VSAM RLS - system error recovery - see “VSAM RLS - system error recovery” on page 293.
- VSAM RLS - serviceability - command to identify hangs and latch contention - see “VSAM RLS - serviceability - latch contention” on page 294.
- PDSE scale ability - increase number of concurrently open PDSE data sets see - “PDSE scaleability - number of concurrently open PDSE data sets” on page 294.
- PDSE performance - retain buffers after a PDSE is closed - see “PDSE performance - retain buffers after a PDSE is closed” on page 295.
- DADSM performance - use DEVMAN to rebuild VTOC index - see “DADSM performance - use DEVMAN to rebuild VTOC index” on page 296.

8.1.1 AMS LISTCAT performance improvement

The LISTCAT command has been improved when used with the LEVEL parameter or when used to list ALL entries of a particular catalog. This has been done by using the generic filter locate (GFL) function.

Usage

The usage information is:

- Input

  There is no change to the syntax of the LISTCAT command.
- Output

  The output format from listing only one catalog will not change.

Migration

If the output from then LISTCAT command is processed after production, the revised order may require attention. Generally, there are no migration considerations.

Coexistence

There are no coexistence considerations.

8.1.2 Catalog Reliability and Service (RAS) - large page data set support

Prior to this enhancement, the size of a page data set was restricted to less than 4 GB.

Usage

The usage information is:

- Allocate a page data set larger than 4 GB.
- Requirements:
  – The data set must be SMS managed.
  – You must specify extended format.
  – You must specify extended addressability.
  – Request greater than 4 GB.

Migration

A page data set that is larger than 4 GB can be specified in the same way as any other page data set, via the PAGE ADD command, or by specification in SYS1.PARMLIB(IEASYSxx).

Coexistence

APAR OA10493 must be installed on lower-level systems to allow a page data set to be allocated anywhere on a volume. An earlier APAR OA07281 is not sufficient. This is not specific to large page data set support.

8.1.3 Catalog Reliability and Service - dynamic service task count

Prior to z/OS V1R8, the maximum number of catalog dynamic service tasks was 180. This may have delayed responses to catalog requests.

In z/OS V1R8 an optional method is provided by which a higher number of dynamic service tasks can be allowed — up to the maximum for dynamic services is 999. By using this you can avoid having catalog requests wait for an available task. This may improve performance for systems currently constrained by catalog performance.

The existing interface is retained if you do not need more than 180 dynamic service tasks. You can determine how many catalogs are open by using the MODIFY CATALOG,REPORT command.

Usage

Requirements:

- To define more than 200 dynamic service tasks, you must create or update SYS1.NUCLEUS(SYSCATxx) column 65 with a three-character EBCDIC value for the number of tasks to be defined.
- If SYSCATxx is implemented, no value (or a value less than 200 is specified), a value of 200 is used.

After IPL using SYSCATxx, you can use the command to verify the new number of tasks:

```
F CATALOG,REPORT
```

Migration

You must implement SYS1.NUCLEUS(SYSCATxx) to use the new function, but in its absence, the system will continue to use an the SYSCAT entry of the LOADxx member.

Coexistence

There are no coexistence considerations because the SYS1.NUCLEUS data set would not be used on a lower-level system.

8.1.4 SMS volume selection performance improvement

On very large systems with many candidate volumes, SMS volume selection may take an unacceptable length of time owing to the number of retries that occur when trying to find an optimal volume.

The SMS fast path volume selection performance enhancement provides a user-selectable approach to speed up volume selection.

Usage

Use the command to verify the new number of tasks:

```
SETSMS FAST_VOLSEL(ON)
```

Alternatively, specify FAST-VOLSEL(ON) in SYS1.PARMLIB(IGDSMSxx).

Use the command to disable fast volume selection:

```
SETSMS FAST_VOLSEL(OFF)
```

It is not necessary to specify FAST_VOLSEL(OFF) in SYS1.PARMLIB(IGDSMSxx), as OFF is the default.

The fast volume selection process uses a set of criteria to speed up the process of volume allocation by reducing the number of retries.

Definitions

The definitions are:

- Best fit allocations

  Initial allocations of space that could span volumes.

- Space constraint relief

  If initial volume selection fails, use the data class attributes of SCR to allocate the space.

- SMS volume statistics

  The SMS configuration maintains space information (total volume capacity, amount of free space available) for each SMS-managed volume.

  If free space on a volume changes, it is updated in the SMS configuration.

Implementation

For the implementation:

- For non-best-fit allocations, exclude volume from selection when requested space exceeds volume size.

  SMS will perform volume selection from the prioritized list until 100 volumes have been rejected by DADSM for insufficient space. When that occurs, SMS will exclude all volumes that do not have sufficient space based on the *volume statistics* in the SMS configuration. This *fast* volume selection approach may drastically reduce the number of candidate volumes, and thus the number of retries, while continuing to select volumes by volume preference ranking as in the normal volume selection approach.
- For both best-fit and non-best-fit allocations, limit the number of retries.

  For best-fit requests, SMS will call DADSM interactively until space requested has been obtained or the maximum number of volumes has been used and requested space has not been completely obtained. In this latter case, SMS will reject *only* the last volume that was used and will re-select another volume.
- Select volumes by the amount of free space in the second best-fit step.

  SMS will re-rank the volumes by the amount of free space in the second best-fit phase, which is also the last phase, of space constraint relief processing if the first best-fit phase has failed to allocate sufficient space.

Verification

Turn fast volume selection on. Look for message IGD17294I, as in Figure 8-1.

```
IGD17294I FAST VOLUME SELECTION IS USED TO ALLOCATE DATA SET dsn
```

Figure 8-1 IGD17294I FAST VOLUME SELECTION message

Migration

No considerations.

Coexistence

No considerations.

8.1.5 SMS ease of use - copy SCDS as ACDS

This enhancement allows users to create an ACDS from any valid SCDS without having to first activate the SCDS. The ability to create an ACDS in this way is to support SMS definitions for another system and for disaster recovery locations.

Usage

Use the **SETSMS COPYSCDS(scds_dsn,acds_dsn)** command.

Requirements:

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

System validation: SMS will verify that:

- The acds-dsn is not the current ACDS.
- The source SCDS scds_dsn is an SCDS.

Migration

No considerations.

Coexistence

No considerations.

Any ACDS created in this way would be usable on any same-level system, or on a lower-level system that had the necessary SMS CDS coexistence maintenance installed.

8.1.6 SMS serviceability - TRACE DEBUG parameter

A new SMS trace option will be provided to reduce the amount of trace data captured while recording key data to expedite problem resolution.

Key modules will be modified to provide trace points, which will be activated when the new SETSMS TRACE DEBUG option is used.

Usage

You must remove any existing traces in effect by using the following command:

```
SETSMS DESELECT(ALL)
```

Then use the following command:

```
SETSMS TRACE(ON) TYPE(ALL) SIZE(size) SELECT(DEBUG)
```

Where *size* sets a value for the size of the trace table (this is not new with the DEBUG parameter).

Migration

No considerations.

Coexistence

No considerations.

8.1.7 VSAM serviceability - dump management

A new operator command will allow the user to specify a return code/reason code on which to take a SVC dump. The new function will automatically take a dump on certain errors. These enhancements are intended to reduce the necessity for the complex process of setting a SLIP when data is needed for problem analysis. This command will usually be issued at the direction of VSAM support.

Usage

No action is required for the automatically generated dumps. The new command is:

```
MODIFY CATALOG,VDUMPON(rc,rsn,pdf,comp)
```

Where:

rc Specifies the VSAM return code in decimal format (one to three characters with a maximum of 255) or two asterisks (**).

rsn Specifies the VSAM reason code in decimal format (one to three characters with a maximum of 255) or **.

pdf Specifies the VSAM problem determination function code (one to three characters with a maximum of 255) or **.

comp Specifies the component code (0-5) or **.

You can omit specification of comp, pdf, or rsn (in that order) or specify them as ** to indicate no constraint on matches. You can specify three of the sub fields as ** (or omit them), but you must end up with one of the fields explicitly set. You can only specify one VDUMPON at a time. After a match occurs the information is cleared and no further dumps will be taken. The new command to verify the new number of tasks is:

```
MODIFY CATALOG,VDUMPOFF
```

Migration
No considerations.

Coexistence
No considerations.

8.1.8 VSAM RLS - first time data capture

Enhancements will document internal logic errors at the earliest point in time possible via dump and messages.

Usage
No user interaction required.

Migration
No considerations.

Coexistence
No considerations.

8.1.9 VSAM RLS - system error recovery

Enhancements provide for continuation of the goal of preventing damage to client data, but to provide for individual subcomponents to be shut down should a problem be detected. Until these enhancements were provided, in order to preserve integrity, a problem with any subcomponent would result in the whole SMSVSAM address space being terminated.

Usage
No user interaction required.

Migration
No considerations.

Coexistence
No considerations.

8.1.10 VSAM RLS - serviceability - latch contention

Threads can hold one or more latches and go on to contend for other resources managed by other types of serialization mechanisms like ENQ, record lock, and special lock. Threads can also be in contention for a latch, potentially waiting for it indefinitely. Latches tend to be a component in a VSAM RLS *hang*, whether it is a real or a perceived one.

This enhancement provides a new command to do a diagnostic display of any latch contention. With the display it should be easier to locate and address the actual problem rather than have to recycle the SYSVSAM address space.

Usage

This command will show where the contention is occurring:

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

Analysis and options

If the holder of a latch that is holding other threads is a cross-memory task from a client address space (like a CICS transaction or batch job) then that task or job can be assessed. Depending on that analysis it might be cancelled, or if it is more important, left to run and the incoming task can wait.

- If the holder of a latch is waiting for an ENQ that is held in another system, then analysis should shift to that other system, possibly issuing the DIAG command there.
- If the holder of a latch is waiting for I/O or any other system resource, analysis should shift to determining why that resource is not available or completion is taking so long.
- If the holder of a latch is an SMSVSAM thread, then the SMSVSAM address space may need to be restarted.

Migration

No considerations.

Coexistence

No considerations.

8.1.11 PDSE scaleability - number of concurrently open PDSE data sets

This enhancement exploits 64-bit addressable storage to increase the number of open PDSE data sets. Prior to this enhancement, or if the new option is not selected, the upper limit of concurrently open PDSE data sets is 1 million.

By changing the PDSE Buffer Management Facility (BMF) and moving the directory buffers into 64-bit addressable space, the number of PDSE data sets that can be open concurrently is significantly increased. The Index Management Facility (IMF) is changed to address the directory blocks in their new location.

Usage

By default, PDSE directory buffers will continue to reside in 31-bit addressable storage.

The commands **SETSMS PDSE_DIRECTORY_STORAGE(size)** and **SETSMS PDSE1_DIRECTORY_STORAGE(size)** are used to specify that 64-bit addressable storage is to be used, and how much, for the standard PDSE address space (PDSE) and the restartable address space (PDSE1). Where:

- (size) is optional. If used, it specifies the amount of storage to be used expressed as nnnM for Megabytes and nnnG for gigabytes.
- The default is 2 G.
- The maximum is 16 G.
- The minimum is 64 M.

The values may also be specified in the SYS1.PARMLIB(IGDSMSxx) member where the same syntax and considerations apply as for the SETSMS command.

The following command can be used to determine the effective settings:

```
D SYS,SMSVSAM
```

Migration

No considerations.

Coexistence

No considerations.

8.1.12 PDSE performance - retain buffers after a PDSE is closed

When a PDSE is closed, the buffers allocated to the PDSE are purged. For data sets that are repeatedly opened and closed, they may lose some of the benefits of caching buffers.

With this optional enhancement, buffers can be retained after the PDSE is closed. The buffers will still be subject to the buffer management algorithms in effect, so if the PDSE is not reopened, in due course the buffer will be purged.

Usage

By default the buffers continue to be purged when the PDSE is closed. Use the following command to specify that buffers are to be retained after a PDSE is closed for the standard PDSE address of the restartable address space:

```
SETSMS PDSE_BUFFER_BEYOND_CLOSE(YES)
SETSMS PDSE1_BUFFER_BEYOND_CLOSE(YES)
```

This function applies to all PDSE data sets, but can be specified selectively for the two address spaces.

The values may also be specified in the SYS1.PARMLIB(IGDSMSxx) member where the same syntax and considerations apply as for the SETSMS command.

The following commands can be used to turn the function off. The command to turn the function off would not normally be specified in the IGDSMSxx member, as the function is off by default.

```
SETSMS PDSE_BUFFER_BEYOND_CLOSE(NO)
SETSMS PDSE1_BUFFER_BEYOND_CLOSE(NO)
```

The following command can be used to determine the effective settings:

```
D SYS,SMSVSAM
```

Migration

No considerations.

Coexistence

No considerations.

8.1.13 DADSM performance - use DEVMAN to rebuild VTOC index

When a VTOC index is disabled, the DSCB chain of format 5 and 7 (free space) has to be rebuilt. This requires each DSCB on the volume to be read. Prior to this enhancement, each DSCB has to be read as one individual operation.

With this enhancement, which is implemented in a new convert routine in the DEVMAN address space, the whole DSCB chain is read into a data space, which will require very few start subchannel requests. This will allow the new VTOC structure to be built more quickly and with fewer I/O operations. The original DADSM code is retained so that if the DEVMAN address space is unavailable, the DSCB chain rebuild can still take place.

Usage

If a problem with the VTOC index occurs, the new function will automatically rebuild the DSCB chain using DEVMAN if it is available.

If a user wants to deliberately disable the VTOC index or rebuild it, 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 if available. ICKDSF R17 is common to prior systems where, if the APAR is installed, it will not find DEVMAN or may find DEVMAN but not find the convert routine, in which case the original DADSM routines will be used.

Example

In Figure 8-2 we show sample JCL to rebuild the indexed VTOC on volume 339131.

```
// job card....
//*************************************************************************
//* THIS STEP INITIALIZES A VOLUME WITH A NEW INDEXED VTOC. IF ONE
//* ALREADY EXISTS IT IS DISABLED THEN A NEW ONE IS BUILT.
//CLEANVOL EXEC PGM=ICKDSF,PARM=NOREPLYU
//SYSPRINT DD SYSOUT=*
//DISK DD UNIT=3390,VOL=SER=339131
//SYSIN DD *
 BUILDIX DDNAME(DISK) OS
 BUILDIX DDNAME(DISK) IX
/*
```

Figure 8-2 ICKDSF INDEXED VTOC BUILD

Message IEC604I is 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 without using DEVMAN appearing on SYSLOG would be as shown in Figure 8-3.

```
14.33.22 SYSTEM1 JOB00016 IEC604I VTOC CONVERT ROUTINE ENTERED ON
0E40,339131,DOS
```

Figure 8-3 IEC604I VTOC convert message not using DEVMAN

A typical example of message IEC604I while using DEVMAN appearing on SYSLOG would be as shown in Figure 8-4.

```
14.34.27 SYSTEM1 JOB00016 IEC604I VTOC CONVERT ROUTINE ENTERED ON
0E40,339131,DOS,DEVMAN
```

Figure 8-4 IEC604I VTOC convert message using DEVMAN

Migration

If ICKDSF is to be used, and the expectation is that the new convert interface is to be called, ICKDSF must have maintenance for APAR PK19625 installed. Without PK19625, ICKDSF will still manage the index rebuild functions, but will not attempt to check whether DEVMAN is available.

Coexistence

No considerations.

8.2 DFSMSdss V1R8 preview

This section describes the enhancement to DFSMSdss in z/OS V1R8 to allow physical data set copies.

8.2.1 Physical data set copy

Currently, the DFSMSdss COPY command can only create logical copies of data sets. DFSMSdss will be enhanced in z/OS V1R8 to physically copy a single volume data set, or the parts of a multi-volume data set, from one DASD volume to another DASD volume. This enhancement will be added to enable DFSMShsm fast replication data set recovery, which is described in 8.3.6, “Fast replication data set recover” on page 305.

Usage

DFSMSdss will perform a physical data set copy if either of the new PHYSINDDNAME or PHYSINDYNAM keywords are specified on the COPY DATASET command. PHYSINDDNAME specifies the name of a DD statement that identifies the volume containing the data set to be copied, and PHYSINDYNAM specifies the input volume to be dynamically allocated. Only one input volume may be specified using these keywords.

The output volume can be specified using either the OUTDDNAME or OUTDYNAM keywords, and must have the same device type as the input volume.

A DFSMSdss conditioned volume created with the DFSMSdss COPY FULL DUMPCONDITIONING command can be specified as a source volume. A non-conditioned volume can also be specified as the source volume. However, the target volume cannot be a conditioned volume.

Figure 8-5 shows a JCL example of a job that creates a physical copy of a data set.

```
//MHLRES2C JOB NOTIFY=&SYSUID,CLASS=A,MSGCLASS=T
//COPY     EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//IN       DD VOL=SER=BH5ST1,UNIT=3390,DISP=OLD
//OUT      DD VOL=SER=SBOX92,UNIT=3390,DISP=OLD
//SYSIN    DD *
  COPY DATASET(INCLUDE(MHLRES2.TEST.DS)) -
       PHYSINDDNAME(IN) -
       OUTDDNAME(OUT)
/*
```

Figure 8-5 DFSMSdss physical data set copy

All the parts of a data set on a volume will be copied by this operation. Multi-volume data sets must be copied one volume at a time, and then recataloged when the copies are complete.

Migration considerations

There are no migration considerations for this enhancement.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.3 DFSMShsm V1R8 preview

This section describes the enhancements to DFSMShsm in z/OS V1R8.

8.3.1 Error handling on alternate duplex tapes

The duplex tape option creates two identical backup or migration tapes in parallel. One tape is designated as the original and the other as the alternate. The intent is that the original tape be kept on site, while the alternate can either be created in a remote tape library or taken off site.

Currently, when an error is detected on an alternate tape during tape duplexing, the alternate tape is dismounted and discarded while DFSMShsm continues to write to the original tape. When DFSMShsm has finished writing to the original tape, a tape copy is scheduled to create a new alternate tape. With high-capacity tapes, this can result in tapes being without a valid duplex copy for a long period of time.

This enhancement will provide the option to mark the original and alternate tapes as full when an error occurs writing an alternate migration tape. Both tapes will be dismounted and the migration operation will be retried on a new set tapes, ensuring that the original and alternate tapes are always in sync. This will apply during the following operations:

- Data set migration
- Volume migration
- Primary space management
- Interval migration
- Migration tape recycle

Usage

The new ERRORALTERNATE option on the SETSYS DUPLEX command can be used to specify the action to be performed when an error occurs writing to an alternate tape. The syntax of the command is as follows:

```
SETSYS DUPLEX(MIGRATION(Y ERRORALTERNATE(CONTINUE | MARKFULL)))
```

CONTINUE is the default option, and specifies that errors that occur on alternate tapes will affect duplexing as they currently do. The alternate tape will be dismounted and discarded, and DFSMShsm will continue to write to the original tape. When DFSMShsm has finished writing to the original tape, it will mark the tape full and schedule a tape copy to create a new alternate tape.

MARKFULL causes DFSMShsm to treat an error on the alternate tape as it would an error on the original tape. This will be one of the following possible actions, depending on the error:

- Both the original and alternate tape volumes are dismounted and marked as full, and the operation is retried on a new set of tapes.
- Both the original and alternate tape volumes are dismounted and the operation will be failed with the corresponding return code.

The current value of the ERRORALTERNATE parameter will be displayed by the QUERY SETSYS command, as shown in Figure 8-6.

```
ARC0442I TAPE OUTPUT PROMPT FOR TAPECOPY=NO, DUPLEX 741
ARC0442I (CONT.) BACKUP TAPES=YES, DUPLEX MIGRATION TAPES=(Y,
ARC0442I (CONT.) ERRORALTERNATE=CONTINUE)
```

Figure 8-6 ERRORALTERNATE value in the QUERY SETSYS command output

Migration considerations

When ERRORALTERNATE(MARKFULL) is specified in the SETSYS DUPLEX command, migration processing will ignore the performance improvements introduced by APAR OA09928, which prevents SYNCDEV being issued as each data set is written to an alternate volume, and APAR OW43224, which prevents SYNCDEV being issued for the alternate tape during recycle processing.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.3.2 Recycle performance improvement

Currently, SYNCDEV is performed after each data set is written to tape during recycle processing. This enhancement will improve the performance of recycle processing by causing SYNCDEV to be issued at specific intervals, depending on the number of data sets and the amount of data that has been written, rather than after every data set.

Usage

This enhancement requires no user intervention.

Migration considerations

APAR OW43224, which allows SYNCDEV to be bypassed for an alternate tape during recycle processing, will continue to be honoured. The exception to this is when a migration tape is being recycled and the ERRORALTERNATE(MARKFULL) option is specified on the SETSYS DUPLEX command.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.3.3 Non-VSAM data set migration performance improvement

When a non-VSAM data set is migrated from level 0 DASD, the data set is deleted from the level 0 volume immediately after it has been successfully migrated. The time taken to delete the data set from the level 0 volume can be a significant component of the total time taken to migrate the data set.

This enhancement will improve the performance of volume migration by creating a queue of scratch requests for non-VSAM data sets that are migrated from level 0. These scratch requests will be processed asynchronously from the other migration steps, after the data set has been successfully migrated. Note that this only applies when DFSMShsm is migrating a level 0 volume, and not an individual data set or a data set that has already been migrated to level 1.

Usage

This enhancement requires no user intervention.

Migration considerations

There are no migration considerations for this enhancement.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.3.4 Individual data set restore for ARECOVER processing

There is currently no way to restore an individual data set from the backup copy of an aggregate group. This enhancement will introduce the ability to recover one or more fully qualified data sets from an aggregate group.

Usage

A new parameter, ONLYDATASET, will be added to the ARECOVER command to allow one or more fully qualified data sets to be restored. The syntax of this new parameter is:

`ONLYDATASET(NAME(`*`dsname`*`) | LISTOFNAMES(`*`listdsname`*`))`

The NAME subparameter allows the single fully qualified data set specified to be recovered from the aggregate group backup.

The LISTOFNAMES subparameter can be used to specify a data set that contains a list of fully qualified data set names to be recovered from the aggregate group backup.

These options can also be specified on the aggregate group recover ISMF panels. Note that all data set name conflict processing applies to the data sets being recovered.

Migration considerations

There are no migration considerations for this enhancement.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.3.5 Fast replication backup to tape

DFSMShsm fast replication backup allows the volumes in a pool storage group to be copied using fast replication to another set of DASD volumes in a *copy pool backup* storage group. The SMS *copy pool* construct is used to define the set of pool storage groups that DFSMShsm can back up and recover using fast replication.

Currently, the only way to dump the volumes from a copy pool to tape is to perform the fast replication backup and then use a DFSMSdss job to dump the copy pool backup volumes to tape. This enhancement will provide the capability to dump the fast replication target volumes to tape during the fast replication process, or at a later time, using either the FRBACKUP command or DFSMShsm auto dump.

Usage

This section describes how to perform fast replication backup to tape and recover a backup copy that has been dumped to tape.

FRBACKUP

The FRBACKUP command will be enhanced with the options highlighted in Figure 8-7 to enable fast replication backup to tape.

```
>>-FRBACKUP-COPYPOOL(cpname)-----------------------------------------------------------><
                              |-EXECUTE--------------------------------- Fragment A -|
                              |         |-TOKEN(token)-| |-NOVTOCENQ-|               |
                              |                                                      |
                              |-PREPARE----------------------------------------------|
                              |-WITHDRAW---------------------------------------------|
                              |-DUMPONLY(-----------------------)- Fragment B -------|
                                         |-TOKEN(token)-------|
                                         |-VERSION(vernum)----|
                                         |-DATE(yyyy/mm/dd)---|
                                         |-GENERATION(gennum)-|

A: >>-------------------------------------------------------------------------------|
       |-FORCE-|  |-DUMP------------------------------------------- Fragment B -|
                          |-RETURNCONTROL(--DUMPEND-------------)-|
                                          |-FASTREPLICATIONEND-|

B: >>-------------------------------------------------------------------------------|
       |-DUMPCLASS(dclass1,...dclass5)-|
```

Figure 8-7 New FRBACKUP command syntax

The options are:

- DUMP

  This option will cause DFSMShsm to first create a DASD copy of the source volumes, and when all relationships have been successfully established, to start dumping the target DASD copies to tape.

  If the number of DASD backup versions specified in the copy pool construct is set to zero, the fast replication relationships are established without initiating a background copy, and the relationships are withdrawn as the volume dumps successfully complete, leaving only the dump copies. Otherwise, if the number of DASD backup versions in the copy pool is non zero, a background copy is initiated when the flash copy relationships are established, and the relationships are withdrawn as the DASD copies finish.

The RETURNCONTROL subparameter specifies when DFSMShsm should return control for user commands where WAIT has been specified. The following values may be used:

- FASTREPLICATIONEND

  Control is returned just before dump processing starts.

- DUMPEND

  Control is returned after all the volumes have been dumped.

- DUMPONLY

  This option is used to dump the fast replication target volumes to tape when a backup copy already exists on the fast replication target volumes. The VERSION, DATE, TOKEN, and GENERATION subparameters are mutually exclusive keywords to specify which DASD version should be dumped. No fast replication relationships will be established when this option is specified.

- FORCE

  This keyword is required if the creation of a new DASD backup version will cause the roll off of a backup version that has only been partially dumped to tape.

- DUMPCLASS

  This option allows between one and five dump classes to be specified. One dump copy of each volume is produced for each dump class. When this option is specified on the FRBACKUP command, any dump classes associated with the copy pool are ignored.

Refer to *DFSMShsm Storage Administration Reference,* SC26-0422, for more information about the current FRBACKUP keyword definitions.

Dump class definitions

The DFSMShsm dump class definition has a new FASTREPLICATIONDUMP option with the following syntax:

```
DEFINE DUMPCLASS(FASTREPLICATIONDUMP(REQUIRED | OPTIONAL))
```

Copy pools with dump classes that are defined with FASTREPLICATIONDUMP(REQUIRED) must have all required dump classes successfully completed in order for the dump version to be considered successful. Dump classes defined as FASTREPLICATIONDUMP(OPTIONAL) do not have to complete successfully in order for the dump version to be considered not to have failed. If an optional dump class fails, the dump version will be marked as *required complete*.

If the FASTREPLICATIONDUMP option is not specified on the dump class definition, the dump class will default to being a required dump class.

The LIST DUMPCLASS command can be used to display the value of this parameter for a particular dump class. Figure 8-8 shows the output of this command for a dump class that has FASTREPLICATIONDUMP(REQUIRED) specified.

```
 CLASS=MHLRESA   UNIT=3590-1   REUSE=Y  RESTORE=N  RESET=N  DISABLE=N  REQD=Y
AVAILABLE=N
 DAY=**  FREQ=007  RETPD=000007  TAPEEXPDT=*******  VTOCCOPIES=001  STACK=01
 DISP=FR REQUIRED DUMP
 HWCOMP=NO   ENCRYPT=NONE   ENCRYPT_TYPE=*********  ICOUNT=*****
 ENC_KEY=****************************************************************
 ARC0140I LIST COMPLETED,        5 LINE(S) OF DATA OUTPUT
```

Figure 8-8   LIST DUMPCLASS output

Auto dump

A copy pool can be dumped by DFSMShsm auto dump if the copy pool construct has auto dump enabled. If a generation zero DASD copy exists in a copy pool that has not been previously successfully dumped, it will be processed by auto dump. The dump classes specified in the copy pool construct will be used by auto dump.

The auto dump settings for the source storage groups in the copy pool will not affect auto dump processing of the copy pool. However, we recommend that the source storage groups all have auto dump disabled unless there is a specific need to dump the volumes both at the storage group and copy pool level.

FRRECOV

The FRRECOV command will be enhanced with the options highlighted in Figure 8-9 to allow backup versions that have been dumped to tape to be restored.

```
>>-FRRECOV---------------------------------------------------------------->
          |-TOVOLUME(volser)-------------------------------------------|
          |                  |-FROMCOPYPOOL(cpname)-| |- Fragment A -| |
          |                                           |- Fragment B -| |
          |                                                            |
          |-COPYPOOL(cpname)-------------------------------------------|
                             |-VERIFY(--Y--)-|
                                      |-N-|

>-----------------------------------------------------------------------><
   |-GENERATION(gennum)-|
   |-VERSION(vernum)----|
   |-DATE(yyyy/mm/dd)---|
   |-TOKEN(token)-------|

A: >>-------------------------------------------------------------------|
       |-FROMDUMP(------------------------------------------)-|
                   |-DUMPCLASS(dclass)-| |-APPLYINCREMENTAL-|
                   |-DUMPVOLUME(dvol)--|

B: >>-------------------------------------------------------------------|
       |-FROMDASD-|
```

Figure 8-9   New FRRECOV command syntax for restoring dumped backup copies

Restriction: FRRECOV will not allow an entire copy pool to be recovered when restoring from a backup that has been dumped to tape. Only single volumes can be recovered from the backup in this situation.

The FROMDUMP option indicates that volume recovery should use a dump tape as input. If more than one dump class was used for the dump, then the DUMPCLASS subparameter can be used to restrict the input to a particular dump class. The DUMPVOLUME subparameter can be used to restrict the input to a particular dump copy. For performance reasons, FROMDUMP cannot be used with the COPYPOOL keyword.

The FROMDASD option indicates that only DASD backup copies should be considered as input to the recovery process.

FRRECOV will recover the most recent DASD backup version available unless it is given a more specific source. If a generation zero copy exists both on DASD and tape, the DASD copy will be recovered. If the FROMDUMP keyword is used without any further specification of which dump to recover from, recovery of the generation zero copy will be attempted. The command will fail if there is no dump of generation zero to recover.

Refer to *DFSMShsm Storage Administration Reference,* SC26-0422, for more information about the current FRRECOV keyword definitions.

Deleting backup copies

Fast replication dump copies can be deleted in any of the following ways:

- Excess dump generations will be rolled-off after 85 generations have been accumulated.
- Dump volumes can be explicitly deleted using the DELVOL command. The new COPYPOOLCOPY keyword will need to be specified as follows:

  `DELVOL volser COPYPOOLCOPY`

  The backup version that the deleted tape was part of will be marked as a partial success after this command is executed so that dump processing for that version can be resumed at a later time if the DASD copy is still valid. Any remaining dump tapes for the version will still be available for recovery processing.
- Copy pool backup versions can be deleted using the FRDELETE command. Two new parameters, DUMPONLY and DASDONLY, can be used to specify that either only dump copies or only DASD copies should be deleted. If none of these options is specified, the default action is to delete both DASD and dump copies of the version specified. The syntax of the command is as follows:

  `FRDELETE COPYPOOL(cpname) VERSIONS(bvn) [DUMPONLY | DASDONLY]`
- Dump class copies of copy pool dump volumes will be expired as part of the dump expiration phase of auto dump processing.

Migration considerations

Once a copy pool created by a lower-level system is processed by a z/OS V1R8 system, it will be converted to a z/OS V1R8 copy pool. Lower-level systems will not be able to delete, withdraw, create, or dump a z/OS V1R8 copy pool. However, lower-level systems will be able to recover DASD backup copies from a z/OS V1R8 copy pool.

Coexistence considerations

Toleration PTFs should be applied to all systems in the HSMplex that are not at z/OS V1R8, as some CDS records will be expanded by this enhancement.

8.3.6 Fast replication data set recover

Currently, FRRECOV can only recover fast replication backup copies at the volume level. Recovering individual data sets takes several steps. This enhancement allows individual data sets to be recovered by the FRRECOV command.

Usage

The FRRECOV command will be enhanced with the options highlighted in Figure 8-10 to allow individual data sets to be recovered from a backup copy.

```
>>-FRRECOV-----------------------------------------------------------><
          |                                |  |-GENERATION(gennum)-|
          |        |-,--|                  |  |-VERSION(vernum)----|
          |-DSNAME(dsname)-----------------|  |-DATE(yyyy/mm/dd)---|
          |                |- Fragment A -|    |-TOKEN(token)-------|

A: >>--------------------------------------REPLACE-------------------->
       |-FROMDASD-----------------------|
       |-FROMDUMP(---------------------)-|
                  |-DUMPCLASS(dclass)-|
                  |-DUMPVOLUME(dvol)--|

>--------------------------------------------------------------------->
    |-FASTREPLICATION(--PREFERRED-)-| |-FROMCOPYPOOL(cpname)-|
                      |-REQUIRED-|
                      |-NONE-----|

>---------------------------------------------------------------------|
    |-NOCOPYPOOLBACKUP(--RC8--)-|
                       |-RC4-|
```

Figure 8-10 New FRRECOV command syntax for restoring individual data sets

The keywords are:

- DSNAME

  This keyword is mutually exclusive with the TOVOLUME and COPYPOOL keywords. A comma-separated list of data sets to be recovered can be specified. The data set names may be fully or partially qualified. Generation zero will be recovered if no generation, version, date, token, or dump volume is specified.

- REPLACE

  This keyword is required to indicate that the data sets being restored will replace the current versions.

- FASTREPLICATION

  This optional keyword specifies whether the data sets should be recovered using fast replication. If this keyword is specified on the FRRECOV command, it overrides any value specified on the SETSYS FASTREPLICATION command. The value set by SETSYS FASTREPLICATION is used if this keyword is not specified.

- NOCOPYPOOLBACKUP

  When a partially qualified data set name is specified and a copy pool backup version cannot be found for a data set that matches the filtering criteria, the default return code is eight. When RC4 is specified with this keyword, a return code of four is returned instead.

All other errors will result in a return code of eight, regardless of the value of this option. This keyword is ignored for fully qualified data set names.

By default, data sets will be recovered from a DASD copy. The FROMDUMP command can be specified to allow backup versions that have been dumped to tape to be considered. The DUMPCLASS and DUMPVOLUME subparameters can be used to limit the tapes used as input to a specific dump class or dump copy.

The data sets being restored must be cataloged and reside on the same volumes that they were on at the time the backup copy was created. VSAM data sets are processed at the cluster level only. The following data sets cannot be recovered individually by FRRECOV:

- User catalogs
- VVDS
- VTOC index data sets
- VSAM key-range data sets
- Migrated data sets

DFSMShsm will not automatically retry a data set that fails recovery because it is unable to obtain serialization for the data set. A data set should not be in use by any application when it is being recovered.

Refer to *DFSMShsm Storage Administration Reference,* SC26-0422, for more information about the current FRRECOV keyword definitions.

Dump class definitions

The DFSMShsm dump class definition will contain a new parameter, FASTREPLICATIONRECOVER, with the following syntax:

```
DEFINE DUMPCLASS(FASTREPLICATIONRECOVER(AVAILABLEFORMOUNT(YES | NO)))
```

The AVAILABLEFORMOUNT subparameter must be specified whenever FASTREPLICATIONRECOVER is specified. This parameter specifies whether a dump class copy is immediately available to be mounted for recovery. If the dump class is immediately available for recovery then it does not need to be specified on the FRRECOV command in order to be used. A dump class that is not immediately available for recovery must be explicitly specified on the FRRECOV command in order to be used for recovery.

If the FASTREPLICATIONRECOVER parameter is not specified for a dump class, the dump class will default to not being immediately available for recover.

The LIST DUMPCLASS command can be used to verify whether a dumpclass has been defined as being immediately available for recovery. Figure 8-11 shows an example of the output from this command for a dump class that is immediately available for recovery.

```
 CLASS=MHLRESB   UNIT=3590-1   REUSE=Y  RESTORE=N  RESET=N  DISABLE=N  REQD=*
AVAILABLE=N
 DAY=**  FREQ=007  RETPD=000007  TAPEEXPDT=*******  VTOCCOPIES=001  STACK=01
 DISP=FR NOT AVAILABLE
 HWCOMP=NO   ENCRYPT=NONE   ENCRYPT_TYPE=*********  ICOUNT=*****
 ENC_KEY=*****************************************************************
 ARC0140I LIST COMPLETED,       5 LINE(S) OF DATA OUTPUT
```

Figure 8-11 LIST DUMPCLASS output

Migration considerations

Currently, data sets are usually recovered from tape. Therefore, the SETSYS MAXDSRECOVERTASKS command not only defines the maximum data sets recovery tasks, but also implies the maximum number of tape devices that can be used for data set recovery.

As this enhancement will allow data sets to be recovered from disk as well as from tape, the SETSYS MAXDSRECOVERTASKS command will be changed to specify the maximum number of data set recovery tasks for both DASD and tape. The maximum number of data set recovery tasks that require tape should be specified using the following new command:

```
SETSYS MAXDSTAPERECOVERYTASKS(numtasks)
```

If the maximum number of data set recovery tasks from tape is not specified, it defaults to the value provided for the maximum number of data set recovery tasks. For this reason, you will not have to make any changes to your existing SETSYS values if you do not plan to use this enhancement.

Coexistence considerations

No coexistence PTFs will be required for this enhancement. A z/OS V1R8 system will be able to recover individual data sets from any copy pool backup created on a z/OS V1R5 system or later.

8.3.7 Fast replication backup security

The RACF facility class names introduced to provide security for fast replication backup and recovery will be modified in z/OS V1R8 to ensure that proper security can be provided through RACF. Currently, the entire copy pool name cannot always be specified in the facility class since the maximum length of a facility class is 39 characters, and the copy pool name alone can be up to 30 characters long. This enhancement will allow the entire copy pool name to be specified in the facility class.

Usage

In order to allow entire copy pool names to be specified in the RACF facility classes, this enhancement will introduce two changes. The first is that the maximum length of copy pool names will be decreased from 30 to 23 characters. The second change is that the static portion of the facility class names is being truncated by six characters. The new facility classes are shown in Table 8-1. The existing facility class names will remain usable for compatibility with pre z/OS V1R8 systems.

Table 8-1 Changed RACF facility classes

Current	**z/OS V1R8**
STGADMIN.ARC.FRBACKUP	STGADMIN.ARC.FB.*
STGADMIN.ARC.FRBACKUP.*cpname*	STGADMIN.ARC.FB.*cpname*
STGADMIN.ARC.FRRECOV	STGADMIN.ARC.FR.*
STGADMIN.ARC.FRRECOV.*cpname*	STGADMIN.ARC.FR.*cpname*
STGADMIN.ARC.FRDELETE	STGADMIN.ARC.FD.*
STGADMIN.ARC.FRDELETE.*cpname*	STGADMIN.ARC.FD.*cpname*

Migration considerations

Copy pools created on a pre-z/OS V1R8 system and with names longer than 23 characters can still be used on a z/OS V1R8 system. However, these copy pools must have their names changed to be no longer than 23 characters if you want to specify their complete names in a RACF facility profile. Copy pools created on a z/OS V1R8 system cannot have names with more than 23 characters.

Coexistence considerations

The current RACF facility profiles will need to remain defined until all systems in the sysplex have been migrated to z/OS V1R8. When all systems in the sysplex are at z/OS V1R8, the current RACF facility profiles can be completely replaced with the new facility classes.

8.4 DFSMSrmm V1R8 preview

In this section we cover the items that are changed in the DFSMSrmm V1R8 component. The following is a list of the line items:

- Enterprise level interface:
  - See 8.4.1, "EnterpriSe Level Interface - resource creation and update via the CIM" on page 308.
  - See 8.4.2, "Enterprise level interface - true e-mail address for RMM NOTIFY function" on page 311.
- See "Time and date management and presentation" on page 311.
- See "Tape data set authorization" on page 313.
- See "VRS policy management simplification" on page 314.
- Usability items
  - See 8.4.6, "Usability - new SEARCHOWNER command" on page 314.
  - See 8.4.7, "Usability - enhanced SEARCHVOLUME" on page 316.
  - See 8.4.8, "Usability - SELECT command support for all search results" on page 317.
  - See 8.4.9, "Usability - new columns for volume, data set, and VRS search results" on page 317.
  - See 8.4.10, "Usability - Rexx variable constraint relief" on page 318.
  - See 8.4.11, "Usability - ISPF and ISMF data set list utility support" on page 318.

8.4.1 EnterpriSe Level Interface - resource creation and update via the CIM

Clients require system management, open standards based, and with a common data model across platforms. Provide a remote instrumentation interface for system management, open standards based, and with a common data model across platforms. DFSMSrmm will use CIM/WEBM.

The Common Information Model (CIM) is a standard data model developed by a consortium of major hardware and software vendors (including IBM) called the Distributed Management Task Force (DMTF) as part of the Web Based Enterprise Management (WBEM) initiative.

WBEM includes a set of standards and technologies that provide management solutions for a distributed network environment. Interoperability is a major focus of WBEM, and using WBEM technologies can help you develop a single set of management applications for a diverse set of resources.

Usage

CIM operations are used by clients to interact with the CIM management data provided through CIM Servers.

The OpenPegasus CIM Server must be up and running in order to issue client commands against it. You access the CIM servers using a client such as wbemcli, available at http://sblim.sourceforge.net/clientapps.html as package sblim-wbemcli, invoked with appropriate operands.

In Figure 8-12 we show the RMM CIM classes.

```
RMM CIM classes
IBMrmm_PhysicalVolume
IBMrmm_LogicalVolume
IBMrmm_Dataset
IBMrmm_Owner
IBMrmm_Location
IBMrmm_ShelfLocation

IBMrmm_PhysicalLogicalVolume (association)
IBMrmm_LogicalVolumeDataset (association)
IBMrmm_LogicalVolumeOwner (association)
IBMrmm_PhysicalVolumeCurrentLocation (association)
IBMrmm_PhysicalVolumeCurrentShelfLocation (association)
IBMrmm_DatasetOwner (association)

IBMrmm_SearchOperands (aux class for search type operations)
IBMrmm_DeleteOperands (aux class for delete operation)
```

Figure 8-12 RMM CIM classes

In Table 8-2 we show the CIM classes supported for DFSMSrmm.

Table 8-2 Supported CIM classes

CIM class	CIM operations	
Main classes	EnumerateInstances EnumerateInstanceNames GetInstance CreateInstance ModifyInstance DeleteInstance	ei ein gi ci mi di
Association classes	Associators AssociatorNames References ReferenceNames	ai ain ri rin

In Table 8-3 we show typical tasks using wbemcli.

Table 8-3 Typical tasks using the CIM interface with RMM (part 1)

List all logical volumes, but limit them to 100 occurrences.	wbemcli mi http://<userid>:<password>@<cimom_url>:<port> /root/cimv2:IBMrmm_SearchOperands. Resource="IBMrmm_LogicalVolume“ Operands="Owner(*) Limit(100)" wbemcli ein http://<userid>:<password>@<cimom_url>:<port> / root/cimv2:IBMrmm_LogicalVolume
List all physical volumes, starting with "A".	wbemcli mi http://<userid>:<password>@<cimom_url>:<port> /root/cimv2:IBMrmm_SearchOperands. Resource="IBMrmm_PhysicalVolume“ Operands="Owner(*) Volume(A*)" wbemcli ein http://<userid>:<password>@<cimom_url>:<port> / root/cimv2:IBMrmm_PhysicalVolume

Where:

<userid> Your CIM client user ID
<password> Your CIM client password
<cimom_url> The IP address of the machine where the CIM Server is running
<port> The port or the CIM Server, typically 5988

In Table 8-4 we show further tasks that the CIM interface can be used on.

Table 8-4 Typical tasks using the CIM interface with RMM (part 2)

List all physical volumes in location "SHELF".	wbemcli ain -ac IBMrmm_PhysicalVolumeCurrentLocation http://<userid>:<password>@<cimom_url>:<port> /root/cimv2:IBMrmm_Location. CdsID="RMM_A", CreationClassName="", LocName="SHELF", Tag=""
List all data sets owned by “ADMIN”.	wbemcli ain -ac IBMrmm_DatasetOwner http://<userid>:<password>@<cimom_url>:<port> /root/cimv2:IBMrmm_Owner. CdsID="*", InstanceID="", OwnerId="ADMIN"

Hardware dependencies

This runs on any hardware that can exploit the Pegasus CIM server (currently z/OS V1R8 and Linux (RedHat, SuSE)).

Software dependencies

In order to exploit the CIM provider API for RMM, you must be running The OpenPegasus Common Information Model Object Manager (CIMOM) implementation, and be using a CIM browser/client or any equivalent product that supports the standards, as specified in the xmlCIM in support of the SNIA Storage Management Initiative - Specification (SMI-S). The CIM server can run either on a non-z/OS server or directly on z/OS V1R7 or later. The CIM client/browser can run on any platform supported by the provider of that client/application.

Migration and coexistence considerations

None.

8.4.2 Enterprise level interface - true e-mail address for RMM NOTIFY function

Prior to DFSMSrmm V1R8, DFSMSrmm used the TSO TRANSMIT command to communicate with end users. DFSMSrmm notification could not be sent to a true e-mail address.

With DFSMSrmm V1R8, RMM notification can be via TSO TRANSMIT or through a SMTP mail server.

The OWNER attributes are extended to include an e-mail address, and the TSO RMM SEARCHOWNER command has been added.

Usage

To receive e-mail notification, you must add `email address` to owner. Use the ADDOWNER or CHANGEOWNER command:

```
NOTIFY(YES)
VOLUME RELEASEACTION(NOTIFY)
```

Hardware dependencies

None.

Software dependencies

You must have an SMTP server configured and started. The default is for DFSMSrmm to try and use an SMTP server called SMTP on the current JES node.

Migration and coexistence considerations

None. There is no support on prior level systems.

8.4.3 Time and date management and presentation

Prior to z/OS V1R8, DFSMSrmm maintained all records in the CDS in local time.

With z/OS V1R8, DFSMSrmm allows the user to specify that all records are to be maintained in UTC (GMT) time by settings made with EDGUTIL UPDATE.

Independent of whether records are being maintained in UTC time, DFSMSrmm V1R8 provides services to present information, using z/OS time services, with time stamps in the common time or local time, or as is, based on settings made with EDGUTIL UPDATE.

Usage

The IBM recommendation, which is independent of DFSMSrmm, is that all systems run machines with the system time of day (TOD) set to UTC and use the CLOCK parameter to set appropriate offsets.

Once all systems are running on UTC, then DFSMSrmm can be updated by means of the EDGUTIL program to specify CONTROL UTC(YES).

TSO: You can specify the TZ operand on all ADD and CHANGE commands. The format of the TZ operand is:

```
TZ([offset]HH[:MM[:SS]])
```

Where:

offset +|- represents the offset direction. Specify a plus sign (+) to indicate that the offset is West of the zero meridian (UT). Specify a minus sign (-) to indicate that the offset is East of the zero meridian (UT). An offset is required.

HH[:MM[:SS]] This specifies the time offset from universal time (UT). An offset is required. An optional colon (:) separates hours from optional minutes and optional seconds. The value range of HH:MM:SS is from 00:00:00 to 15:00:00. MM and SS value range is 00 to 59.

Rexx variable:

- Use the SYSAUTH.EDGTZ rexx variable to request that dates and times are specified and returned in your chosen time zone.
- By default, DFSMSrmm returns dates and times in the system local time zone.
- Set SYSAUTH.EDGTZ to a valid time zone offset.

  Refer to the use of the TZ subcommand operand under TSO for acceptable values.
- You can set SYSAUTH.EDGTZ from an EXEC and avoid using the TZ operand to specify the offset. If specified, it takes priority.
- You can use SYSAUTH.EDGTZ regardless of the UTC setting.

Verification

ISPF:

- To see what time zone data will be displayed in, use DFSMS rmm Dialog Option 0.1 (User).
- To see data in your time zone when looking at a system running in a different time zone, use DFSMSrmm Dialog option 0.1 to select the option to change the time zone.
- Change the time zone by over typing the field contents.
- Alternatively, use the DFSMSrmm fast path command TZ.

Migration and coexistence

Toleration APARS OA13577, OA14136, and OA15873 are required on lower-level systems. These should be treated as minimum level APARs.

When toleration maintenance is installed:

- Toleration is provided for TZ SFI and the TZ subcommand operand to enable vendors to update their RMM-based applications and tools. The SYSAUTH.EDGTZ variable is ignored on lower-level systems.
- Toleration supports a mixed environment where some systems have toleration and the rest have z/OS V1R8 installed. Optionally, you can enable UTC(YES) when running coexistence.

On toleration systems and z/OS V1R8, when the TZ operand is specified on a subcommand and UTC is not enabled, RMM converts specified dates and times through common time to the running system local time.

DFSMSrmm report extract file records are still created using local times so reporting tools should not have to be updated.

Additional considerations:

- Running RMM commands using times from other systems/platforms

  You can add the TZ subcommand operand to these commands to indicate the time reference used.

- Using the results from the DFSMSrmm API

  The dates and times returned in structured fields are in local time.

  If you need to use these dates and times related to other date and time values that may be in local time you must use the TZ SFI value to make the adjustment before the date or time can be used.

- Using the API XML results

  Additional date and time formats can now be considered because the time zone offset is available.

  This is a change that occurs regardless of the DFSMSrmm UTC setting.

  Using the z/OS SET command with DATE or TIME keywords, or replying to message IEA888A to run the system on future or past dates, can affect the way DFSMSrmm calculates timed. To get correct results, alter the TOD clock and keep the time zone offset as before.

Daylight savings time considerations

We recommend that you QUIESCE DFSMSrmm when you make your daylight savings time changes on the system. This ensures that time and data fields in DFSMSrmm CDS records can be handled consistently.

When you switch *from* daylight savings time there are dates and times in the system that are repeated. DFSMSrmm journal records are time stamped with local time values, so you should either QUIESCE DFSMSrmm until there is no chance of repeated times, or accept that during forward recovery with journal records from this time change period you will get message EDG6429W. We recommend the latter approach.

8.4.4 Tape data set authorization

Prior to DFSMSrmm V1R8, there are limitations of RACF TVTOCS, overhead in using the TAPEVOL profiles, and no easy way to use the same DATASET CLASS profiles for tape data sets while fully protecting tape data sets.

In DFSMSrmm V1R8 increased tape data set flexibility is provided by:

- Providing tape data set authorization independent of the TAPEVOL class and TAPEDSN option enabling generic DATASET profiles to be used for both DASD and tape data sets.
- Allowing DFSMSrmm to use the ERASE field in RACF generic data set profiles to determine if tape data sets should be erased when deleted or the tape is returned to scratch status.

This support is provided by a combination of new DFSMSrmm and RACF support.

Usage

Details of the implementation of a new protection environment for tape data sets is beyond the scope of this preview, and since no changes come into effect until you start making changes, you should plan progression to the new process for after the cut over to DFSMSrmm V1R8 is completed.

Migration and coexistence

Changes are required to PARMLIB DEVSUPxx and in the RACF database, so care must be taken to check interaction at implementation time if these are shared with lower-level systems.

8.4.5 VRS policy management simplification

Prior to DFSMSrmm V1R8, setting up retention and movement policies can require a large number of VRS because each data set name mask also contains retention and movement information.

In DFSMSrmm V1R8, data set name masks can be separated from the policy specifications. This may reduce the number of VRSs required to manage retention and movement of volumes because some of those policies could be common to more than one data set name.

Usage

The new processes can only be used when using the DFSMSrmm VRSEL(NEW) environment. (This is not new with DFSMSrmm, so transition can be arranged in advance.)

You will need to review the existing VRS policies and determine which can be converted.

The keyword COUNT(0) on data set name VRSs and retention NAME VRSs make them into filter VRSs. Then you link the filter VRS to the policy VRSs.

Migration and coexistence

You must migrate the DFSMSrmm environment to VRSEL(NEW) before you will be able to use the DFSMSrmm V1R8 new functions. You can continue to run with VRESL(OLD) on DFSMSrmm V1R8, but you will get a warning message each time VRSEL is run, and the EDGHSKP processing ends with a RC4. Migration to VRSEL(NEW) is supported on prior level systems.

You will need to update any VRS that uses JOBNAME(OPEN) or JOBNAME(ABEND) if these are related to actual JOB names, because OPEN and ABEND are used internally by DFSMSrmm. Change these to JOBNAME(OPEN*) or JOBNAME(ABEND*).

Toleration for lower-level systems will be provided via APAR OA13355.

If you do not use any VRS with COUNT(0), and you do not use JOBNAME(ABEND/OPEN), then you can run VRSEL on either a toleration system or on DFSMSrmm V1R8.

8.4.6 Usability - new SEARCHOWNER command

Prior to DFSMSrmm V1R8 there was no SEARCHOWNER command.

DFSMSrmm V1R8 provides a SEARCHOWNER command. It was provided for use with the Enhanced Enterprise Level interface, but is available in the standard TSO environment.

Usage

There is no ISPF dialog support for SEARCHOWNER.

The TSO command is available for use with TSO. It was primarily provided for use with the CIM interface.

In Figure 8-13 we show the syntax of the new SEARCHOWNER command.

```
                                            |      10         |
>> SEARCHOWNER ---------------------|--LIMIT(|  search_limit--| --->
    SO                              |        |       ,        |
                                    |--END(end_owner)

                                                      | LIST   |
>-- CLIST (---prefix_string-- . -----suffix_string---) | NOLIST |

              -- * --------------------
>-- OWNER ( -- full_or_generic_owner ---)
```

Figure 8-13 SEACHOWNER syntax

Where:

- LIMIT(*search_limit*,*)

  Specifies to limit how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk (*) to list all entries matching your search criteria. The default value is 10.

- END(*end_owner*)

  Using the END operand as an alternative to the LIMIT operand allows you to specify both the starting and ending point of the owner search. END identifies the last entry to be returned. This parameter is mutually exclusive with the LIMIT parameter.

- CLIST(*prefix_string,suffix_string*)

  Specifies CLIST to create a data set of executable commands. You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). The CLIST that is allocated is called *prefix*.EXEC.RMM.CLIST unless you pre-allocate a data set to the RMMCLIST FILE/DD NAME.

- 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.

- OWNER(*full_or_generic_owner_name*)

  Specifies an owner name. A full owner name is one to eight characters. Specify an asterisk (*) to list owners regardless of name. An asterisk is the default for OWNER.

Migration

No considerations.

Compatibility

No considerations.

8.4.7 Usability - enhanced SEARCHVOLUME

New with DFSMSrmm V1R8 is enhancement of the SEARCHVOLUME command.

Usage

The usage is:

- ISPF - panels enhanced to allow for new operands
- TSO - SEARCHVOLUME command has the following new operands:
  - RELEASEACTION
  - JOBNAME
  - NOJOBNAME
  - LOCATION enhanced to add generic_location_name

Updates to the SEARCHVOLUME syntax are shown in Figure 8-14.

```
>>-- SEARCHVOLUME ------------------|---------------------------------|------>
      SR                            |                                 |
                                    |                    | *        | |
                                    |-- RELEASEACTION(--| ALL      |--)|
                                                         | ERASE    |
                                                         | INIT     |
                                                         | NOTIFY   |
                                                         | REPLACE  |
                                                         | RETURN   |
                                                         | SCRATCH  |
>---------------------------------------------------------------------------->

      |-- LOCATION(--| SHELF                   |--)
                     | LOCAL                   |
                     | DISTANT                 |
                     | REMOTE                  |
                     | library_name            |
                     | LOCDEF_location_name    |
                     | generic_location_name   |


>--------------------------------------------------------------------------->
      |-- JOBNAME(  | full_create_job_name    |       |
      |             | generic_create_job_name | --)   |
      |             | *                       |       |
      | NOJOBNAME ----------------------------------|
```

Figure 8-14 SEARCHVOLUME new and updated options

Where:

RELEASEACTION Specify to search for volumes that have the specified release action set. Release actions are those attributes that will be set as pending actions when a volume is released. Use the ACTION operand to

	search for volumes that have the specified pending action. The sub options on ACTION and RELEASEACTION are the same.
JOBNAME	Specify this operand to search for volumes that have been created by the specified job name. You may use a full or generic name. If you do not specify JOBNAME, job name is not used as a selection. If you specify JOBNAME(*), DFSMSrmm returns all volumes that match the search values and that have a creating job name. Volumes that do not have a creating job name are not listed.
NOJOBNAME	Specify to list volumes that have no creating job name.
LOCATION	In addition to the existing sub operands, you may use a *generic_location_name*.

Migration
No considerations.

Coexistence
No considerations - no support on lower-level systems.

8.4.8 Usability - SELECT command support for all search results

New with DFSMSrmm V1R8 is the SELECT primary command.

Usage
Once a results list has been obtained by using one of the SEARCH commands while using the ISPF interface, you can use the SELECT primary command to create a subset of entries to be processed. For example, in a volume list, SELECT * I in a volume list sets the I line command against each row of the results table.

Migration
No considerations.

Coexistence
No considerations - no support on lower-level systems.

8.4.9 Usability - new columns for volume, data set, and VRS search results

Updated with DFSMSrmm are the panels that display the results of volume, data set, and VRS search results.

Usage
No user action required. The information is displayed when the panels are used.

Migration
No considerations.

Coexistence
No considerations - no support for updated panels on lower-level systems.

8.4.10 Usability - Rexx variable constraint relief

Prior to DFSMSrmm V1R8, functions may not have completed due to running out of storage.

With z/OS V1R8, DFSMSrmm gets the benefit of the new TSO PROFILE attribute VARSTORAGE(HIGH), which allows variables to be stored above 16 M.

Usage

Issue the following command to enable this function:

```
TSO PROFILE VARSTORAGE(HIGH)
```

The default is VARSTORAGE(LOW).

Issue the following command to restore the profile to the same as prior to z/OS V1R8:

```
TSO PROFILE VARSTORAGE(LOW)
```

Migration

No considerations.

Coexistence

No considerations.

8.4.11 Usability - ISPF and ISMF data set list utility support

Prior to DFSMSrmm V1R8, ISPF and ISMF do not directly support tape data sets.

In z/OS V1R8, new support allows you to configure ISPF to enable DFSMSrmm to support a limited set of the available line commands for tape data sets.

Line commands supported:

- I - displays a search results list showing all data sets in the multi-volume 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 specified 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 the results list.
- M - displays a search results list showing all data sets on the same volume as the selected data set.
- D - releases the volume. If the volume is part of a multi-volume set there is the option to release all volumes in the set.

Usage

You must use the ISPF configuration utility to update the ISPF configuration table. Please refer to z/OS V1R8 ISPF documentation on the ISPF configuration utility.

To enable the new function, select the **Enable RM/TAPE Commands** option.

Migration

No considerations.

Coexistence

No considerations.

8.5 OAM V1R8 preview

This section describes the enhancements to OAM in z/OS V1R8.

8.5.1 Binary large object support

OAM binary large object support will allow objects larger than 32 KB to be stored using DB2's large object (LOB) support, and the binary large object (BLOB) data type. Currently, these large objects are stored in multiple rows in the 32-KB table. A 256-MB object, for example, will be stored in over 8000 rows. This enhancement will allow large objects to be stored directly in a DB2 BLOB field, rather than being stored over multiple rows, therefore reducing the number of SQL calls for storing and retrieving these types of objects.

The new LOB environment is intended for use by objects larger than 32 KB. Objects up to 32 KB in length will continue to be stored in the existing 4-KB and 32-KB tables.

Usage

The migration job CBRILOB in samplib should be modified and submitted to create the LOB storage structures required for OAM to exploit DB2's LOB support. Another migration job, CBRSMR18, should be modified and submitted to add a new column, ODLOBFL, to the DB2 object directory tables to indicate whether a DASD resident object resides in a LOB storage structure.

Binary large object support will be enabled using the new LOB parameter on the OAM1 statement of the IEFSSNxx parmlib member. The syntax is as follows:

```
LOB=A|P|N
```

A This value enables LOB support for all storage groups.

P This value enables LOB support at the storage group level. It will be possible for some storage groups to store large objects in the 32-KB table, while large objects in other storage groups will be directed to the LOB storage structure.

N This value disables LOB for all storage groups. This is the default.

A storage group must have a V_OSM_LOB_BASE_TBL view in order for it to be LOB enabled. This view must be created using the CBRILOB migration job for all storage groups if the LOB parameter is set to A, otherwise large object operations will fail. However, if the LOB parameter is set to P, storage groups without this view will continue to store their large objects in the 32-KB table. In this case, the V_OSM_LOB_BASE_TBL view must be created using the CBRILOB migration job for all storage groups for which you wish to have LOB enabled.

Once LOB is enabled for a storage group, all objects larger than 32 KB will be stored in the new LOB storage structure. Additionally, any existing object larger than 32 KB that is transitioned, recovered, or recalled to disk will be stored in the new LOB storage structure.

Migration considerations

DB2 Version 6.1 or later is required before OAM can use DB2's LOB support.

If an OAMplex is being used, and is sharing data across systems, all systems must be capable of LOB support before this feature is enabled on any system in the OAMplex.

Coexistence considerations

Coexistence PTFs will be provided to allow lower-level systems to detect and fail a request to access an object that is stored using DB2's LOB support. These PTFs should be installed regardless of whether an OAMplex is being used or whether the z/OS V1R8 system has LOB support enabled.

A migration job, CBRSMR18, will be provided to add a new column, ODLOBFL, to the DB2 object directory tables to indicate whether a DASD resident object resides in a LOB storage structure. This job should be modified and submitted on any systems where this column does not already exist.

DB2 large object support coexistence PTF

As of z/OS V1R8, OAM will be enhanced to use DB2's large object (LOB) support. z/OS V1R7 systems will not be able to use this support, but coexistence PTFs will be provided by APAR OA12683 for lower-level systems. If this PTF is installed on a pre-z/OS V1R8 system, migration job CBRSMR18, provided in SYS1.SAMPLIB, must be edited and submitted to add a new column, ODLOBFL, to the object directory to indicate whether a DASD resident object resides in a LOB storage structure.

During the installation process, CBRSMR18 should be submitted after creating the DB2 databases and tables, but before creating and binding the DB2 packages. If this job is not submitted, job CBRPBIND will fail with SQL code -206 as part of the installation process, and OSMC processing will fail once OAM is installed.

8.5.2 Tape recycle

The selection of tape volumes to be recycled using the MOVEVOL utility is currently a manual process. This enhancement will allow MOVEVOL to automatically select tapes eligible for recycling based on installation-specified thresholds and limits. The recycling of optical volumes will be unaffected by this enhancement.

Usage

This enhancement will add new parameters to the SETOAM command in the CBROAMxx parmlib member, and change the syntax of the command used to start a recycle task.

New SETOAM parameters

The following parameters will be added to the SETOAM command in the CBROAMxx parmlib member:

- MAXRECYCLETASKS(*nn*)

  This keyword can be specified at the global level to set the maximum number of MOVEVOL tasks that can be run concurrently by the recycle function. Valid values are between 0 and 15, and the default value is 1. A value of 0 indicates that no recycle operations can be run at the storage group or global level.

- SGMAXRECYCLETASKS(*nn*)

  This keyword can be specified at the storage group level to set the maximum number of MOVEVOL tasks that can be run concurrently by the recycle function for a particular storage group. Valid values are between 0 and 15, and the default value if 1. The value for this parameter cannot exceed the value specified for MAXRECYCLETASKS. A value of 0 indicates that no recycle operations can be run for the given storage group.

- SGMAXTAPERETRIEVETASKS(*nn*)

  This keyword can be specified at the storage group level only, and is identical to the current MAXTAPERETRIEVETASKS keyword when it is specified at the storage group level. This new keyword is the preferred naming convention. However, MAXTAPERETRIEVETASKS can still be used for historical compatibility.

  These optional keywords specify the maximum number of tasks that can concurrently read objects from tape volumes belonging to a specific object or object backup storage group. Valid values are between 1 and 100, and the default is 1. The value specified with these keywords cannot exceed the maximum number of tape retrieve tasks specified with the MAXTAPERETRIEVETASKS parameter at the global level.

- SGMAXTAPESTORETASKS(*nn*)

  This keyword can be specified at the storage group level only, and is identical to the current MAXTAPESTORETASKS keyword when it is specified at the storage group level. This new keyword is the preferred naming convention. However, MAXTAPESTORETASKS can still be used for historical compatibility.

  These optional keywords specify the maximum number of tasks that can concurrently write objects to tape volumes belonging to a specific object or object backup storage group. Valid values are between 1 and 100, and the default is 1. The value specified with these keywords cannot exceed the maximum number of tape store tasks specified with the MAXTAPESTORETASKS parameter at the global level.

- PERCENTVALID(*nnn*)

  This keyword can only be specified at the global level. It provides the global default percentage of the valid data threshold, which is used to determine whether a full tape volume is a candidate for recycle processing. Valid values are between 0 and 100, and the default is 0.

These parameters can be updated dynamically with the MODIFY OAM,UPDATE command, as shown in Table 8-5.

Table 8-5 Modify OAM commands for recycle processing

SETOAM parameter	MODIFY OAM command
MAXRECYCLETASKS	`F OAM,UPDATE,SETOAM,ALL,MAXRECYC,`*nn*
SGMAXRECYCLETASKS	`F OAM,UPDATE,SETOAM,`*strgrp*`,SGMAXREC,`*nn*
PERCENTVALID	`F OAM,UPDATE,SETOAM,ALL,PERCENTV,`*nn*

The current values of these parameters can be displayed at the global level, storage group level, or both using the MODIFY OAM,DISPLAY command, as follows:

```
F OAM,DISPLAY,SETOAM,[ALL | GLOBAL | storagegroup]
```

Recycle command changes

The syntax of the command used to start a recycle task will be changed as follows:

```
F OAM,START,RECYCLE,scope {,PV=xxx} [,LIM=yy | ,DISPLAY]
```

- scope

  This parameter indicates which set of volumes should be considered candidates for recycling. One of the following values may be provided:

  – An object or object backup storage group name

    Only tape volumes belonging to the storage group specified will be considered candidates for recycling.

  – ALLGRP

    All tape volumes belonging to a primary object storage group will be considered candidates for recycling.

  – ALLBKP

    All tape volumes belonging to an object backup storage group will be considered candidates for recycling.

  – ALLBK1

    All tape volumes belonging to a first backup storage group will be considered candidates for recycling.

  – ALLBK2

    All tape volumes belonging to a second backup storage group will be considered candidates for recycling.

- PV=*xxx*

  This optional keyword provides the valid data threshold to be used in determining if a volume is a candidate for recycling. Valid values for the threshold are between 0 and 100. Full tape volumes that have a percentage of valid data less than or equal to the specified value will be candidates for recycling. If this keyword is not specified, the percentage of valid data that will be used to determine whether a volume is eligible for recycling will be derived from the global default value defined with the PERCENTVALID parameter of the SETOAM command in the CBROAMxx PARMLIB member.

- LIM=*yy*

  This keyword indicates the maximum number of volumes that can be selected for recycling. Valid values are between 1 and 40. If this keyword is not specified, then DISPLAY must be specified.

- DISPLAY

  This keyword causes a list of volumes that meet the recycle criteria to be produced. This list is sorted by the percentage of valid data on each volume. No recycling of volumes will be performed when this keyword is specified. If this keyword is not specified, then LIM=*yy* must be specified.

The recycle process may be stopped at any time by issuing the following command:

```
F OAM,RECYCLE,STOP
```

Migration considerations

There are no migration considerations for this enhancement.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.5.3 Immediate backup copy

Each object stored by OAM will have a primary copy and one or two backup copies. The backup copies are usually created in the evening or on weekends when the object storage management cycle runs.

This enhancement will allow objects to be backed up when they are stored, rather than waiting for the next storage management cycle to perform the backup. The first backup copy of an object will be created immediately after the primary copy is stored. If a second backup is required, it will be created during the next storage management cycle. As the first backup will be created sooner, this enhancement will provide more assurance of the availability of the object being stored.

Usage

An object will be eligible for immediate backup copy if its management class has auto backup enabled and specifies a value of zero for the backup frequency. Objects that do not meet this criteria will continue to have their first backup copies created during the first storage management cycle after the primary copy is created.

Migration considerations

There are no migration considerations for this enhancement.

Coexistence considerations

There are no coexistence considerations for this enhancement.

8.6 z/OS DFSMSnfs V1R8 preview

In this section we cover the items that are changed in the DFSMSnfs V1R8:

Exports file net group support

This support is:

- NFS server checklist function enhancements
- Exports file format changes
- Dynamic refresh of EXPORTS file information
- SHOWMOUNT command

NFS Server V4 protocol support continuation

Continue implementation of the z/OS NFS Server portion of the NFS V4 protocol:

- Internationalization - UTF-8 and stringprep support
- RPCBIND-based registration

  Change NFS Server registration from portmapper to RPCBIND for IPv6 support.

Serviceability improvements

In the continuing effort to improve the quality and usability of z/OS NFS, several serviceability improvements have been identified. These improvements are grouped under the serviceability improvements line item category.

- Convert NFS Client external trace file to CTRACE.

  Conversion of the z/OS NFS Client from use of its own proprietary debug trace mechanism to exploiting the system component trace (CTrace) facility.

- List mounts operator command host name support.

  Modify the LIST MOUNTS operator command to return the client host name.

- NFS Client stop pfs operator command support.

  Implement support for proper function vector cleanup when the NFS Client is terminated.

> **Note:** Starting with NFS V1.8, both the z/OS NFS Server and z/OS NFS Client have the same FMID of HDZ118N.

8.6.1 Exports file net group support

z/OS NFS provides multiple mechanisms for specifying the authentication to be used when validating an NFS Client's authority to access a z/OS file system. One of these mechanisms (selected via the SECURITY(EXPORTS | SAFEXP) site attribute) uses an exports file to specify access rules. When this mechanism is used, only the file systems listed in this file may be accessed via NFS. Further restrictions may be placed on the access when desired. If only some NFS Clients are to be allowed access to a given file system, those NFS clients may be listed on parameters (that is, Access, ro, rw) to indicate that only those clients may access the specified file system.

Pre-z/OSV 1.8, the format of the NFS Client specification is restricted to lists of single host names. Other platforms support multiple definition formats:

- Single host

  This is the most common format. You may specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

- Netgroups

  NIS netgroups may be given as @group. Only the host part of each netgroup member is considered in checking for membership. Empty host parts or those containing a single dash (-) are ignored.

- Wildcards

  Machine names may contain the wildcard characters of an asterisk (*) and a question mark (?). These can be used to make the exports file more compact. For instance, '*.cs.foo.edu' matches all hosts in the domain 'cs.foo.edu'. However, these wildcard characters do not match the dots in a domain name, so the above pattern does not include hosts such as 'a.b.cs.foo.edu'.

- IP networks

  You can also export directories to all hosts on an IP (sub-) network simultaneously. This is done by specifying an IP address and netmask pair as address/netmask, where the netmask can be specified in dotted-decimal format or as a contiguous mask length (for example, either '/255.255.252.0' or '/22' appended to the network base address result in identical subnetworks with 10 bits of host).

- To extend z/OS NFS usability and compatibility with other platforms, this line item extends the z/OS NFS support to include the above specification formats with the following restriction: The z/OS platform does not support the NIS environment. z/OS UNIX does support the /etc/netgroup file. Therefore, z/OS NFS is restricted to only support use of the local /etc/netgroup file for definition of net groups. Otherwise, the design will follow UNIX netgroup concepts and other UNIX server's implementations of netgroup support.

  A new *root* keyword will be added to the syntax to support the root authorization specification cited above.

NFS server checklist function enhancements

The NFS Server also supports the use of a checklist data set with which MVS data sets can be specified for which SAF authorization checking is to be bypassed, implying that all NFS users have read/write access to the data set, regardless of what is specified in SAF.

The checklist data set also uses NFS Client specifications, like the exports file, and has the same current specification restrictions. The client specification enhancements being implemented for the exports file would also be of value for the checklist file.

Further, pre-z/OS 1.8, the checklist file can only be refreshed by restarting the NFS Server. This is a severe restriction on the 7x24 high availability environments becoming common on client configurations. The checklist support should be enhanced to provide refresh capability without it being necessary to restart the NFS Server.

Client configurations are moving to environments where the system administrator is managing the system from designated remote hosts. These administrators manage the systems via special root-level authorizations. Currently, the z/OS NFS server does not allow root-level access via NFS. However, with these remote administration environments, it would be very helpful to the system administrators to allow specific hosts to be identified as having root-level authority to some file systems.

In order to accommodate the above checklist requirements and provide common extensibility to the checklist and exports file data sets in the future, this line item will merge the checklist and exports file data sets.

Restriction: Not all functions of the export list are supported for the checklist.

Dynamic refresh of EXPORTS file information

The z/OS NFS server's exports file information can be refreshed from the exports data set with the EXPORTFS keyword on the **`Modify mvsnfs`** operator command. This line item does not change this function. However, with the merge of the checklist function into the exports file, the checklist information can now also be dynamically refreshed without restarting the NFS Server.

This line item does not change the exports file refresh mechanism. Therefore, the refreshed values will not affect directories that are already mounted. It will be necessary to unmount and remount the directories for changed values to take effect.

SHOWMOUNT command

The SHOWMOUNT command with operand -e displays the list of exported directories. With this line item, the command will support displaying directory names without their suffixes (because directory suffixes correspond to checklist data). It will also support displaying expanded format host names (that is, single host name, netgroup name, single IP address, IP network, wildcards) with their suffixes (client suffixes). Host specifications will be shown as they are in the EXPORTS data set.

8.6.2 NFS Server V4 Protocol support continuation

The Network File System is a distributed file system that provides transparent processing capability for data and information about worldwide and heterogeneous networks. z/OS NFS provides the implementation that allows the z/OS platform to participate in these networks. A new industry-wide version of the communication protocol, V4, has been defined. The project described by this functional specification is intended to extend the z/OS NFS support to incorporate the new protocol. In z/OS V1R7 most of the NFSV4 protocol functions were implemented in the NFS server. However, some functions had to be deferred.

> **Note:** The UTF-8 support was done in 1.7. Only stringprep is being added in 1.8.

NFS V2 and V3 protocols are limited in the character encoding of strings. In the NFS protocols, strings are used for file and directory names, and symbolic link contents. Although the XDR definition [RFC1832] limits strings in the NFS protocol to 7-bit US-ASCII, common usage is to encode file names in 8-bit ISO-Latin-1. However, there is no mechanism available to tag XDR character strings to indicate the character encoding used by the client or server. Obviously, this limits NFS's usefulness in an environment with clients that may operate with various character sets.

The primary issue in which NFS V4 needs to deal with internationalization is with respect to file names and other strings as used within the protocol. The choice of string representation must allow reasonable name/string access to clients that use various languages. The UTF-8 encoding of the UCS as defined by [ISO10646] allows for this type of access and follows the policy described in "IETF Policy on Character Sets and Languages" [RFC2277].

[RFC3454-Preparation of Internationalized Strings ("stringprep")], otherwise known as "stringprep," documents a framework for using Unicode/UTF-8 in networking protocols, so as "to increase the likelihood that string input and string comparison work in ways that make sense for typical users throughout the world."

UTF-8 encoding and stringprep introduced in the NFS V4 protocol will be fully supported by the z/OS NFS server. Refer to NFS V4 protocol RFC3530 for additional information about UTF-8 and stringprep usage.

NFS conversion of UTF-8 character strings requires adding support of UTF-8 [CCSID 1208] conversion tables to the z/OS Unicode Services.

RPCDIND-base registration

In order for NFS Client systems to access the NFS Server, the server must register its supported program versions with TCP/IP. For the IPv4 Internet protocol, NFS Clients query the NFS Server presence via the portmapper service (Version 2 of the port mapping protocol). However, for the expanded IPv6 Internet protocol, this query is done via the RPCBIND service (using Version 3 or 4 of the port mapping protocol). The RPCBIND service is being implemented by z/OS Communication Services in z/OS V1R8.

Prior to NFS V1.8, the z/OS NFS Server has registered with TCP/IP via the portmapper service. This will not be adequate for the NFS Clients to determine whether the NFS Server supports IPv6 communications. Therefore, it will be necessary to modify the z/OS NFS Server registration function to register via RPCBIND once the RPCBIND service is available on the system.

To avoid a total dependency on the existence of RPCBIND, the z/OS NFS Server must be modified to first query the presence of the RPCBIND service. If it is available, then that will be used for server registration. Otherwise, the existing portmapper service will be used instead to allow continued use of existing functionality.

8.6.3 Serviceability improvements

In this section we discuss serviceability improvements.

Convert NFS client external trace file to CTrace

Prior to NFS V1.8, the z/OS NFS Client stores messages and diagnostic debug information in the z/OS NFS client log data sets specified in the NFSCMSG1 and NFSCMSG2 DD statements of the client's startup procedure.

NFSCMSG1 is associated with the primary log, while NFSCMSG2 is associated with the secondary log. When the client is started, the primary log is used first. When the primary log is full, the logging is automatically switched to the secondary log and a console message is displayed. The last log buffer is lost when the switch takes place. When the secondary log is also full, the logging is switched back to the primary log and the original primary log content is overwritten.

In this release the NFS client will now add support to record diagnostic information in a z/OS component trace buffer. Depending on which debug controls are turned on will determine whether the diagnostics are written to the existing log data sets or to the new trace buffer. If both are turned on, it will be recorded to both. For performance reasons, we highly recommend that the component trace buffer be used, not the log data sets. Error, warning, and informational messages will continue to be recorded to the log data sets. They will also be written to the component trace if selected.

List mounts operator command host name support

The z/OS NFS "List=Mounts" operand of the MODIFY operator command displays all of the mount points that are currently active on the z/OS NFS Server. However, it does not give any information about which mount points are being accessed by which NFS Clients. Therefore, if a mount point problem occurs, the client cannot determine which clients are using that mount point.

This line item will expand the console output from this operator command to follow each mount point with the list of clients that currently have active mounts to this mount point. If available, the client's host name will be displayed. If not, then the IP address will be given. The number of clients in the list will be less than or equal to the number reported on the ACTIVE parameter of the GFSA910I, or GFSA911I, message, which displays the mount point. The list of clients may be less than the active count because a client can have more than one active mount to a given mount point.

NFS client stop PFS operator command support

Currently, the z/OS NFS client can be stopped by the **`p mvsnfsc`** operator command. However, if the client has active mount points at the time, use of this operator command can cause user's shells to terminate abnormally. This may occur because the NFS client cannot inform z/OS UNIX LFS that the PFS (that is, NFS) is terminating, and LFS may subsequently try to call one of the NFS client functions, which will then no longer be present.

In z/OS V1R8, z/OS UNIX is implementing a new operator command **`MODIFY OMVS STOPPFS=pfsname`** (that is, pfsname = "NFS" for the NFS Client). The purpose of this new operator command allows z/OS UNIX LFS to control the shutdown of PFSs. This z/OS NFS line item implements the NFS Client changes necessary to support this new function.

With the implementation of this improved function for stopping PFSs, like the NFS Client, z/OS NFS client support for the **STOP mvsnfsc** operator command will be removed.

Note: The new operator command is a new z/OS UNIX operator command, not a new NFS Client operator command. Availability of the new operator command is dependent on its implementation by z/OS UNIX.

A

Code samples

This appendix contains sample code that may be useful when implementing DFSMS V1.7.

Sample program to display OAM recall to DB2 records

This sample program produces a printout of some of the fields from the SMF record type 85 subtype 38. This is a new record with DFSMSdfp OAM corresponding to the new feature that allows objects to be recalled from tape and be stored in DB2 for subsequent faster retrieval if required again.

In order to use this code, SMF records type 85 must be being recorded.

The program and related code use the prefix SMF85TO.

This material is intended to be cut and pasted into a z/OS data set. The example was built in a PDS, so it will be easiest to update the JCL if the parts are reconstructed into a PDS.

JCL to assemble and link the program

There are seven steps required to run the SMF85TO program and review the output.

Step 1: Create PDS member SMF85TOJ in your PDS

This is the JCL to assemble and link the module.

Cut and paste the contents of Figure A-1 into the SMF85TOJ member of your PDS.

```
//
/*PRIORITY  15
//MHLRES1L JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
//*OUT OUTPUT DEST=WTSCMXA.EFLETCH,CLASS=J,FORMS=STD,DEFAULT=YES
//OUT OUTPUT DEST=WTSCMXA.EFLETCHP,CLASS=J,FORMS=STD,DEFAULT=YES
//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.JOBS.JCL(SMF85TOA)
/*
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TOJ.PDS,DISP=SHR TLG,DELETE)
//LKED.SYSIN   DD *
 SETSSI 00004000
 NAME SMF85TO(R)
```

Figure A-1 JCL to assemble and link SMF85TO

The first time this JCL is run, temporarily alter the //LKED.SYSLMOD line so that it creates the PDS.

Change:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TOJ.PDS,DISP=SHR TLG,DELETE)
```

to:

```
//LKED.SYSLMOD  DD DSN=MHLRES1.SMF85TOJ.PDS,DISP=(,CATLG,DELETE)
```

Note: Change the line back after the first run before any re-runs.

Step 2: Create PDS member SMF85TOA in your PDS

Cut and paste the contents of Figure A-2 on page 333 and all subsequent ones through to Figure A-11 on page 342 one after the other into PDS member SMF85TOA. The result should contain 420 lines.

```
 MACRO
&NAME    SEGSTART &SAVE=(14,12),&T=,&IDENT=*,&BASE=12,&EQU=YES,&RENT=, *
               &R1=R1,&R2=R2,&R3=R3,&R4=R4,&R5=R5,&R6=R6,&RB=RB,       *
               &R7=R7,&R8=R8,&R9=R9,&R10=R10,&R11=R11,&R12=R12,        *
               &R13=R13,&R14=R14,&R15=R15,&R0=R0,&AREA=SAVEAREA,       *
               &SP=0,&SAREA=
         GBLB  &NREN,&NSAR
         GBLC  &SEGSECT,&SUBP,&ARNAME,&RBASE
         LCLA  &WK
         LCLC  &A,&B,&TC,&ID2
         LCLC  &R0L,&R1L,&R2L,&R3L,&R4L,&R5L,&R6L,&R7L,&R8L,&R9L,&R10L
         LCLC  &R11L,&R12L,&R13L,&R14L,&R15L
         AIF   ('&NAME' NE '').OK
         MNOTE 16,'THERE MUST BE AN ENTRY IN THE NAME FIELD'
         MEXIT
.OK      ANOP
&ID2     SETC  '&IDENT'
         AIF   ('&ID2' NE '*').NOID2
&ID2     SETC  '&NAME'
.NOID2   ANOP
&NREN    SETB  ('&RENT' EQ 'NO')
&NSAR    SETB  ('&SAREA' EQ 'NO')
&SEGSECT SETC  '&NAME'
&RBASE   SETC  '&RB'
&NAME    CSECT
         USING *,&BASE
&TC      SETC  '&T'
         AIF   ((&NREN) OR (&NSAR)).SAVO
&A       SETC  '&SAVE(1)'
         AIF   (N'&SAVE EQ 1).SAV1
         AIF   (N'&SAVE NE 2).SAVE
&B       SETC  '&SAVE(2)'
.SAV1    AIF   (&A GE 3).NOTALL
         AIF   ('&A' EQ '0').CHK1
         AIF   (N'&SAVE EQ 2).SET0
&B       SETC  '&A'
         AGO   .SET0
.CHK1    AIF   (N'&SAVE EQ 1).SET1
         AIF   ('&B' NE '0').SET0
.SET1    ANOP
&B       SETC  '1'
.SET0    ANOP
         AIF   ('&TC' EQ 'T').SET14
         SAVE  (0,&B),&TC,&ID2
         AGO   .SAVT
.SET14   SAVE  (14,&B),,&ID2
         AGO   .SAVT
.NOTALL  AIF   (&A GE 14).TEST2
         SAVE  (0,1),&TC,&ID2
         SAVE  &SAVE
         AGO   .SAVT
```

Figure A-2   SMF85TO assembler source (part 1 of 10)

```
.TEST2   AIF   (N'&SAVE EQ 1).SB1
         AIF   ((&B GE 1) AND (&B LE 12)).SAVE
.SB1     ANOP
&B       SETC  '1'
.SAVE    AIF   ('&B' EQ '').SAVO
         SAVE  (&SAVE(1),&B),&TC,&ID2
         AGO   .SAVT
.SAVO    SAVE  &SAVE,&TC,&ID2
.SAVT    ANOP
         LR    &BASE,15
         AIF   (&NSAR).CHKEQU
         AIF   ('&RENT' EQ 'NO').NORENT
&SUBP    SETC  '&SP'
         GETMAIN R,LV=SEGWORKL,SP=&SUBP
         ST    1,8(13)
         ST    13,4(1)
         LM    0,1,20(13)
         L     13,8(13)
         USING SEGWORK,13
SEGWORK  DSECT
         DS    18A
&NAME    CSECT
         AGO   .CHKEQU
.NORENT  ST    13,&AREA+4
         AIF   (&BASE EQ 3).SET4
&WK      SETA  3
         AGO   .LR
.SET4    ANOP
&WK      SETA  4
.LR      LR    &WK,13
         CNOP  0,4
         BAL   13,*+76
&ARNAME  SETC  '&AREA'
&ARNAME  DC    18F'0'
         ST    13,8(&WK)
.CHKEQU  AIF   ('&EQU' NE 'YES').NOEQU
&R0L     SETC  '&R0'(1,8)
&R1L     SETC  '&R1'(1,8)
&R2L     SETC  '&R2'(1,8)
&R3L     SETC  '&R3'(1,8)
&R4L     SETC  '&R4'(1,8)
&R5L     SETC  '&R5'(1,8)
&R6L     SETC  '&R6'(1,8)
&R7L     SETC  '&R7'(1,8)
&R8L     SETC  '&R8'(1,8)
&R9L     SETC  '&R9'(1,8)
&R10L    SETC  '&R10'(1,8)
&R11L    SETC  '&R11'(1,8)
&R12L    SETC  '&R12'(1,8)
```

Figure A-3 SMF85TO assembler source (part 2 of 10)

```
&R13L    SETC  '&R13'(1,8)
&R14L    SETC  '&R14'(1,8)
&R15L    SETC  '&R15'(1,8)
         AIF   (&BASE NE 12).N12
&R12L    SETC '&RB'(1,8)
         AGO   .EQUS
.N12     AIF   (&BASE NE  2).N2
&R2L     SETC '&RB'(1,8)
         AGO   .EQUS
.N2      AIF   (&BASE NE  3).N3
&R3L     SETC '&RB'(1,8)
         AGO   .EQUS
.N3      AIF   (&BASE NE  4).N4
&R4L     SETC '&RB'(1,8)
         AGO   .EQUS
.N4      AIF   (&BASE NE  5).N5
&R5L     SETC '&RB'(1,8)
         AGO   .EQUS
.N5      AIF   (&BASE NE  6).N6
&R6L     SETC '&RB'(1,8)
         AGO   .EQUS
.N6      AIF   (&BASE NE  7).N7
&R7L     SETC '&RB'(1,8)
         AGO   .EQUS
.N7      AIF   (&BASE NE  8).N8
&R8L     SETC '&RB'(1,8)
         AGO   .EQUS
.N8      AIF   (&BASE NE  9).N9
&R9L     SETC '&RB'(1,8)
         AGO   .EQUS
.N9      AIF   (&BASE NE 10).N10
&R10L    SETC '&RB'(1,8)
         AGO   .EQUS
.N10     AIF   (&BASE NE 11).N11
&R11L    SETC '&RB'(1,8)
         AGO   .EQUS
.N11     AIF   (&BASE NE 13).N13
&R13L    SETC '&RB'(1,8)
         AGO   .EQUS
.N13     AIF   (&BASE NE 14).N14
&R14L    SETC '&RB'(1,8)
         AGO   .EQUS
.N14     AIF   (&BASE NE 15).N15
&R15L    SETC '&RB'(1,8)
         AGO   .EQUS
.N15     AIF   (&BASE NE  0).N0
&R0L     SETC '&RB'(1,8)
         AGO   .EQUS
.N0      AIF   (&BASE NE  1).EQUS
&R1L     SETC '&RB'(1,8)
```

Figure A-4   SMF85TO assembler source (part 3 of 10)

```
.EQUS    ANOP
&R0L     EQU   0
&R1L     EQU   1
&R2L     EQU   2
&R3L     EQU   3
&R4L     EQU   4
&R5L     EQU   5
&R6L     EQU   6
&R7L     EQU   7
&R8L     EQU   8
&R9L     EQU   9
&R10L    EQU   10
&R11L    EQU   11
&R12L    EQU   12
&R13L    EQU   13
&R14L    EQU   14
&R15L    EQU   15
.NOEQU   ANOP
         MEND
         MACRO
&NAME    SEGEND &RENT=,&RC=0,&SP=,&SAVE=(0),&BSM=YES,&SAR
         LCLC  &RETC,&SAV0,&SAV1
         GBLC  &SEGSECT,&SUBP,&ARNAME
         GBLB  &NREN,&NSAR
         AIF   ('&SYSECT' EQ '&SEGSECT').CHKR
&SEGSECT CSECT
.CHKR    AIF   ('&NAME' EQ '').NONAME
&NAME    DS    0H
.NONAME  AIF   ('&RENT' EQ '').CHKRETC
&NREN    SETB  ('&RENT' EQ 'NO')
.CHKRETC ANOP
&RETC    SETC  '&RC'
         AIF   ('&RC'(1,1) NE '(').RCCON
&RETC    SETC  '&RC'(2,K'&RC-2)
.RCCON   ANOP
         AIF   ('&SAREA' EQ '').USENSAR
&NSAR    SETB  ('&SAREA' EQ 'NO')
.USENSAR AIF   (&NSAR).RETR
         AIF   (&NREN).NORENT
&SAV0    SETC  '&SAVE(1)'
&SAV1    SETC  '&SAVE(2)'
         AIF   (K'&SAV0 EQ 0).R0OK
         AIF   ('&SAV0'(1,1) NE 'R').R0OK
&SAV0    SETC  '&SAV0'(2,K'&SAV0-1)
.R0OK    AIF   (K'&SAV1 EQ 0).R1OK
         AIF   ('&SAV1'(1,1) NE 'R').R1OK
&SAV1    SETC  '&SAV1'(2,K'&SAV1-1)
.R1OK    ANOP
         AIF   (('&SAV0' NE '1') AND ('&SAV1' NE '1')).NS1
         LR    14,1
```

Figure A-5 SMF85TO assembler source (part 4 of 10)

```
.NS1     ANOP
         LR    1,13
         L     13,4(13)
         AIF   (('&SAV0' NE '0') AND ('&SAV1' NE '0')).NOT0
         ST    0,20(13)
.NOT0    AIF   (('&SAV0' NE '1') AND ('&SAV1' NE '1')).NOT1
         ST    14,24(13)
.NOT1    ANOP
         AIF   ('&RC'(1,1) NE '(').NS15
         ST    &RETC,16(13)
.NS15    ANOP
         AIF   ('&SUBP' NE '').NO
&SUBP    SETC  '0'
.NO      AIF   (T'&SP EQ '0').FR1
&SUBP    SETC  '&SP'
.FR1     FREEMAIN R,LV=SEGWORKL,A=(1),SP=&SUBP
         LM    14,12,12(13)          RESTORE REGISTERS
         AIF   ('&RC'(1,1) EQ '(').SETI
         LA    15,&RETC       SET UP R15
         AGO   .SETI
.*
.NORENT  ANOP
         L     13,&ARNAME.+4
.RETR    ANOP
         AIF   ('&RC'(1,1) EQ '(').R15NR
         LA    15,&RETC       SET UP RETURN CODE
         LM    2,12,28(13)          RESTORE REGS
         L     14,12(13)
         AGO   .SETI
.R15NR   ANOP
         AIF   (('&RETC' EQ '15') OR ('&RETC' EQ 'R15')).R15OK
         LR    15,&RETC
.R15OK   ANOP
         LM    2,12,28(13)          RESTORE REGS
         L     14,12(13)
.*
.SETI    ANOP
         OI    15(13),X'01'          SET RETURNED INDICATION
         AIF   ('&BSM' EQ 'YES').BSM      GO DO BSM INSTEAD
         BR    14                    RETURN
```

Figure A-6 SMF85TO assembler source (part 5 of 10)

```
 AGO   .DUMM
.BSM     ANOP
         BSM   0,14                 RETURN
.DUMM    ANOP
         AIF   (&NSAR).LTORG
         AIF   (&NREN).LTORG
SEGWORK  DSECT
         DS    0D
SEGWORKL EQU   *-SEGWORK
         DS    (((SEGWORKL+4095)/4096)*4096-SEGWORKL)X
&SYSECT  CSECT
.LTORG   LTORG
         MEND
```

Figure A-7 SMF85TO assembler source (part 6 of 10)

```
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
*  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
```

Figure A-8   SMF85TO assembler source (part 7 of 10)

```
* 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 FLGO ON OR OFF
*
*************************************************************
* 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)
         UNPK  OLEN(11),DWORD+2(6)
         PUT   PRINTDCB,PRINTL2
*
         MVC   VSN,ST38VSN
         MVC   MT,ST38MT
```

Figure A-9 SMF85TO assembler source (part 8 of 10)

```
*  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
*  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  RC=0
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'
```

Figure A-10 SMF85TO assembler source (part 9 of 10)

```
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 A-11 SMF85TO assembler source (part 10 of 10)

Step 4: Create PDS member IFASMFDP in your PDS

This JCL creates an extract from the system SMF records containing only SMF record type 85 subtype 38 records.

```
//MHLRES1S JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES1
// EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DISP=SHR,DSN=SMFDATA.ALLRECS.G0046V00
//OUTDD  DD DSN=MHLRES1.SMF8538E.TEXT,
//*OUTDD  DD DSN=MHLRES1.SMF4216B.TEXT,
// SPACE=(CYL,(10,5)),
// RECFM=VB,LRECL=5096,
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA
//SYSIN DD *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(OUTDD,TYPE(85(38)))
//
```

Figure A-12 SMF job to select records type 85 subtype 38

Step 5: Create PDS member SMF85TOE in your PDS

```
//
/*PRIORITY  15
//MHLRES1E JOB (1234567,COMMENT),MHLRES1,TIME=10,
// MSGCLASS=J,
// MSGLEVEL=1,CLASS=A,
// NOTIFY=MHLRES1
//*OUT OUTPUT DEST=WTSCMXA.EFLETCH,CLASS=J,FORMS=STD,DEFAULT=YES
//  EXEC PGM=SMF85TO
//STEPLIB DD DISP=SHR,DSN=MHLRES1.SMF85TOJ.PDS
//SYSUDUMP DD SYSOUT=A
//*SMFIN   DD DISP=SHR,DSN=UAALFO.AAAR.SMF.D2004325.EXTR
//SMFIN   DD DISP=SHR,DCB=BFTEK=A,
//  DSN=MHLRES1.SMF8538E.TEXT
//PRINT   DD SYSOUT=A,RECFM=UA
```

Figure A-13 JCL to execute program SMF85TO

Step 6: Build and run the SMFTO85 program

After building all of the members of the PDS, check and adjust data set names to suit your requirements. Then run:

- SMF85TOJ to build the program expect return code 0
- IFASMFDP to create the SMF extract
- SMF85TOE to create the report

Step 7: Review the output from the SMF85TO program

In Figure A-14 we show an example of the report produced.

```
SMF TYPE 85 SUBTYPE 38 RECORDS
COLN/CNID:            MHLRES2.COLLECT                              00000000001
OBJN/SGN/OLEN:        MHLRES2.OBJECT                               SGOBJECT 000
VSN/MT/TKN/VT/BT/FLGS:TST016 06 00000000004             FLGO ON

COLN/CNID:            MHLRES2.COLLECT                              00000000001
OBJN/SGN/OLEN:        MHLRES2.OBJECT1                              SGOBJECT 000
VSN/MT/TKN/VT/BT/FLGS:TST016 06 00000000070             FLGO ON
```

Figure A-14 Sample output from program SMF85TO

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 the OAM recall to DB2. In the report the field names are abbreviations of the names found in the SMF records. In the SMF records, as shown in Figure A-15, 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.

```
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
ST38FLGS DS     BL4'0'          PROCESSING FLAGS
*
ST38FLGO EQU    X'80'           WHEN ON, OBJECT RECALL
*                               WAS SUCCESSFUL.
*                               WAS SUCCESSFUL.
```

Figure A-15 SMF record type 85 subtype 38 significant fields (extract from CBRSMF macro)

Example of NFS V4 attribute data set

The following is an example of a Nfs Server attribute data set.

Example: A-1 Nfs Server attribute data set

```
#####################################################################
#                                                                   #
#  Z/OS Network File System Server Sample Attribute Table      @L6C #
#####################################################################
#                                                                   #
# change activities:                                                #
# 1.  5/10/91 - Release it for MVS/DFP V3                           #
# 2.  1/30/92 - Updates for VSCR                                    #
# 3.  8/06/92 - Change mintimeout default                           #
# 4.  8/31/92 - Add PCNFSD                                          #
# 5.  9/20/92 - R2 updates                                          #
# $L3X=NFS,HDZ11NP,931230,SJPLJST: Change NFSTASKS default and   @L3XA#
#                                  add XLAT keyword              @L3XA#
# $L3L=KA90033,HDZ11NP,940405,SJPLJST: Add RESTIMEOUT keyword    @L3LA#
#                                                                   #
# $L33=NFS,HDZ11NP,940613,SJPLTEM:  Add SMF keyword              @L33A#
# $01=OW12199,HDZ11NP,950323,SJPLTEM: Add HFS keywork             @01A#
# $P1=KA00045,HDZ11SP,960111,SJPLTEM: Updates for DFSMS 1.3       @P1A#
# $P2=KA00107,HDZ11SM,960415,SJPLTEM: Remove MODEL attribute      @P2A#
# $L59=NFS,HDZ11TS,970226,SJPLBPF: File Ext. Support             @L59A#
# $P3=KAB0033,HDZ11TS,970701,SJPLPKL: Add # comment char after    @P3A#
#                                      xlat keyword               @P3A#
# $P4=KAB0114,HDZ11TS,971030,SJPLTEM: Chg DFSMS/MVS->OS/390       @P4A#
# $L53=NFS,HDZ11TS,971031,SJPLBPF: WebNFS Support                @L53A#
# $P5=KAB0379,HDZ11TS,980512,SJPLBPF: Default changes             @P5A#
# $L5D=NFS,HDZ11TS,980821,SJPLBPF: NC Support            OW34846=@L5DA#
# $L5X=NFS,HDZ11TS,980820,SJPLTEM: Filename delimiter    OW34846=@L5XA#
# $LA1=OW38745,HDZ11TS,981209,SJPLRMS: nfstasks(n,m,o)           @LA1A#
# $02=OW40268,HDZ11TS,990727,SJPLRMS: Comment out nfstasks flag   @02A#
# $03=OW42036,HDZ11TS,991213,SJPLBPF: New readdirtimeout keyword  @03A#
# $04=OW43829,HDZ11TS,000410,SJPLBPF: Lower readdirtimeout limit  @04A#
# $LA7=OW46949,HDZ11TS,000921,IBSKEK: TEXT/BINARY on a single    @LA7A#
#                                     mount point                @LA7A#
# $P6=KAD0016,HDZ11TS,001106,SJPLBPF: Allow up to 100 hfs tasks   @P6A#
# $05=OW48939,HDZ11TS,010415,SJPLJST: rddr cookie verifier        @05A#
# $LC1=OW49104,HDZ11TS,010115,IBSNIV: File Tagging Support       @LC1A#
# $06=OW51358,HDZ11TS,010921,IBSPIV : New hfsfbtimeout keyword    @06A#
# $07=OW54351,HDZ11TS,020422,IBSKVV : New upcase and              @07A#
#                                     mixcase keywords            @07A#
# $08=OW55830,HDZ11TS,020819,IBSKVV :SMF activate at NFSS startup @08A#
# $L6=NFS,HDZ11US,030405,SJPLMB:      Changed OS/390 to Z/OS      @L6A#
# $L66=NFS,HDZ11US,030303,IBSMVB: NFSS 878 abend handling        @L66A#
# $LCE=OW55734,HDZ11TS,020701,IBSNIV: File Tagging Support       @LCEA#
# $L74=NFS,HDZ11VS,031015,SJPLJST: NFS ver 4 protocol support    @L74A#
# $09=OA03523,HDZ11TS,030515,IBSVKR: Extend RETRIEVE attr to HFS  @09A#
# $0A=OA05684,HDZ11TS,031208,IBSNIV: New remount/noremount keyword@0AA#
# $L76=NFS,HDZ11VS,040119,IBSDYP: NFS Server DHCP Support        @L76A#
# $L74=NFS,HDZ11VS,040322,SJPLSLH: Added MVSMNT PRocessing Attr @L74BA#
# $L74=NFS,HDZ11VS,040805,SJPLMB:  Added DENYRW/NODENYRW Attrs @L74M5A#
# $L7E=NFS,HDZ11VS,041214,SJPLMB:                                @L7EA#
```

```
#          Legal Rqmt: Change "OPEN EDITION" to "z/OS UNIX"      @L7EA#
# $P07=KAJ0262,HDZ11VS,050204,SJPLMB:                            @P07A#
#          1. Add missing comment delimiters                     @P07A#
#          2. Move MVSMNT to Processing Attributes Section       @P07A#
#          2. Remove nohfs option.  It is no longer supported.   @P07A#
# $0B=OA08867,HDZ11US,040929,IBSKYL: Correct typo for @L66A       @0BA#
# $P08=KAJ0243,HDZ11VS,050331,SJPLSCA: 64bit fileid support      @P08A#
# $0C=OA12850,HDZ11VS,050811,SJPLSLH: MVSMNT and SAF text added   @0CA#
# $0D=OA14044,HDZ11VS,051208,IBSNIV: Correct nfstasks(n, ,o)      @0DA#
# $L74=NFS,HDZ11VS,041020,SJPLKU:                               @L74AA#
#             Added mvssec(),hfssec() and pubsec().             @L74AA#
# $0E=OA12994,HDZ11VS,060106,SJPLRAS:                             @0EA#
#             Added nordrcache                                    @0EA#
#                                                                    #
######################################################################
#
# This is a prototype site defaults attribute file for the
#  Z/OS Network File System Server Sample Attribute Table         @L6C#
#
# '#' character starts a comment.  Comments can appear anywhere.
# White space is ignored when parsing the file.

# Default values are illustrated in the examples in this file


#
# Keywords are not case sensitive. 'BLKS' is the same as 'blks' is
# the same as 'BlKs'.

# All time values are in seconds.

######################################################################
# The following are known as data set creation attributes.           #
######################################################################
space(100,10), blks
norlse
recfm(vb), blksize(0), lrecl(8196)
dsorg(ps)
dir(27)
mgmtclas(mgmt_class_name)
volume(volser)
unit(unit_name)
# The following attributes are used to control VSAM data set
# creation. They are used only if the DSORG parameter defines
# the data set to be type INDEXED, NONINDEXED or NUMBERED.
#
# Refer to appropriate IBM MVS documentation for a more
# complete description of these and other data set creation
# attributes.
keys(64,0)
recordsize(512,4K)
nonspanned
shareoptions(1,3)
######################################################################
# The following are known as processing attributes.                  #
######################################################################
```

```
attrtimeout(120), readtimeout(90), writetimeout(30)
binary
mapped
LF
blankstrip
mapleaddot
maplower
retrieve
nofastfilesize
setownerroot
executebitoff
xlat(member)
nofileextmap
notag
cln_ccsid(819)
srv_ccsid(1047)
convserv(LRE)
NORDRCACHE
nordrverf
########################################################################
# The following are known as site attributes.                          #
########################################################################
security(safexp,safexp,safexp)
nopcnfsd
leadswitch
mintimeout(1)
nomaxtimeout
logout(1800)                    # 30 minutes (30 * 60)
readdirtimeout(30)
nfstasks(8,16,8)
mintasks(4,8,4)
restimeout(48,0)
cachewindow(112)
hfs(/hfs)
fileidsize(64)
logicalcache(16M)
bufhigh(32M)
percentsteal(20)
readaheadmax(16K)
maxrdforszleft(32)
smf(none,off)
sfmax(0)
PUBLIC(legacy_path,hfs_path)
nochecklist
fn_delimiter(,)
hfsfbtimeout(60)
upcase
rec878
nonlm
leasetime(120)
remount
nodhcp
DENYRW
```

```
mvssec(krb5,krb5i,krb5p,sys)
hfssec(krb5,krb5i,krb5p,sys)
pubsec(krb5,krb5i,krb5p,sys)
```

B

APAR text

This appendix contains the text of APARs that must be reviewed to successfully implement DFSMS V1.7.

APARs referenced in the book

OA13465

Problem summary:

****

* USERS AFFECTED: Callers of FASTDATA (IEWBFDAT) services *

****

* PROBLEM DESCRIPTION: IEWBFPRC fails to check for proper *

* positioning of the cursor with multi- *

* section modules *

****

* RECOMMENDATION: *

****

Abend S0C4 in IEWBFDATwhen retrieving text from multiple

classes.

Problem Conclusion:

Added checking to IEWBFPRC to validate the cursor and

maintain proper positioning of the cursor when handling

multi-section modules

OA11875

NEW SERVER FUNCTION SUPPORT.

ERROR DESCRIPTION:

NEW SERVER FUNCTION SUPPORT.

Implementation of RPCSEC_GSS authentication flavor employing Generic Security Services API using the Kerberos V5 Security Mechanism, as mandated by the NFS version 4 protocol.

OA11875 introduces a new set of z/OS NFS Server (release 1.7) attributes related to Kerberos security services support. To be able to see the z/OS NFS Server (release 1.7) attributes from an NFS Client session by the "showattr" command, users should use the showattr utility built from SYS1.NFSTARB source code.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 352. Note that some of the documents referenced here may be available in softcopy only.

- *z/OS V1R6 DFSMS Technical Guide*, SG24-6651
- *Partitioned Data Set Extended Usage Guide*, SG24-6106

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
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405
- *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

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 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

H

I

J

K

L

M

N

z/OS V1R7 DFSMS Technical Update

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

z/OS V1R7 DFSMS Technical Update

Understand the features and functions in z/OS DFSMS V1.7

Get 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 Redbook provides a technical overview of the functions and enhancements in z/OS V1R7 DFSMS. It provides you with the information 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 have been integrated into z/OS DFSMS V1R7.

This book 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 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

SG24-7225-00 ISBN 073848976X