Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xiii.

First Edition (July 1994)

This edition applies to Version 1, Release Number 2, Modification Level 0 of DFSMS/MVS, Program Number 5695-DF1.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader’s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 471 Building 070B
5600 Cottle Road
San Jose, California 95193-0001

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1994. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Abstract

This document describes the implementation of the compression facility that is one of the functional capabilities in DFSMS/MVS Version 1 Release 2.0. It provides an overview of sequential data set (BSAM and QSAM) and VSAM KSDS compression, performance measurements, and hints and tips about how to start using the new facilities.

This document is intended for IBM customer technical professional personnel.

A knowledge of MVS/ESA, DFSMS/MVS, and IBM ES/9000 711-, 511-, and 211-based processors is assumed.

ST  

(131 pages)
## Contents

Abstract ................................................................. iii
Figures ........................................................................................ ix
Tables ................................................................................................ xi
Special Notices .................................................................................. xiii
  Trademarks ......................................................................................... xiv
Preface ................................................................................................. xv
  How This Document Is Organized ....................................................... xv
Related Publications .............................................................................. xvii
  International Technical Support Organization Publications .......... xvii
  ITSO Storage-Related Publication Offerings: June 1994 ................. xviii
Acknowledgments .................................................................................. xix

### Chapter 1. Introduction

1.1 Extended Remote Copy and Peer-to-Peer Remote Copy ............... 1
1.2 Support of Distributed FileManager/MVS Server Implementation .... 1
1.3 VSAM Partial Release of Space and Secondary Volume Space Allocation .... 1
1.4 Improvements in DFSMShsm ................................................. 2
1.5 Improvements in DFSMSrmm ................................................. 2
1.6 System-Managed Tape Enhancements .................................... 2
1.7 Backup-While-Open Improvements for CICS VSAM Data Sets .... 2
1.8 Support for More than Eight MVS Images in an SMSplex ........ 3
1.9 Device Allocation Enhancements .......................................... 3
1.10 Object Access Method Object Tape Support ......................... 3
1.11 OAM Statistics ............................................................... 3
1.12 OAM Support for the 3995 Optical Library Dataserver Model 133 ... 3
1.13 Support for OpenEdition MVS ............................................ 4
1.14 BSAM, QSAM, and VSAM KSDS Data Compression ............ 4

### Chapter 2. DFSMS/MVS Version 1 Release 2.0 System Changes

2.1 Hardware and Software Requirements ................................ 7
2.2 Compression Processes and Procedures .............................. 7
  2.2.1 Compression Services Activation .................................. 8
  2.2.2 Compression Management Services .............................. 8
2.3 System Services and Facilities ......................................... 10
  2.3.1 System Services ...................................................... 10
  2.3.2 Subsystems .......................................................... 11
2.4 System Utilities and DFSMS/MVS Components .................... 11

### Chapter 3. Implementation of the Extended Format

3.1 New Block Formats ....................................................... 13
3.2 Implementation for Compressed Format .............................. 14
3.3 DASD Space Usage Considerations .................................. 14
3.4 Application Program Dependencies ................................. 15
# Chapter 4. Storage Management Administration

4.1 ISMF .................................................. 17
4.2 SMS Classes ........................................... 20
4.3 VSAM Partial Release and Secondary Volume Space Allocation ... 23
4.4 Device Allocation ....................................... 26

# Chapter 5. Tools and Utilities

5.1 Compression Evaluation Tool .............................. 29
5.1.1 DASD Space Savings .................................. 29
5.1.2 CPU Cost and Batch Elapsed Time Savings .......... 30
5.1.3 CICS VSAM Analysis .................................. 32
5.2 Utilities to Create an Extended and Compressed Format Data Set .... 33
5.2.1 IDCAMS .................................................. 33
5.2.2 DFSMSdss ............................................... 35
5.2.2.1 VSAM KSDSs ......................................... 35
5.2.2.2 Sequential Data Sets .................................. 36
5.2.3 DFSMSshm .............................................. 38
5.2.4 IEHLLIST ................................................ 38
5.2.5 IEBGENER ............................................... 39
5.2.6 DFSORT .................................................. 39
5.3 Summary ................................................... 39

# Chapter 6. Rules to Create and Use Extended Format Data Sets

6.1 DFSMS/MVS System Environment .......................... 43
6.1.1 System Setup .......................................... 43
6.2 Data Set Creation ........................................ 44
6.2.1 VSAM Data Set Eligibility ............................. 44
6.2.2 Sequential Data Set Eligibility ......................... 45
6.2.3 Sequential Data Set and VSAM KSDS Eligibility .......... 45
6.2.4 VSAM KSDS Space Considerations ....................... 46
6.3 Data Set Usage ............................................ 46
6.3.1 Application Programs ................................... 46
6.3.2 New Messages .......................................... 47
6.4 Data Availability ......................................... 48
6.4.1 Local Backup and Recovery ............................. 48
6.4.2 Disaster Backup and Recovery ......................... 49

# Chapter 7. Performance Measurements

7.1 DASD Space .............................................. 51
7.2 CPU Load .................................................. 52
7.2.1 Sequential Data Set Usage ............................. 53
7.2.2 VSAM KSDS Usage ...................................... 55
7.2.3 Considerations .......................................... 58
7.3 Implementation .......................................... 58

# Appendix A. Informational APAR II07816

# Appendix B. FLASH 4014 IMS and DFSMS/MVS 1.2

# Appendix C. Performance Measurement Details

C.1 Test Environment .......................................... 69
C.1.1 Hardware Configuration .................................. 69
C.1.2 SMS Classes ............................................ 70
C.1.3 ACS Routines ............................................ 71
C.1.4 Customer Data ............................................ 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.2 Objectives of the Measurements</td>
<td>77</td>
</tr>
<tr>
<td>C.2.1 Utilities and Tools</td>
<td>78</td>
</tr>
<tr>
<td>C.2.2 Data Collected</td>
<td>78</td>
</tr>
<tr>
<td>C.2.3 Compression Estimation Tool</td>
<td>78</td>
</tr>
<tr>
<td>C.3 Measurement Results</td>
<td>79</td>
</tr>
<tr>
<td>C.3.1 DASD Space Allocation</td>
<td>79</td>
</tr>
<tr>
<td>C.3.2 Effect of Block Size or CI Size on Space Allocations</td>
<td>81</td>
</tr>
<tr>
<td>C.3.3 Read I/O Measurements for New Format Data Sets</td>
<td>88</td>
</tr>
<tr>
<td>C.3.4 Write I/O Measurements for New Format Data Sets</td>
<td>97</td>
</tr>
<tr>
<td>C.3.5 Elapsed Times and CPU Times</td>
<td>104</td>
</tr>
<tr>
<td>List of Abbreviations</td>
<td>123</td>
</tr>
<tr>
<td>Index</td>
<td>125</td>
</tr>
</tbody>
</table>
Figures

1. Components Where Hardware Compaction or Compression Takes Place 5
2. Overview of Compression Management Services .............................. 9
3. Extended Format Data Sets and Compression ................................. 13
4. Changed Data Set Selection Entry Panel ..................................... 18
5. New Data Set Selection Entry Panel ........................................... 18
6. Data Set Selection Panel with Compression Attribute ..................... 19
7. New Compression Attributes in Data Set List ................................ 19
8. New DDM and Data Set Type in Data Set List ................................. 20
9. Information Messages Issued When Using Extended Preferred ........... 21
10. Data Class Definition Required Attributes: Page 1 of 2 .................. 21
11. Data Class Definition Required Attributes: Page 2 of 2 .................. 22
12. Storage Class Define Panel: Multistriped Extended Format Data Set .. 23
13. New Parameters Associated with VSAM Allocations ...................... 25
14. DASD Space Savings Output for Sequential Data Set .................... 30
15. DASD Space Savings Output for VSAM KSDS ............................... 30
16. CPU Report Details .................................................................... 31
17. Data Set Analysis Report ........................................................ 31
18. Cost of Resource Analysis .......................................................... 32
19. LISTCAT Display ..................................................................... 34
20. DCOLLECT Support ................................................................. 34
21. IEHLIST Changes .................................................................... 39
22. IEHLIST Sample ...................................................................... 39
23. Successful Allocation of Compressed Format Data Set ................... 47
24. Allocation in Extended Format Only, No Compression .................... 47
25. Allocation in Standard Format, No Extended Format ....................... 48
26. Space Analysis Summary for Sequential Data Sets ....................... 51
27. Space Analysis Summary for VSAM KSDSs ................................. 52
28. Average I/O Activity: Sequential Data Sets ................................. 53
29. Average CPU Load: Sequential Data Sets ..................................... 54
30. Average Elapsed Time: Sequential Data Sets ............................... 55
31. Average I/O Activity: VSAM KSDs .............................................. 56
32. Average CPU Load: VSAM KSDs ................................................ 57
33. Average Elapsed Time: VSAM KSDs ............................................ 58
34. Test System Configuration .......................................................... 70
35. Data Class ACS Routine Used in Test System ............................... 72
36. Storage Class ACS Routine Used in Test System ......................... 73
37. Management Class ACS Routine Used in Test System ................. 73
38. Storage Group ACS Routine Used in Test System ....................... 74
39. JCL Used to Test Block Sizes for Sequential Data Sets .................. 82
40. Result of Space Measurements for Sequential Data Sets ............... 84
41. JCL Used to Test Cl Size for VSAM KSDSs ................................. 85
42. Result of Allocated Space Measurements for VSAM KSDSs .......... 87
43. Result of Used Space Measurements for VSAM KSDSs ................... 88
44. JCL Used to Read a VSAM KSDS with IDCAMS ............................ 89
45. IDCAMS I/O Load for Reading VSAM KSDs ................................. 91
46. JCL Used to Read a VSAM KSDS with DFSMSdss ....................... 91
47. DFSMSdss I/O Load for Reading VSAM KSDs ....................... 92
48. JCL Used to Read a Sequential Data Set with DFSMSdss ............... 93
49. DFSMSdss I/O Load for Reading Sequential Data Sets .................. 95
50. JCL Used to Read a Sequential Data Set with IEBGENER .......... 95
51. IEBGENER I/O Load for Reading Sequential Data Sets ................ 97
Tables

1. Utilities That Manage Extended and Compressed Data Sets .............. 40
2. DFSMS/MVS Components That Manage New Format Data Sets ............ 41
3. Data Classes Used for Measurements .................................. 70
4. Storage Classes Used for Measurements .................................. 71
5. Management Classes Used for Measurements ................................ 71
6. Characteristics of Data Sets Used for Measurements .................... 75
7. VSAM KSDSs .................................................................. 76
8. Sequential Data Sets ......................................................... 77
9. Estimated Compression Factors ............................................ 79
10. DASD Space Measurements for VSAM KSDSs .......................... 80
11. DASD Space Measurements for Sequential Data Sets .................. 81
12. Space Occupancy Based on Sequential Data Set Block Size
    (Megabytes) .................................................................. 83
13. Space Occupancy Based on VSAM CI Size ................................ 86
14. Read I/O Measurements on VSAM KSDSs: IDCAMS ................... 90
15. Read I/O Measurements on VSAM KSDSs: DFSMSdss ............... 92
16. Read I/O Measurements on Sequential Data Sets: DFSMSdss ....... 94
17. Read I/O Measurements on Sequential Data Sets: IEBGENER ....... 96
18. Write I/O Measurements on VSAM KSDSs: IDCAMS ................. 98
19. Write I/O Measurements on Sequential Data Sets: IEBGENER ...... 100
20. Write I/O Measurements on Sequential Data Sets: DFSMSdss ...... 103
21. Elapsed and CPU Time Measurements When Reading VSAM KSDSs .. 105
22. Elapsed and CPU Time Measurements When Reading Sequential Data
    Sets ........................................................................ 110
23. Elapsed and CPU Time Measurements When Writing VSAM KSDSs .. 115
24. Elapsed and CPU Time Measurements Writing Sequential Data Sets .. 118
Special Notices

This publication is intended to help the customer plan the implementation of the DFSMS/MVS support for compression. The information in this publication is not intended as the specification of any programming interfaces that are provided by DFSMS/MVS and/or MVS/ESA. See the PUBLICATIONS section of the IBM Programming Announcement for DFSMS/MVS and/or MVS/ESA for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM’s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM’s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.
Trademarks

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR  CICS
DB2  DFSMS/MVS
DFSMSdss  DFSMS/hrsm
DFSMSrmm  ES/9000
ESCON  Hiperbatch
IBM  IMS/ESA
MVS/DFP  MVS/ESA
OpenEdition  OS/2
Resource Measurement Facility  RMF
S/390  VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

SAS  SAS Institute, Inc.
Preface

This document is written for people who are planning to implement DFSMS/MVS Version 1 Release 2.0 support for data compression.

How This Document Is Organized

The document is organized as follows:

• Chapter 1, “Introduction”
  This chapter provides a short introduction to the new functions in DFSMS/MVS Version 1 Release 2.0.

• Chapter 2, “DFSMS/MVS Version 1 Release 2.0 System Changes”
  This chapter briefly describes the hardware and software requirements for compression, reviews the services and explains what compression is and how it works.

• Chapter 3, “Implementation of the Extended Format”
  This chapter describes the extended format data sets and how to implement the new format.

• Chapter 4, “Storage Management Administration”
  This chapter describes the enhancements to the Interactive Storage Management Facility to support the new format data sets.

• Chapter 5, “Tools and Utilities”
  This chapter describes the techniques and utilities you can use to create an extended format data set.

• Chapter 6, “Rules to Create and Use Extended Format Data Sets”
  This chapter describes some techniques that are useful for implementing data compression in an installation and presents some guidelines for use in an existing production environment.

• Chapter 7, “Performance Measurements”
  This chapter summarizes and comments on some of the results we obtained through our tests. In particular we make some observations on DASD space, CPU load, and implementation.

• Appendix A, “Informational APAR II07816”
  This appendix is the “Info APAR for BSAM/QSAM Temporary Data Sets in DFSMS/MVS 1.2.0 Compression Support.”

• Appendix B, “FLASH 4014 IMS and DFSMS/MVS 1.2”
  This appendix is the Dallas National Marketing and Technical Support Center Document Number F4014.

• Appendix C, “Performance Measurement Details”
  This appendix describes our methodology and the results of the performance measurement tests we ran on extended format and compressed data sets.
Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document:

- *ESA/390 Principles of Operation*, SA22-7201
- *ESA/390 Data Compression*, SA22-7208
- *Support for MVS/ESA Compression Services*, GC28-1094
- *DFSMS/MVS Version 1 Release 2.0 Compression*, GG24-4056
- *A Technical Guide to ESA/390 Compression*, GG24-4130
- *MVS/ESA SP V5 System Management Facilities (SMF)*, GC28-1457

International Technical Support Organization Publications

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:


How to Order Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their IBM branch office.

Customers may order hardcopy redbooks individually or in customized sets, called GBOFs, which relate to specific functions of interest. You may also order redbooks in online format on CD-ROM collections, which contain the redbooks for multiple products.
Recently available IBM ITSO Redbooks related to storage are listed below.

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Redbook Title</th>
<th>Date</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GG24-4221-00</td>
<td>ADSM Advanced Implementation Experiences</td>
<td>05/94</td>
<td>140</td>
</tr>
<tr>
<td>GG24-4402-00</td>
<td>IBM 3495 Tape Library Dataserver Model M10</td>
<td>04/94</td>
<td>124</td>
</tr>
<tr>
<td>GG24-4400-00</td>
<td>DFSMS/MVS Version 1 Release 2.0 S/390 Parallel Sysplex Support</td>
<td>04/94</td>
<td>94</td>
</tr>
<tr>
<td>GG24-4404-00</td>
<td>DFSMS/MVS Version 1 Release 2.0 Storage Management Enhancements</td>
<td>03/94</td>
<td>124</td>
</tr>
<tr>
<td>GG24-4403-00</td>
<td>DFSMS/MVS Version 1 Release 2.0 Distributed FileManager/MVS</td>
<td>03/94</td>
<td>172</td>
</tr>
<tr>
<td>GG24-4401-00</td>
<td>DFSMS/MVS Version 1 Release 2.0 Support for OpenEdition MVS</td>
<td>03/94</td>
<td>152</td>
</tr>
<tr>
<td>GG24-4130-00</td>
<td>A Technical Guide to ESA/390 Compression</td>
<td>03/94</td>
<td>138</td>
</tr>
<tr>
<td>GG24-4056-00</td>
<td>DFSMS/MVS Version 1 Release 2.0 Compression</td>
<td>03/94</td>
<td>160</td>
</tr>
<tr>
<td>GG24-4045-00</td>
<td>Introduction to Storage Performance Tools and Techniques for MVS/ESA</td>
<td>01/94</td>
<td>162</td>
</tr>
<tr>
<td>GG24-3993-00</td>
<td>Disaster Recovery Library Database Recovery</td>
<td>01/94</td>
<td>152</td>
</tr>
<tr>
<td>GG24-3990-00</td>
<td>Implementing Concurrent Copy</td>
<td>01/94</td>
<td>168</td>
</tr>
<tr>
<td>GG24-4146-00</td>
<td>ADSTAR Distributed Storage Manager Presentation Guide</td>
<td>12/93</td>
<td>386</td>
</tr>
</tbody>
</table>

Redbooks are “how to” books written by the most experienced IBM technical professionals from all over the world. The ITSO, IBM’s International Technical Support Organization, with Centers close to IBM’s Development Divisions, publishes the Redbooks.

To improve quality, users of ITSO publications are encouraged to send their comments on the feedback form provided in the Redbooks.
Acknowledgments

This publication is the result of a residency conducted at the International Technical Support Organization, San Jose Center.

This project was designed and managed by:

Seppo Rinne
International Technical Support Organization, San Jose Center.

We want to acknowledge the excellent work of:

- Federico Fumagalli, IBM Italy
- Jane McMinn, IBM UK.

Thanks to the following people for the invaluable advice and guidance they provided in the production of this document:

- Stevan Allen, Storage Systems Division Software Products - San Jose
- Cecilia Carranza-Lewis, Storage Systems Division Software Products - San Jose
- Geoff Littlewood, IBM - UK
- Bruce Haake, Storage Systems Division - San Jose
- Dave Hernandez, Storage Systems Division Software Products - San Jose
- Jim Kwon, Storage Systems Division - San Jose
- Mark McClintock, Storage Systems Division Software Products - San Jose
- Lyle Merithew, Storage Systems Division Software Products - San Jose
- Don Miller, Storage Systems Division Software Products - Tucson
- Jerry Pence, Storage Systems Division Software Products - Tucson
- Joerg Pohlmann, IBM - Canada
- Wayne Rhoten, Storage Systems Division Software Products - San Jose
- Debbie Saintonge, MVS Systems Management and Solutions Design - Poughkeepsie
- Sanjay Shyam, Storage Systems Division Software Products - San Jose
- Dave Thompson, Storage Systems Division Software Products - San Jose
- John Thompson, Storage Systems Division Software Products - Tucson
- Nancy Young, Storage Systems Division Software Products - San Jose
- Pong Wang, Storage Systems Division Software Products - San Jose.

Thanks are also due to the many other collaborators and reviewers who contributed to improve this book. We especially wish to acknowledge our technical editor, Maggie Cutler, whose work substantially improved the quality of this book.

International Technical Support Organization - San Jose Center
July 1994
Chapter 1. Introduction

DFSMS/MVS* Version 1 Release 2.0 introduces new functions and services and data compression. In this chapter we briefly review the most important functions and present an overview of compression.

1.1 Extended Remote Copy and Peer-to-Peer Remote Copy

Enhanced disaster recovery is available through the new remote copy extended function of the IBM 3990 Model 6 in concert with DFSMS/MVS Version 1 Release 2.0. Remote copy comes in two forms: extended remote copy and peer-to-peer remote copy, at channel (OEMI or ESCON*) distances. The main difference between the two forms is this: extended remote copy allows immediate completion of an update and then asynchronously updates the second device; peer-to-peer remote copy does not allow the update to complete until updates are copied by the primary device to the secondary device synchronously.

1.2 Support of Distributed FileManager/MVS Server Implementation

Distributed FileManager/MVS enables transparent remote access to the pool of MVS/ESA* data and storage resources by workstation applications. Data accessed can be in byte or record format. Thus workstation applications can benefit from the vast amount of MVS/ESA’s reliability, security, and availability as well as take advantage of storage resources such as data sharing, system-managed storage, centralized data access, file security access, and high-performance storage access. Distributed FileManager/MVS adds Distributed Data Management (DDM) server capability to MVS/ESA. The DDM architecture offers a vocabulary and a set of rules for sharing and accessing data among like and unlike computer systems. The collection of the various components involved in DDM product implementation is called Distributed File Management. OS/2* client support is provided by Data Access Services (DAS), which is available as a component of ADSTAR* Distributed Storage Manager (ADSM).

1.3 VSAM Partial Release of Space and Secondary Volume Space Allocation

VSAM partial release of space allows the partial release of DASD space for VSAM extended format data sets. Extraneous space at the end of a VSAM extended format data set can be released either at CLOSE processing of the data set or during DFSMShsm* primary space management.

VSAM secondary volume space allocation enables the specification of the secondary amount of space when extending to subsequent volumes. Before DFSMS/MVS Version 1 Release 2.0, extension to subsequent volumes for VSAM data sets used the primary space amount. Now there is a choice.

---

1 Peer-to-peer remote copy is ESCON only.
1.4 Improvements in DFSMShsm

Problems associated with virtual storage constraint, tape recycle performance, and tape output processing have been addressed. The DFSMShsm load module has been moved to the extended private address space, thus raising the maximum concurrent tasking level from 25 to greater than 60, based on a 7MB region. Tapes are now recycled in order of emptiest to fullest rather than in collating sequence. Up to 15 tasks can process a single recycle command, and a new parameter allows the continuous allocation of the input tape drive, minimizing time delays from drive contention. Multiple tape output buffers will improve performance for primary level 0 DASD direct-to-tape processing when DFSMSdss is the data mover.

1.5 Improvements in DFSMSrmm

DFSMS/MVS Version 1 Release 2.0 provides full support for system-managed tape enhancements, including volume entry for manual tape libraries, provision of data needed for decisions in use of the volume-not-in-library installation exit, and provision of a new return-to-scratch exit, which enables interfacing to an external inventory manager. DFSMSrmm support for scratch pools is extended to allow installation exit control of pool selection using controls such as job name and data set name.

1.6 System-Managed Tape Enhancements

There are four new enhancements in this area:

- Volume-not-in-library installation exit
- Full support for the manual system-managed tape library—the IBM 3495 Tape Library Dataserver Model M10 function installation tape exits
- Systemwide default, which allows users to delete unexpired system-managed DASD data sets through JCL DD statements.

1.7 Backup-While-Open Improvements for CICS VSAM Data Sets

Backup-while-open (BWO) improvements incorporate code from CICS, CICSVR, DFSMSdss, DFSMShsm, Catalog Services, and VSAM to allow users of the CICS database to make dumps (backup copies) of their data sets that are in open-for-update mode and at the same time preserve data integrity. BWO with concurrent copy solves all known data integrity and security problems. Before this release, there were major limitations in the use of BWO. Now concurrent copy used along with BWO by DFSMSdss allows backups to be taken with integrity even when control area (CA) and control interval (CI) splits and data additions are occurring for VSAM key-sequenced data sets (KSDSs).

CICSVR uses the BWO stamp for the data set in the integrated catalog facility (ICF) set by DFSMSdss and DFSMShsm when the data set is restored, to determine the forward recovery start point on the log. The controls built into the BWO design allow CICS to prevent access to the data set after a data set restore and before the forward recovery logs are applied by CICSVR to ensure integrity of the data accessed by the user.
1.8 Support for More than Eight MVS Images in an SMSplex

A Storage Management Subsystem complex (SMSplex) is a system (MVS image) or collection of systems that share a common configuration including a common active control data set (ACDS) and communications data set (COMMDS) pair. DFSMS/MVS Version 1 Release 2.0 supports more than eight systems while maintaining the format of the control data sets. This new support allows the definition of system group names in the DFSMS/MVS Version 1 Release 2.0 configuration. A system group name may represent multiple systems.

The SMSplex function supports IBM’s S/390 Parallel Sysplex architecture and applies only to JES2 environments.

1.9 Device Allocation Enhancements

The volume selection algorithm has been modified to use a “best fit” for performance criteria (for volume selection) for new data set allocations and data set extensions. This change will accommodate future device types and allows allocation to better determine device selection based on performance items specified in the selected storage class. Such an improvement optimizes the use of the storage hierarchy.

1.10 Object Access Method Object Tape Support

Object Access Method (OAM) now supports objects on IBM cartridge system tapes. These tapes may be standard or enhanced capacity, and they may be written in compacted or noncompacted form. The tapes may be inside a tape library or used with stand-alone tape devices. OAM tape support extends the storage hierarchy for storing objects to tape, as well as DASD and optical, and enables primary and backup copies of objects and object transition up and down the storage hierarchy.

1.11 OAM Statistics

The OAM statistics feature meets customers’ requirements for additional data for system monitoring of transaction activity and workload measurements. This enhancement supports the recording function of the Resource Measurement Facility (RMF) through subcomponents of OAM.

1.12 OAM Support for the 3995 Optical Library Dataserver Model 133

OAM is enhanced to support the IBM 3995 Optical Library Dataserver Model 133 and the Model 133 as ESCON attached. The Model 133 includes the capability to read and write single-capacity rewritable cartridges written by the Model 131, single-capacity write-once-read-many (WORM) cartridges written by the Model 132, and double-capacity rewritable cartridges and double-capacity WORM cartridges.
1.13 Support for OpenEdition MVS

DFSMS/MVS Version 1 Release 2.0 improves the connection of heterogeneous systems to MVS/ESA by providing in OpenEdition MVS a hierarchical file system (HFS). The file system conforms to POSIX standard 1003.1 and a subset of 1003.1a, providing an application programming interface (API) for both C and Assembler languages. Application portability is at the source level; therefore POSIX applications compiled and executed on one POSIX-conforming system can be recompiled on another POSIX-conforming system and executed.

1.14 BSAM, QSAM, and VSAM KSDS Data Compression

DFSMS/MVS Version 1 Release 2.0 uses MVS compression services (with QSAM and BSAM data sets, and VSAM KSDSs) to exploit the hardware compression facility when present and provide software compression services when it is not present. See Appendix A, “Informational APAR II07816” on page 61 for information about BSAM and QSAM temporary data sets.²

The rest of this book deals with the data compression function, which reduces the space required to store, on any storage media, certain kinds of data.

Depending on the type of implementation, compression functions can be located in different places within the chain of hardware and software components that manage the data, such as the processor, channel subsystem, control unit, and transmission facilities. Compression is achieved by passing data through an algorithm developed for this purpose. This algorithm can be implemented as a software function, a hardware function, or a mixed hardware and software function.

Figure 1 on page 5 highlights areas where hardware compression previously took place. Improved data recording capability (IDRC) has been available on IBM cartridge system tape control units since 1988. The data is compacted in the control unit, typically with a ratio of 3:1 compaction depending on the data content. Compression or compaction can also take place at the workstation and/or at the modem, reducing data stored locally as well as the amount of data transmitted over the network.

Compression can also take place within the IBM ES/9000 processors using the S/390 hardware compression feature.

Some subsystems use this compression function directly, for example, DB2 and IMS/ESA, but the main topic of this book is the implementation of MVS compression services through DFSMS/MVS Version 1 Release 2.0 using access method compression for DASD sequential data sets and VSAM KSDSs.

² A temporary data set is a data set for which the user has not defined a data set name or has defined a data set name that begins with an ampersand (&).
IBM has implemented compression in access methods, a function within the system software that manages access to the data (the access method) and uses, when available, the processor hardware compression facility.

The implementation of compression fulfills the following requirements:

- Improved DASD space utilization
- Improved cache utilization
- Reduced I/O processing to access data
- Reduced channel, controller, and device utilization to process a given amount of data.

In some cases the implementation of compression relieves the design restriction of 4GB as the maximum size of a VSAM KSDS because you can have more data in each CI without changing the size of the 4-byte relative byte address (RBA). The increase is in proportion to the compression ratio.
The many system changes are best put into perspective by looking briefly at the setup and processes DFSMS/MVS Version 1 Release 2.0 uses to provide compression.

In this chapter we briefly review the hardware and software prerequisites as well as changes to other aspects of managing and creating compressed data sets.

We also review the compression process and cover the changes made in the everyday tools end users and storage administrators use in their data management tasks.

### 2.1 Hardware and Software Requirements

The hardware compression feature is in 711-, 511-, and 211-based IBM ES/9000 processors. This feature enables hardware compression at specific Licensed Internal Code levels in conjunction with DFSMS/MVS Version 1 Release 2.0. IBM ES/9000 711-based processors require SEC C35945 level code, and 511-based processors require SEC 228250 level code.

Other processors can provide compression through software emulation of the hardware feature only if they support MVS SP V4.3 with PTF UY91011 or higher and run DFSMS/MVS Version 1 Release 2.0.

Currently, compressed data sets can be created and processed only on the following ESCON-attached DASD control units:

- IBM 3990 Model 3 with ESCON attachment and feature code #6201, #6202, #6203, #6204, or #9203
- IBM 3990 Model 6 ESCON attachment
- ESCON-attached 9340 cached controller model DC4 with EC 486392.

### 2.2 Compression Processes and Procedures

With DFSMS/MVS Version 1 Release 2.0 installed, during the IPL of the system a Compression Services Available message is generated indicating verification of the necessary software.

Next come the actual processes involved in compressing the data, such as sampling the data content and matching appropriate dictionary tokens.

The ability to compress data in this system environment is called variously access method compression, DFSMS/MVS Version 1 Release 2.0 compression, or hardware compression, according to the different aspects or facilities used to compress the data. In this book we use the term *Compression Management Facility*, which is the term the system uses when DFSMS/MVS Compression Services are present.

The Compression Management Facility consists of two parts:

- Compression Services Activation, which covers checking and setting up the required compression environment.
• Compression Management Services (CMS), which covers the actual process a data set goes through when compression is requested.

2.2.1 Compression Services Activation

Compression Services Activation is invoked during SMS address space initialization and determines that the compression facilities are available.

This activity requires that the new dictionary building block (DBB) distribution library, SYS1.DBBLIB, an integral part of DFSMS/MVS Version 1 Release 2.0, be accessible through the master catalog. The DBB distribution library can be either a partitioned data set (PDS) or a partitioned data set extended (PDSE). It is through the consultation and selection of these building blocks that compression is eventually achieved as CMS matches the most suitable building blocks to form tokens associated with the compression of the data set. Information about the mix of DBBs used is kept with the data set in the catalog entry so that the data set can be easily compressed and decompressed when necessary.

2.2.2 Compression Management Services

CMS provides and carries out the following services used by VSAM, QSAM, and BSAM compression:

• Candidate data set verification
• Dictionary selection
• Data compression and decompression
• Generic dictionary creation.

Figure 2 depicts the activities of Compression Management Services. These activities are described in detail in the text that follows.
Candidate data set verification provides extensive filtering of the candidates for compression. It ensures the optimal use of system resources by rejecting small candidates or candidates that do not compress efficiently. For example, Compression Management Facility tests the viability of compressing the data set, and it may not compress the data set because of data set size, diverse data content, low compression ratios, or small records.

Although the data class can request compression, certain requirements must be met before the data set content is sampled for compression and best dictionary match (see 6.2, “Data Set Creation” on page 44).

The verification details for both sequential and VSAM eligibility are covered in 6.2, “Data Set Creation” on page 44. Candidate data set verification is used after a data set is allocated with a data class requesting both compression and extended format.

Dictionary selection involves the sampling and interrogation of the eligible data set. A sample from the eligible data set is taken and compared to the DBBs for similar content and compression efficiency. Up to 64KB of the data set can be sampled before the data set is compressed. The first time a compressed format data set is written to disk, the first bytes are written in noncompressed form. Because the dictionary is created during the sampling and/or interrogation, the dictionary does not exist when the first bytes of the data set are written. Once the dictionary is built, the data can use this dictionary to compress the rest of the data set. Information about the mix of DBBs selected during the sampling and interrogation process is kept in the catalog. This information enables the decompression of the data set when required but does not require a dictionary to be stored with the data set.
Data compression and decompression are invoked whenever read and/or write or get and/or put is done on a mated data set. A mated data set is one that has acquired a suitable dictionary token through successful interrogation and sampling processes. In other words, suitable building blocks have been found and selected from the DBB distribution library, and their combination constitutes the dictionary token associated with the data set and held in the catalog entry as well. Compression and decompression of a mated data set use the dictionary built from the dictionary token associated with the data set entry in the extended format cell of the catalog.3

The dictionary is customized to the data set through the dictionary selection process. However, the dictionary is not stored with the data. Only the token information is stored because the dictionary is a table dynamically built in storage from the token information that enables the compression and decompression of a data set when it is opened.

2.3 System Services and Facilities

To support the new functions DFSMS/MVS Version 1 Release 2.0 introduces changes in several system service components and in different subsystems.

2.3.1 System Services

Catalogs  ICF catalogs are not eligible for compression but are enhanced to contain additional information about compressed format data sets in the extended format cell of the catalog entry.

The extended format cell is part of the VSAM volume data set (VVDS) component of the ICF catalog.

The extended format cell holds:
- Number of stripes
- Compression flags
- Physical block size
- Noncompressed user data set size in bytes
- Compressed user data set size in bytes
- Active dictionary token
- Compression characteristic record.

(Figure 19 on page 34 shows the extended format cell information from a LISTCAT as well as the catalog additions.)

SMF  System Management Facility (SMF) is enhanced to record compression information in the type 14 and 15 records when CLOSE is performed on a sequential compressed format data set. Information on the dictionary token selected and the compressed and noncompressed data set size is added to the SMF type 14 and 15 records. New fields have also been added to SMF type 64 records for VSAM compression. These new fields record the size of the data before and after compression, flags for extended format and compression, and dictionary tokens used for compression. You can find more detailed information on the SMF record layout in the MVS/ESA SP V5 System Management Facilities (SMF), GC28-1457.

3 There is only one dictionary token per data set. The token consists of the DBB identifiers.
This information is also maintained in the extended format cell in the catalog.

Note that the type 14 and 15 records contain new information that is not related to compression. For instance, a new field, containing the date of OPEN, has been added. The field is inserted in the middle of the record for all data sets.

**DADSM**
Another new facility introduced in DFSMS/MVS Version 1 Release 2.0 is the partial release for VSAM extended format KSDSs. This function is built into DADSM and is activated through either JCL or management class.

**VTOC**
There is now a compression indicator, DS1COMPR, and an extended format data set indicator, DS1STRP, in the VTOC. Because a compressed format data set must be an extended format data set, both indicators are on for compressed format data sets.

### 2.3.2 Subsystems

**IMS**
IMS uses the hardware compression feature independent of access method compression services implemented in DFSMS/MVS Version 1 Release 2.0. IMS/ESA allows the use of hardware compression for segments in its databases. Therefore, IMS provides its own compression support directly and does use extended format or compressed format data sets.


**DB2**
DB2 uses the hardware compression feature independent of access method compression services implemented in DFSMS/MVS Version 1 Release 2.0. DB2 implements hardware compression for its data tables and provides its own compression support directly. It does not use extended format or compressed format data sets.

**CICS**
CICS does not use the hardware compression facility directly but can use compressed format VSAM KSDSs without changes.

**VTAM**
VTAM can also use hardware-assisted compression but not through DFSMS/MVS Version 1 Release 2.0 access method compression services.

### 2.4 System Utilities and DFSMS/MVS Components

Many utilities and DFSMS/MVS components have been enhanced to support compressed format data sets, without new or modified interfaces. Below is a summary of the supporting functions. The use of the utilities with compressed format data set support is covered in detail in 5.2, “Utilities to Create an Extended and Compressed Format Data Set” on page 33.

---

4 Partial release works on extended format VSAM KSDSs regardless of whether or not they are compressed.

5 VTAM has implemented several algorithms, not only the hardware-assisted algorithm.
**IDCAMS**

IDCAMS allows you to create and manage the new data set formats. The most important changes are:

- **IMPORT/EXPORT**—can be used to create a compressed format data set, but care should be taken when using this facility to move data to another system that may not have the same compression facilities available.

- **REPRO**—has been changed to read and write the compressed format data set without decompressing and compressing when written to a like device and with the same data set attributes. REPRO can be used to create a compressed format data set.

- **DCOLLECT**—is enhanced to report on overallocated space at the end of a VSAM extended format data set. It also reports on the user data size and the compressed user data size as discussed in 5.2, “Utilities to Create an Extended and Compressed Format Data Set” on page 33. In other words DCOLLECT supports the additional information held in the extended format cell of the catalog but does not report on the number of stripes a data set has.

**IEHLIST**

IEHLIST is enhanced to show whether a data set is in compressed format.

**DFSMSdss**

DFSMSdss supports the movement of compressed format data sets but cannot create compressed format data sets as none of its functions allows the specification of the required data class attributes.

**DFSMShsm**

DFSMShsm supports the migration, recall, backup, and recover of compressed format data sets. DFSMShsm bypasses its own compression facilities when managing these data sets, which are already compressed. DFSMShsm also supports VSAM KSDS partial release through management class.

**ISMF**

ISMF supports the additional information on the compressed format data sets. These changes are covered in Chapter 4, “Storage Management Administration” on page 17.

**IEB-**

IEBGENER, IEBPTPCH, IEBCOMPR, IEBDG, IEBEDIT, IEBUPDTE, and IEBCOPY support compressed format data sets without changes.

**IEH-**

IEHPROGM supports compressed format data sets without changes.
Chapter 3. Implementation of the Extended Format

Compression is a new method of physically storing data on DASD. Currently compression can be used for storing permanent sequential data sets and VSAM KSDSs. It is based on a new data set format called extended format. Its most remarkable characteristics are:

- All data set physical blocks have the same length.
- The system adds to each physical block a suffix of 32 bytes before storing the block on DASD. This suffix contains control information that the system uses.
- The block format and the suffix are transparent to the application, that is, the application does not require internal or external modifications to create and use the new data set format.
- Extended format data sets must be system-managed.

Extended format data sets are the prerequisite for creating a striped or a compressed data set. Figure 3 shows the relationships among these data set structures.

**Note:** A sequential extended format data set is also known as an extended sequential data set (introduced in DFSMS/MVS Version 1 Release 1.0).

---

**Figure 3. Extended Format Data Sets and Compression**
From Figure 3 on page 13 you can see that an extended format data set can also be regarded as a single stripe data set; however, when allocated on more than one volume, it becomes a multistriped data set. The multistriped format is available only for sequential data sets.

Programs that use the standard access methods can access an extended format data set. However, execute channel program (EXCP) processing cannot be used.

3.1 New Block Formats

When a data set is allocated in an extended format or is compressed, the system introduces some modifications to the record structure that are required to hold the necessary control information.

Because these modifications are not available to the user or the programming interface, you will not be required to modify programs or procedures that use the new format data sets. However, in some cases this additional control information may require you to review the criteria you use to calculate and allocate space for the new data sets.

The modifications are as follows:

- For the extended format
  - A 32-byte suffix is added to each physical record (block)

- In addition, for the compressed format
  - A 3-byte record prefix is added to each logical record (sequential data set and VSAM KSDS) or to each segment of a spanned record (sequential data set only)
  - A 5-byte record prefix is added to each segment of a spanned record (VSAM KSDS only).

3.2 Implementation for Compressed Format

Once a sequential data set or VSAM KSDS allocation has met the data set eligibility requirements, the candidate data set content is sampled during the write of the data to the eligible DASD device by the CMS routines. The data set is said to be compression eligible at this point, because it has been assessed as eligible for compression, and the catalog has noted that the data set is eligible for compression in the extended format cell.

Up to the first 64KB of the data set can be written in noncompressed form to the DASD during the sampling and interrogation of the content. The sampling and interrogation enable a dictionary token, which is made up of a set of DBB identifiers, to be associated with the data set rather than the whole dictionary. These token codes are kept in the catalog and form part of the extended format cell in the catalog entry for the data set.

A compression-eligible data set can exist in one of three forms:

- Dictionary selection The data set is eligible but its makeup is yet to be determined. The dictionary tokens are compared with the data set contents during interrogation and sampling. Interrogation maps bytes into alpha, numeric, upper- or lower-case, and sampling evaluates DBB compression efficiency.
The data set is mated with the appropriate dictionary; this concludes the sampling and interrogation processes.

A suitable dictionary match was not found during sampling and interrogation, so compression is bypassed, or for a sequential data set the data set was closed before a token could be selected. Note that the data set is in compressed format.

3.3 DASD Space Usage Considerations

The suffix for extended format data sets (not compressed format) is transparent to the application accessing the data, but there may be the unlikely occasion when the presence of the suffix changes the number of blocks written on a track, and hence the space usage on DASD. More details about the DASD space usage is covered in Appendix C, “Performance Measurement Details” on page 69.

3.4 Application Program Dependencies

Application programs can use an extended format data set if they are not dependent on the precise physical structure of the data. In fact, the application program does not see the 32-byte suffix or the new physical block or CI structure. Those three elements are managed by the access method only.
Chapter 4. Storage Management Administration

Many of the new functions in DFSMS/MVS Version 1 Release 2.0 are implemented through the use of SMS constructs, that is, the classes and automatic class selection (ACS) routines. The data class now supports compression of DASD data sets as well as compaction of tape data sets. The main interface for the storage administrator, that is, ISMF, has been enhanced to support the new format data sets. In this chapter we outline the changes a storage administrator should be aware of. We cover such VSAM enhancements as partial release of space and multivolume support, as well as device allocation enhancements.

4.1 ISMF

ISMF is enhanced to support extended and compressed format data sets, and many panels have been updated to provide information about these new facilities.

ISMF is an integral part of the systems management of data and the recommended interface to view all information, especially space details, about extended and compressed format data sets.

A few new attributes are associated with the allocation and data set information held about extended and compressed format data sets, and the panel enhancements have been made to support these. You, as a storage administrator, can now use these new attributes for sort, list, view, and data class define to identify, monitor, and assess compression information in your installation.

We do not recommend using the ISPF menu 3.4 line operator short form (S) and the more detailed form (I) when you obtain data set space information about extended and compressed data sets. Instead, we recommend that you use ISMF. The data set list application in ISMF is enhanced and now contains five selection panels (previously there were four) that support the new attributes associated with the implementation of DFSMS/MVS Version 1 Release 2.0.

The Data Set Selection Entry Panel in Figure 4 on page 18 is useful for identifying data sets with varying compression values. The panel shows how to identify low compression ratio data sets, which you may want to exclude from being compressed in the future.
To further limit the generated list, specify a single value or range of values in any of the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>REL OP</th>
<th>VALUE</th>
<th>AND/OR REL OP</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocated Space</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block/CI Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creation Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expiration Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Backup Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Reference Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Extents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Stripes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal Block/CI Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Space Not Used</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% User Data Reduction</td>
<td></td>
<td></td>
<td>LE 25</td>
<td></td>
</tr>
<tr>
<td>Record Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secondary Allocation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use ENTER to perform selection; use UP/DOWN command for other selection panels; use HELP command for help; use END command to exit.

Figure 4. Changed Data Set Selection Entry Panel

Figure 5 shows the additional panel added to the data set list application to further refine selection criteria. This panel is especially useful when associated with the many other changes in DFSMS/MVS Version 1 Release 2.0 concerning management of distributed data and data types new to MVS and systems management.

The Data Set Selection Entry Panel in Figure 6 on page 19 is used to select only those data sets in compressed format. A data class could be specified in this selection panel as all compressed data sets have to have a data class with
COMPACTION=YES and DATA SET NAME TYPE=EXTENDED required or preferred.

TO FURTHER LIMIT THE GENERATED LIST, SPECIFY A SINGLE VALUE OR LIST OF VALUES IN ANY OF THE FOLLOWING:

<table>
<thead>
<tr>
<th>REL OP</th>
<th>VALUE</th>
<th>VALUE</th>
<th>VALUE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCATION UNIT</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGE INDICATOR</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPACTION FORMAT</td>
<td>=&gt;</td>
<td>EQ</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>DATA CLASS NAME</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA SET ENVIRONMENT</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA SET NAME TYPE</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA SET ORGANIZATION</td>
<td>=&gt;</td>
<td>(1 to 8 values)</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>DDM ATTRIBUTES</td>
<td>=&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVICE TYPE</td>
<td>=&gt;</td>
<td>(1 to 8 values)</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>ENTRY TYPE</td>
<td>=&gt;</td>
<td>(1 to 12 values)</td>
<td>=&gt;</td>
<td></td>
</tr>
</tbody>
</table>

USE ENTER TO PERFORM SELECTION; USE UP/DOWN COMMAND FOR OTHER SELECTION PANELS; USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

Figure 6. Data Set Selection Panel with Compression Attribute

Once you have entered your selections for these attributes, the relevant data set list panels are presented with the new attributes incorporated into them. All associated LIST, VIEW, and SORT panels support the new and changed attributes by displaying new columns as follows:

- Compressed Format—values of YES or NO, indicating whether the data set is in compressed or noncompressed format.
- % User Data Reduction—achieved by the access method compression services.

The Data Set List panel in Figure 7 shows the columns mentioned in a VIEW selected through the VIEW facility from a data set list selection.

Figure 7. New Compression Attributes in Data Set List
The Data Set List panel in Figure 8 shows the rest of the new and changed attributes through another VIEW of the information available:

- **Data Set Name Type**—new value of EXTENDED now associated with compressed data sets as well as extended format data sets.
- **DDM attributes**—associated with DDM facilities introduced in DFSMS/MVS Version 1 Release 2.0.
- **CCSID Description**—the coded character set identifier used and associated with new facilities introduced in the DFSMS/MVS Version 1 Release 2.0 character data representation architecture (CDRA) repository.

![Figure 8. New DDM and Data Set Type in Data Set List](image)

### 4.2 SMS Classes

Compressed format data sets require that a data class be assigned to them, and some new and changed parameters in DFSMS/MVS Version 1 Release 2.0 support this requirement.

For a data set to be considered for compression, it must be given a data class with the attributes that support compression. You can either code the data class in the JCL, or filter and select in the data class ACS routine and in the storage class ACS routine for hardware dependencies. Further discussion and recommendations are in Chapter 6, “Rules to Create and Use Extended Format Data Sets” on page 43 onwards.

Regardless of the selection method you use you need to create new data classes for the sequential data sets and VSAM KSDSs you want to be considered for compression. The data classes must have the following parameters:

- **Data Set Name Type = EXTENDED**
- **Extended = Required or Preferred**
- **Compaction = YES.**

The Data Class Define panel in Figure 10 on page 21 shows the selection of extended for the data set name type. Note also the new option for the HFS introduced in DFSMS/MVS Version 1 Release 2.0 in support of OpenEdition/MVS.

If the data set name type is extended, you must choose either REQUIRED or PREFERRED. Be careful in your selection here as the parameters have two distinct outcomes:
• If you select **REQUIRED**, the allocation will fail if all compression selection and verification criteria are not met and a suitable volume is not available.

• If you select **PREFERRED**, the allocation will continue as nonextended and noncompressed when the compression criteria and the hardware dependencies are not met, and the message shown in Figure 9 is issued in the job output.

```
IG017000I DATA SET CUST05.EXYCNS.VSAM.DS01 IS NOT ELIGIBLE FOR EXTENDED
FORMAT, ALLOCATION AS NON-EXTENDED CONTINUES

IG017163I COMPRESSION REQUEST NOT HONORED FOR DATA SET CUST05.EXYCNS.VSAM.DS03
BECAUSE DATA SET CHARACTERISTICS DO NOT MEET COMPRESSION CRITERIA, ALLOCATION CONTINUES
```

**Figure 9. Information Messages Issued When Using Extended Preferred**

The Data Class Define panels in Figure 10 and Figure 11 on page 22 indicate how to select the parameters for a data set to be considered for compression.

```
DATA CLASS DEFINE

COMMAND ==> SEPP02.RES.SCDS
DATA CLASS NAME: TDC

TO DEFINE DATA CLASS, SPECIFY:

DESCRIPTION ===>

RECORG ===> (KS, ES, RR, LS or blank)
RECFM ===> (any valid RECFM combination or blank)
LRECL ===> (1 to 32761 or blank)
KEYLEN ===> (0 to 255 or blank)
KEYOFF ===> (0 to 32760 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)

DATA SET NAME TYPE ===> EXT
                    (EXT, HFS, LIB, PDS or blank)

IF EXT ===> R
            (P=Preferred, R=Required, or blank)
```

**Figure 10. Data Class Definition Required Attributes: Page 1 of 2**
For completeness Figure 12 on page 23 is included to show the requirements for allocating multistriped extended format data sets. For a multistriped extended format data set to be allocated across more than one volume, the Sustained Data Rate value in the storage class has to be more than the data transfer rate associated with the devices used. Currently these values are 3 and 4 for IBM 3380 type DASD and IBM 3390 type DASD, respectively.
4.3 VSAM Partial Release and Secondary Volume Space Allocation

VSAM partial release, another new facility introduced in DFSMS/MVS Version 1 Release 2.0, releases overallocated space on extended format VSAM KSDSs. This facility is particularly useful when allocating VSAM compressed format KSDSs because the user cannot determine the allocation size before allocation.

Partial release of excess space on extended format VSAM KSDSs is activated in the following two ways:

- Coding SPACE=(.,RLSE) in the DD statement
- Assigning a management class with partial release of values:
  - CI - conditional immediate
  - YI - yes immediate.

VSAM CLOSE will request partial release processing of DADSM when either the CI or YI value is coded, provided that the data set is in extended format and opened for OUTPUT processing.

DFSMShsm space management performs partial space release when other values like YES and CONDITIONAL are in the management class selected for the VSAM KSDS.6

---

6 DFSMShsm releases space for single-volume VSAM KSDSs only; in contrast, at VSAM CLOSE, space is released regardless of whether the data set is a single-volume or multivolume data set.
You have to decide which partial release methods to allow in your installation. It is common to code release in JCL statements, but to date not for VSAM allocations, so JCL changes are required if you use this implementation method. If you select a management class with CI or YI, which can be done either through the ACS routines or coded in the JCL of VSAM definitions, alteration to the ACS routines and minor JCL changes may be required.

The following list summarizes the conditions necessary to ensure successful release of unused space at the end of a VSAM data set:

- Partial release currently supports only extended format VSAM KSDSs allocated in a single stripe (compressed or not).
- Only the data component of the VSAM cluster is eligible for partial release.
- Partial release processing does not support data sets defined with guaranteed space.  
  
- Alternate indexes (AIXs) opened for path or upgrade processing are not eligible for partial release. The data component of an AIX, when opened as a cluster, can be eligible for partial release.
- Partial release processing is not supported for CLOSE TYPE=T.
- Partial release processing is supported for VSAM CLOSE provided that:
  - Partial release is specified in the management class or the JCL space parameter in the DD statement.
  - The data set is defined as extended format.
  - The data set is opened for OUTPUT processing.
  - This is the last access method control block (ACB) closing for the data set, which includes all closes in the current address space, and other address spaces in the same system and other systems.

When partial release is invoked using either method, all space after the high-used RBA is released on a CA boundary up to the high allocated boundary. If the high-used RBA is not on a CA boundary, the high used amount is rounded to the next CA.

Another change in DFSMS/MVS Version 1 Release 2.0 is associated with VSAM and space allocation: multivolume VSAM allocations. The change applies only to extended format and compressed format VSAM KSDSs as the facility has the following prerequisites:

- RECORG = KS
- Data Set Name Type = EXT
- Volume count must be greater than 1.

A new parameter in the data class definition panels determines the allocation amount on subsequent volumes for multivolume allocations for VSAM KSDSs. The new parameter is called Add’l Volume Amount and is highlighted on the Data Class Define panel in Figure 13 on page 25, along with the other required parameter values.

The Add’l Volume Amount can have the following values to determine the space allocation on subsequent volumes:

---

7 At CLOSE, space is released for VSAM multivolume single-striped KSDSs that are defined with nonguaranteed space.
• P—primary allocation amount is used
• S—secondary allocation amount is used
• Blank—the system will default to using the primary allocation amount.

COMMAND ===>

DATA CLASS DEFINE Page 2 of 2

SCDS NAME: SEPP02.RES.SCDS
DATA CLASS NAME: TDC

TO DEFINE DATA CLASS, SPECIFY:

RETPOD OR EXPOT ===> (0 to 9999, YYYY/MM/DD or blank)
VOLUME COUNT ===> 3 (1 to 59 or blank)
ADDL VOLUME AMOUNT ===> P (P=Primary, S=Secondary or blank)
IMBED ===> (Y, N or blank)
REPLICATE ===> (Y, N or blank)
CISIZE DATA ===> (1 to 32768 or blank)
% FREESPACE CI ===> (0 to 100 or blank)
CA ===> (0 to 100 or blank)
SHAREOPTIONS XREGION ===> (1 to 4 or blank)
XSYSTEM ===> (3, 4 or blank)
COMPACI ON ===> (Y, N or blank)
MEDIA INTERCHANGE
MEDIA TYPE ===> (MEDIA1, MEDIA2 or blank)
RECORDING TECHNOLOGY ===> (18TRACK, 36TRACK or blank)

USE ENTER TO PERFORM VERIFICATION; USE UP COMMAND TO VIEW PREVIOUS PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

Figure 13. New Parameters Associated with VSAM Allocations

End users should be informed of a new data class with multivolume support so they can request the facility by coding the data class in the JCL. The alternative would be to assign a data class through the ACS routines.

A few points are worthy of consideration when implementing compression and using the new VSAM facilities, but as every installation is unique, it is difficult to make recommendations that would apply to all installations. New data classes have to be created to support compressed format data sets, but what about the data sets' subsequent management? It may not be necessary to create new management classes to support compressed format data sets because the operational profile of the data, for example, the access pattern and data set reference pattern, has not changed.8 The only change is in the physical format of the data set, so existing service levels should suffice. What is worthy of more consideration is the placement of the data sets and the backup and recovery procedures currently in operation. The hardware dependencies should be taken into account along with the utilities used. You can find more information on the utilities and recovery in 6.1, "DFSMS/MVS System Environment" on page 43.

---

8 On the other hand, you could take advantage of less migration in a homogeneous DASD type environment.
4.4 Device Allocation

Device allocation is enhanced to further encourage the mixture of the same device type but different performance capabilities within a storage group as well as adhere more closely to the service and performance objectives requested through the storage class.

The main benefit of the device allocation enhancement is to allow storage administrators to reduce the number of storage groups. The reduction in the number of storage groups allows more volumes to be considered in the selection process. More volumes considered in the selection process can increase the success rate for multivolume allocations and the volume utilization.

The changes apply only to nonextended noncompressed data sets. A new volume selection algorithm refines the list of volumes selected for allocation to match more closely the millisecond response time (MSR) in the storage class associated with allocation, rather than currently where the list of volumes selected contains not only the volumes that meet but also the volumes that exceed the MSR objective. This results in the selection of volumes with higher performance than the requested service level for the data and may not optimize the utilization of high performance devices.

Before DFSMS/MVS Version 1 Release 2.0 the primary volume list contained volumes that matched or exceeded the MSR in the storage class. This could result in an allocation on a cached device capable of an MSR of 6 even though an MSR of 25 was in the storage class.

Currently the device allocation algorithm selects volumes and groups them as follows into three groups of candidate volumes:

- Primary
- Secondary
- Tertiary.

The primary list consists of devices that fit the storage class requirements best. The least busy volume will be selected. It is possible that none of the devices in the storage groups selected meets the performance objective. It is also possible that not enough space is available on any of the candidate volumes to be placed on the primary list (see “How Does SMS Select Volumes” in MVS/ESA Storage Management Library Implementing System-Managed Storage, SC26-3123).

The secondary list contains volumes that can be used for allocation although they do not necessarily meet all of the objectives of the storage class. The secondary list is used when the allocation cannot be satisfied from the primary list and the order in which the volumes are processed is as follows:

1. Prevention of ABEND X37
   - Volumes that are below the high threshold of occupancy specified in the volume’s storage group are placed above volumes that are at or above the threshold level.

2. Volumes used for spill or overflow
   - Volumes that are in ENABLED status are placed ahead of volumes in QUIESCED status.
3. Performance characteristics

The requested MSR number is used to sort the volumes in ascending order so that the closest volume to your request is favored. If no volume can meet or exceed your request, the fastest volume is selected.

4. Capacity

Devices with larger amounts of free space are placed ahead of devices with a smaller amount of free space.

Within each major group the devices are sorted further into minor groups, for example:

1. Below threshold enabled
2. Below threshold quiesced
3. Above threshold enabled
4. Above threshold quiesced.

The tertiary group of volumes is ordered in the same way as the secondary group. A storage group that has fewer volumes than the specified volume count will have all volumes in that storage group marked as tertiary. If you use Tape Mount Management (TMM) DASD buffers, we recommend that you review their usage, because the use of the buffers has changed slightly. If the TMM DASD buffer has two volumes in its own storage group and the use of five volumes in a spill storage group selected through the ACS routines, consider the following scenarios:

- A tape allocation requesting more than seven volumes through TMM would mark both storage groups as tertiary.

  When there are no primary and secondary volumes, the enabled TMM volumes are preferred over the quiesced volumes.

- A tape allocation requesting five volumes through TMM would make the quiesced volumes preferred over the TMM volumes because the quiesced volumes are considered part of the secondary group in this instance.

In summary, when mixing devices with varying performance, you must analyze the performance of the data sets to be placed in the storage group. For example, if data sets that require fast performance occupy approximately 30% of the storage group capacity, 30% of the storage group volumes should be fast devices.

You must review the assignment of a storage class to data sets if the same storage class is assigned to all data sets allocated to the same storage group, when that storage group comprises devices of varying performance. In such circumstances the devices that are closest to the MSR will be overutilized, and the other devices will be underutilized.

When specifying an initial access response time (IART) value greater than zero (optical requested), you should specify an MSR that reflects the optical performance. If an optical device is not available, a fixed DASD device is selected based on the MSR value specified. When an MSR is not specified, if an optical device is not available, the default MSR prefers a 3390 with cache or DASD fast write active.

Note: When there are more spill volumes than the enabled storage group volumes, the spill storage group could be placed on the secondary list—if it can satisfy the volume count—and the enabled storage group volumes
could be placed on the tertiary list (cannot meet volume count). This would make spill volumes more preferred (secondary is preferred over tertiary).

If there are less or equal numbers of spill volumes than the enabled storage group, the enabled storage group is always preferred.

With TMM, we suggest dedicating spill volumes only for TMM. To get around the spill storage group being preferred, you can place the spill volumes (quiesced) inside the TMM storage group. This would result in your having only one storage group, which has both enabled and quiesced (spill) volumes. The storage group would be enabled. You can only do this if you have enough spare volumes to dedicate to TMM only.
Chapter 5. Tools and Utilities

In this chapter we describe the techniques and utilities you can use to create an extended format data set.

The extended format data set is a prerequisite for implementing data compression and multistripe data sets. Therefore, in this chapter we deal mainly with the techniques related to extended format, and we mention the compressed format and the multistripe organization only when they are meaningful.

5.1 Compression Evaluation Tool

The compression evaluation tool consists of documentation and code (available from your IBM representative) that analyze your existing data sets to determine the approximate compression ratio that is likely to be achieved and the associated resources required.\(^9\) We used some of the output from this tool in our measurements and so include a brief overview of the tool for your reference.

The compression evaluation tool helps assess the likely benefits of compression to an installation in three main areas:

- DASD space savings
- CPU cost and batch elapsed time savings
- CICS VSAM analysis.

5.1.1 DASD Space Savings

The primary tangible benefit of compression is DASD space savings. The tool assesses the likely compression factor for candidate data sets.

This part of the compression evaluation tool requires MVS/DFP* Version 3 Release 3 or later. The code is available as part of the compression evaluation tool package through your local IBM representative. The predicted compression factors from the customer data used for this book are shown in Table 9 on page 79. They can be compared favorably with the actual compression factors achieved by compressing the data using DFSMS/MVS compression services.

This tool actually samples the data sets identified by the customer and assesses the compression that could be achieved. The user can control the sampling rate and amount.

Figure 14 and Figure 15 on page 30 show sample DASD space savings output for sequential and VSAM KSDSs.

---

\(^9\) The evaluation tool is classified IBM Internal Use Only, so the IBM representative can only give extracts to customers, not all of the reports; and the tool must be removed from the customers’ systems.
5.1.2 CPU Cost and Batch Elapsed Time Savings

The CPU Report estimates the cost of CPU time required to process compressed format data sets. The analysis is based on SMF records extracted from a customer installation.

This part of the tool requires SAS** and an SMF extract of relevant records types from the customer environment. For VSAM analysis SMF record types 30, 62, and 64 are required, and for BSAM and QSAM, SMF record types 14, 15, and 30 are required.

Significant reduction in elapsed time may result for jobs that sequentially read compressed format data sets.

We used some of the information from the reports generated by SAS from SMF records extracted during the measurements. Once SMF records are extracted they can be processed to produce the reports shown in Figure 16 on page 31, Figure 17 on page 31, and Figure 18 on page 32. We show a report for the jobs writing sequential data sets in Figure 16 on page 31. This CPU report is by job showing the task control block (TCB) and service request block (SRB) times, which we added together to give our CPU time used in the measurement results in Appendix C, "Performance Measurement Details" on page 69.

This is the only part of the compression tool output that we used extensively as input to the relative measurements of the resources required to process ordinary
format, extended format, and compressed format data sets. To use the estimation process you need the CPU time but you also need to use the Data Set Analysis Report, which tells you how many times each data set was accessed. The report also tells you the CPU consumed by a job or performance group and which types of data were accessed by the job. You can use the report as input to the overhead estimator. To estimate the performance, you need to know the CPU consumed and the reads and writes done during a particular time period.

<table>
<thead>
<tr>
<th>JOB</th>
<th>TCB TIME</th>
<th>SRB TIME</th>
<th>ACTIVE TIME</th>
<th>RESIDENT TIME</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TW11PCG</td>
<td>4.38</td>
<td>2.07</td>
<td>16.07</td>
<td>16.07</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW12PCG</td>
<td>1.04</td>
<td>0.35</td>
<td>4.13</td>
<td>4.13</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW21PCG</td>
<td>8.03</td>
<td>0.40</td>
<td>23.21</td>
<td>23.21</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW31PCG</td>
<td>0.08</td>
<td>0.01</td>
<td>1.91</td>
<td>1.91</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW32PCG</td>
<td>67.10</td>
<td>1.82</td>
<td>176.65</td>
<td>176.65</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW41PCG</td>
<td>105.14</td>
<td>3.61</td>
<td>291.16</td>
<td>291.16</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW42PCG</td>
<td>153.65</td>
<td>16.14</td>
<td>394.91</td>
<td>394.91</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW51PCG</td>
<td>20.69</td>
<td>8.95</td>
<td>100.59</td>
<td>100.59</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW61PCG</td>
<td>3.23</td>
<td>0.87</td>
<td>9.28</td>
<td>9.28</td>
<td>BSAM</td>
</tr>
<tr>
<td>TW71PCG</td>
<td>2.43</td>
<td>1.62</td>
<td>8.12</td>
<td>8.12</td>
<td>BSAM</td>
</tr>
</tbody>
</table>

---

**ALL JOBS**

<table>
<thead>
<tr>
<th>TCB TIME</th>
<th>SRB TIME</th>
<th>ACTIVE TIME</th>
<th>RESIDENT TIME</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>365.77</td>
<td>35.84</td>
<td>1026.02</td>
<td>1026.02</td>
</tr>
</tbody>
</table>

**BSAM/QSAM JOBS = 10**

<table>
<thead>
<tr>
<th>TCB TIME</th>
<th>SRB TIME</th>
<th>ACTIVE TIME</th>
<th>RESIDENT TIME</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>365.77</td>
<td>35.84</td>
<td>1026.02</td>
<td>1026.02</td>
<td></td>
</tr>
</tbody>
</table>

Figure 16. CPU Report Details

The Data Set Analysis Report (Figure 17) provides useful information about the activity to and from the data set as well as some of the data set attributes.

The compression evaluation tool does not analyze or report on temporary data sets or data sets created and deleted in the same job or job step.

We did not use the Data Set Analysis Report information extensively in our measurements as all of the utilities did not use, perform, or access the data set in the same manner. For example, DFSMSdss does not perform data set OPEN or CLOSE when logically dumping VSAM KSDSs, so there is no information in the SMF record types 62 and 64, VSAM open and close, respectively. Therefore, there is no detailed information to use at the data set level because many of the tests that created and deleted the data sets within the job were not analyzed and thus are not included in any reports.

<table>
<thead>
<tr>
<th>CLUSTER NAME</th>
<th>WRITES</th>
<th>READS</th>
<th>EXCPS</th>
<th>CI SIZE</th>
<th>MAX LRCL</th>
<th>BLKSIZE</th>
<th>MBS USED</th>
<th>ACC/MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST03.STNCS.VSAM.DS01.DA TA</td>
<td>1685291</td>
<td>82773</td>
<td>11773</td>
<td>4096</td>
<td>1016</td>
<td>4096</td>
<td>323.53</td>
<td>5209.1</td>
</tr>
<tr>
<td>CUST05.STNCS.VSAM.DS02.DA TA</td>
<td>1300279</td>
<td>11773</td>
<td>24576</td>
<td>25258</td>
<td>24576</td>
<td>306.56</td>
<td>4270.8</td>
<td></td>
</tr>
<tr>
<td>CUST02.STNCS.VSAM.DS01.DA TA</td>
<td>218562</td>
<td>4752</td>
<td>16384</td>
<td>350</td>
<td>16384</td>
<td>74.53</td>
<td>2932.5</td>
<td></td>
</tr>
<tr>
<td>CUST01.STNCS.VSAM.DS02.DA TA</td>
<td>135367</td>
<td>1110</td>
<td>18432</td>
<td>135</td>
<td>18432</td>
<td>22.15</td>
<td>6111.8</td>
<td></td>
</tr>
<tr>
<td>CUST04.STNCS.VSAM.DS01.DA TA</td>
<td>207124</td>
<td>96</td>
<td>4096</td>
<td>22</td>
<td>4096</td>
<td>4.92</td>
<td>42082.3</td>
<td></td>
</tr>
<tr>
<td>CUST05.STNCS.VSAM.DS01.DA TA</td>
<td>207124</td>
<td>96</td>
<td>4096</td>
<td>22</td>
<td>4096</td>
<td>4.92</td>
<td>42082.3</td>
<td></td>
</tr>
</tbody>
</table>

**TOTALS**

<table>
<thead>
<tr>
<th>WRITES</th>
<th>READS</th>
<th>EXCPS</th>
<th>CI SIZE</th>
<th>MAX LRCL</th>
<th>BLKSIZE</th>
<th>MBS USED</th>
<th>ACC/MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3589355</td>
<td>101606</td>
<td>733.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 17. Data Set Analysis Report

The cost of the Resource Analysis Report is the basis of the SMF output report and can be further edited and analyzed to provide an estimated resource cost associated with access method compression. A comparison can be made of the resources required using the hardware feature in conjunction with compression.
services compared to the resources required using only the compression software services, as shown in Figure 18.

<table>
<thead>
<tr>
<th>PERF GRP</th>
<th>SECONDS</th>
<th>WITH HARDWARE</th>
<th>SOFTWARE ONLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEGPSG</td>
<td>67.0600</td>
<td>434.0 %</td>
<td>1803.5 %</td>
</tr>
</tbody>
</table>

CCOST511 INPUTS FOR: SEGPSG

CPU TIME: 67.0600 SECS
CURRENT PROCESSOR: 620
SYSTEM: SMS
NUMBER OF INPUTS: 10

<table>
<thead>
<tr>
<th>DATA</th>
<th>READS</th>
<th>WRITES</th>
<th>AVG REC</th>
<th>%BYTES</th>
<th>%HW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST03.S</td>
<td>1712619.00</td>
<td>0.00</td>
<td>200</td>
<td>50</td>
<td>75.08</td>
</tr>
<tr>
<td>CUST04.S</td>
<td>5512290.00</td>
<td>0.00</td>
<td>133</td>
<td>50</td>
<td>182.41</td>
</tr>
<tr>
<td>CUST04.S</td>
<td>2360948.00</td>
<td>0.00</td>
<td>241</td>
<td>50</td>
<td>119.03</td>
</tr>
<tr>
<td>CUST02.S</td>
<td>225456.00</td>
<td>0.00</td>
<td>121</td>
<td>50</td>
<td>7.03</td>
</tr>
<tr>
<td>CUST02.S</td>
<td>165986.00</td>
<td>0.00</td>
<td>256</td>
<td>50</td>
<td>8.76</td>
</tr>
<tr>
<td>CUST05.S</td>
<td>8884.00</td>
<td>0.00</td>
<td>22528</td>
<td>50</td>
<td>32.21</td>
</tr>
<tr>
<td>CUST06.S</td>
<td>191601.00</td>
<td>0.00</td>
<td>80</td>
<td>50</td>
<td>4.71</td>
</tr>
<tr>
<td>CUST07.S</td>
<td>112360.00</td>
<td>0.00</td>
<td>76</td>
<td>50</td>
<td>2.69</td>
</tr>
<tr>
<td>CUST01.S</td>
<td>66659.00</td>
<td>0.00</td>
<td>151</td>
<td>50</td>
<td>1.64</td>
</tr>
<tr>
<td>CUST03.S</td>
<td>12950.00</td>
<td>0.00</td>
<td>151</td>
<td>50</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Figure 18. Cost of Resource Analysis

5.1.3 CICS VSAM Analysis

This part of the compression evaluation tool deals with CICS VSAM systems accessing system-managed compressed records. It involves understanding access rates to VSAM KSDSs through the use of CICS statistics for an understanding of compression effects. The CICS VSAM analysis requires more in-depth knowledge of the particular CICS systems running in the customer environment but goes through a similar analysis path as the CPU Report.
5.2 Utilities to Create an Extended and Compressed Format Data Set

As we describe in Chapter 3, “Implementation of the Extended Format” on page 13, the most important prerequisite for creating an extended format data set is that it be system-managed. Such management is required because the definitions of extended format are in the SMS data class construct. This requirement leads us to the considerations discussed below.

5.2.1 IDCAMS

The functions listed below have been enhanced to support extended format and compressed format data sets.

REPRO  There are no external changes to REPRO. Internally REPRO can read and write in a compressed state, but only when it is writing the data set to a like format and a like device. To do this reading and writing REPRO takes into consideration the physical characteristics of the data set used as the target of the operation as well as the compression token in use on both source and target data sets. If both source and target data sets are eligible for compression, REPRO bypasses decompression and compression and replaces the target token with the source token; otherwise REPRO decompresses and compresses each record.

IMPORT  EXPORT and IMPORT can be used to convert a standard VSAM KSDS data set to a compressed format VSAM KSDS.

EXPORT and IMPORT are primarily used to move data sets to other systems or environments; in other words they create portable data sets. IMPORT and EXPORT move compressed data, and the extended format cell and the compression information are saved and propagated during the EXPORT and IMPORT process. Be aware that if the cluster is predefined and IMPORT is used with the INTOEMPTY parameter, extended format and compression information are not propagated. However, if the predefined data set is eligible for compression, IMPORT loads the records into the data set in compressed format. Exported compressed VSAM KSDSs are actually in uncompressed format, and thus it does not matter where to IMPORT them.

PRINT  PRINT supports compressed data sets with no changes and prints decompressed records.

LISTCAT  Extra information is added to the catalog entry when a data set is extended and compressed. This extra information is held in the extended format cell, which is in the VVDS part of the catalog. There is an extended format cell for every extended format data set entry in the catalog. For multivolume data sets, the extended format cell is in the VVDS of the first volume of the data set.

---

10 The output data set can be new and thus have no token.
The information printed from the extended format cell is as follows:

- Number of stripes
- Compression flags
- Physical CI size
- Noncompressed user data set size in bytes
- Compressed user data set size in bytes
- Active dictionary token.

New fields have been added to LISTCAT. Figure 19 shows a sample of a LISTCAT of a sequential extended format data set.

```bash
==> listcat ent('cust04.exycns.seq.file01')
NONVSAM --------- CUST04.EXYCNS.SEQ.FILE01
SMSDATA
  STORAGECLASS ----SC00000  MANAGEMENTCLASS--MC00000
  DATACLASS -------PSEXY01  LBACKUP ----0000.000.0000
ATTRIBUTES
  STRIPE-COUNT---------1
  ACT-DIC-TOKEN----X'40000009050005FE08FE0DFE0DE0AFE002000400F'
  COMP-FORMAT EXTENDED
  CCSID---------------------65535
STATISTICS
  USER-DATA-SIZE-----------------------------568952077
  COMP-USER-DATA-SIZE----------------------139188247
  SIZES-VALID----------(YES)
```

**Figure 19. LISTCAT Display**

LISTCAT displays new attributes for extended format and compression for extended format data sets. New statistics are the sizes before and after compression. SIZES-VALID indicates whether the reported sizes are valid.

**DCOLLECT** The user data set size and the compressed user data set size are reported in the output data set in two new fields, as shown in Figure 20. If DCDBDSZ is set, it indicates that the reported sizes are invalid. These values can be used for billing purposes. DCOLLECT does not report the number of stripes a data set has.

```bash
DCDCSID DS XL2  CODED CHARACTER SET IDENTIFIER
  DS CL2  RESERVED
  ORG DCUOORTH+262
DCDUDSZ DS XL8  USER DATA SET SIZE
DCDUCDSZ DS XL8  COMPRESSED USER DATA SET SIZE
DCDEXFLG DS XL2
DCDBDSZ EQU B'10000000'
DCDADSIE DS 0C  END OF DCUDSET
```

**Figure 20. DCOLLECT Support**
5.2.2 DFSMSdss

DFSMSdss in DFSMS/MVS Version 1 Release 2.0 supports extended format and compressed format data sets. The support DFSMSdss provides for sequential data sets differs from the support for VSAM.

5.2.2.1 VSAM KSDSs

The following DFSMS/MVS Version 1 Release 2.0 DFSMSdss functions support extended format VSAM KSDSs, AIXs, and compressed format KSDSs:

- Physical volume DUMP/RESTORE
- Physical volume COPY
- Stand-alone volume RESTORE
- Volume DEFRAG
- PRINT (compressed logical records are not expanded)
- COPYDUMP
- Logical data set DUMP (validate)
- Logical data set RESTORE
- Logical data set COPY.

The following DFSMS/MVS Version 1 Release 2.0 DFSMSdss functions do not support extended format VSAM KSDSs, AIXs, and compressed format KSDSs:

- CONVERTV (message ADR878E 99 is issued)
- Physical data set DUMP/RESTORE (message ADR972E is issued)
- Logical data set DUMP (novalidate) (message ADR971E 1 is issued)

There are some restrictions to the above:

- Logical data set COPY
  - If the source is an extended format VSAM KSDS, the target must be an extended format VSAM KSDS.
  - If the source is an extended format AIX, the target must be an extended format AIX.
  - If the source is a compressed format VSAM KSDS, the target must be a compressed format VSAM KSDS.
  - Source and target CI size must be equal.
  - IDCAMS REPRO is always used for copy.
  - The TOLERATE keyword (ignores an enqueue failure) is not supported.
  Note that you can do the conversion with IDCAMS REPRO outside DFSMSdss.

- Logical data set RESTORE
  - If the source is an extended format VSAM KSDS, the target must be an extended format VSAM KSDS.
  - If the source is an extended format AIX, the target must be an extended format AIX.
  - If the source is a compressed format VSAM KSDS, the target must be a compressed format VSAM KSDS.
  - Source and target CI size must be equal.

The following DFSMS/MVS Version 1 Release 1.0 DFSMSdss and DFDSS 2.5.0 functions support extended format VSAM KSDSs, AIXs, and compressed format KSDSs:
• Physical volume DUMP/RESTORE
• PRINT
• Volume DEFRAG
• Physical volume COPY
• Stand-alone volume RESTORE
• COPYDUMP.

The following DFSMS/MVS Version 1 Release 1.0 DFSMSdss and DFDSS 2.5.0 functions do not support extended format VSAM KSDSs, AIXs, and compressed format KSDSs:

• CONVERTV (message ADR878E 99 is issued)
• Physical data set DUMP (message ADR778E 4 is issued)
• Physical data set RESTORE (messages ADR285E 4 and ADR778E 4 are issued)
• Logical data set DUMP (validate) (message ADR778E 4 is issued)
• Logical data set DUMP (novalidate) (message ADR778E 4 is issued)
• Logical data set RESTORE (message ADR778E 4 is issued)
• Logical data set COPY (message ADR778E 4 is issued).

5.2.2.2 Sequential Data Sets

The following DFSMS/MVS Version 1 Release 2.0 DFSMSdss functions support extended format and compressed format sequential data sets:

• Physical volume DUMP/RESTORE
• Physical volume COPY
• Stand-alone volume RESTORE
• Volume DEFRAG
• PRINT (compressed records are not expanded)
• COPYDUMP
• Logical data set DUMP
• Logical data set RESTORE
• Logical data set COPY
• CONVERTV REDETermine the volume
• Physical data set DUMP/RESTORE.

Conversion is possible during logical restore:

• Extended format data set

DFSMSdss requests allocation of the extended format data set. If the data set cannot be allocated in extended format, it is allocated as a sequential data set.

• Compressed format data set (requires MVS/ESA with the proper level of CMS)

DFSMSdss requests allocation of the compressed format data set. If the allocate fails, DFSMSdss requests allocation as a sequential data set.

• Extended format data set to non-system-managed volume

When BYPASSACS and NULLSTORCLAS are specified, DFSMSdss requests allocation of a non-system-managed sequential data set. Note that DFSMSHsm specifies BYPASSACS and NULLSTORCLAS to DFSMSdss when FORCENONSMS is specified on a RECALL. Therefore, an extended format or compressed format data set can be recalled as a sequential data set to a non-system-managed volume.
Note: An extended format or compressed format sequential data set cannot be converted during copy.

The following DFSMS/MVS Version 1 Release 1.0 DFSMSdss functions (with OY67882) support extended format sequential data sets:

- Physical volume DUMP/RESTORE
- Physical volume COPY
- Stand-alone volume RESTORE
- Volume DEFRAG
- PRINT (compressed records are not expanded)
- COPYDUMP
- Logical data set DUMP
- Logical data set RESTORE
- Logical data set COPY
- CONVT TV REDE Termine of the volume
- Physical data set DUMP/RESTORE.

The following DFSMS/MVS Version 1 Release 1.0 DFSMSdss functions (with OY67882) support compressed format sequential data sets:

- Physical volume DUMP/RESTORE
- Physical volume COPY
- Stand-alone volume RESTORE
- Volume DEFRAG
- PRINT (compressed records are not expanded)
- COPYDUMP
- CONVT TV REDE Termine of the volume.

The following DFSMS/MVS Version 1 Release 1.0 DFSMSdss functions do not support compressed format sequential data sets:

- Logical data set DUMP
- Logical data set RESTORE
- Logical data set COPY
- CONVT NONS MS
- Physical data set DUMP/RESTORE.

Conversion is possible during logical restore

- Extended format data sets
  DFSMSdss requests allocation of the extended format data set. If the data set cannot be allocated in extended format, it will be allocated as a sequential data set.
- Extended format data set to non-system-managed volume
  When BYPASSACS and NULLSTORCLAS are specified, DFSMSdss requests allocation of a non-system-managed sequential data set. Note that DFSMSHshm specifies BYPASSACS and NULLSTORCLAS to DFSMSdss when FORCENONS MS is specified on a RECALL. Therefore, an extended format or compressed format sequential data set can be recalled as a sequential data set to a non-system-managed volume.

Note: An extended format sequential data set cannot be converted during copy.

The following DFDSS 2.5.0 functions (with PN48274) support extended format and compressed format sequential data sets:
• Physical volume DUMP/RESTORE
• Physical volume COPY
• Stand-alone volume RESTORE
• Volume DEFRAG
• PRINT (compressed records are not expanded)
• COPYDUMP
• CONVERTV REDETerm of the volume.

The following DFDSS 2.5.0 functions do not support extended format and compressed format sequential data sets:
• Logical data set DUMP
• Logical data set RESTORE
• Logical data set COPY
• CONVERTV NONSMS
• Physical data set DUMP/RESTORE.

5.2.3 DFSMShsm

DFSMShsm does not try to compact a compressed format data set on either backup or migration.

DFSMShsm uses DFSMSdss as the data mover to process compressed format data sets and does not request DFSMSdss to reblock a compressed data set when DFSMShsm is performing recall or recover.

A toleration PTF for DFSMS/MVS 1.1 prevents recall and recover from processing compressed format data sets.

DFSMShsm records in the control data set (CDS) inventory that the data set is compressed. DFSMShsm retrieves information from the ICF catalog extended format cell (which allows billing on a bytes-used basis) and stores the size of the user data set in a new field in the CDS record. This information then becomes available to DCOLLECT, which subsequently updates the DCOLLECT M and B records.

5.2.4 IEHLIST

IEHLIST indicates the data set that is listed as extended format and gives the extended attributes. Figure 21 on page 39 shows the various options in the SMS indicator fields. Note that the E indicator has changed from extended sequential to extended format—because extended format includes both extended sequential data sets and extended format VSAM KSDSs.
SMS-IND SYSTEM-MANAGED STORAGE ATTRIBUTES

- S = SMS-MANAGED DATA SET
- U = NO BCS ENTRY EXISTS FOR DATA SET
- R = DATA SET IS REBLOCKABLE
- B = OPTIMAL BLOCK SIZE SELECTED BY DASDSM CREATE
- I = DATA SET IS A PDSE
- E = DATA SET IS IN EXTENDED FORMAT
- C = DATA SET IS IN COMPRESSED FORMAT

Figure 21. IEHLIST Changes

The storage attributes are shown in the sample in Figure 22.

Figure 22. IEHLIST Sample

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>SER NO</th>
<th>SEQNO</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST02.SNCMP.SEQ.FILE01</td>
<td>VM5Q22</td>
<td>1</td>
<td>1994</td>
</tr>
<tr>
<td>SMS.IND LRECL KEYLEN INITIAL ALLOC 2ND ALLOC EXTEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S R 256 TRKS 22 1MB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTENTS NO LOW(C-H) HIGH(C-H)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 6 0 56 10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---ON THE ABOVE DATA SET, THERE

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>SER NO</th>
<th>SEQNO</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST04.ECOMP.SEQ.FILE01</td>
<td>VM5Q22</td>
<td>1</td>
<td>1994</td>
</tr>
<tr>
<td>SMS.IND LRECL KEYLEN INITIAL ALLOC 2ND ALLOC EXTEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S R E C 241 TRKS 33 1024MB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTENTS NO LOW(C-H) HIGH(C-H)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 56 11 239 8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---ON THE ABOVE DATA SET, THERE

5.2.5 IEBGENER

IEBGENER supports compressed format data sets and can transform data sets from one format to another as shown in Table 1 on page 40.

5.2.6 DFSORT

DFSORT uses the noncompressed user data set size information from the extended format cell of the catalog to determine the required size of the sort work data sets.

5.3 Summary

Table 1 summarizes the utilities discussed in this chapter. You can use this table as a quick reference to select the appropriate method of converting from one data set format to another.
<table>
<thead>
<tr>
<th>Utility</th>
<th>Function</th>
<th>Input Format</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Standard Format</td>
</tr>
<tr>
<td>IDCAMS (VSAM KSDS)</td>
<td>REPRO (DEFINE CLUSTER followed by REPRO)</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>IMPORT (data set not predefined)</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td>IMPORT (data predefined, INTOEMPTY specified)</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Supported</td>
</tr>
<tr>
<td>IEBGENER or ICEGENER</td>
<td>Data set copy</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td>(sequential)</td>
<td></td>
<td>Extended</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Supported</td>
</tr>
</tbody>
</table>

Table 2 on page 41 shows the DFSMS/MVS subsystem components that manage the new data sets formats.

**Note:** Input format for IMPORT means the VSAM KSDS cluster that is exported, not the portable data set that is input to IMPORT. There are two input data sets: the cluster that is input for the EXPORT command, and the portable data set, which is the output of the EXPORT command and the input to IMPORT.
<table>
<thead>
<tr>
<th>Utility</th>
<th>Function</th>
<th>Input Format</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Standard Format</td>
</tr>
<tr>
<td>DFSMSdss</td>
<td>Physical data set dump and restore</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Logical data set dump and restore</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Supported (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Supported (1)</td>
</tr>
<tr>
<td></td>
<td>Data set copy</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Not supported</td>
</tr>
<tr>
<td>DFSMShsm</td>
<td>MIGRATE and RECALL</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Supported (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Supported (1)</td>
</tr>
<tr>
<td></td>
<td>RECOVER from a data set backup copy</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td>RECOVER from a physical dump copy (2)</td>
<td>Standard</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressed</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>

**Note:**

(1) Sequential only.

(2) A DFSMSHsm recover of the entire volume, which translates to a physical full volume restore to DFSMSdss, is supported. Recover of an extended sequential data set (compressed or not), which translates to a physical data set restore to DFSMSdss, is supported. Recover of an extended VSAM KSDS (compressed or not), which translates to a physical data set restore to DFSMSdss, is not supported.

**Note:** Data set restore from a physical volume dump has the same values as physical data set restore in Table 2. Logical volume dump is nothing more than a logical data set dump of all of the data sets on a volume. Therefore, the values for data set restore from a logical volume dump are the same as the values for logical data set restore in the table.
Chapter 6. Rules to Create and Use Extended Format Data Sets

In this chapter we describe some techniques that are useful for implementing data compression in an installation and present some usage guidelines in an existing production environment.

As mentioned in Chapter 3, “Implementation of the Extended Format” on page 13, implementing the new data set formats in most cases does not require that you modify the programs and the procedures that access the data sets. However, you should consider that the new data set format has some hardware and software dependencies that, in some cases, may require you to review the existing environment.

6.1 DFSMS/MVS System Environment

The new format data sets can exist only on a system-managed volume, under SMS management. This implies that you must update your SMS system before implementing extended format data sets.

The necessary modifications are not many, and, as in all SMS systems, they can be implemented in different ways, according to your system environment and your needs. In this section we provide some guidelines that can help you in the implementation task.

6.1.1 System Setup

Before starting to implement the extended format, you must update your SMS system environment in the following way:

- Define one or more data classes with the EXT option in the DATA SET NAME TYPE field and with R or P (see 6.2.1, “VSAM Data Set Eligibility” on page 44) in the IF EXT field.
- One or more of these data classes must have Y in the COMPACTION field, to allow you to create compressed format data sets.
- Define one or more storage classes with a value greater than 4 in the SUSTAINED DATA RATE field. This allows you to create multistriped data sets that are defined with nonguaranteed space.
- Update your ACS routines with the necessary coding to recognize and manage the new allocations.

Another important thing you should consider is the need to modify your storage group definitions. You can:

- Create a storage group where you can address all of the allocations of extended format data sets
  or
- Allow the data sets in the new format to share the existing storage groups.

We recommend the first alternative when you have only a limited number of hardware resources that support the new formats. In this case you should control where the new format data set allocations are directed to avoid possible incompatibilities that result in failing the allocation. Of course, this requires that
you implement some coding in your ACS routines to be able to recognize and address these new conditions.

The second alternative is easier to implement because it requires fewer modifications to the existing environment. You can implement it if your storage subsystem consists mainly of devices that have the required characteristics to support the allocation of extended format data sets.

For both alternatives, you probably do not need to do anything at the management class level—unless you want to use partial release for VSAM KSDSs.

### 6.2 Data Set Creation

You can create a VSAM sphere in the extended format provided that you observe conditions related to the space allocation, data set organization, and the environment where the allocation is performed. The extended format sequential data sets can be striped, but the extended format VSAM KSDSs cannot be striped. The extended format data sets can be compressed, that is, they are in compressed format, but an extended format VSAM AIX data set cannot be in compressed format.

### 6.2.1 VSAM Data Set Eligibility

More variables and permutations are associated with VSAM than with sequential data set definition or allocation. Therefore, a few more rules must be met for successful allocation and processing of compressed VSAM data sets:

- **DSNTYPE=EXTended** through data class and R in the If EXT field.
- **COMPACTION=YES** through data class.
- Primary space allocation must be at least 5MB if a secondary extent is requested.
- Primary space allocation must be at least 8MB if no secondary extent is requested.
- The data set must be a KSDS.
- The data set must be a permanent data set.
- AIXs must be allocated as extended format data sets when the base is extended format.
- IMBED indexes are not allowed.
- Data cannot be split in KEYRANGES.
- The maximum record length for compressed nonspanned data sets is 3 bytes less than the maximum record length allowed with the standard format; it is the CI size minus 10 bytes because of extra control information.
- The minimum record length must be equal to the key offset plus the key length plus 40 bytes.
- For spanned data sets the key offset plus the key length must be less than or equal to the CI size minus 15.
- Device must be eligible to support compressed data sets.
If a KSDS is allocated as compressed format, both the data and index portion are in extended format, although only the data portion is compressed.

### 6.2.2 Sequential Data Set Eligibility

Before examining the content of a sequential data set, the following requirements have to be met for sequential data set compression:

- **DSNTYPE=EXTended** through the data class.
- **COMPACTION=YES** through the data class.
- Device must be eligible to support compressed data sets.
- Primary space allocation must be at least 5MB if a secondary extent is requested.
- Primary space allocation must be at least 8MB if no secondary extent is requested.
- If RECFM=F, V, or U, the minimum record size, excluding the record descriptor word (RDW) and block descriptor word (BDW), must be at least 40 bytes. For blocked records, BLKSIZE (- BDW and RDW) must be at least 40.

### 6.2.3 Sequential Data Set and VSAM KSDS Eligibility

The allocation has to be made with a data class that has the correct attributes specified and meets the following eligibility requirements:

- You must develop at least one data class construct that defines the extended attributes, either as preferred or required (must be required for VSAM).
- You can either develop an ACS data class routine to assign the extended attributes data class to selected data sets or decide to modify the procedures that allocate the data sets, adding the parameter DATACLASS that asks for the data class construct.

Remember that in some cases these two techniques do not work. They would not work, for example, if you were to create a data set using a utility or function that does not drive the ACS data class routine and does not allow you to specifically request the data class with the allocation request. All DFSMSdss functions behave in this way.

The following are exceptions or items to be aware of:

- Temporary sequential data sets are not eligible for compression.
- ICF catalogs, temporary, and other system VSAM KSDSs are not eligible for compression.
- A compressed KSDS is a short name for compressed extended format KSDS. Because it is an extended format data set, it is represented in the catalog as having a stripe count of 1. Although a compressed sequential data set can be multistriped, a compressed KSDS in DFSMS/MVS Version 1 Release 2.0 has a stripe count of 1.
- Any data sets that are accessed before SMS is active must not be compressed.
6.2.4 VSAM KSDS Space Considerations

In some cases, the modifications to the data set record structures that are required to support the extended and compressed formats can affect the way you define the KSDS clusters.

When you use DEFINE CLUSTER with the keyword MODEL, the space parameters used for the new cluster are taken from the primary and secondary allocations of the original data set, provided no other space-related parameter, such as RECORDS or SPACE=, is specified.

When creating an extended format VSAM KSDS from a standard format input, be aware that the space allocation for the extended format data set should be increased from that of the original standard format allocation. This increase may be considerably more than the extra control information associated with an extended format data set, especially when the space is requested in records. This increase mostly applies to data sets that are in noncompressed format.\(^{11}\)

6.3 Data Set Usage

Hiperbatch\(^*\) processing is not supported if the data set is an extended format data set.

6.3.1 Application Programs

Some restrictions apply when you write a program that accesses an extended format data set. These restrictions are related to the technique you use to read or write the VSAM KSDS:

- When data is compressed, you cannot determine the RBA of another record using the length of the current record.
- When data is compressed, the length of a stored record may change after an update without any length change.
- CI mode processing is not permitted unless you open the data set to perform a VERIFY. Any attempt to access the data using a request parameter list (RPL) that specifies CI processing will result in a logical error.
- Locate mode processing is supported but requires a larger number of internal work areas to process. In fact, when data is in a compressed format, VSAM must acquire an additional work area to expand the data from the I/O buffer before passing the data pointer to the program. The number of work areas needed for this action is equal to the number of strings (RPLs) used for locate mode processing. When shared resources are used with this processing mode, the number of work areas needed may grow to the number of strings specified for the resource pool. If you plan to use locate mode, consider that:
  - A storage area must be obtained for each string doing locate processing.
  - This storage area must be associated with the job step TCB.
  - At least one supervisor call instruction (SVC) is issued.
  - The area needed may be of large size, up to 32K.

\(^{11}\) When a compressed data set does not compress well or has a rejection token, the data set will be in compressed format and will probably incur more space than in noncompressed format.
The number of areas needed may be 255 for each ACB or each resource pool.

6.3.2 New Messages

Many new messages support compression. Figure 23 shows some successful allocation messages using compression and striping. The sample also shows that the new data set has passed through Compression Services Activation selection and met the requirements for sequential data set compression.

There are several new messages to be aware of, and occasionally they should be checked because even when a data class with EXTENDED REQUIRED and COMPACTION is selected for the allocation, the data set may not be compressed because the attributes did not meet the compression requirements.

Figure 24 highlights some of the messages produced when you are expecting a compressed allocation but, because the data set does not meet the compression requirements, it is not compressed. The system, however, does not fail the allocation because the data set did not compress; instead, the data set has been allocated in the extended format (VSAM KSDS can only have 1 stripe).

We recommend that you know which data sets become compressed, and which data sets you expected to be compressed but do not for eligibility or hardware dependency reasons. This information should be incorporated into operational and disaster backup plans and documentation as there are extra hardware and software dependencies for recovery of compressed data sets.

The method we used during testing to create compressed format data sets requested the necessary data class based on the data set name. There is no indication during data set processing of the kind of format the data sets have other than the messages in the job output. In reality the method we used may not be practicable, but you should be aware of the consequences and check the type of data set using LISTCAT, ISMF, or SMF records, or view the job output.
Note: For a sequential data set, you can issue the ISITMGD macro, after the data set has been successfully opened, to determine whether the data set is extended format or compressed format.

In Figure 25 an IDCAMS DEFINE job requesting compression for the data set to be written by using REPRO received the message shown. The data set is not even eligible to be allocated as extended and therefore is not considered for compression at all.

```
DEF CLUSTER (NAME(CUST04.EXNCNS.VSAM.DS01) +
              MODEL(CUST04.STNCNS.VSAM.DS01) )

IGD17080I DATA SET CUST04.EXNCNS.VSAM.DS01 IS NOT ELIGIBLE FOR EXTENDED FORMAT, ALLOCATION AS NON-EXTENDED CONTINUES

IDC05081I DATA ALLOCATION STATUS FOR VOLUME VM5Q21 IS 0
IDC05091I INDEX ALLOCATION STATUS FOR VOLUME VM5Q21 IS 0
IDC05121I NAME GENERATED-(D) CUST04.EXNCNS.VSAM.DS01.DATA
IDC05121I NAME GENERATED-(I) CUST04.EXNCNS.VSAM.DS01.INDEX
IDC0181I STORAGECLASS USED IS EXTNSTR
IDC0181I MANAGEMENTCLASS USED IS MC00000
IDC0181I DATACLASS USED IS VSEXN01
```

*Figure 25. Allocation in Standard Format, No Extended Format*

### 6.4 Data Availability

To guarantee recovery of the new data set structures when developing backup and recovery procedures you must consider their characteristics while also taking into account the hardware and software dependencies.

#### 6.4.1 Local Backup and Recovery

You should review operational or local backup and recovery plans to include the new data sets formats. Compressed format data sets have hardware and software dependencies, and these should be taken into account when covering aspects of operational failure. For example, hardware dependencies on control unit type and level may influence the placement of the compressed format data sets, the backup utility used, and the recovery utility used.

We recommend that you isolate or at least be aware of the volumes that are eligible for extended format data set allocation and review the recovery procedures in place for:

- The loss of an extended format compressed data set
- The loss of a volume with extended format compressed data sets
- The loss of head disk assembly (HDA) with extended format data sets
- The loss of a channel to the above devices as it is possible to mix parallel and ESCON channels, and there may be ESCON dependencies
- The loss of the controller.
6.4.2 Disaster Backup and Recovery

Disaster recovery usually involves recovery in a different hardware and software environment, whether another partition or another site.

If the hardware is different, you must be aware of where compressed format data sets are and the limitations of the location to which they can be recovered.

If the software is lower than DFSMS/MVS Version 1 Release 2.0, you must be aware that a toleration PTF must be applied to stop access to the extended format compressed data sets from the lower-level system.

In a disaster recovery environment, DFSMSdss is the utility used most to quickly restore customer data. DFSMSdss has several facilities you can use. However, the only DFSMSdss facilities that can change the format of the data set are logical data set dump and restore, and thus they can be used in an environment with DFSMS/MVS Version 1 Release 2.0 and an MVS/ESA SP that supports compression to relieve compressed sequential data sets of the hardware dependencies by restoring them to older devices.

Table 2 on page 41 summarizes the facilities used to manage extended format and extended format compressed data sets.
Chapter 7. Performance Measurements

We tested the new feature of compression available in DFSMS/MVS Version 1 Release 2.0 in different conditions and against different customer data sets to develop some guidelines regarding performance, usability, and management.

In this chapter we summarize and comment on some of the results we obtained through our tests. In particular we consider:

- DASD space
- CPU load
- Implementation.

7.1 DASD Space

Figure 26 summarizes the data in Table 12 on page 83.

![DASD Space Analysis - Sequential Data Sets](image)

Figure 26. Space Analysis Summary for Sequential Data Sets

Figure 27 on page 52 summarizes the space allocation data that we measured (see Table 13 on page 86).
Figure 26 on page 51 and Figure 27 allow us to make some observations regarding space allocation:

- Implementation of the compressed format allows you to define the block size of your data sets independently of the device type where the data set resides. In fact, for sequential compressed format data sets, there is no correlation between the block size that you define for the data set and the actual physical block size used to write the data.

- The compressed format allows an average reduction of 50% of the original space used in the data set.

- Implementation of the extended noncompressed format requires that you increase the amount of space assigned to the data set only when short blocks are used (for sequential) or when the block size has a value such that it occupies the full DASD track.

7.2 CPU Load

We performed several tests using the most common utilities to measure the differences between the standard, extended, and compressed formats when a program processes a data set.
7.2.1 Sequential Data Set Usage

Figure 28 shows the I/O activity of IEBGENER and DFSMSdss. The data is based on average read and write activity on all of the data sets we used. The data is calculated by dividing the measured number of I/O operations by the data set size in megabytes.

![IEBGENER and DFSMSdss Comparison](image)

Figure 28. Average I/O Activity: Sequential Data Sets

We used the same algorithm to calculate the average CPU load and to compare IEBGENER and DFSMSdss (see Figure 29 on page 54).
The additional CPU time incurred when DFSMSdss accesses an extended format data set needs to be explained. In fact, as we mentioned in Chapter 5, "Tools and Utilities" on page 29, DFSMSdss does not perform any format conversion during a logical data set dump and restore. The increase in CPU time when DFSMSdss processes an extended format data set compared with the value of processing a standard format data set is caused by the different technique used to perform the I/O activity. When the data set is in an extended format, DFSMSdss does not directly perform I/O operations to read or write the data; it uses the system data mover service. When data is compressed, this overhead is reduced because there are fewer records to copy or dump.

Figure 30 on page 55 shows the average elapsed time per megabyte for sequential data sets when using IEBGENER and DFSMSdss.
7.2.2 VSAM KSDS Usage

Figure 30 on page 56 depicts the I/O activity. The graph was built using the same algorithm we used to present the results of the tests on sequential data sets.

Note: In the graphs IDCAMS means IDCAMS REPRO. Note also that striping was not used.
In the same way we built the graph in Figure 32 on page 57, which compares the CPU time used by IDCAMS and DFSMSdss for processing a VSAM KSDS. The increase in CPU time for IDCAMS and not for DFSMSdss can be explained by the fact that IDCAMS uses the data compression services, whereas DFSMSdss—which does not support the change of data set format during dump or copy—does not.
Figure 32. Average CPU Load: VSAM KSDSs

Figure 33 on page 58 compares the elapsed times of DFSMSdss and IDCAMS when they process VSAM KSDSs. The reduction in elapsed time when data is in a compressed format is attributable to the reduced size of the data sets.
7.2.3 Considerations

From the data in the graphs presented in this chapter we can make the following observations about using DFSMSdss, IEBGENER, and IDCAMS to process data sets in extended and compressed format:

- By using IEBGENER and IDCAMS to process your sequential and VSAM data set, you can achieve a good reduction in the elapsed time of the jobs, but you must pay for that reduction by a sizable increase in CPU load.
- By using DFSMSdss to back up or copy your data, you can achieve good improvements in performance if you convert the data to compressed format. However, remember that the use of DFSMSdss can affect your recovery procedures.

7.3 Implementation

There are several reasons for why you should want to implement the new data set formats. The most important are:

- Extended noncompressed format
  - Offer some new ways to control and manage data integrity and to protect against failures
  - Are prerequisite to compressed and multistripe formats
- Provide I/O benefits for multistriped extended format data sets.

- Extended compressed format (in addition to the benefits listed for the noncompressed format):
  - DASD space savings
  - Reduced job elapsed times
  - Reduced I/O load on the path (channel, control unit, device).

You should also consider the following drawbacks:

- The new formats require that you review the local and remote backup procedures to guarantee data recoverability in case of errors.

- In some cases there is an increased processor overhead, sometimes higher than 100%.\(^\text{12}\)

- You must review your SMS storage group design to be sure that the required hardware resources are available every time a new format data set is allocated, copied, or restored.

---

\(^{12}\) There is increased processor overhead, which in some instances can be substantial. For compressed data sets we suggest that you run the analysis tools to understand whether the increase in processor overhead is worth the savings in both DASD and elapsed time of batch jobs.
Appendix A. Informational APAR II07816

******************************************************************************
* INFO APAR FOR BSAM/QSAM TEMPORARY DATA SETS IN DFSMS/MVS 1.2.0 *
* COMPRESSION SUPPORT. *
******************************************************************************

START OF DESCRIPTION IN INFO APAR II07816

In DFSMS/MVS 1.2.0, BSAM/QSAM temporary data sets will not be allocated as compressed format data sets or as multivolume single striped extended sequential data sets. These restrictions are enforced in the HDZ11B0 level of the SMS component (5695DF101).

Details of these restrictions:
The following rules apply for BSAM/QSAM temporary data sets when DSNTYPE = EXT is specified in data class.
1. If the SMS-derived volume count <= 1,
   the data set may be allocated as an extended sequential (noncompressed format) data set (containing one or more stripes).
   The COMPACTION parameter is ignored.

2. If the SMS-derived volume count > 1,
   the data set will be allocated as a physical sequential data set.
   The DSNTYPE = EXT parameter is ignored.
   The COMPACTION parameter is ignored.

NOTE: SMS derives the volume count based on the largest of the values specified in JCL as:
- volume count in the VOLUME= keyword,
- number of volsers in the VOLUME= keyword,
- unit count in the UNIT= keyword.
If the above are not specified, the volume count is set to the volume count in data class, if specified, or defaulted to one, if not specified.

These restrictions are to be included in the following publications:
SC26-4922-01 DFSMS/MVS 1.2 Using Data Sets
sections:
1.1.2 Using Access Methods
1.2.4.1.2 Allocating an Extended Sequential Data Set
3.7.10 Characteristics of Extended Sequential Data Sets
3.7.10.1 Compressed Format Data Sets
SC26-4919-01 DFSMS/MVS 1.2 Planning for Installation
sections:
3.3.13 QSAM or BSAM Compression

END OF DESCRIPTION IN INFO APAR II07816

******************************************************************************
Attached are excerpts of the sections to be updated. The updates are flagged with ‘*’ in column 1.

SC26-4922-01 Using Data Sets

1.1.2 Using Access Methods Copyright IBM Corp. 1979, 1994

Basic Sequential Access Method (BSAM)
BSAM arranges records sequentially in the order in which they are entered. Data sets organized this way are called sequential data sets. The user blocks records with other records into blocks. This is called basic access.

Extended Sequential Data Set
Extended sequential data sets have a different internal storage format from a sequential data set that is not extended. This storage format gives extended sequential data sets additional usability characteristics. Extended sequential data sets must be SMS-managed and must reside on DASD. You cannot use an extended sequential data set for certain system data sets.

* A temporary data set will not be allocated as an extended sequential data set if the SMS-derived volume count > 1.

An extended sequential data set may be allocated in the compressed format (may be referred to as a compressed format data set). A compressed format data set is a type of extended sequential data set that has an internal storage format which allows for record level compression.

* A temporary data set will not be allocated as a compressed format data set. The data class COMPACTION parameter is ignored for BSAM/QSAM temporary data sets.

***Note: The above line also is to be added under heading ‘Queued Sequential Access Method (QSAM)’ in same section.

1.2.4.1.2 Allocating an Extended Sequential Data Set
Copyright IBM Corp. 1979, 1994

Extended sequential data sets must be system-managed. The DSNTYPE parameter (a DSNTYPE of EXTENDED) in a data class definition determines whether the data set is allocated as extended sequential. The DSNTYPE parameter on other sources does not allow the allocation of an extended sequential data set.

* Note that DSNTYPE=EXT is ignored for BSAM/QSAM temporary data sets if the SMS-derived volume count > 1.

In addition to a DSNTYPE of EXTENDED, COMPACTION=YES in a data class definition must be specified to request allocation of an extended sequential data set in the compressed format (known as a compressed format data set).

* Note that COMPACTION=YES is ignored for BSAM/QSAM temporary data sets.

3.7.10 Characteristics of Extended Sequential Data Sets
The following are characteristics of extended sequential data sets:

- In most cases, sequential data striping does not require any changes to existing JCL. Your installation’s ACS routines must request a data class which specifies data set name type of EXTENDED.

- In most cases, no changes to applications are needed to access extended sequential data sets.

- The following types of data sets cannot be allocated as extended sequential data sets:
  - Nonsequential data sets
  - Non-SMS-managed data sets
  - VIO data sets.

- The following types of data sets should not be allocated as extended sequential data sets:
  - System data sets
  - System dump data sets
  - GTF trace
  - DFSORT work data sets
  - Data sets used with Hiperbatch
  - Data sets opened with SAM and accessed with EXCP
  - Data sets used with checkpoint/restart.

- Sequential data striping is exclusive to SMS-managed storage. Extended sequential data sets can be allocated only on SMS-managed volumes. An extended sequential data set will not be allocated in a storage group containing mixed device types.

- A temporary data set will not be allocated as an extended sequential data set if the SMS-derived volume count > 1.

- Extended sequential data sets may be allocated only on certain devices.

3.7.10.1 Compressed Format Data Sets

An extended sequential data set may be allocated in the compressed format by also specifying COMPACTION = YES in the data class. Extended sequential data sets allocated in the compressed format may be referred to as compressed format data sets. A compressed format data set is created with a data format which allows for record level compression. Unless otherwise stated, all characteristics which apply to extended sequential data sets continue to apply to compressed format data sets. However, due to the differences in data format, there are some differences in characteristics.

The following are characteristics specific to compressed format data set:

- A compressed format data set may or may not contain compressed records.
- The data format for a compressed format data set consists of physical blocks whose length has no correlation to the logical block size of the data set found in DCB BLKSIZE and DSIBLKLN in the data set label. The actual physical block size is calculated by the system and never returned to the user. However, the system maintains the user's block boundaries when the data set is created so that the original user blocks can be returned to the user when the data set is read.

- A compressed format data set cannot be opened for update.

* - A BSAM/QSAM temporary data set will not be allocated as a compressed format data set. The data class COMPACTION attribute is ignored for BSAM/QSAM temporary data sets.

---

3.3.13 QSAM or BSAM Compression Copyright IBM Corp. 1979, 1994

BSAM or QSAM compression provides a technique for compressing physical sequential data sets that are processed with BSAM or QSAM to reduce the amount of storage required to house the data on DASD.

BSAM or QSAM compression exploits the hardware-based compression facility available on selected processors and uses a software compression facility when running on processors that do not have the hardware compression available.

Only SMS-managed extended format data sets can be compressed. A data set is compressed if it is assigned a data class when created that specifies EXT for the DSNTYPE attribute and YES for the COMPACTION attribute. The only method in which JCL can be used to create a compressed data set is through the DD statement keyword LIKE, where the parameter references a data set that is already in the compressed format. The DSNTYPE parameter in the JCL and on the TSO ALLOCATE command cannot be used to explicitly allocate an extended format data set.

Functional Components Enhanced
DFSMSdfp, DFSMSdss, DFSMSshsm

Software Dependencies
BSAM or QSAM compression requires MVS/ESA SP Version 4 Release 3 with the Compression Facility SPE, or MVS/ESA SP Version 5.

Hardware Dependencies
Extended sequential data sets (eligible for compression) must reside on DASD attached to cached storage controls with the Extended Platform.

For optimal performance, this support uses the data compression feature available on some processors, for example, the 9021 Model 711 and 9121 Model 511 processors. If the hardware is not available, however, it uses the software compression facility to perform data compression.

Restrictions
Extended sequential data sets defined as compressible cannot be opened
for update processing.

* A BSAM/QSAM temporary data set will not be allocated as a compressed
  format data set. The data class COMPACTION attribute is ignored for
  BSAM/QSAM temporary data sets.

  Other restrictions for extended sequential data sets also apply.

====================================================================
Appendix B. FLASH 4014 IMS and DFSMS/MVS 1.2

DFSMS/MVS Version 1 Release 2 was announced on March 1, 1994, via announcement letter 294-095. Among the enhancements contained in this release of DFSMS/MVS was support for hardware data compression with QSAM and BSAM data sets and VSAM key-sequenced data sets (KSDSs). DFSMS/MVS hardware data compression for KSDSs is not supported by IMS/ESA and therefore should not be requested for IMS database data sets. DFSMS/MVS compressed KSDSs will have a different internal CI format, and IMS/ESA does not support that different format. It should be noted that IMS hardware data compression can be used to compress segments in HDAM, HIDAM, and HISAM databases that use VSAM KSDSs and ESDSs. DFSMS/MVS also supports extended format data sets. It is recommended that the extended format attribute not be specified for IMS database data sets. Finally, DFSMS/MVS 1.2 is using a word in the RPL that was previously defined for VTAM. Unfortunately, IMS has also been using this word. Therefore, an IMS APAR must be installed prior to the installation and use of DFSMS/MVS 1.2. The APAR numbers, by IMS release, are:

<table>
<thead>
<tr>
<th>IMS Release</th>
<th>APAR Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>PN43715</td>
</tr>
<tr>
<td>3.1</td>
<td>PN43718</td>
</tr>
<tr>
<td>4.1</td>
<td>PN43721</td>
</tr>
</tbody>
</table>

Note: An extended format data set includes both extended format sequential data sets and extended format VSAM KSDSs.
Appendix C. Performance Measurement Details

This appendix describes our methodology and the results of the performance measurement tests we ran on extended format and compressed format data sets.

For testing we developed a set of procedures that use the standard system utilities. Our purpose was to measure the following:

- The differences in DASD space allocation among the three data set formats.
- The compression factors for different types of data sets.
- Some significant physical characteristics of the I/O access patterns to the three data set formats.
- The differences in duration and CPU usage of jobs accessing the new data set formats compared with the traditional data set format.

C.1 Test Environment

We performed all tests on a dedicated MVS/ESA system. The installed software levels were:

- MVS/ESA Version 5 Release 1.0
- DFSMS/MVS Version 1 Release 2.0.

C.1.1 Hardware Configuration

Figure 34 on page 70 shows the hardware configuration we used. On this system we defined a small SMS environment to control the allocations of the test data sets. The characteristics of the SMS environment are described in detail below.
Table 3 lists the SMS data classes that we used to provide the required allocation parameters for the different types of data sets we needed to create.

<table>
<thead>
<tr>
<th>Data Class Name</th>
<th>Recorg</th>
<th>Data Set Name Type</th>
<th>IF EXT</th>
<th>Volume Count</th>
<th>Add'1 Volume Amount</th>
<th>Compaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSTN01</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>PSEXN01</td>
<td></td>
<td>EXT</td>
<td>R</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>PSEXY01</td>
<td></td>
<td>EXT</td>
<td>R</td>
<td>1</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>VSSTN01</td>
<td></td>
<td>KS</td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>VSEXN01</td>
<td></td>
<td>KS</td>
<td>EXT</td>
<td>R</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>VSEXY01</td>
<td></td>
<td>KS</td>
<td>EXT</td>
<td>R</td>
<td>1</td>
<td>Y</td>
</tr>
</tbody>
</table>
We used four different storage classes to transfer the required data sets under SMS control, one for each data set format, and one with the guaranteed space option to drive some allocations to a selected volume.

Table 4 lists the characteristics of the storage classes.

<table>
<thead>
<tr>
<th>Storage Class Name</th>
<th>MSR</th>
<th>Bias</th>
<th>Sustained Data Rate</th>
<th>Availability</th>
<th>Accessibility</th>
<th>Guaranteed Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC00000</td>
<td></td>
<td>0</td>
<td>STD</td>
<td>S</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>GSPACE</td>
<td></td>
<td>0</td>
<td>STD</td>
<td>S</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>STDNSTR</td>
<td></td>
<td>0</td>
<td>STD</td>
<td>S</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>EXTNSTR</td>
<td></td>
<td>0</td>
<td>STD</td>
<td>S</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

We also defined two management classes (see Table 5). They differ only by the partial release option, enabled in one of them as YI (release space immediately after close). The option allowed us to test the partial release function, which now is also available for VSAM KSDSs.

<table>
<thead>
<tr>
<th>Management Class Name</th>
<th>Expiration</th>
<th>Partial Release</th>
<th>Migration L1</th>
<th>Migration L2</th>
<th>Backup Frequency</th>
<th>Backup Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC00000</td>
<td>NOLIMIT</td>
<td>2</td>
<td>60</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>VSPARTR</td>
<td>NOLIMIT</td>
<td>YI</td>
<td>60</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

We also defined two storage groups. We do not show their characteristics because they were not meaningful for the tests.

### C.1.3 ACS Routines

We developed a set of simple ACS routines to support the creation of different data set types in the SMS test environment. Figure 35 on page 72 shows the data class ACS routine we used in the system.
PROC DATACLS
/*---------------------------------*/
/* DFSMS 1.2.0 - TEST SYSTEM */
/* -EXTENDED DATASET FORMAT */
/* -COMPRESSION */
/*---------------------------------*/
SELECT
  WHEN(&DATACLAS = ‘’ ) SET &DATACLAS = &DATACLAS
  WHEN(&HLQ = CUST* ) SET &DATACLAS = &DATACLAS
  WHEN(&DSN = *.STNCNS.SEQ.** ) SET &DATACLAS = ‘PSSTNO1’
  WHEN(&DSN = *.STNCNS.VSAM.**) SET &DATACLAS = ‘VSSTNO1’
  WHEN(&DSN = *.EXNCNS.SEQ.** ) SET &DATACLAS = ‘PSEXNO1’
  WHEN(&DSN = *.EXNCNS.VSAM.**) SET &DATACLAS = ‘VSEXNO1’
  WHEN(&DSN = *.EXNCNS.SEQ.** ) SET &DATACLAS = ‘PSEXNO1’
  WHEN(&DSN = *.EXNCNS.VSAM.**) SET &DATACLAS = ‘VSEXNO1’
  OTHERWISE SET &DATACLAS = &DATACLAS
END
END

Figure 35. Data Class ACS Routine Used in Test System

Figure 36 on page 73 shows the storage class ACS routine. As you can see, this routine implements automatic actions for permanent data only when it encounters a specific naming convention. In this case we assigned a storage class according to the contents of the second qualifier of the data set name. Data could also be passed to SMS management when we explicitly requested it by associating a storage class name with the allocation request. Thus we were able to easily control what we passed to SMS as well as the associated management characteristics.
PROC STORCLAS
/*---------------------------------*/
/* DFSMS 1.2.0 - TEST SYSTEM */
/* -EXTENDED DATASET FORMAT */
/* -COMPRESSION */
/*---------------------------------*/
SELECT
WHEN(&STORCLAS = '') SET &STORCLAS = &STORCLAS
WHEN(&DSTYPE = 'TEMP') SET &STORCLAS = 'SC00000'
WHEN(&DSN = '*.SPF.*') SET &STORCLAS = 'SC00000'
WHEN(&DSN = '*.ISR*.BACKUP') SET &STORCLAS = 'SC00000'
WHEN(&HLQ = CUST*) SET &STORCLAS = ''
WHEN(&DSN = '*.ST*.**') SET &STORCLAS = 'STDNSTR'
WHEN(&DSN = '*.EX*.**') SET &STORCLAS = 'EXTNSTR'
OTHERWISE SET &STORCLAS = ''
END

PROC MGMTCLAS
/*---------------------------------*/
/* DFSMS 1.2.0 - TEST SYSTEM */
/* -EXTENDED DATASET FORMAT */
/* -COMPRESSION */
/*---------------------------------*/
SELECT
WHEN(&MGMTCLAS = '') SET &MGMTCLAS = &MGMTCLAS
WHEN(&DSN = CUST*.EX*.**) SET &MGMTCLAS = 'VSPARTR'
OTHERWISE SET &MGMTCLAS = 'MC00000'
END

Figure 36. Storage Class ACS Routine Used in Test System

We used the management class ACS routine to associate the partial release option to all of the data sets in the new formats. Figure 37 shows the coding.

PROC MGMTCLAS
/*---------------------------------*/
/* DFSMS 1.2.0 - TEST SYSTEM */
/* -EXTENDED DATASET FORMAT */
/* -COMPRESSION */
/*---------------------------------*/
SELECT
WHEN(&MGMTCLAS = '') SET &MGMTCLAS = &MGMTCLAS
WHEN(&DSN = CUST*.EX*.**) SET &MGMTCLAS = 'VSPARTR'
OTHERWISE SET &MGMTCLAS = 'MC00000'
END

Figure 37. Management Class ACS Routine Used in Test System

The storage group ACS routine assigns different storage groups on the basis of the data set type, temporary or permanent. It also supports the requests for a guaranteed space allocation (see Figure 38 on page 74).
C.1.4 Customer Data

The tests we performed were based on data we received from nine different customers. The data consisted of 21 data sets; 11 were VSAM KSDSs, and the other 11, sequential data sets. Table 6 on page 75 shows their characteristics. As you can see, we could use data sets of different sizes, record formats, and record lengths.
### Table 6. Characteristics of Data Sets Used for Measurements

<table>
<thead>
<tr>
<th>No.</th>
<th>Data Set Name</th>
<th>Data Set Organization</th>
<th>RECFM</th>
<th>Number of Records</th>
<th>LRECL</th>
<th>KEYLEN</th>
<th>RKP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CUST01.*.VSAM.DS01</td>
<td>VS</td>
<td>F</td>
<td>361,930</td>
<td>288</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST01.*.VSAM.DS02</td>
<td>VS</td>
<td>F</td>
<td>135,367</td>
<td>135</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>CUST01.*.SEQ.FILE01</td>
<td>PS</td>
<td>F</td>
<td>225,000</td>
<td>121</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>CUST01.*.SEQ.FILE02</td>
<td>PS</td>
<td>F</td>
<td>66,085</td>
<td>80</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td>CUST02.*.VSAM.DS01</td>
<td>VS</td>
<td>F</td>
<td>218,562</td>
<td>350</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST02.*.SEQ.FILE01</td>
<td>PS</td>
<td>F</td>
<td>165,753</td>
<td>256</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>3</td>
<td>CUST03.*.VSAM.DS01</td>
<td>VS</td>
<td>V</td>
<td>1,685,291</td>
<td>17-1016</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.VSAM.DS02</td>
<td>VS</td>
<td>F</td>
<td>442</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.SEQ.FILE01</td>
<td>PS</td>
<td>F</td>
<td>12,646</td>
<td>151</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.SEQ.FILE02</td>
<td>PS</td>
<td>F</td>
<td>1,712,343</td>
<td>200</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>4</td>
<td>CUST04.*.VSAM.DS01</td>
<td>VS</td>
<td>F</td>
<td>207,124</td>
<td>22</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>CUST04.*.SEQ.FILE01</td>
<td>PS</td>
<td>F</td>
<td>2,360,797</td>
<td>241</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>CUST04.*.SEQ.FILE02</td>
<td>PS</td>
<td>F</td>
<td>5,512,044</td>
<td>133</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>CUST05.*.VSAM.DS01</td>
<td>VS</td>
<td>V</td>
<td>3,732</td>
<td>300-600</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST05.*.VSAM.DS02</td>
<td>VS</td>
<td>V</td>
<td>1,309,279</td>
<td>1024-22528</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST05.*.VSAM.DS03</td>
<td>VS</td>
<td>V</td>
<td>2,879</td>
<td>1024-26000</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CUST05.*.SEQ.FILE01</td>
<td>PS</td>
<td>V</td>
<td>18,953</td>
<td>22528</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>6</td>
<td>CUST06.*.SEQ.FILE01</td>
<td>PS</td>
<td>F</td>
<td>190,980</td>
<td>80</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>7</td>
<td>CUST07.*.SEQ.FILE01</td>
<td>PS</td>
<td>V</td>
<td>239,400</td>
<td>76</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>8</td>
<td>CUST08.*.VSAM.DS01</td>
<td>VS</td>
<td>V</td>
<td>1,387,006</td>
<td>150-1500</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>CUST09.*.VSAM.DS01</td>
<td>VS</td>
<td>V</td>
<td>155,423</td>
<td>750-1360</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:**
* This qualifier indicates the data set format. It can have the following values:
  - STNCNS - standard format
  - EXNCNS - extended format, noncompressed, one stripe
  - EXYCNS - extended format, compressed, one stripe.

Table 7 on page 76 and Table 8 on page 77 show the name and contents of each data set we use in the tables and graphs in this manual.
<table>
<thead>
<tr>
<th>No.</th>
<th>Data Set Name</th>
<th>Data Description</th>
<th>Names in Subsequent Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CUST01.*.VSAM.DS01</td>
<td>Data is in binary format.</td>
<td>C1VS1</td>
</tr>
<tr>
<td></td>
<td>CUST01.*.VSAM.DS02</td>
<td>Data is in binary format.</td>
<td>C1VS2</td>
</tr>
<tr>
<td>2</td>
<td>CUST02.*.VSAM.DS01</td>
<td>Data is in binary format with short numeric strings in character format embedded in the records.</td>
<td>C2VS1</td>
</tr>
<tr>
<td>3</td>
<td>CUST03.*.VSAM.DS01</td>
<td>Character format data records.</td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.VSAM.DS02</td>
<td>Character format with a short binary field at the beginning of the records.</td>
<td>C4VS1</td>
</tr>
<tr>
<td>4</td>
<td>CUST04.*.VSAM.DS01</td>
<td>Character format with a short binary field at the beginning of the records.</td>
<td>C4VS1</td>
</tr>
<tr>
<td>5</td>
<td>CUST05.*.VSAM.DS01</td>
<td>Binary format records with small embedded text.</td>
<td>C5VS1</td>
</tr>
<tr>
<td></td>
<td>CUST05.*.VSAM.DS02</td>
<td>Binary format records.</td>
<td>C5VS2</td>
</tr>
<tr>
<td></td>
<td>CUST05.*.VSAM.DS03</td>
<td>Binary format records with small embedded text.</td>
<td>C5VS3</td>
</tr>
<tr>
<td>8</td>
<td>CUST08.*.VSAM.DS01</td>
<td>SLR database.</td>
<td>C8VS1</td>
</tr>
<tr>
<td>9</td>
<td>CUST09.*.VSAM.DS01</td>
<td>Binary data with short text strings embedded in the records.</td>
<td>C9VS1</td>
</tr>
</tbody>
</table>

Note:

* This qualifier indicates the data set format. It can have the following values:
  - STNCNS - standard format
  - EXNCNS - extended format, noncompressed, one stripe
  - EXYCNS - extended format, compressed, one stripe.

We could not use the following two data sets for the tests:

**CUST03.*.VSAM.DS01** As you can see from Table 6 on page 75, this data set has a minimum record length of 17 bytes, 12 of which are the record key. The key position allows only 1 byte of data for the shortest records, making the data set ineligible for compression.

**CUST03.*.VSAM.DS02** This is a key range data set (record length equals the key length). This type of data set is not supported for compression.

The following two data sets had characteristics that allowed us to perform the tests, but the measured data was not consistent with all other measurements:

**CUST04.*.VSAM.DS01** As you see in Table 6 on page 75 this data set has a short record length of 22 bytes, 16 of which are used by the key. The relative key position is such that only 1 byte of data is available. For these reasons this data set is not eligible for compression.
The data set is just large enough (9MB) to be eligible for compression, but it contains a small amount of data (free space is 90%). The measurement data for this data set is in the tables, but we decided not to use the data for the graphs.

Table 8. Sequential Data Sets

<table>
<thead>
<tr>
<th>No.</th>
<th>Data Set Name</th>
<th>Data Description</th>
<th>Names in Subsequent Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CUST01.*.SEQ.FILE01</td>
<td>Output listing data set. Data is in character format.</td>
<td>C1PS1</td>
</tr>
<tr>
<td></td>
<td>CUST01.*.SEQ.FILE02</td>
<td>System definition data set. Data is in character format.</td>
<td>C1PS1</td>
</tr>
<tr>
<td>2</td>
<td>CUST02.*.SEQ.FILE01</td>
<td>Output listing format with character strings embedded in the text. Data is in character format.</td>
<td>C2PS1</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.SEQ.FILE01</td>
<td>Binary data with short numeric field in character format.</td>
<td>C3PS1</td>
</tr>
<tr>
<td></td>
<td>CUST03.*.SEQ.FILE02</td>
<td>Output listing format with long character numeric strings at the end of each record.</td>
<td>C3PS2</td>
</tr>
<tr>
<td>4</td>
<td>CUST04.*.SEQ.FILE01</td>
<td>Binary data set with record prefixed by a short string of data in character format.</td>
<td>C4PS1</td>
</tr>
<tr>
<td></td>
<td>CUST04.*.SEQ.FILE02</td>
<td>Compiler output listing. Data is in character format.</td>
<td>C4PS2</td>
</tr>
<tr>
<td>5</td>
<td>CUST05.*.SEQ.FILE01</td>
<td>Mixed character and binary data records.</td>
<td>C5PS1</td>
</tr>
<tr>
<td>6</td>
<td>CUST06.*.SEQ.FILE01</td>
<td>Non-formatted text data set. Data is in upper- and lower-case character format.</td>
<td>C6PS1</td>
</tr>
<tr>
<td>7</td>
<td>CUST07.*.SEQ.FILE01</td>
<td>Non-formatted text data set. Data is in upper- and lower-case character format.</td>
<td>C7PS1</td>
</tr>
</tbody>
</table>

Note:
* This qualifier indicates the data set format. It can have the following values:
  - STNCNS - standard format
  - EXNCNS - extended format, noncompressed, one stripe
  - EXYCNS - extended format, compressed, one stripe.

We did not use the CUST03.(*)SEQ.FILE01 data set for the tests because its size made it ineligible for compression.

C.2 Objectives of the Measurements

The procedures and the testing environment we developed to perform the measurements presented in this book were set up to:

- Identify the cases that do not allow conversion to an extended format
- Measure the impact of CPU time overhead for extended format and compressed format
- Measure the reduction in elapsed time when accessing a compressed format data set
• Evaluate the effect of block size on sequential data sets and of CI size on VSAM KSDSs when the data sets are in extended format and compressed format.

C.2.1 Utilities and Tools

We ran all measurement tests using the following system utilities:

• IEBGENER to read and write sequential data sets
• DFSMSdss to read and write sequential data sets
• DFSMSdss to read VSAM KSDSs
• IDCAMS REPRO to read and write VSAM KSDSs.

Examples of our JCL procedures are given in the sections of this chapter where we report the results of the measurements.

C.2.2 Data Collected

We obtained the measurement data from:

• Output job logs, for some data describing the data set structures
• RMF reports, for data regarding the I/O activity and characteristics
• SMF records, for CPU and elapsed time data
• ISMF lists, for the data on DASD space and allocation characteristics
• Catalog lists from LISTCAT IDCAMS command.

C.2.3 Compression Estimation Tool

This tool is available to IBM technical professionals to evaluate the effects of compression on selected data sets before implementing the feature. We describe the tool in Chapter 5, “Tools and Utilities” on page 29.

As a first test we ran this tool against the data sets we planned to use for the measurements, to have an initial estimate of compression efficiency. Table 9 on page 79 shows the results.
### Table 9. Estimated Compression Factors

<table>
<thead>
<tr>
<th>VSAM KSDs</th>
<th>Sequential Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set Name</td>
<td>Expected Compression (%)</td>
</tr>
<tr>
<td>CUST01.STNCNS.VSAM.DS01</td>
<td>76</td>
</tr>
<tr>
<td>CUST01.STNCNS.VSAM.DS02</td>
<td>55</td>
</tr>
<tr>
<td>CUST02.STNCNS.VSAM.DS01</td>
<td>32</td>
</tr>
<tr>
<td>CUST03.STNCNS.VSAM.DS01</td>
<td>24</td>
</tr>
<tr>
<td>CUST03.STNCNS.VSAM.DS02</td>
<td>-</td>
</tr>
<tr>
<td>CUST04.STNCNS.VSAM.DS01</td>
<td>-</td>
</tr>
<tr>
<td>CUST04.STNCNS.VSAM.DS02</td>
<td>-</td>
</tr>
<tr>
<td>CUST05.STNCNS.VSAM.DS01</td>
<td>13</td>
</tr>
<tr>
<td>CUST05.STNCNS.VSAM.DS02</td>
<td>29</td>
</tr>
<tr>
<td>CUST05.STNCNS.VSAM.DS03</td>
<td>57</td>
</tr>
<tr>
<td>CUST08.STNCNS.VSAM.DS01</td>
<td>10</td>
</tr>
</tbody>
</table>

**Note:**
(1) This value is a tool prediction. In actuality, the data set was ineligible for compression.

### C.3 Measurement Results

In this section we describe the different test environments and the results we obtained.

For each set of measurements we present examples of the procedures we ran, one or more tables showing the details of the data collected, and one or more graphs summarizing the results.

#### C.3.1 DASD Space Allocation

To evaluate the differences in DASD space allocation, we created three copies of each available data set: the first in standard format; the second, in an extended single-stripe format; the third, in an extended single-stripe and compressed format. In all cases we used the system-determined block size for sequential data sets and a value close to the optimum CI size for VSAM KSDSs to allow the system to optimize the DASD space utilization.
Table 10 shows the results for VSAM KSDSs. The values in the table are copied from ISMF lists and IDCAMS LISTCAT output.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Number of Bytes Following the Key</th>
<th>CI Size (Data Component)</th>
<th>Number of Records</th>
<th>Megabytes (before compression)</th>
<th>Megabytes (after compression)</th>
<th>% Reduction</th>
<th>Number of Records (before compression)</th>
<th>Number of Records (after compression)</th>
<th>% Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1VS1</td>
<td>258</td>
<td>18432</td>
<td>361,930</td>
<td>125.631</td>
<td>25.219</td>
<td>79.9</td>
<td>162</td>
<td>38</td>
<td>76.5</td>
</tr>
<tr>
<td>C1VS2</td>
<td>109</td>
<td>18432</td>
<td>135,367</td>
<td>22.695</td>
<td>10.088</td>
<td>55.5</td>
<td>29</td>
<td>15</td>
<td>48.3</td>
</tr>
<tr>
<td>C2VS1</td>
<td>328</td>
<td>16384</td>
<td>218,562</td>
<td>76.748</td>
<td>49.236</td>
<td>35.8</td>
<td>107</td>
<td>69</td>
<td>35.5</td>
</tr>
<tr>
<td>C4VS1</td>
<td>1</td>
<td>4096</td>
<td>207,124</td>
<td>4.986</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>C5VS1</td>
<td>280-580</td>
<td>18432</td>
<td>3,732</td>
<td>0.828</td>
<td>0.738</td>
<td>13.0</td>
<td>(1)</td>
<td>(1)</td>
<td>0</td>
</tr>
<tr>
<td>C5VS2</td>
<td>1000-22504</td>
<td>24576</td>
<td>1,309,279</td>
<td>314.802</td>
<td>210.110</td>
<td>33.3</td>
<td>441</td>
<td>295</td>
<td>33.1</td>
</tr>
<tr>
<td>C5VS3</td>
<td>1000-25976</td>
<td>32768</td>
<td>2,879</td>
<td>33.841</td>
<td>13.396</td>
<td>60.4</td>
<td>49</td>
<td>20</td>
<td>59.2</td>
</tr>
<tr>
<td>C8VS1</td>
<td>50-1400</td>
<td>12288</td>
<td>1,387,006</td>
<td>439.490</td>
<td>319.770</td>
<td>27.2</td>
<td>605</td>
<td>441</td>
<td>27.1</td>
</tr>
<tr>
<td>C9VS1</td>
<td>745-1355</td>
<td>18432</td>
<td>155,423</td>
<td>128.358</td>
<td>64.962</td>
<td>49.4</td>
<td>156</td>
<td>89</td>
<td>42.9</td>
</tr>
</tbody>
</table>

Note: (1) The data set was ineligible for compression.

Table 11 on page 81 shows the results for sequential data sets. In this case we copied the values in the table from ISMF lists.
### Table 11. DASD Space Measurements for Sequential Data Sets

| Data Set | Record Length | Standard Format |  |  |  |  |  |  |  |
|----------|---------------|----------------|---|---|---|---|---|---|
|          |               | Block Length   | Mega-bytes | Block Length | Mega-bytes | Block Length | Mega-bytes | Space Saved with Compression (%) |
| C1PS1    | 121           | 27951          | 27.004     | 27951        | 27.004     | 27951        | 12.340     | 59 |
| C1PS2    | 80            | 27920          | 5.257      | 27920        | 5.257      | 27920        | 2.324      | 64 |
| C2PS1    | 256           | 27904          | 42.111     | 27904        | 42.111     | 27904        | 13.779     | 70 |
| C3PS1    | 151           | 27935          | 1.937      | 27935        | 1.937      | (1)          | (1)        |  |
| C3PS2    | 200           | 27800          | 340.869    | 27800        | 340.869    | 27800        | 90.142     | 76 |
| C4PS1    | 241           | 27956          | 563.099    | 27956        | 563.099    | 27956        | 151.786    | 76 |
| C4PS2    | 133           | 27930          | 726.229    | 27930        | 726.229    | 32718        | 316.964    | 61 |
| C5PS1    | 22528         | 27998          | 197.660    | (2)          | 27998      | 104.032     | 34         |  |
| C6PS1    | 80            | 27920          | 15.162     | 27920        | 15.162     | 27920        | 7.692      | 57 |
| C7PS1    | 76            | 27998          | 8.411      | (2)          | 27998      | 6.198       | 36         |  |

**Note:**
- (1) Ineligible for compression
- (2) Data not available

### C.3.2 Effect of Block Size or Cl Size on Space Allocations

The effect of the data set block size for sequential data sets on the amount of DASD space required for allocation is known. With this test we wanted to see how the effect changes when the data set is in an extended format or in a compressed format.

We tested the space used by the different data set formats, varying the block size for sequential data sets and the Cl size for VSAM KSDSs.

We performed the block size test on sequential data sets by using the JCL shown in Figure 39 on page 82.
CRE EXEC PGM=IEBGENER
/*-----------------------------------------------------*/
/* CREATE A STANDARD FORMAT DATA SET WITH: */
/* .BLKSIZE = 80 */
/*-----------------------------------------------------*/
SYSPRINT DD SYSOUT=* 
SYST1 DD DISP=SHR,DSN=CUST06.STNCNS.SEQ.TEST 
SYST2 DD DISP=(,CATLG,DELETE),DSN=CUST06.STNCNS.SEQ00080.TEST, 
// SPACE=(80,(263000,1),RLSE),AVGREC=U,STORCLAS=GSPACE, 
// RECFM=FB,LRECL=80,BLKSIZE=80,VOL=SER=VM5Q21 
SYSSN DD DUMMY 
/*
COPY1 EXEC PGM=IEBGENER
/*-----------------------------------------------------*/
/* THIS STEP CREATES AN EXTENDED AND NON-COMPRRESSED */
/* SEQUENTIAL DATA SET */
/* .BLKSIZE = 80 */
/*-----------------------------------------------------*/
SYSPRINT DD SYSOUT=* 
SYST1 DD DISP=SHR,DSN=CUST06.STNCNS.SEQ.TEST 
SYST2 DD DISP=(,CATLG,DELETE),DSN=CUST06.EXNCNS.SEQ00080.TEST, 
// SPACE=(80,(263000,1),RLSE),AVGREC=U, 
// STORCLAS=GSPACE,VOL=SER=VM5Q24, 
// RECFM=FB,LRECL=80,BLKSIZE=80 
SYSSN DD DUMMY 
/*
COPY2 EXEC PGM=IEBGENER
/*-----------------------------------------------------*/
/* THIS STEP CREATES AN EXTENDED AND COMPRESSED */
/* SEQUENTIAL DATA SET */
/* .BLKSIZE = 80 */
/*-----------------------------------------------------*/
SYSPRINT DD SYSOUT=* 
SYST1 DD DISP=SHR,DSN=CUST06.STNCNS.SEQ.TEST 
SYST2 DD DISP=(,CATLG,DELETE),DSN=CUST06.EXYCNS.SEQ00080.TEST, 
// SPACE=(80,(263000,1),RLSE),AVGREC=U, 
// STORCLAS=GSPACE,VOL=SER=VM5Q23, 
// RECFM=FB,LRECL=80,BLKSIZE=80 
SYSSN DD DUMMY 
/*
Figure 39. JCL Used to Test Block Sizes for Sequential Data Sets

This JCL, through the RLSE parameter, activates a partial release at the end of
the job step let the data set use only the minimum required space to contain the
data. We modified this procedure to change the value of the block size and ran
the procedure once for each selected block size value. Table 12 on page 83
shows the block size value we selected and the measured data. 
### Table 12. Space Occupancy Based on Sequential Data Set Block Size (Megabytes)

<table>
<thead>
<tr>
<th>Block Size</th>
<th>Standard Format</th>
<th>Extended Format (Non-compressed)</th>
<th>Ext/Std (%)</th>
<th>Extended Format (Compressed)</th>
<th>Cmp/Std (%)</th>
<th>Cmp/Ext (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>186.039</td>
<td>193.454</td>
<td>+ 4.0</td>
<td>10.514</td>
<td>- 94.3</td>
<td>- 94.6</td>
</tr>
<tr>
<td>320</td>
<td>61.478</td>
<td>63.636</td>
<td>+ 3.5</td>
<td>10.514</td>
<td>- 82.9</td>
<td>- 83.5</td>
</tr>
<tr>
<td>1280</td>
<td>31.320</td>
<td>31.320</td>
<td>0.0</td>
<td>10.514</td>
<td>- 66.4</td>
<td>- 66.4</td>
</tr>
<tr>
<td>3120</td>
<td>24.846</td>
<td>24.846</td>
<td>0.0</td>
<td>10.514</td>
<td>- 57.7</td>
<td>- 57.7</td>
</tr>
<tr>
<td>6480</td>
<td>22.411</td>
<td>22.411</td>
<td>0.0</td>
<td>10.514</td>
<td>- 53.1</td>
<td>- 53.1</td>
</tr>
<tr>
<td>13680</td>
<td>21.249</td>
<td>28.332</td>
<td>+ 33.3</td>
<td>10.514</td>
<td>- 50.5</td>
<td>- 62.9</td>
</tr>
<tr>
<td>Half track(1)</td>
<td>20.806</td>
<td>20.806</td>
<td>0.0</td>
<td>10.514</td>
<td>- 49.5</td>
<td>- 49.5</td>
</tr>
</tbody>
</table>

**Note:**
(1) System-determined block size. The system-determined block size for a sequential compressed format data set is not necessarily half-track because the user-specified block size has no correlation to the actual physical block size used to write the data.

The results for a block size of 13680 require some explanation. As you can see, there is an unusual difference in space allocation between the standard format and the extended format. The reason for this difference is the block size that we used. In fact, this block size is only 2 bytes less than the maximum value that allows the system to write four blocks per track.

As you know, the extended format adds a suffix of 32 bytes to each physical block. In this case the block becomes some bytes longer than the four blocks per track maximum, forcing the system to write only three blocks per track. This requires an allocated space increase of one third, that is, the value you see in the table.

Figure 40 on page 84 shows the results of the measurements for sequential data sets based on the data presented in Table 12.
It is also important to note that the implementation of compression, in some cases, can nullify the negative effects on space allocation caused by an inefficient block size. In fact, when data is compressed, the user-specified block size becomes a logical characteristic of the data structure. In this case the physical block size is defined by the system.

We used the JCL in Figure 41 on page 85 to measure the effects of different CI sizes on the space allocation of a VSAM KSDS cluster.

Figure 40. Result of Space Measurements for Sequential Data Sets
In this case as well we changed the value of the CI size several times and ran this procedure one time for each selected CI size. Table 13 on page 86 presents the data we collected with this measurement.

---

Figure 41. JCL Used to Test CI Size for VSAM KSDSs
### Table 13. Space Occupancy Based on VSAM CI Size

<table>
<thead>
<tr>
<th>CI Size</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Allocated</td>
<td>Used</td>
<td>Allocated</td>
</tr>
<tr>
<td>512</td>
<td>83.723</td>
<td>25.778</td>
<td>58.822</td>
</tr>
<tr>
<td>1024</td>
<td>53.455</td>
<td>25.453</td>
<td>42.664</td>
</tr>
<tr>
<td>4096</td>
<td>34.253</td>
<td>22.448</td>
<td>25.952</td>
</tr>
</tbody>
</table>

**Note:**

(1) Based on the allocated space
(2) ISMF suggested optimal block size

Figure 42 on page 87 shows the differences among the three data set formats based on the allocated space indicated in Table 13.
Figure 42. Result of Allocated Space Measurements for VSAM KSDSs

Figure 43 on page 88 shows the differences among the three data set formats based on the used space indicated in Table 13.
As you can see from Figure 42 on page 87 and Figure 43, changing the CI size has less effect on space allocation for a VSAM cluster than changing the block size for a sequential data set.

In addition, the availability of the partial release option for VSAM extended format clusters now allows you to define the space values when allocating a cluster such that you cannot underestimate the value and you can thereby avoid wasting unused space within the cluster.

Note that in Figure 42 on page 87 and Figure 43 the value selected for block size or CI size does not affect the space used by the data set when it is in compressed format.

### C.3.3 Read I/O Measurements for New Format Data Sets

We performed read I/O measurements for extended format and compressed format data sets using three utilities:

- IDCAMS REPRO for VSAM KSDSs
- DFSMSdss logical dump for sequential data sets and VSAM KSDSs
- IEBGENER for sequential data sets.

We report the results in tables and graphs. In particular we show:

- I/O load data when reading VSAM KSDSs with IDCAMS
- I/O load data when reading VSAM KSDSs with DFSMSdss
• I/O load data when reading sequential data sets with DFSMSdss
• I/O load data when reading sequential data sets with IEBGENER.

For the first set of measurements we used IDCAMS to read VSAM KSDSs. Figure 44 shows the JCL procedure we used. We ran the procedure three times for each data set: one time to read the data set in its standard format, the second time to read it in its extended noncompressed format, and the third time to read it in its compressed format.

```
//TR11VSI JOB SYS1,'FRED'.MSGLEVEL=(1,1),MSGCLASS=Z,
 // NOTIFY=FRED
//READ EXEC PGM=IDCAMS
/*---------------------------------------------*/
/* -READ TEST:
 */.VSAM KSDS INPUT
/* .IDCAMS (REPRO)
/*---------------------------------------------*/
//SYSPRINT DD SYSOUT=*  
//INVSAM DD DISP=OLD,DSN=CUST01.STNCNS.VSAM.DS01  
//OUTSEQ DD DISP=(,DELETE,DELETE),DSN=&VSAMR,  
 // SPACE=(1,(800)),AVGREC=M,  
 // DCB=(RECFM=FB,LRECL=288),  
 // STORCLAS=GSPACE,VOL=SER=VM5Q21  
//SYSIN DD *  
 // REPRO IFILE(INVSAM) OFILE(OUTSEQ)  
/*
```

Figure 44. JCL Used to Read a VSAM KSDS with IDCAMS

For each run we measured the number of physical I/O operations, the average service time, and the average connect time. Table 14 on page 90 shows the results. We obtained the data from the RMF monitor, which we activated before executing each job.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Noncompressed</th>
<th>Extended Compressed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
</tr>
<tr>
<td>C1VS1</td>
<td>6,538</td>
<td>11</td>
<td>5.3</td>
</tr>
<tr>
<td>C1VS2</td>
<td>1,164</td>
<td>11</td>
<td>5.2</td>
</tr>
<tr>
<td>C2VS1</td>
<td>4,884</td>
<td>11</td>
<td>4.9</td>
</tr>
<tr>
<td>C4VS1</td>
<td>280</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>C5VS1</td>
<td>48</td>
<td>4</td>
<td>3.1</td>
</tr>
<tr>
<td>C5VS2</td>
<td>12,261</td>
<td>18</td>
<td>7.0</td>
</tr>
<tr>
<td>C5VS3</td>
<td>1,059</td>
<td>20</td>
<td>9.1</td>
</tr>
<tr>
<td>C8VS1</td>
<td>1,461</td>
<td>89</td>
<td>79.7</td>
</tr>
<tr>
<td>C9VS1</td>
<td>6,432</td>
<td>11</td>
<td>5.4</td>
</tr>
</tbody>
</table>

**Note:**

(1) Ineligible for compression

We used some of the values in Table 14 to build a graph showing the I/O activity (see Figure 45 on page 91). Because the number of I/O operations changes according to the data set size, we used a relative value calculated as the number of I/Os per megabyte.
We repeated the test we performed reading a VSAM KSDS with IDCAMS using DFSMSdss logical data set dump. Figure 46 shows the JCL we used.

```
//TR11VSD JOB SYS1,'FRED',MSGLEVEL=(1,1),MSGCLASS=Z,
// NOTIFY=FRED
//STEP1 EXEC PGM=ADRDSSU,REGION=5M
//*-----------------------------------------------------*
//* -READ TEST: *
//* .SEQUENTIAL INPUT *
//*-----------------------------------------------------*
//SYSPRINT DD SYSOUT=* 
//OUTDD1 DD DISP=(*,DELETE,DELETE),DSN=&&DSSOUT, 
// SPACE=(1,(600)),AVGREC=M,STORCLAS=GSPACE, 
// VOL=SER=VM5Q21 
//SYSIN DD * 
DUMP DATASET(INCLUDE(CUST01.STNCNS.VSAM.DS01)) + 
OUTDONAME(OUTDD1) 
//*
```

Table 15 on page 92 shows the data we measured. The data describes the physical characteristics of the I/O operations that DFSMSdss performed.
We processed the data from Table 15 to build the graph you see in Figure 47 on page 93.

We calculated the numbers of I/O per megabyte for the compressed format using the real data set size, not the size after compression.
We used DFSMSdss logical data set dump to access sequential data sets. Figure 48 shows the JCL we used.

```jcl
//TR11PSD JOB SYS1,'FRED',MSGLEVEL=(1,1),MSGCLASS=Z,
// NOTIFY=FRED
//STEP1 EXEC PGM=ADRDSSU,REGION=5M
//*-----------------------------------------------------*
//* -READ TEST: *
//* .SEQUENTIAL INPUT *
//* .DFSMSdss *
//*-----------------------------------------------------*
//SYSPRINT DD SYSOUT=* 
//OUTDD1 DD DSN=&DSSOUT,SPACE=(1,(800)),AVGREC=M 
//SYSIN DD * 
DUMP DATASET(INCLUDE=CUST01.STNCNS.SEQ.FILE01) + 
OUTDDNAME(OUTDD1) 
//*
```

Table 16 on page 94 shows the data we measured. We could not perform the measurements on two data sets for extended format and one for compressed format because the data set structures did not allow the allocation of these formats.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th></th>
<th>Extended Format (Noncompressed)</th>
<th></th>
<th>Extended Format (Compressed)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
</tr>
<tr>
<td>C1PS1</td>
<td>513</td>
<td>24</td>
<td>14.4</td>
<td>515</td>
<td>22</td>
<td>15.0</td>
</tr>
<tr>
<td>C1PS2</td>
<td>122</td>
<td>27</td>
<td>12.1</td>
<td>115</td>
<td>22</td>
<td>13.2</td>
</tr>
<tr>
<td>C2PS1</td>
<td>782</td>
<td>35</td>
<td>14.7</td>
<td>782</td>
<td>25</td>
<td>15.2</td>
</tr>
<tr>
<td>C3PS1</td>
<td>53</td>
<td>25</td>
<td>10.3</td>
<td>24</td>
<td>6</td>
<td>3.6</td>
</tr>
<tr>
<td>C3PS2</td>
<td>6,218</td>
<td>44</td>
<td>14.8</td>
<td>5,593</td>
<td>25</td>
<td>15.2</td>
</tr>
<tr>
<td>C4PS1</td>
<td>10,261</td>
<td>32</td>
<td>15.0</td>
<td>7,872</td>
<td>27</td>
<td>15.5</td>
</tr>
<tr>
<td>C4PS2</td>
<td>13,229</td>
<td>35</td>
<td>15.0</td>
<td>11,649</td>
<td>21</td>
<td>15.5</td>
</tr>
<tr>
<td>C5PS1</td>
<td>3,616</td>
<td>25</td>
<td>12.2</td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6PS1</td>
<td>296</td>
<td>40</td>
<td>14.2</td>
<td>294</td>
<td>29</td>
<td>14.6</td>
</tr>
<tr>
<td>C7PS1</td>
<td>172</td>
<td>29</td>
<td>13.6</td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression
(2) Test not performed

The DFSMSdss read I/O data for sequential data sets in Table 16 is plotted in Figure 49 on page 95. The graph shows the same behavior as the graph for VSAM KSDSs in Figure 47 on page 93.
Figure 49. DFSMSdss I/O Load for Reading Sequential Data Sets

We repeated the read I/O test on sequential data sets using IEBGENER. Figure 50 shows the JCL we used for this test.

```
//TR11PSG JOB SYS1, 'FRED', MSGLEVEL=(1,1), MSGCLASS=Z,
// NOTFY=FRED
//COPY EXEC PGM=IEBGENER
//*-----------------------------------------------------*
//* -READ TEST: *
//* .SEQUENTIAL INPUT *
//* .IEBGENER *
//*-----------------------------------------------------*
//SYSPRINT DD SYSOUT=* 
//SYSUT1 DD DISP=OLD, DSN=CUSTO1.STNCNS.SEQ.FILE01
//SYSUT2 DD DSN=&CPYOUT, SPACE={(1,(800)), AVGREC=M}
//SYSIN DD DUMMY
//*
```

Figure 50. JCL Used to Read a Sequential Data Set with IEBGENER

Table 17 on page 96 shows the measured data. In this case as well we could not measure the compressed format of one data set because its characteristics did not allow the creation of the compressed format.
### Table 17. Read I/O Measurements on Sequential Data Sets: IEBGENER

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td>Number of I/Os (Read)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td></td>
</tr>
<tr>
<td>C1PS1</td>
<td>598</td>
<td>23</td>
<td>12.0</td>
<td>588</td>
<td>12</td>
<td>11.4</td>
<td>226</td>
<td>14</td>
<td>13.1</td>
<td></td>
</tr>
<tr>
<td>C1PS2</td>
<td>104</td>
<td>32</td>
<td>13.6</td>
<td>117</td>
<td>12</td>
<td>11.1</td>
<td>46</td>
<td>14</td>
<td>12.2</td>
<td></td>
</tr>
<tr>
<td>C2PS1</td>
<td>947</td>
<td>19</td>
<td>11.9</td>
<td>915</td>
<td>12</td>
<td>11.3</td>
<td>253</td>
<td>14</td>
<td>13.1</td>
<td></td>
</tr>
<tr>
<td>C3PS1</td>
<td>45</td>
<td>18</td>
<td>4.5</td>
<td>44</td>
<td>12</td>
<td>10.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3PS2</td>
<td>8,752</td>
<td>15</td>
<td>10.2</td>
<td>7,396</td>
<td>12</td>
<td>11.3</td>
<td>1,633</td>
<td>14</td>
<td>13.3</td>
<td></td>
</tr>
<tr>
<td>C4PS1</td>
<td>14,472</td>
<td>15</td>
<td>10.3</td>
<td>15,689</td>
<td>9</td>
<td>8.9</td>
<td>2,747</td>
<td>14</td>
<td>13.3</td>
<td></td>
</tr>
<tr>
<td>C4PS2</td>
<td>18,542</td>
<td>15</td>
<td>10.3</td>
<td>17,746</td>
<td>10</td>
<td>10.1</td>
<td>5,731</td>
<td>14</td>
<td>14.83</td>
<td></td>
</tr>
<tr>
<td>C5PS1</td>
<td>5,491</td>
<td>10</td>
<td>7.9</td>
<td></td>
<td></td>
<td></td>
<td>1,884</td>
<td>14</td>
<td>13.3</td>
<td></td>
</tr>
<tr>
<td>C6PS1</td>
<td>340</td>
<td>20</td>
<td>6.7</td>
<td>330</td>
<td>12</td>
<td>11.4</td>
<td>143</td>
<td>15</td>
<td>12.9</td>
<td></td>
</tr>
<tr>
<td>C7PS1</td>
<td>183</td>
<td>20</td>
<td>12.3</td>
<td></td>
<td></td>
<td></td>
<td>116</td>
<td>15</td>
<td>12.9</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

1. Ineligible for compression
2. Test not performed

Figure 51 on page 97 shows the I/O load per megabyte of data set size. The graph also shows the benefit of compressing data in terms of reduced total I/O load to read the data set.
C.3.4 Write I/O Measurements for New Format Data Sets

We performed write I/O measurements for extended format and compressed format data sets using three utilities:

- IDCAMS REPRO
- DFSMSdss copy
- IEBGENER copy.

We ran three jobs for each data set and each utility used. We ran the first job to process the data set in its standard format; the second, to process the data set in its extended format; and the third, to process the data set in its compressed format.

We present the results in tables and graphs. In particular we show:

- I/O load data when writing VSAM KSDSs with IDCAMS
- I/O load data when writing sequential data sets with IEBGENER
- I/O load data when writing sequential data sets with DFSMSdss.

Figure 52 on page 98 shows the JCL we used to write a VSAM KSDS with IDCAMS. Note that this procedure always takes as input a standard format cluster and creates as output standard, extended, or compressed format clusters according to the type of measurements we wanted to perform.
Figure 52. JCL Used to Write a VSAM KSDS with IDCAMS REPRO

Table 18 shows the results of these measurements.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Write)</td>
<td>Service Time</td>
<td>Connect Time</td>
</tr>
<tr>
<td>C1VS1</td>
<td>10,054</td>
<td>17</td>
<td>7.4</td>
</tr>
<tr>
<td>C1VS2</td>
<td>1,850</td>
<td>16</td>
<td>7.2</td>
</tr>
<tr>
<td>C2VS1</td>
<td>5,536</td>
<td>17</td>
<td>7.2</td>
</tr>
<tr>
<td>C4VS1</td>
<td>512</td>
<td>14</td>
<td>6.7</td>
</tr>
<tr>
<td>C5VS1</td>
<td>156</td>
<td>9</td>
<td>3.8</td>
</tr>
<tr>
<td>C5VS2</td>
<td>15,396</td>
<td>16</td>
<td>8.8</td>
</tr>
<tr>
<td>C5VS3</td>
<td>2,586</td>
<td>23</td>
<td>8.9</td>
</tr>
<tr>
<td>C8VS1</td>
<td>27,370</td>
<td>19</td>
<td>10.5</td>
</tr>
<tr>
<td>C9VS1</td>
<td>10,009</td>
<td>16</td>
<td>8.0</td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression

Figure 53 on page 99 shows some of the data collected during the IDCAMS write I/O test for VSAM KSDSs. The graph shows the same behavior as the graph for VSAM KSDS read I/O in Figure 45 on page 91.
Because DFSMSdss does not support the change of the data set format during copy or restore operations, we did not perform measurements for write I/O on VSAM KSDSs.

![Write I/O Analysis: IDCAMS](image)

Figure 53. IDCAMS I/O Load for Writing VSAM KSDSs

Figure 54 shows the JCL we used to test the write operations on different data set formats using IEBGENER.

```jcl
//TW1IPSG JOB SYS1, 'FRED', MSGLEVEL=(1,1), MSGCLASS=Z,
//   NOTIFY=FRED
//COPY EXEC PGM=IEBGENER
//*-----------------------------------------------------*
//* -WRITE TEST *
//* .SEQUENTIAL STANDARD OUTPUT *
//* .IEBGENER *
//*-----------------------------------------------------*
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DISP=OLD, DSN=CUST01.STNCNS.SEQ.FILE01 
//SYSUT2 DD DSN=CUST01.STNCNS.SEQ.TEST, DISP=(NEW,DELETE,DELETE), 
//   SPACE=(1,(800)), AVGREC=M, 
//   SPACE=STORCLASS=GSPACE, VOL=SER=VM5Q21 
//SYSIN DD DUMMY 
//*
```

Figure 54. JCL Used to Write a Sequential Data Set with IEBGENER
Table 19 on page 100 shows the details of the data collected with this set of measurements.

### Table 19. Write I/O Measurements on Sequential Data Sets: IEBGENER

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Write)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td>Number of I/Os (Write)</td>
<td>Service Time</td>
<td>Connect Time</td>
<td>Number of I/Os (Write)</td>
<td>Service Time</td>
<td>Connect Time</td>
</tr>
<tr>
<td>C1PS1</td>
<td>437</td>
<td>50</td>
<td>20.1</td>
<td>419</td>
<td>47</td>
<td>21.0</td>
<td>261</td>
<td>43</td>
<td>13.1</td>
</tr>
<tr>
<td>C1PS2</td>
<td>123</td>
<td>36</td>
<td>14.5</td>
<td>103</td>
<td>40</td>
<td>17.2</td>
<td>80</td>
<td>30</td>
<td>8.9</td>
</tr>
<tr>
<td>C2PS1</td>
<td>650</td>
<td>48</td>
<td>20.8</td>
<td>637</td>
<td>48</td>
<td>21.4</td>
<td>287</td>
<td>35</td>
<td>13.3</td>
</tr>
<tr>
<td>C3PS1</td>
<td>75</td>
<td>24</td>
<td>9.5</td>
<td>53</td>
<td>29</td>
<td>12.9</td>
<td></td>
<td></td>
<td>(1)</td>
</tr>
<tr>
<td>C3PS2</td>
<td>4,982</td>
<td>51</td>
<td>22.0</td>
<td>5,033</td>
<td>50</td>
<td>21.8</td>
<td>1,667</td>
<td>34</td>
<td>14.7</td>
</tr>
<tr>
<td>C4PS1</td>
<td>8,193</td>
<td>50</td>
<td>22.1</td>
<td>8,186</td>
<td>51</td>
<td>22.2</td>
<td>2,782</td>
<td>39</td>
<td>14.8</td>
</tr>
<tr>
<td>C4PS2</td>
<td>10,563</td>
<td>51</td>
<td>22.2</td>
<td>10,532</td>
<td>50</td>
<td>22.1</td>
<td>5,773</td>
<td>40</td>
<td>14.9</td>
</tr>
<tr>
<td>C5PS1</td>
<td>2,908</td>
<td>47</td>
<td>18.6</td>
<td></td>
<td>(2)</td>
<td></td>
<td>1,918</td>
<td>40</td>
<td>14.8</td>
</tr>
<tr>
<td>C6PS1</td>
<td>267</td>
<td>44</td>
<td>18.7</td>
<td>244</td>
<td>47</td>
<td>20.1</td>
<td>161</td>
<td>36</td>
<td>12.3</td>
</tr>
<tr>
<td>C7PS1</td>
<td>170</td>
<td>41</td>
<td>16.5</td>
<td></td>
<td>(2)</td>
<td></td>
<td>150</td>
<td>32</td>
<td>11.7</td>
</tr>
</tbody>
</table>

**Note:**
- (1) Ineligible for compression
- (2) Test not performed

Figure 55 on page 101 shows some of the IEBGENER write I/O data from Table 19. The plotted values are calculated as I/O per megabyte of data contained in the data sets.
We performed another write test on sequential data sets using DFSMSdss. In this case we used the logical copy function, with the input and output data sets in the same format. Figure 56 on page 102 shows the JCL we used.
Figure 56. JCL Used to Write a Sequential Data Set with DFSMSdss

Table 20 on page 103 shows the data we collected with this series of tests.
### Table 20. Write I/O Measurements on Sequential Data Sets: DFSMSdss

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of I/Os (Write)</td>
<td>Service Time</td>
<td>Connect Time</td>
</tr>
<tr>
<td>C1PS1</td>
<td>706</td>
<td>21</td>
<td>11.0</td>
</tr>
<tr>
<td>C1PS2</td>
<td>314</td>
<td>14</td>
<td>5.9</td>
</tr>
<tr>
<td>C2PS1</td>
<td>980</td>
<td>26</td>
<td>12.0</td>
</tr>
<tr>
<td>C3PS1</td>
<td>253</td>
<td>13</td>
<td>3.8</td>
</tr>
<tr>
<td>C3PS2</td>
<td>6,380</td>
<td>39</td>
<td>14.4</td>
</tr>
<tr>
<td>C4PS1</td>
<td>10,398</td>
<td>33</td>
<td>14.7</td>
</tr>
<tr>
<td>C4PS2</td>
<td>13,329</td>
<td>35</td>
<td>14.7</td>
</tr>
<tr>
<td>C5PS1</td>
<td>3,791</td>
<td>40</td>
<td>11.7</td>
</tr>
<tr>
<td>C6PS1</td>
<td>501</td>
<td>27</td>
<td>9.2</td>
</tr>
<tr>
<td>C7PS1</td>
<td>369</td>
<td>22</td>
<td>7.3</td>
</tr>
</tbody>
</table>

**Note:**
- (1) Ineligible for compression
- (2) Test not performed

We plotted some of the DFSMSdss write I/O data from this table in the graph in Figure 57 on page 104.
As you know, DFSMSdss does not convert the format of the data set during a copy operation. In most cases DFSMSdss does not activate the compress or decompress function. The reduction in the number of I/Os per megabyte that you see in Figure 57 when the compressed format is copied is attributable to the reduced data set size achieved with compression.

C.3.5 Elapsed Times and CPU Times

During the tests we describe in C.3.4, "Write I/O Measurements for New Format Data Sets" on page 97, we also recorded the following SMF records that contain information related to the jobs we executed:

- Type-30—step-end and job-end information
- Type-14—input or readback data set activity
- Type-15—output or update data set activity
- Type-62—VSAM component open
- Type-64—VSAM component close.

We processed these SMF records with analysis CLISTs to calculate values of:

- Job CPU times
- Job elapsed times.
We present the results of the measurements in the form of tables and graphs. In particular we show:

- CPU and elapsed time data when reading VSAM KSDSs with IDCAMS
- CPU and elapsed time data when reading VSAM KSDSs with DFSMSdss
- CPU and elapsed time data when reading sequential data sets with DFSMSdss
- CPU and elapsed time data when reading sequential data sets with IEBGENER
- CPU and elapsed time data when writing VSAM KSDSs with IDCAMS
- CPU and elapsed time data when writing sequential data sets with DFSMSdss
- CPU and elapsed time data when writing sequential data sets with IEBGENER.

Table 21 shows the elapsed and CPU time measurements when reading VSAM KSDSs with DFSMSdss and IDCAMS.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DFSMSdss</td>
<td>IDCAMS</td>
<td>DFSMSdss</td>
</tr>
<tr>
<td>Time Resident</td>
<td>Total CPU Time</td>
<td>Time Resident</td>
<td>Total CPU Time</td>
</tr>
<tr>
<td>C1VS1</td>
<td>130.6</td>
<td>5.62</td>
<td>128.1</td>
</tr>
<tr>
<td>C1VS2</td>
<td>22.9</td>
<td>1.18</td>
<td>24.2</td>
</tr>
<tr>
<td>C2VS1</td>
<td>92.8</td>
<td>3.76</td>
<td>91.5</td>
</tr>
<tr>
<td>C4VS1</td>
<td>9.7</td>
<td>0.62</td>
<td>8.5</td>
</tr>
<tr>
<td>C5VS1</td>
<td>5.7</td>
<td>0.17</td>
<td>2.1</td>
</tr>
<tr>
<td>C5VS2</td>
<td>371.5</td>
<td>14.94</td>
<td>339.9</td>
</tr>
<tr>
<td>C5VS3</td>
<td>53.0</td>
<td>1.58</td>
<td>34.3</td>
</tr>
<tr>
<td>C8VS1</td>
<td>485.5</td>
<td>20.97</td>
<td>290.1</td>
</tr>
<tr>
<td>C9VS1</td>
<td>166.8</td>
<td>5.43</td>
<td>176.4</td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression

We plotted the values recorded for IDCAMS (Figure 58 on page 106). As you can see, there is an increase in the CPU load when the data set is compressed. The increase is caused by the activation of the decompression algorithm.
Figure 59 on page 107 shows how the elapsed times (also called resident times) change when accessing the different data set formats. As you can see, the trend is a general reduction in the duration of the jobs when the data sets are in compressed format.
Figure 59. IDCAMS Elapsed Times When Reading VSAM KSDSs

Figure 60 on page 108 shows the CPU used by DFSMSdss to perform the same read activity. In this case the CPU load does not increase when the data set is compressed because DFSMSdss does not perform any format conversion and therefore does not call the compression services.
Figure 60. CPU Used by DFSMSdss When Reading VSAM KSDSs

Figure 61 on page 109 shows the differences in elapsed time when processing the extended format and compressed format data sets and the elapsed time required to process the same data sets in standard format. As you can see, elapsed time increases for extended format processing. This increase is caused by the different technique DFSMSdss uses to process the extended format. You can also note that the elapsed times show a reduction when the data set is in a compressed format. This reduction is caused by the smaller data set size.
Table 22 on page 110 shows the results of the CPU and elapsed time measurements when reading sequential data sets with DFSMSdss and IEBGENER. Figure 62 on page 111, Figure 63 on page 112, Figure 64 on page 113, and Figure 65 on page 114 graph the data.
Table 22. Elapsed and CPU Time Measurements When Reading Sequential Data Sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DFSMSdss</td>
<td>IEBGENER</td>
<td>DFSMSdss</td>
</tr>
<tr>
<td></td>
<td>Time Resident</td>
<td>Total CPU Time</td>
<td>Time Resident</td>
</tr>
<tr>
<td>C1PS1</td>
<td>28.0</td>
<td>0.6</td>
<td>22.2</td>
</tr>
<tr>
<td>C1PS2</td>
<td>9.0</td>
<td>0.19</td>
<td>4.6</td>
</tr>
<tr>
<td>C2PS1</td>
<td>37.7</td>
<td>0.86</td>
<td>34.9</td>
</tr>
<tr>
<td>C3PS1</td>
<td>4.5</td>
<td>0.12</td>
<td>2.4</td>
</tr>
<tr>
<td>C3PS2</td>
<td>285.1</td>
<td>6.68</td>
<td>269.4</td>
</tr>
<tr>
<td>C4PS1</td>
<td>441.2</td>
<td>10.48</td>
<td>444.0</td>
</tr>
<tr>
<td>C4PS2</td>
<td>619.9</td>
<td>13.88</td>
<td>575.6</td>
</tr>
<tr>
<td>C5PS1</td>
<td>156.9</td>
<td>3.47</td>
<td>140.4</td>
</tr>
<tr>
<td>C6PS1</td>
<td>16.8</td>
<td>0.37</td>
<td>12.7</td>
</tr>
<tr>
<td>C7PS1</td>
<td>19.3</td>
<td>0.27</td>
<td>6.8</td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression
(2) Test not performed
Figure 62. CPU Used by DFSMSdss When Reading Sequential Data Sets

Figure 63 on page 112 shows how the elapsed times change when processing different data set formats. The comments we make for Figure 61 on page 109 also apply in this case.
Figure 64 on page 113 shows the CPU used by IEBGENER to read sequential data sets. Note the increase in CPU when compressed data sets are used. The need to interface the compression services to decompress the data after read causes the increase.
Figure 64. CPU Used by IEBGENER When Reading Sequential Data Sets

Figure 65 on page 114 shows the changes in job elapsed time when comparing extended format and compressed format with standard format. As you can see the new formats allow a reduction in processing time when the data sets are converted to the new formats.
Table 23 on page 115 shows the results of the measurements performed when writing VSAM KSDSs with IDCAMS.
Table 23. Elapsed and CPU Time Measurements When Writing VSAM KSDSs

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Total CPU Time</td>
<td>Time</td>
</tr>
<tr>
<td>C1VS1</td>
<td>255.3</td>
<td>11.51</td>
<td>283.8</td>
</tr>
<tr>
<td>C1VS2</td>
<td>47.9</td>
<td>3.05</td>
<td>46.6</td>
</tr>
<tr>
<td>C2VS1</td>
<td>150.8</td>
<td>6.77</td>
<td>146.5</td>
</tr>
<tr>
<td>C4VS1</td>
<td>20.4</td>
<td>3.72</td>
<td>25.9</td>
</tr>
<tr>
<td>C5VS1</td>
<td>8.8</td>
<td>0.20</td>
<td>5.1</td>
</tr>
<tr>
<td>C5VS2</td>
<td>449.9</td>
<td>30.83</td>
<td>496.2</td>
</tr>
<tr>
<td>C5VS3</td>
<td>94.12</td>
<td>1.67</td>
<td>92.7</td>
</tr>
<tr>
<td>C8VS1</td>
<td>711.7</td>
<td>36.72</td>
<td>744.2</td>
</tr>
<tr>
<td>C9VS1</td>
<td>289.5</td>
<td>8.30</td>
<td>263.5</td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression

The values from Table 23 are shown in Figure 66 on page 116. As you see, the use of compression services by IDCAMS increases the CPU load when the output data set is in compressed format.
The elapsed times measured with this set of tests decrease when the data set is converted from standard format to compressed format (see Figure 67 on page 117).
Figure 67. IDCAMS Elapsed Times When Writing VSAM KSDSs

Table 24 on page 118 shows the results of the write test on sequential data sets using DFSMSdss and IEBGENER.
Table 24. Elapsed and CPU Time Measurements Writing Sequential Data Sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Standard Format</th>
<th>Extended Format (Noncompressed)</th>
<th>Extended Format (Compressed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DFSMSdss</td>
<td>IEBGENER</td>
<td>DFSMSdss</td>
</tr>
<tr>
<td></td>
<td>Time Resident</td>
<td>Total CPU Time</td>
<td>Time Resident</td>
</tr>
<tr>
<td>C1PS1</td>
<td>27.6</td>
<td>0.52</td>
<td>21.9</td>
</tr>
<tr>
<td>C1PS2</td>
<td>10.5</td>
<td>0.3</td>
<td>4.5</td>
</tr>
<tr>
<td>C2PS1</td>
<td>48.9</td>
<td>0.7</td>
<td>31.8</td>
</tr>
<tr>
<td>C3PS1</td>
<td>8.5</td>
<td>0.29</td>
<td>2.0</td>
</tr>
<tr>
<td>C3PS2</td>
<td>275.7</td>
<td>3.73</td>
<td>253.9</td>
</tr>
<tr>
<td>C4PS1</td>
<td>472.2</td>
<td>6.05</td>
<td>415.1</td>
</tr>
<tr>
<td>C4PS2</td>
<td>555.0</td>
<td>7.78</td>
<td>537.8</td>
</tr>
<tr>
<td>C5PS1</td>
<td>158.9</td>
<td>2.22</td>
<td>136.6</td>
</tr>
<tr>
<td>C6PS1</td>
<td>19.1</td>
<td>0.44</td>
<td>11.8</td>
</tr>
<tr>
<td>C7PS1</td>
<td>11.8</td>
<td>0.34</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Note:
(1) Ineligible for compression
(2) Test not performed

The DFSMSdss data from this table is shown in Figure 68 on page 119. As you see, the differences among the three data set formats are similar to those shown in Figure 62 on page 111 when reading the same data sets.
Figure 68. CPU Used by DFSMSdss When Writing Sequential Data Sets

The elapsed time decreases when the data set is in compressed format.
Figure 70 shows the IEBGENER data from Table 24 on page 118. In this case the CPU load increases when the data set is in compressed format.
Figure 70. CPU Used by IEBGENER When Writing Sequential Data Sets

Appendix C. Performance Measurement Details

Figure 71 on page 122 shows the changes in IEBGENER elapsed time when writing sequential data sets.
Figure 71. IEBGENER Elapsed Times When Writing Sequential Data Sets
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACB</td>
<td>access method control block</td>
</tr>
<tr>
<td>ACDS</td>
<td>active control data set</td>
</tr>
<tr>
<td>ACS</td>
<td>automatic class selection</td>
</tr>
<tr>
<td>ADSM</td>
<td>ADSTAR Distributed Storage Manager</td>
</tr>
<tr>
<td>AIX</td>
<td>alternate index</td>
</tr>
<tr>
<td>APAR</td>
<td>authorized program analysis report</td>
</tr>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>BDW</td>
<td>block descriptor word</td>
</tr>
<tr>
<td>BWO</td>
<td>backup while open</td>
</tr>
<tr>
<td>BSAM</td>
<td>basic sequential access method</td>
</tr>
<tr>
<td>CA</td>
<td>control area</td>
</tr>
<tr>
<td>CDS</td>
<td>control data set</td>
</tr>
<tr>
<td>CI</td>
<td>control interval</td>
</tr>
<tr>
<td>CDRA</td>
<td>character data representation architecture</td>
</tr>
<tr>
<td>CMS</td>
<td>Compression Management Services</td>
</tr>
<tr>
<td>COMMDS</td>
<td>communication control data set</td>
</tr>
<tr>
<td>DADSM</td>
<td>direct access device space management</td>
</tr>
<tr>
<td>DAS</td>
<td>Data Access Services</td>
</tr>
<tr>
<td>DBB</td>
<td>dictionary building block</td>
</tr>
<tr>
<td>DDM</td>
<td>Distributed Data Management</td>
</tr>
<tr>
<td>EOV</td>
<td>end of volume</td>
</tr>
<tr>
<td>ESCON</td>
<td>Enterprise Systems Connection</td>
</tr>
<tr>
<td>EXCP</td>
<td>execute channel program</td>
</tr>
<tr>
<td>HDA</td>
<td>head disk assembly</td>
</tr>
<tr>
<td>HFS</td>
<td>hierarchical file system</td>
</tr>
<tr>
<td>IART</td>
<td>initial access response time</td>
</tr>
<tr>
<td>ICF</td>
<td>integrated catalog facility</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines Corporation</td>
</tr>
<tr>
<td>IDRC</td>
<td>improved data recording capability</td>
</tr>
<tr>
<td>IPL</td>
<td>initial program load</td>
</tr>
<tr>
<td>ISMF</td>
<td>Interactive Storage Management Facility</td>
</tr>
<tr>
<td>ITSO</td>
<td>International Technical Support Organization</td>
</tr>
<tr>
<td>KSDS</td>
<td>key-sequenced data set</td>
</tr>
<tr>
<td>LIC</td>
<td>licensed internal code</td>
</tr>
<tr>
<td>MSR</td>
<td>millisecond response time</td>
</tr>
<tr>
<td>OAM</td>
<td>Object Access Method</td>
</tr>
<tr>
<td>OEMI</td>
<td>original equipment manufacturer interface</td>
</tr>
<tr>
<td>PDS</td>
<td>partitioned data set</td>
</tr>
<tr>
<td>PDSE</td>
<td>partitioned data set extended</td>
</tr>
<tr>
<td>PTF</td>
<td>program temporary fix</td>
</tr>
<tr>
<td>QSAM</td>
<td>queued sequential access method</td>
</tr>
<tr>
<td>RBA</td>
<td>relative byte address</td>
</tr>
<tr>
<td>RDW</td>
<td>record descriptor word</td>
</tr>
<tr>
<td>RMF</td>
<td>Resource Measurement Facility</td>
</tr>
<tr>
<td>RPL</td>
<td>request parameter list</td>
</tr>
<tr>
<td>SMF</td>
<td>System Management Facility</td>
</tr>
<tr>
<td>SMS</td>
<td>Storage Management Subsystem</td>
</tr>
<tr>
<td>SVC</td>
<td>supervisor call instruction</td>
</tr>
<tr>
<td>TCB</td>
<td>task control block</td>
</tr>
<tr>
<td>TMM</td>
<td>Tape Mount Management</td>
</tr>
<tr>
<td>VVDS</td>
<td>VSAM volume data set</td>
</tr>
<tr>
<td>WORM</td>
<td>write-once-read-many</td>
</tr>
<tr>
<td>XRC</td>
<td>extended remote copy</td>
</tr>
<tr>
<td>PPRC</td>
<td>peer-to-peer remote copy</td>
</tr>
</tbody>
</table>
Index

Numerics
3390 Model 2 and 3  70
3495  2
3990 Models 3 and 6, or cached 9340
  3990 Model 3  7
  feature codes  7
  3990 Model 6  1
  extended remote copy  1
  peer-to-peer remote copy  1
  ESCON  1, 3, 7, 48, 70
  OEMI  1
3995 Models 131, 132, and 133  3
  See also DFSMS/MVS, OAM, support for the 3995
4GB restriction
  See VSAM, KSDS, restriction of 4GB
6201, 6202, 6203, or 9203 feature code
  See also 3990 Models 3 and 6, or cached 9340,
  3990 Model 3, feature codes
9340 Model DC4  7
  EC 486392  7
9021 711-based processors
  See ES/9000 processors, 9021 711-based
  processors
9121 511-based processors
  See ES/9000 processors, 9121 511-based
  processors
9221 211-based processors
  See ES/9000 processors, 9221 211-based
  processors

A
ACB
  See VSAM, KSDS, ACB
access method control block
  See VSAM, KSDS, ACB
ACDS
  See DFSMS/MVS, SMS, ACDS
ACS routines
  See DFSMS/MVS, ISMF, ACS routines
active control data set
  See DFSMS/MVS, SMS, ACDS
ADR-messages
  See message, ADR-messages
ADSM  1
ADSTAR Distributed Storage Manager
  See ADSM
AIX
  See VSAM, KSDS, AIX
alternate index
  See VSAM, KSDS, AIX
APAR
  II07816  4, 61
  OY67882  37
  APAR (continued)
    PN43715  67
    PN43718  67
    PN43721  67
    PN48274  37
API
  See DFSMS/MVS, OpenEdition MVS, API
application programming interface
  See DFSMS/MVS, OpenEdition MVS, API
Assembler
  See languages, Assembler
  automatic class selection
    See DFSMS/MVS, ISMF, ACS routines
B
backup-while open
  See CICS, BWO
basic sequential access method
  See BSAM
BDW  45
  block descriptor word
    See BDW
BSAM  4, 8, 30–31, 61, 67
BWO
  See CICS, BWO
BYPASSACS
  See DFSMS/MVS, DFSMSdss, BYPASSACS
C
C
  See languages, C
CA
  See VSAM, CA
candidate data set verification
  See Compression Management Facility,
  Compression Management Services, candidate
  data set verification
catalog
  See system services, catalog
CDS
  See DFSMS/MVS, DFSMSshm, CDS
CI
  See VSAM, CI
CI freespace
    See VSAM, CI, freespace
CI mode processing
    See VSAM, CI, CI mode processing
CI size
    See VSAM, CI, size
CI split
    See VSAM, CI, split
CICS  2, 11, 29, 32
  BWO  2
CICS (continued)
   CICSVR 2
   See CICS, CICSVR
   CLOSE 1, 15, 23–24, 31
CMS
   See Compression Management Facility,
   Compression Management Services
COMMDS
   See DFSMS/MVS, SMS, COMMDS
communications control data set
   See DFSMS/MVS, SMS, COMMDS
compaction
   See DFSMS/MVS, ISMF, compaction
compression feature
   See ES/9000 processors
compression flags
   See system services, catalog, compression flags
compression indicator
   See system services, VTOC, DS1COMPR
Compression Management Facility 7, 9
   Compression Management Services 8
   candidate data set verification 8
   data compression 8, 10, 29, 43, 56
   data decompression 8, 10, 12, 33, 104–105, 112
   dictionary sampling 8, 14
   dictionary selection 8
   compression services activation 8, 29, 32, 47, 107, 112, 115
   building blocks 8
   SYS1.DBBLIB 8
compression-eligible data set
   dictionary selection 14
   mated 10, 15
   rejected 15
concurrent copy
   See DFSMS/MVS, DFSMSdss, concurrent copy
control interval
   See VSAM, CI
control area
   See VSAM, CA
control data set
   See DFSMS/MVS, DFSMSHshm, CDS
control interval freespace
   See VSAM, CI, freespace
control interval size
   See VSAM, CI, size
control interval split
   See VSAM, CI, split
CONVERTV
   See DFSMS/MVS, DFSMSdss, CONVERTV
COPYDUMP
   See DFSMS/MVS, DFSMSdss, COPYDUMP

D
DADSM
   See system services, DADSM
DAS
   See OS/2, DAS
DASD buffers
   See TMM, DASD buffers
Data Access Services
   See OS/2, DAS
data and storage resources
   See MVS/ESA, data and storage resources
data compression
   See Compression Management Facility,
   Compression Management Services, data compression
data decompression
   See Compression Management Facility,
   Compression Management Services, data decompression
data mover
   See DFSMS/MVS, DFSMSdss, data mover
Data Set Name Type
   See DFSMS/MVS, ISMF, Data Set Name Type
DB2 4, 11, 32
DBB
   See Compression Management Facility,
   compression services activation, SYS1.DBBLIB
DBB distribution library
   See Compression Management Facility,
   compression services activation, SYS1.DBBLIB
DCollect
   See system utilities, IDCAMS, DCollect
DDM 1
DEFRAG
   See DFSMS/MVS, DFSMSdss, DEFRAG
device allocation
   See DFSMS/MVS, SMS, device allocation
DFDSS 35, 37
DFSMS/MVS
   DFSMSdp 1, 64
   Distributed FileManager/MVS 1
DFSMSdss
   BYPASSACS 36–37, 102
   concurrent copy 2
   CONVERTV 31–38
   COPYDUMP 31–38
   data mover 2, 38, 54
   DEFRAG 31–38
   logical data set dump 31–38
   logical data set restore 31–38
   NULLSTORCLAS 36–37
   physical volume COPY 31–38
   physical volume DUMP/RESTORE 31–38
   PRINT 31–38
   stand-alone volume restore 31–38
DFSMSHshm 1, 12, 38, 41, 64
   extended private address space 2
   multiple tape output buffers 2
DFSMS/MVS (continued)

DFSMShsm (continued)
primary level 0 2
primary space management 1
tape output processing 2
tape recycle performance 2
virtual storage constraint 2

DFSMSrmm 2
return-to-scratch exit 2
scratch pools 2
volume-not-in-library installation exit 2

ISMF
ACS routines 17, 20, 24, 27, 43—45, 63, 71—74
compaction 17, 19—20, 22, 25, 43—45, 47, 61, 70
Data Set Name Type 19—21, 24, 43, 61, 70
guaranteed space 22
If Ext 20—21, 43—44, 70
MSR 22, 26—27
sustained data rate 22, 43, 71

OAM 3
statistics 3
support for the 3995 3
tape support 3
WORM 3

OpenEdition MVS 4
API 4
HFS 4
POSIX standards 4

SMS
ACDS 3
COMMDS 3
device allocation 3
SMS-managed 39, 61
SMSplex 3
system group names 3
system-determined block size 79, 83

DFSMSdip
See DFSMS/MVS, DFSMSdip

DFSMShsm
See DFSMS/MVS, DFSMShsm

DFSMSrmm
See DFSMS/MVS, DFSMSrmm

DFSORT 39, 63
dictionary sampling
See Compression Management Facility, Compression Management Services, dictionary sampling
dictionary selection
See Compression Management Facility, Compression Management Services, dictionary selection
See compression-eligible data set, dictionary selection
dictionary token
See system services, catalog, dictionary token
direct access device space management
See system services, DADSM

Distributed Data Management
See DDM
Distributed FileManager/MVS
See DFSMS/MVS, DFSMSdip, Distributed FileManager/MVS

DS1BLKL
See system services, VTOC, DS1BLKL

DS1COMPR
See system services, VTOC, DS1COMPR

DS1STRP
See system services, VTOC, DS1STRP

E
EC 486392
See 6201, 6202, 6203, or 9203 feature code, 9340 Model DC4

ES/9000 processors 4
9021 711-based processors 7
SEC C35945 7
9121 511-based processors 7
SEC 228250 7
9221 211-based processors 7
compression feature 4

ESCON
See 3990 Models 3 and 6, or cached 9340, ESCON
EXCP 14, 31, 61
execute channel program
See EXCP

EXPORT
See system utilities, IDCAMS, EXPORT
extended format cell
See system services, catalog, extended format cell
extended format data set indicator
See system services, VTOC, DS1STRP
extended private address space
See DFSMS/MVS, DFSMShsm, extended private address space
extended remote copy
See 3990 Models 3 and 6, or cached 9340, 3990 Model 6, extended remote copy

F
Flash 4014 IMS and DFSMS/MVS 1.2 11, 67
forward recovery
See CICS, forward recovery
freespace
See VSAM, CI, freespace

G
guaranteed space
See DFSMS/MVS, ISMF, guaranteed space

H
hardware compression feature
See ES/9000 processors, compression feature
See DFSMS/MVS, OpenEdition MVS, HFS hierarchical file system

See DFSMS/MVS, OpenEdition MVS, HFS

Hiperbatch 46, 63

ICF 2, 10, 38, 45

IDCAMS
See system utilities, IDCAMS

IDRC 4

IEBCOMPR
See system utilities, IEBCOMPR

IEBCOPY
See system utilities, IEBCOPY

IEBDG
See system utilities, IEBDG

IEBEDIT
See system utilities, IEBEDIT

IEBGENER
See system utilities, IEBGENER

IEBPROGM
See system utilities, IEBPROGM

IEBPTPC
See system utilities, IEBPTPC

IEBUPDTE
See system utilities, IEBUPDTE

IEHLIST
See system utilities, IEHLIST

If Ext
See DFSMS/MVS, ISMF, If Ext

IGD-messages
See message, IGD-messages

II07816
See APAR
See temporary data sets

IMBED
See VSAM, KSDS, IMBED

IMPORT
See system utilities, IDCAMS, IMPORT

improved data recording capability
See IDRC

IMS/ESA 4, 11, 32, 67

initial program load
See IPL

integrated catalog facility
See ICF

internal CI format
See VSAM, CI, internal format

IPL 7

ISMF
See DFSMS/MVS, ISMF

job control language
See JCL

key sequenced data set
See VSAM, KSDS

KEYRANGES
See VSAM, KSDS, KEYRANGES

KSDS
See VSAM, KSDS

languages
Assembler 4
C 4

Licensed Internal Code 7

LISTCAT
See system services, catalog, LISTCAT

general data set dump
See DFSMS/MVS, DFSMSdss, logical data set dump

logical data set RESTORE
See DFSMS/MVS, DFSMSdss, logical data set RESTORE

mated
See compression-eligible data set, mated

message 7, 21, 35–36, 47–48

ADR-messages 35–36

IGD-messages 21, 47–48

millisecond response time
See DFSMS/MVS, ISMF, MSR

MSR
See DFSMS/MVS, ISMF, MSR

multiple tape output buffers
See DFSMS/MVS, DFSMSshm, multiple tape output buffers

multistriped data set
See striping, multistriped data set

MVS/DFP 29

MVS/ESA
data and storage resources 1

RMF 3, 78, 89

non-system-managed 36–37

NULLSTORCLAS
See DFSMS/MVS, DFSMSdss, NULLSTORCLAS

number of stripes
See system services, catalog, number of stripes

O

OAM
See DFSMS/MVS, OAM
OAM tape support
See DFSMS/MVS, OAM, tape support
Object Acces Method
See DFSMS/MVS, OAM
OEMI
See 3990 Models 3 and 6, or cached 9340, OEMI
OPEN 11, 31
OS/2 1
DAS 1
OY67882
See APAR

P

partial release of space
See VSAM, KSDS, partial release of space
partitioned data set
See PDS
partitioned data set extended
See PDSE
PDS 8
PDSE 8
peer-to-peer remote copy
See 3990 Models 3 and 6, or cached 9340, 3990 Model 6, peer-to-peer remote copy
physical block size
See system services, catalog, physical block size
physical volume COPY
See DFSMS/MVS, DFSMSdss, physical volume COPY
physical volume DUMP/RESTORE
See DFSMS/MVS, DFSMSdss, physical volume DUMP/RESTORE
PN43715
See APAR
PN43718
See APAR
PN43721
See APAR
PN48274
See APAR
POSIX standards
See DFSMS/MVS, OpenEdition MVS, POSIX standards
primary level 0
See DFSMS/MVS, DFSMSshm, primary level 0
primary space management
See DFSMS/MVS, DFSMSshm, primary space management
PRINT
See DFSMS/MVS, DFSMSdss, PRINT
program temporary fix
See PTF
PTF
UY91011 7

Q

QSAM 4, 8, 30
queued sequential access method
See QSAM

R

RBA
See VSAM, KSDS, RBA
RDW 45
record descriptor word
See RDW
rejected
See compression-eligible data set, rejected
relative byte address
See VSAM, KSDS, RBA
REPRO
See system utilities, IDCAMS, REPRO
request parameter list
See VSAM, KSDS, RPL
Resource Measurement Facility
See MVS/ESA, RMF
restriction of 4GB
See VSAM, KSDS, restriction of 4GB
return-to-scratch exit
See DFSMS/MVS, DFSMSrmm, return-to-scratch exit
RMF
See MVS/ESA, RMF
RPL
See VSAM, KSDS, RPL

S

S/390 4
S/390 hardware compression feature 4
See also ES/9000 processors, compression feature sampling
See Compression Management Facility, Compression Management Services, dictionary sampling
SAS 30
scratch pools
See DFSMS/MVS, DFSMSrmm, scratch pools
SEC 228250
See ES/9000 processors, 9121 511-based processors, SEC 228250
SEC C35945
See ES/9000 processors, 9021 711-based processors, SEC C35945
service request block
See SRB
single-striped data set
See striping, single-striped data set
SMF
See system services, SMF
SMS
See DFSMS/MVS, SMS
SMS-managed
See DFSMS/MVS, SMS, SMS-managed
SMSplex
See DFSMS/MVS, SMS, SMSplex
sphere
See VSAM, sphere
spill volume 26, 28
SRB 30–31
stand-alone volume RESTORE
See DFSMS/MVS, DFSMSdss, stand-alone volume RESTORE
standards
See DFSMS/MVS, OpenEdition MVS, POSIX standards
Statistical Analysis System
See SAS
Storage Management Subsystem
See DFSMS/MVS, SMS
striped data set
See striping, striped data set
striping
multistriped data set 13, 22–23, 29, 43, 45, 59
single-striped data set 24, 61, 79
striped data set 13
supervisor call instruction
See SVC
support for the 3995
See DFSMS/MVS, OAM, support for the 3995
sustained data rate
See DFSMS/MVS, ISMF, sustained data rate
SVC 46
SYS1.DBBLIB
See Compression Management Facility, compression services activation, SYS1.DBBLIB
system group names
See DFSMS/MVS, SMS, system group names
System Management Facility
See system services, SMF
system services
catalog 2, 8–12, 14, 33, 38–39, 45, 78
compression flags 10, 34
dictionary token 7–6, 10, 14–15, 33–34, 46
extended format cell 10–12, 14, 33–34, 38–39
LISTCAT 10, 33–34, 47, 78, 80
number of stripes 10, 12, 17, 34, 45, 47, 61, 75–77
physical block size 10, 12, 17, 34, 45, 47, 61, 75–77
VDDS 10, 33
DADSM 11, 23
SMF 10, 30–31, 47, 78, 104
type 14 10, 30, 104
type 15 10, 30, 104
type 30 30, 104
type 62 30–31, 104
type 64 10, 30–31, 104
VTCLC 11
DS1BLKL 63
DS1COMPR 11
DS1STRP 11
system utilities
IDCAMS 11, 33, 35, 39, 48, 55, 58, 78–117
dcollect 12, 34, 38
export 11, 33, 40
import 11, 33, 40
repro 12, 33–39, 48, 55, 78–98
iebcompr 12
iebcopy 12
iebdg 12
iebedit 12
iebgener 12, 39, 53–55, 58, 78–121
iebprogram 12
iebptpch 12
iebupdate 12
iehlalist 12, 38
system-determined block size
See DFSMS/MVS, SMS, system-determined block size
system-managed 1, 2, 13, 32, 38, 43, 61
system-managed tape 2
T
tape output processing
See DFSMS/MVS, DFSMSHsm, tape output processing
tape recycle performance
See DFSMS/MVS, DFSMSHsm, tape recycle performance
task control block
See TCB
TCB 30–31, 46
temporary data sets 4, 61
TMM 27–28
DASD buffers 27
type 14
See system services, SMF, type 14
type 15
See system services, SMF, type 15
type 30
See system services, SMF, type 30
type 62
See system services, SMF, type 62
type 64
See system services, SMF, type 64
U
UY91011
See PTF
V
virtual storage constraint
See DFSMS/MVS, DFSMSHsm, virtual storage constraint
volume-not-in-library installation exit
See DFSMS/MVS, DFSMSrmm, volume-not-in-library installation exit
VSAM
  CA  2, 22, 24—25
CI
  CI mode processing  46
  freespace  22, 25
  internal format  67
  size  18, 25, 31, 34—35, 44, 78—88
  split  2
KSDS
  ACB  24, 47
  AIX  24, 35—36, 44
  IMBED  22, 25, 44
  KEYRANGES  44
  partial release of space  2, 11—12, 17, 23—24, 44, 71, 73, 82, 88
  RBA  5, 24, 46
  restriction of 4GB  2
  RPL  46, 67
  sphere  44
VSAM volume data set
  See system services, catalog, VVDS
VTAM  11
VTOC
  See system services, VTOC
VVDS
  See system services, catalog, VVDS

W
WORM
  See DFSMS/MVS, OAM, WORM
write-once-read-many
  See DFSMS/MVS, OAM, WORM

X
XRC
  See 3990 Models 3 and 6, or cached 9340, 3990
  Model 6, extended remote copy
DFSMS/MVS Version 1 Release 2.0
Data Compression Implementation Guide
Publication No. GG24-4251-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. Please fill out this questionnaire and return it using one of the following methods:

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction

Organization of the book
Accuracy of the information
Relevance of the information
Completeness of the information
Value of illustrations
Grammar/punctuation/spelling
Ease of reading and understanding
Ease of finding information
Level of technical detail
Print quality

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:
   Do you provide billable services for 20% or more of your time? Yes___ No___
   Are you in a Services Organization? Yes___ No___

b) Are you working in the USA? Yes___ No___

c) Was the Bulletin published in time for your needs? Yes___ No___

d) Did this Bulletin meet your needs? Yes___ No___
   If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name

Address

Company or Organization

Phone No.