Redbooks
ibm.com/redbooks

DFSMS: Extended Address Volume

Keith Winnard

Alvaro Salla

Jose Gilberto Biondo Junior

Barbara White McDonald

z Systems

IBM

Redpaper

International Technical Support Organization

DFSMS: Extended Address Volume

July 2016

REDP-5364-00

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

First Edition (July 2016)

This edition applies to z/OS Version 2, Release 2 - Data Facility Storage Management Subsystem (DFSMS) (product number 5694-A01).

This document was created or updated on June 30, 2016.

Contents

Notices v
Trademarks vi

IBM Redbooks promotions vii

Preface ix
Authors ix
Now you can become a published author, too! x
Comments welcome x
Stay connected to IBM Redbooks x

Chapter 1. Introduction to extended address volumes 1
1.1 What to do about too much data 2
1.1.1 Data problem solutions 2
1.2 About the EAV solution 3
1.2.1 EAV value 3
1.3 EAV design 4
1.3.1 Parallel access volume and HyperPAV 4
1.3.2 Space management 4
1.3.3 Track addresses 4
1.4 EAS data sets 6
1.4.1 Allocating space on EAV 8
1.5 Volume serial and VTOC 9

Chapter 2. Planning for extended address volumes 11
2.1 Hardware and software requirements for EAV 12
2.2 Planning steps for EAV 13
2.2.1 Using the EATTR value 13
2.2.2 Using the Generic Tracker Facility 13
2.2.3 Migration considerations 16
2.2.4 Configuring and enabling EAV volumes 17
2.2.5 Applying BPV 17
2.2.6 Implementation tips 18
2.2.7 Providing test environment to validate readiness 18
2.2.8 Optionally migrate data 19
2.3 Aiding migration with dynamic volume expansion 19

Chapter 3. Implementation tips 21
3.1 Introduction to DFSMShsm 22
3.2 Implementation considerations 23
3.2.1 EAVs and DFSMShsm 23
3.2.2 Migration subtasking in z/OS V2R1 23
3.2.3 DFSMS storage tiers enhancements 23
3.2.4 EAV and Catalog/IDCAMS 24
3.2.5 EAV and DFSORT 24
3.2.6 EAV and Data Set Services 25
3.2.7 Other considerations 25
3.3 Implementing EAV 26

Appendix A. System macros and more 29
Manipulating track changes 30
IECTRKAD macro 31
OBTAIN, CAMLST, and CAVFDIR macros 31
CVAFDIR macro 32
DEVTYPE (SVC 24) macro 33
EAS-eligible - Finding affected programs 33
LSPACE macro 34
DATATYPE keyword 34
DADSM compatibility issues 35

Related publications 37
IBM Redbooks 37
Other publications 37
Online resources 37
Help from IBM 38

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®	IBM z™	Redpaper™
DB2®	IBM z Systems™	Redbooks (logo)  ®
DS8000®	IBM z13™	RMF™
ECKD™	IMS™	z Systems™
FlashCopy®	MVS™	z/OS®
IBM®	Redbooks®	z13™

The following terms are trademarks of other companies:

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

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get personalized notifications of new content
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

The rapid growth of big data and the storage of all that data is creating a critical problem for many organizations with IBM® z Systems™ environments. This situation occurs because the data that is stored is using all of the addressable device storage that is available.

This IBM Redpaper™ publication describes how extended addressable volume (EAV) for IBM 3390 Direct Access Storage Device (DASD) devices can solve the lack of addressable device storage space problem. The paper also describes the design points of EAV, the value of implementing EAV, and the use of EAV.

Authors

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

Keith Winnard is the IBM z/OS® Project Leader at the International Technical Support Organization (ITSO), Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks® publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional landscape of mainframe.

Alvaro Salla is a Senior IT Consultant for the ITSO. He has more than 40 years of teaching and developing educational material experience that includes the z/OS mainframe platform and consulting services that focus on performance. Alvaro co-authored Redbooks about IBM DS8000®, DFSMS, WLM, SYSPLEX, ABCs, IBM RMF™, and disaster recovery. He started with IBM in 1969.

Jose Gilberto Biondo Junior is an IT Specialist in Integrated Technology Delivery, Server Systems Operations/Storage Management in IBM Brazil. He has eight years of experience with z/OS, working with storage management since 2007. Jose works mostly with IBM storage products (DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm) and with OEM software products. Jose's areas of expertise include installing and maintaining storage products and process automation.

Barbara White McDonald started working for IBM in San Jose, CA, as a software developer for the Storage Division that is focused on DFSMS software development, specifically with access methods and device support for storage hardware. She refocused her attention to distributed storage and spent several years as a project manager bringing to market a new product for storage management. She returned to z/OS where she helped to define and enhance DFSMS strategic direction and roadmaps. Today, she works closely with various development teams to drive plans to deliver solutions that enable market success and growth for z Systems Storage software.

Thanks to LindaMay Patterson of the International Technical Support Organization, Rochester Center, for her contributions to this project.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks
- Send your comments in an email to:

  redbooks@us.ibm.com
- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction to extended address volumes

In this chapter, we describe possible solutions to the issue that is caused by the current growth rates of data. Although several approaches are available, your solution might be only one of these solutions or a combination of solutions that are described in this chapter. Ultimately, the solution you choose depends on your individual situation.

This chapter includes the following topics:

- 1.1, “What to do about too much data” on page 2
- 1.2, “About the EAV solution” on page 3
- 1.3, “EAV design” on page 4
- 1.4, “EAS data sets” on page 6
- 1.5, “Volume serial and VTOC” on page 9

1.1 What to do about too much data

The data that many organizations store and analyze far extends beyond the data that is commonly used in business processes. This extra data comes from many new sources, such as social media and sensing and mobile devices. All of this data in various formats combined becomes what is known as *big data*.

The rapid growth of big data and the storage of all that data is creating a critical problem for many organizations with IBM z™ Systems environments. This situation occurs because the data that is stored is using all of the addressable device storage that is available. The following specific technical issues are related to this addressable storage problem:

- Two-bytes (4 hex digit) device number (and subchannel number) limits is 65,280 devices per IBM z13™ logical channel subsystem (LCSS).
- Data compression when done with Central Processor Assist for Cryptographic Functions (CPACF) inbound coprocessor uses many Central Processing Unit (CPU) cycles.
- Less effective device storage occupancy (the amount of data that is stored versus the amount of data being used). Also consider that efficiency is lost if data must be migrated or recalled from tape to read.
- Each volume is limited by the IBM zArchitecture to up to 54 GB or up to 64 K cylinders.

1.1.1 Data problem solutions

The following methods can be used to handle the system limits that are caused by storing huge volumes of data:

- If you are using an IBM z13, you can institute up to six sets of subchannel numbers (and device numbers) per IBM z13 logical channel subsystem. Each set is known as a *subchannel set*. The first set has up to 65,280 entries and the other sets have 64 K devices each. Consider the following points:
  - The first set is used for real online devices.
  - The other five sets can be used for overflowing parallel access volume (PAV), alias addresses, and secondary offline peer-to-peer remote copy (PPRC) devices.
- Improve storage management subsystem (SMS) best practices policies for scratching temporary and deleted data sets, enforce partial release function, use tape devices more often for backups, logs, and Hierarchical Storage Management (HSM) ML2. Another alterative is to use space-efficient Flashcopy functionality.
- Use of shorter logical records for new applications; for example, define numeric files as decimal-packed instead of EBCDIC.
- Apply IBM z Systems™ Data Compression (zEDC) for z/OS coprocessor at the z13 PCIe I/O drawer, which saves many CPU data compression cycles.
- Implement extended addressable volume (EAV) for IBM 3390 Direct Access Storage Device (DASD) devices.

1.2 About the EAV solution

Over the years, IBM grew the number of available volumes for storing data by increasing the number of cylinders and gigabytes (GBs) capacity. However, the track that addressed architecture limited the growth to relatively small GB capacity volumes, with a maximum DASD storage per volume of 65,520 cylinders, which puts pressure on the four-digit device number limit. The rapid growth in data drove the z/OS platform to confront this limitation. Also, the demand for business resilience solutions for continuous availability caused z/OS to face this limitation.

An EAV is a volume with more than 65,520 cylinders. A volume of this size must be configured in the IBM DS8000 systems as an IBM 3390 model A. However, a 3390 Model A is not always an EAV. A 3390 Model A is any device that is configured in the DS8000 to have 1 - 268,434,453 cylinders. This number of cylinders is an architectural maximum.

EAV significantly increases the addressable disk storage and provides constraint relief for z/OS environments that are approaching the four-digit device number limit. EAV increases the amount of addressable DASD storage per volume beyond 65,520 cylinders by changing how tracks are addressed. This increase is accomplished with a new track address format within the Volume Table of Contents (VTOC), which is referred to as 28-bit cylinder track address and a new way to manage this extra space as multicylinder units. The following EAV increases are available for recent z/OS releases:

- For z/OS 1.12, the largest supported EAV volume is 262,668 cylinders
- For z/OS 1.13, the largest supported EAV volume is 1,182,006 cylinders

With EAV, IBM implements an architecture that provides capacities of hundreds of terabytes (TB) for a single volume. The current maximum is a 1 TB volume or 1,182,006 cylinders.

1.2.1 EAV value

EAV defines larger volumes by increasing the number of cylinders beyond 65,520, which enables the following features:

- Space management in multicylinder units; that is, cylinder-managed space known as *extended addressable space* (EAS). The multicylinder unit is 21 cylinders.
- A new track address format that consists of 28-bit cylinder track addresses (CCCCcccH)
- Compatibility with programs by not changing 3390 track or cylinder geometry, including 3390 extended count key data (IBM ECKD™) track format with up to 57 KB, and 15 tracks per cylinder.

The EAV enhancement provides the following values:

- Relieves storage constraints and helps simplify storage management by managing fewer, larger volumes as opposed to many small volumes.
- Enables the migration to fewer volumes, especially the three/four site disaster recovery configurations that use many device addresses.
- Increases z/OS addressable disk storage capacity and (with z/OS release 12 EAV function) supports extra data set types as EAS eligible.

1.3 EAV design

This section describes the various design points of the EAV.

1.3.1 Parallel access volume and HyperPAV

An EAV is managed by the system that allows it to be a general-purpose volume. This approach works especially well for applications with large files. PAV and HyperPAV technologies help prevent volume and device contention by allowing I/O rates to scale as a volume gets larger.

PAVs were created to improve I/O response times and are designed to allow multiple I/O operations to be processed for a single logical device. PAVs allow customers to reduce overall I/O queuing delays to devices that often are busy. This delay is because all I/O operations for a single device often are processed from a single queue. PAVs are managed by creating multiple addresses for a sole logical device.

HyperPAV technology takes PAVs a step further. HyperPAV was designed so customers can virtually eliminate the queuing of I/O operations. HyperPAV empowers the I/O supervisor component of z/OS to dynamically assign I/O resources instantaneously, on demand, as the workload requires based on customer-specified goals.

1.3.2 Space management

The 3390 track format by image size and tracks per cylinders remains the same as previous 3390 models. For example, an application that uses data sets on an EAV are comparable to how it runs today.

All volumes today feature *track-managed space*. The track-managed space is managed in track and cylinder increments. Track-managed space ends at cylinder address 65,519. Each data set occupies an integral multiple of tracks. The track-managed space allows programs, allocations, and physical migration products to continue to work.

The *cylinder-managed space* is managed only in multicylinder units. Cylinder-managed space begins at cylinder address 65,520. Each data set occupies an integral multiple of multicylinder units. Space requests that are targeted for the cylinder-managed space are rounded up to the next multicylinder unit. Currently, a multicylinder unit on an EAV is 21 cylinders and the number of the first cylinder in each multicylinder unit is a multiple of 21.

You might ask why the multicylinder unit has 21 cylinders? The 21 cylinders value for a multicylinder unit is derived from being the smallest unit that can map out the largest possible EAV and stay within the index architecture (with a block size of 8192 bytes). This value divides evenly into the 1 GB storage segments of a DS8000. These 1 GB segments are equivalent to 1113 cylinders.

1.3.3 Track addresses

The EAS refers to those cylinders of a volume that are beyond the first 65,520. An EAV has cylinders up to (but not including) cylinder 65,520 and are in the base addressing space of the EAV. Cylinders starting with cylinder 65,520 are in the EAS of the EAV.

Data sets that are allocated in cylinder-managed space are described with a new type of data set control blocks (DSCBs) in the VTOC. Tracks that are allocated in this space are addressed by using the new track address. Programs that are not changed do not recognize these new DSCBs and are protected from seeing how the tracks in cylinder-managed space are addressed.

Currently, the CCHH notation is used when referring to a track address (see Figure 1-1).

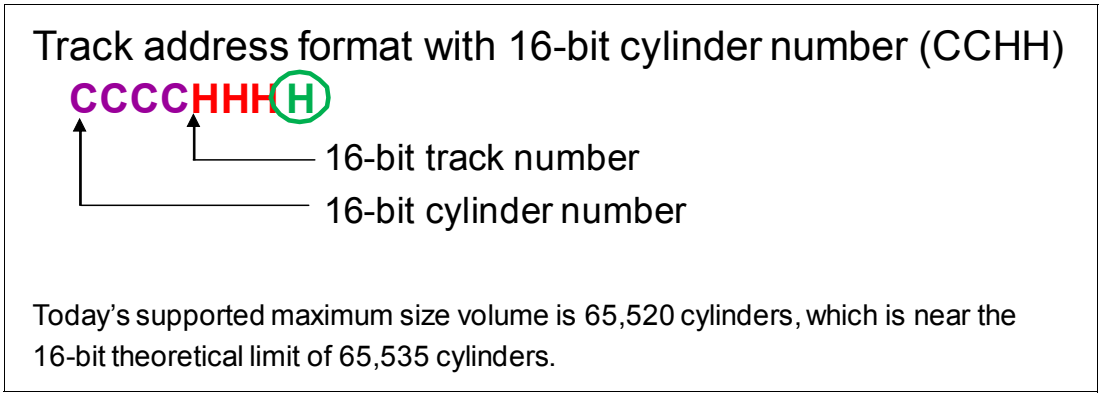

Figure 1-1 Traditional way of address tracking

The current track address is a 32-bit number that addresses each track within a volume. Each cylinder and track number uses a 16-bit number. However, only the low-order 4 bits are used for the track number, where the high-order 12 bits of the track number is not.

With the EAV, track addresses use hexadecimal nibbles that consist of the CCCCcccH notation (see Figure 1-2). For compatibility with older programs, the ccc portion is hexadecimal 000 for tracks in the base addressing space (first 65,520 cylinders). After that mark, the ccc is used to address the high-order 12 bits of 28-bit cylinder number.

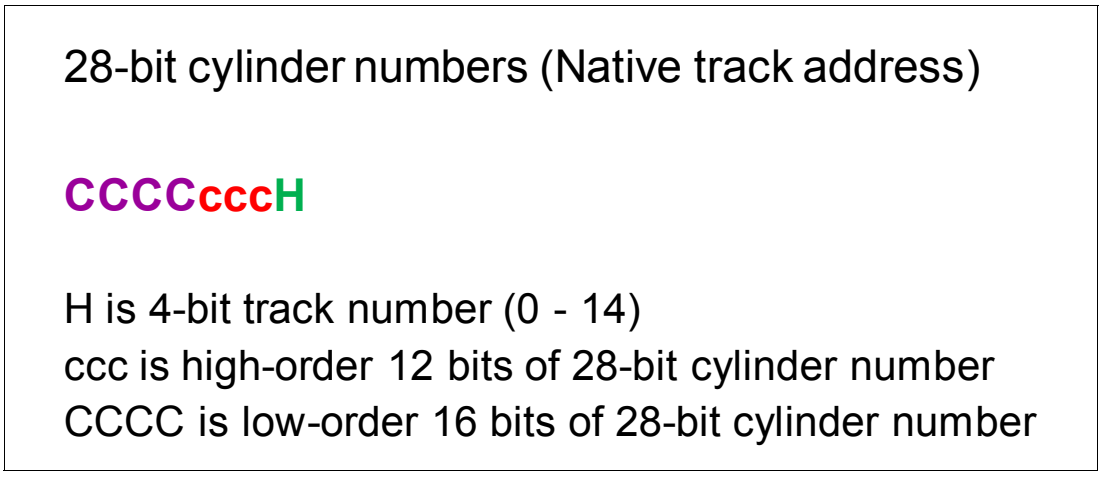

Figure 1-2 Native (28-bit) address tracking

To understand 28-bit cylinder numbers, a normalized cylinder-track address can be used for printing. The normalization process rearranges the bits to a more readable format and a linear 28-bit cylinder number (see Figure 1-3). The presence of a colon before the last hex nibble identifies the track address as being normalized.

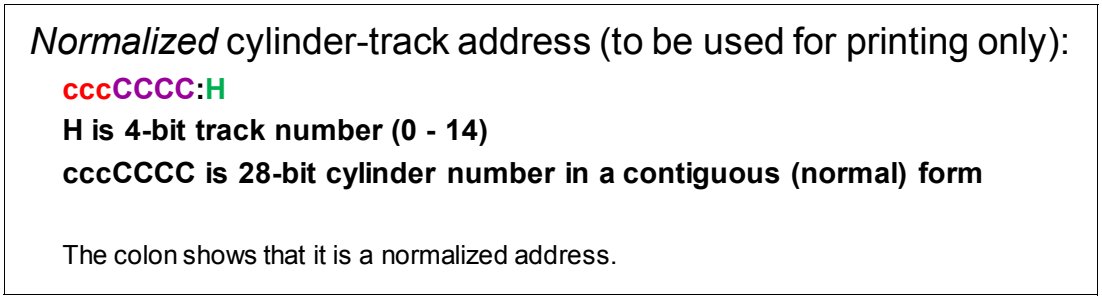

Figure 1-3 Normalized cylinder-track address

Note: Not all messages and reports normalize the output. For example, IDCAMS LISTCAT reports are in the native format.

Some reports were enhanced to display the larger number of cylinders, such as the LITVTOC function of the IEHLIST utility. The messages ICKDSF and DSS normalize.

1.4 EAS data sets

The EAS-eligible data set types on z/OS V2R2 can have extents in the extended addressing space and are described by extended attribute DSCBs (format-8/9). They can be in a track-managed or cylinder-managed space and be SMS-managed or non-SMS managed. This condition is sometimes referred to as *cylinder-managed space*.

The following data set types are supported:

- VSAM data types (key-sequenced data set (KSDS), entry-sequenced data set (ESDS), relative-record data sets (RRDS) and linear):
  - This support covers IBM DB2®, IBM IMS™, IBM CICS®, z/OS File System (zFS) and Network File System (NFS)
  - Control area (CA) sizes are 1, 3, 5, 7, 9 and 15 tracks
- Sequential (Extended Format)
- Sequential (Basic and Large Format)
- Direct Basic Direct Access Method (BDAM)
- Partitioned data set (PDS) and partitioned data set extended (PDSE)
- Catalog (VSAM Volume Data Set [VVDS] and basic catalog structure [BCS])

Some data sets might be on an EAV but cannot have extents (through create or extend) in the extended addressing space or include extended attribute DSCBs. The following data sets are not EAS eligible:

- VSAM data sets with incompatible CA sizes
- VSAM data sets with IMBED or KEYRANGE attributes
- VTOC (continues to be restricted to within the first 64 K-1 tracks)

- VTOC index
- Page data sets
- Hierarchical file system (HFS)
- Extended Remote Copy (XRC) Control, Master, or Cluster non-VSAM data sets:
  - State data set EAS eligible in z/OS V1R12
  - Journal data set EAS eligible in z/OS V1R11 and V1R12

DSCBs are VTOC entries that describe data set attributes and allocated extent information. These extent descriptors can contain 28-bit cylinder number for their track addresses. DSCBs also contain metadata in the format-1 and DSCBs that are the characteristics or attributes of the allocated data set. No more space is available in the format-1 DSCB to add attributes.

Before EAV, data sets were described with a format-1 DSCB, which optionally pointed to a format-3 DSCB. So that a data set can be in the EAS of an EAV, these types of data sets are allocated with format-8 DSCBs instead of format-1 DSCBs. The format-8 DSCB is almost identical to a format-1 DSCB except in the following respects:

- Has a control block ID of F8 instead of F1
- Points to a format-9 DSCB instead of a format-3 DSCB
- Cylinder numbers in the extent descriptions can include large cylinder numbers

The format-9 DSCB is a place for more attribute information and contains direct pointers to each possible format-3 DSCB. The logical DSCB chain is a format-1 and up to 10 possible format-3 DSCBs. The logical DSCB chain for an EAS eligible data set on an EAV can be a format-8, one or more format-9s, and up to 10 possible format-3 DSCBs.

With new services in the system (such as, OBTAIN and CVAFDIR macro), the entire logical DSCB chain for a data set can be read in one call. There are no more loops to read DSCBs until the chain pointer is zero.

EAS-eligible data sets on creation include format-8/9 DSCBs regardless of from where the actual extents where allocated. When extended, they can convert a format-1 DSCB for an EAS eligible data set to a format-8/9 DSCB pair.

In general, applications that have awareness of the DSCBs cannot access EAS eligible data sets without code changes. The `EADSCB=OK` keyword was added to the following services:

- OBTAIN (CAMLST macro)
- CVAFDIR
- CVAFFILT
- CVAFDSM
- CVAFSEQ
- OPEN (DCBE macro) – opening VTOC or VSAM data set with EXCP access.

Not specifying EADSCB=OK causes these services to fail if issued to a data set that supports extended attribute DSCBs or a volume that supports cylinder-managed space (CVAFDSM and OPEN).

For more information about coding applications that support EAVs and EAS eligible data sets, see *DFSMSdfp Advanced Services*:

`http://publibz.boulder.ibm.com/epubs/pdf/dgt3s301.pdf`

1.4.1 Allocating space on EAV

Allocating space on EAV can be accomplished by using the following methods:

- ESMS-managed or non SMS-managed:
  - Mix of EAV and non-EAV in storage groups supported
  - Specific and non-specific targets
  - With no changes, VSAM files that are allocated to an EAV are EAS eligible
- **USEEAV(YES|NO)** controls whether EAV is allowed for data set initial allocations and end of volume (EOV) extends to new volumes:
  - SMS parmlib (IGDSMSxx), change by using the **SETSMS** command
  - Can prevent EAV usage for all SMS and non-SMS requests
- BreakPointValue (BPV) directs the preferred placement of EAS-eligible data sets on EAV:
  - Set by SMS storage group, SMS parmlib, or **SETSMS** command
  - Value in cylinders (system default is 10 cylinders)
  - Cylinder-managed space is preferred if requested space >= BPV
  - Track-managed space is preferred if requested space < BPV
- Space is rounded up to the next multicylinder unit if the extent is allocated in cylinder-managed space. Individual extents must always start and end on multicylinder unit boundary in cylinder-managed space.
- Space that is released on multi-cylinder unit boundaries:
  - Partial release can release a portion or none of the unused space
  - VSAM stripes must have a common release point (RBA)
- An extent can straddle where cylinder-managed space begins. This approach is useful for data sets that can be a single extent only.
- Exact space is obtained if any extents are allocated in track-managed space.
- If the requested space is not available from the preferred managed space, the system can allocate the space from cylinder-managed and track-managed space.
- To help consolidate extents for SMS VSAM, the current last extent is enlarged if any of the newly acquired extents are contiguous to it.

Current algorithms in the search for space continue to apply. The system selects the first free space extent from the start of the preferred managed space that can satisfy the requested quantity. When the space cannot be satisfied in one extent, more extents can be used to satisfy the request where the used extents are ranked from the largest to smallest in the preferred managed space.

When the request cannot be satisfied from the preferred managed space, the available free space extents from the entire volume are ranked and used. Space might come from the non-preferred area or the track-managed and cylinder-managed spaces.

For VSAM extended format, the system reviews all newly acquired extents for an extent that is contiguous to the current last extent. Today's traditional code looks at only the first returned extent from the extend of space.

1.5 Volume serial and VTOC

Volume serial (VOLSER) is another identification of a 3390 device, beyond the device number. VOLSER is at block 2, track 0, and cylinder 0. After being defined as DS8000, the 3390 device is initialized by the IBM MVS™ utility ICKDSF that stores the VOLSER and creates the VTOC. Optionally, ICKDSF creates IPL bootstraps at record 0 and 1 of track 0 of cylinder 0 (for an IPL-able volume) and IPL text is loaded from record 3 on.

VTOC is unique and a mandatory 3390 data set that is defined at physical record 2, track 0, cylinder 0 (see Figure 1-4). The VTOC describes other data sets in the 3390 device and device free extents space. An extent is a contiguous set of free or occupied tracks. There is only one VTOC per 3390 volume. The VTOC is formatted in 140-byte blocks that are named DSCBs.

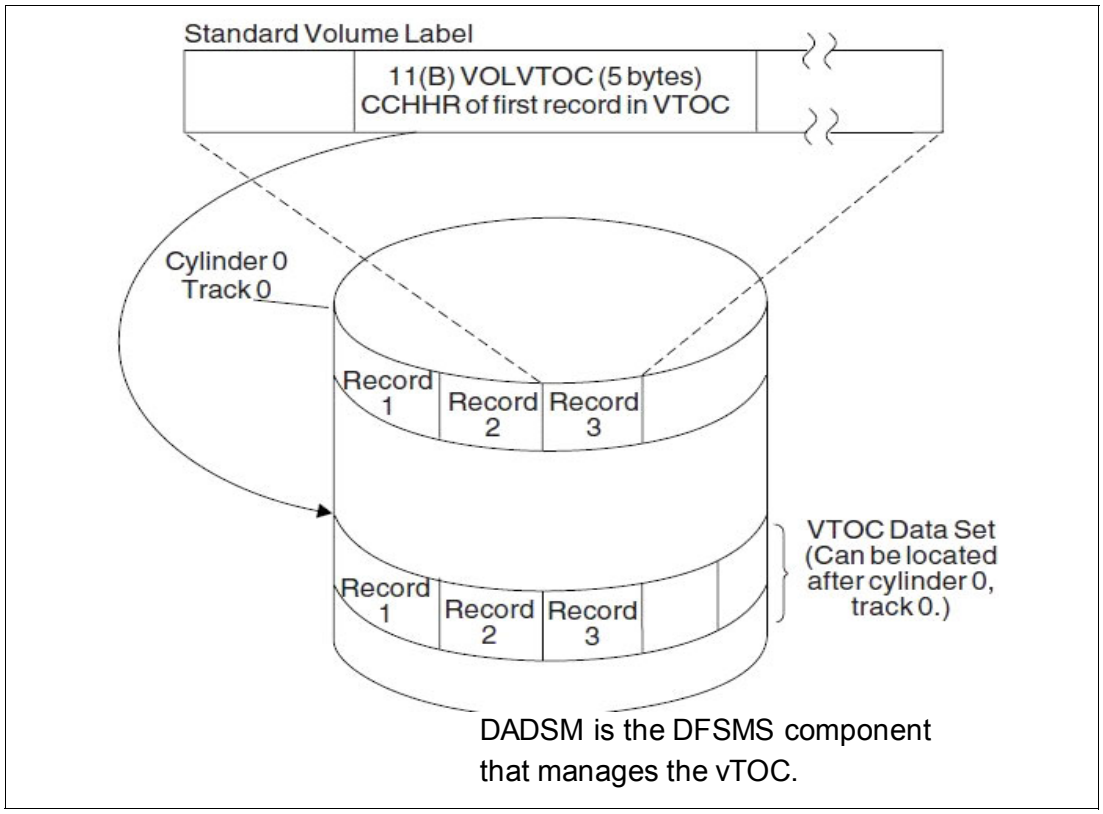

Figure 1-4 VTOC

For a faster DSCB search, the VTOC is associated with a B-tree (another data set) that is known as the Index VTOC. If the device contains VSAM data sets or is an SMS volume, the VTOC has an extension that is named VSAM Volume Data Set (VVDS).

Figure 1-5 shows the format and layout for the DSCB.

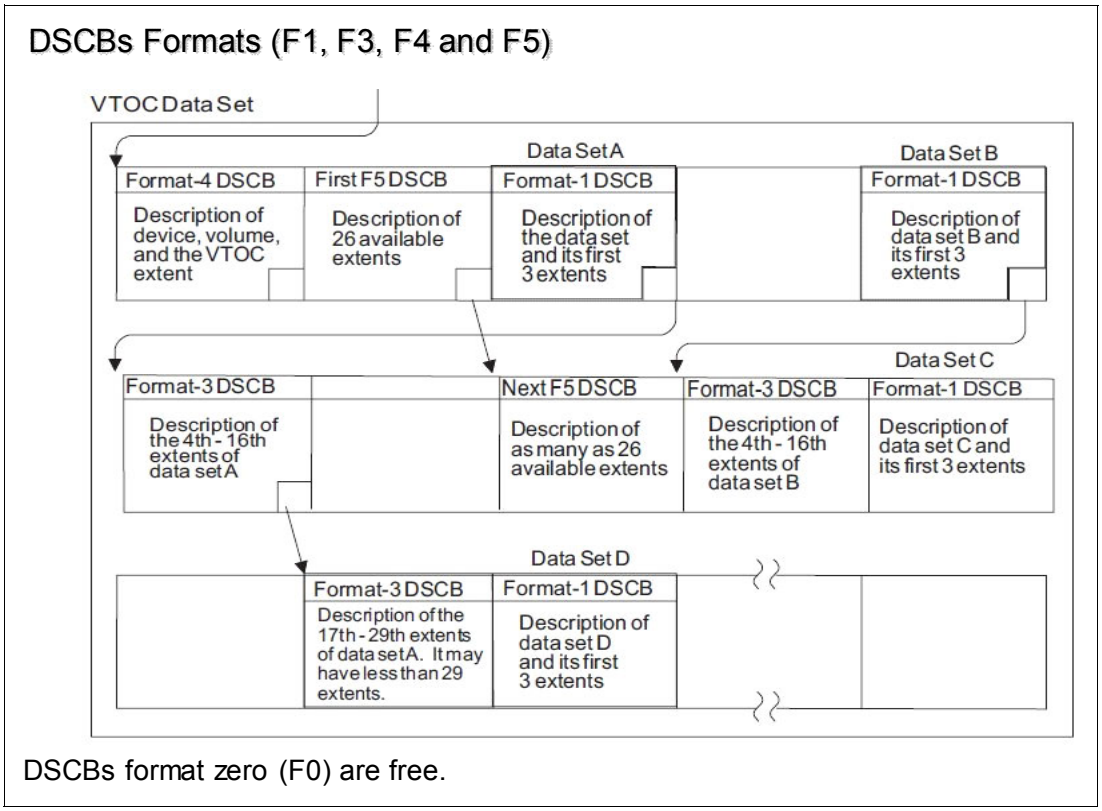

Figure 1-5 DSCB formats

2

Planning for extended address volumes

Introducing extended address volumes (EAVs) into your environment requires careful planning to optimize the effect of the EVAs. In this chapter, we describe many of the factors that must be considered as an EVA implementation is planned.

This chapter includes the following topics:

- 2.1, “Hardware and software requirements for EAV” on page 12
- 2.2, “Planning steps for EAV” on page 13
- 2.3, “Aiding migration with dynamic volume expansion” on page 19

2.1 Hardware and software requirements for EAV

To implement EAVs in your IBM z Systems environment, ensure that your environment meets the following requirements:

- IBM z/OS V1R10 or higher
- A licensed internal code (LIC) upgrade to an IBM DS8000 so an EAV can be configured in the storage subsystem

Coexistence features the following requirements:

- Systems before z/OS V1R10 cannot bring an EAV online. Verify that applications that use an EAV include the proper software levels installed.
- Previous releases of z/OS provide toleration of EAVs, which allows EAVs to be defined in the input/output definition file (IODF) on lower-level systems but are not allowed to come online. Therefore, an EAV volume can be defined in a configuration file that is shared among several LPARs, but they cannot be online on systems before z/OS V1R10 (which is possible by applying the PTF for APAR OA22449).
- Modify the following operator commands (for compliant systems see PTF for APAR OA21487):
  - The **DEVSERV PATHS** operator command is updated to report a new field for the number of cylinders.
  - The **DEVSERV QDASD** operator command is changed to recognize and display the larger device size for an EAV. This larger size is reflected in the CYL column of the DEVSERV QDASD display.
- LSPACE macro processing on pre-V1R10 systems is changed to accept the extended parameter list and not fail it with return code 12. LSPACE macro processing is also changed to treat EXPDATA= or XEXPMSG= requests as DATA= or EXPMSG=, respectively. This change allows LSPACE programs that are assembled by using the new z/OS V1R10 keywords to run on pre-R10 systems.
- DFSMSdss allows restoring a data set by using dump from an EAV on V1R10 to a non-EAV on V1R9 or lower. If you attempt to restore a data set by using dump from an EAV on R10 to a non-EAV on R9 without this support (or lower), you can encounter unpredictable results, including the following examples:
  - DFSMSdss full volume and tracks dumps of EAVs are not compatible with dumps of volumes that are 64 K cylinders or fewer because of changes that are required to format the extended-address space in the dump.
  - DFSMSdss provides limited restore capability on supported lower levels of z/OS for data that is dumped from an EAV on z/OS V1R10 or higher levels.
  - The following restore functions and limitations are supported via a coexistence PTF on pre-z/OS V1R10 systems:
    - Versions of DFSMSdss before V1R10 do not perform a full volume restore of a full volume dump from an EAV. In addition, tracks that are restored where track 0 is included fail and tracks restore, not including track 0, restore only the track-managed space from an EAV. Logical data set restore and physical data set restore convert format-8 and -9 DSCBs to format-1 DSCBs.
    - DFSMSdss Stand-Alone Restore do not perform a full volume or tracks restore of an EAV to a non-EAV.
- DFSMShsm adds support for recall or recover on z/OS R9 or lower of an EAS data set with a format-8 DSCB that was migrated or backed up on z/OS V1R10.

2.2 Planning steps for EAV

For more information about determining which applications might need to be modified to support EAVs, see “Appendix C. Using the extended address volume (EAV) migration assistance tracker” in *z/OS DFSMSdfp Advanced Services*:

http://publibz.boulder.ibm.com/epubs/pdf/dgt3s301.pdf

The process to implement EAV includes the following high-level planning steps:

1. Specify the EATTR=OPT|NO parameter for data sets (as appropriate) to override system default for EAS eligibility.
2. Use the Generic Tracker Facility.
3. Review any migration considerations.
4. Configure and enable EAV volumes and add them to your storage group or pools.
5. Apply BreakPointValue.
6. Review the implementation tips.
7. Provide a test environment to validate readiness.
8. Migrate data (optional).

2.2.1 Using the EATTR value

The EATTR value was added to allow a user to control whether a data set can have the extended attribute data set control blocks (DSCBs) and whether it can be allocated in EAS. EATTR is specifiable for all data set types. The EATTR value is encoded and written in the format-1 or -8 DSCBs for all data set types and in the VSAM Volume Data Set (VVDS) for VSAM clusters. The EATTR value that is recorded for a data set type that is not supported as being EAS-eligible has no effect until a future time when the system might begin supporting that data set type for EAS.

EATTR is used to override system determined EAS-eligibility. The following parameter values are used:

- NO: No extended attributes

  When this parameter is used, the data set cannot have extended attributes (format-8 and -9 DSCBs) or optionally be in EAS and have the default behavior for non-VSAM data sets.
- OPT: Extended attributes are optional

  A data set can have extended attributes and can optionally be in EAS. This value is the default behavior for VSAM data sets.

2.2.2 Using the Generic Tracker Facility

The use of the Generic Tracker Facility includes the following objectives:

- Identify executions of select systems services by job and program name where the starting programs might require analysis for changes to use new services that are provided by the system service. These program calls and the reported output are not considered in error as valid information is returned. These outputs are considered as informational instances.
- Identify the possible improper use of returned information, such as parsing 28-bit cylinder numbers in output as 16-bit cylinder numbers. These instances are considered warnings.

- Identify instances that failed or identified with an informational message if they are run on an extended address volume. These instances are considered as error instances. These errors apply to the following functions when the target volume of the operation is a EAV and the function that was started did not specify the EADSCB=OK keyword:
  - DADSM OBTAIN
  - CVAFDIR, CVAFSEQ, CVAFFILT
  - EXCP OPEN of an EAS-eligible data set

  The other instances are volume-oriented and are not affected by other data set types becoming eligible for EAS, including the following functions:
  - CVAFVSM, CVAFDSM
  - LISTCAT, DCOLLECT
  - IEHLIST
  - DEVTYPE
  - LSPACE
  - LISTDATA PINNED
  - EXCP OPEN VTOC

Note: Generic Tracker Facility (GTZ) is an optional z/OS started task. For z/OS environments before z/OS V2R1, use the Console Tracking Facility.

Generic Tracker Facility (GTZ) identifies programs that are making non-compliant system service invocations by job and program name (+offset). There is no need to install or make available any physical EAVs when it is run. For example, it reports instances for data sets (VSAM or non-VSAM) where EADSCB=OK was not specified on system services. z/OS otherwise fails this action on an EAV.

The instances can be errors, warnings, or informational instances of certain functions that are running in the system as follows. Consider the following points:

- Errors are instances that are failed if they are run on an EAV.
- The warning and informational instances do not fail the function. All instances are collected in the tracking facility without the use of an EAV.

Instances are recorded by items, such as, job, program, program offset, and select address space identifier (ASID) service. Instances are recorded include the following examples:

- Interfaces that access the VTOC and should be upgraded to have EADSCB=OK, such as OBTAIN, CVAFDIR, CVAFDSM, CVAFVSM, CVAFSEQ, CVAFFILT, OPEN to VTOC, and OPEN EXCP.
- Possible improper use of returned information, such as parsing 28-bit cylinder numbers in output as 16-bit cylinder numbers (Warning messages) including IEHLIST LISTVTOC, IDCAMS LISTCAT, and IDCAMS LISTDATA PINNE.D.
- Programs that might want to use new services (Informational messages), such as LSPACE, DEVTYPE, and IDCAMS DCOLLECT.

Use Generic Tracker externals to collect and report such instances in the following ways:

- S GTZ, which is the started task (STC) uses member GTZPRMxx of PARMLIB
- DISPLAY GTZ, TRACKDATA | EXCLUDE | STATUS | DEBUG (unknown instances)
- SETGTZ TRACKING | DEBUG | CLEAR | EXCLUDE
- GTZPRINT is the batch reporting utility; GTZQUERY is a batch API
- SET GTZ=xx to change the GTZPRMxx parmlib configuration settings

You can start the Generic Tracker by using the **S GTZ=xx** command, where xx is the GTZPRMxx PARMLIB member suffix. The GTZPRMxx member specifies the **SETGTZ TRACKING=ON** command, which is issued from the console or use system automation.

You can exclude specific instances by using the **SETGTZ EXCLUDE** command. The EXCLUDE statements are in GTZPRMxx in PARMLIB.

To display the GTZ heart beat and tracking data, use any of the following commands:

- **D GTZ,STATUS**, which is the heart beat of the GTZ started task and instance count
- **D GTZ,TRACKDATA**, to avoid possible console buffer exhaustion if recorded instance data is voluminous; run the **D GTZ,STATUS** command first
- **GTZPRINT**, which is a batch job that writes instance data to a disk or spool

Generic tracker examples

Example 2-1 shows excluding instances.

Example 2-1 Excluding instances

```
SETGTZ EXCLUDE command or EXCLUDE statements in GTZPRMxx PARMLIB
/* ISPF EXPMSG= */
 EXCLUDE(
EVENTDESC='SMS-I:3 LSPACE EXPMSG=*'
  HOMEJOB=*
  PROGRAMTYPE=NOPATH PROGRAM=ISRUDA
 )
```

To use the wildcard capability, run the **EVENTDESC='SMS-I*'** command, which excludes all DFSMS informational instances for EAV.

Example 2-2 shows displaying tracking data.

Example 2-2 Tracking data

```
D GTZ,TRACKDATA
------------------------------------------------------------------------
 INSTANCE:      2                   COUNT:         2
 EVENTDESC:     'SMS-I:3 LSPACE EXPMSG='
 OWNER:         IBMCNZ              SOURCE:        CNZTRKR
 EVENTDATA:     x0000000000000000   x0000000000000803
 PROGRAM:       ISRUDA              PROGRAMOFFSET: x000000000003B69C
 HOMEJOB:       IBMUSER             HOMEASID:      x0034
 EVENTJOB:      IBMUSER             EVENTASID:     x0034
```

For more information, see the following resources:

- z/OS Downloads website for the most current copy of the GTZPRMxx member:

  http://www.ibm.com/systems/z/os/zos/downloads/
- For DFSMS-related tracked events z/OS V2R2.1, see *DFSMSdfp Advanced Services*, SC23-6861, which is available at this website:

  https://ibm.biz/Bdrcux
- For the generic tracker facility, see IBM Knowledge center at this website:

  https://ibm.biz/BdrcuR

- *z/OS MVS Diagnosis: Tools and Service Aids*, GA32-0905, which is available at this website:

  http://www-01.ibm.com/support/docview.wss?uid=pub1ga32090500
- *z/OS MVS Initialization and Tuning Reference*, SA23-1380, which is available at this website:

  http://publibz.boulder.ibm.com/epubs/pdf/iea3e204.pdf
- GTZCNIDT conversion tool that is used to convert pre-V2R1 version CNIDTRxx parmlib records to the newer z/OS 2.1 GTZPRMxx parmlib format:

  https://ibm.biz/BdrcuX
- Generic Tracker instances can be converted into z/OS Health Checks with the sample GTZSHCK program

  https://ibm.biz/BdrcuH

2.2.3 Migration considerations

Consider the following suggested tasks for migration:

- Look for OBTAIN, REALLOC, CVAFDIR, CVAFSEQ, CVAFDSM, and CVAFFILT macro calls.
- Look for programs that calculate volume or data set size by any means, including reading a volume table of contents (VTOC) or VTOC index directly with a basic sequential access method (BSAM) or EXCP DCB. The system cannot distinguish between programs that read DSCBs for space information and those that read them for metadata. Both kinds of program require a new option.
- Look for EXCP, XDAP, and STARTIO macros for DASD channel programs and other programs that examine DASD channel programs or track addresses.
- Look for programs that examine any of the many operator messages that contain a DASD track or block address or data set or volume size. Such track or block addresses often are represented in the documentation as cchh, cchhr, mbbcchhr, or track-address.
- Look for programs that access the VTOC index (`SYS1.VTOCIX.volser`) and VVDS data sets (`SYS1.VVDS.Vvolser`).
- You should upgrade system service calls that read DSCBs, read free space, or attempt to open the VTOC or a VSAM data set with EXCP access, such as OBTAIN, CVAFDIR, CVAFDSM, CVAFVSM, CVAFSEQ, CVAFFILT, OPEN to VTOC, and OPEN EXCP. Add EADSCB=OK to indicate support for format-8 DSCBs and 28-bit cylinder numbers that might be present in extent descriptors and free space. Optionally, upgrade system service calls to use new information that describes an EAV, such as LSPACE, DEVTYPE, and IDCAMS DCOLLECT.
- Upgrade programs that parse formatted output that changed with EAV, including IEHLIST LISTVTOC, IDCAMS LISTCAT, and IDCAMS LISTDATA PINNED.

- Use SMF records to identify programs that might need to be changed and upgrade programs that use changed SMF records or access the VVDS to support 28-bit cylinder. Also, upgrade programs that process the VTOC index to support the larger sized space map records.

2.2.4 Configuring and enabling EAV volumes

An EAV can be a new volume configured in the DS8000 or an existing volume that can be expanded in place without disruption. When a volume is configured with more than 65,520 cylinders, the total size of the volume is restricted to an integral multiple of 1113 cylinders. This space is equivalent to a 1 GB segment in the DS8000. With this rule, space in the backing storage is fully used.

Example 2-3 shows creating CKD volumes by using the **mkckdvol** command.

Example 2-3 Creating CKD volumes by using the mkckdvol command

```
dscli> mkckdvol -extpool p2 -cap 262668 0080
Date/Time: 08 April 2007 17:04:31 IBM DSCLI Version: 5.1.0.204 DS:
IBM.2107-7503461
CMUC00021I mkckdvol: CKD Volume 0000 successfully created.
```

You must perform the following actions when migrating:

- Configure the storage subsystem that is select 3390 Model A:
  - Do not use a new device type
  - Allow for 1 to more than 65,520 cylinders
  - Apply in increments of 1,113 cylinders
- Enable the use of EAVs in your software environment by using the IGDSMSxx PARMLIB member:
  - USEEAV(YES|NO) with YES for new allocations on EAVs
  - The BreakPointValue (BPV) is 0 - 65520 cylinders

2.2.5 Applying BPV

The BPV is applicable only to data sets that are EAS-eligible and directs to the preferred location of EAS-eligible data sets on a EAV. Data sets that are not EAS-eligible must be in the track-managed region of a volume.

BPV is an optional parameter. If not specified at the storage group level, the system uses the value that is specified in the IGDSMSxx member of parmlib or a default value of 10 cylinders if no BPV is specified in the IGDSMSxx member.

The BPV value includes the following definitions:

- IGDSMSxx (system level) can be modified by using the **SETSMS BreakPointValue (0 - 65520)** command
- Storage Group (SG)

  SG BPV overrides system-level BPV and IGDSMSxx BPV value overrides system default (10 cylinders).

BPV applies to non-SMS and SMS-managed volumes. Consider the following points:

- Cylinder-managed space is preferred if requested space >= BPV.
- Track-managed space is preferred if requested space < BPV.
- If not enough track-managed space is available, the system uses cylinder-managed space or both track-managed space and cylinder-managed space.

A value of 0 means that multi-cylinder unit (MCU) space is always preferred. A value of 65520 means that track managed space is always preferred.

2.2.6 Implementation tips

For more information about implementation tips, see 3.2, “Implementation considerations” on page 23.

2.2.7 Providing test environment to validate readiness

After you complete all of the steps that are necessary to implement and use EAV volumes on your environment, run some tests before introducing EAV into production systems. To create a basic working environment, you can set up some storage groups with EAV volumes and define a BPV for each of group. Update the Automatic Class Selection (ACS) routines to redirect your test data sets to the appropriate pools.

A sample SG definition that is used for testing EAV in a specific system is shown in Figure 2-1. In this scenario, the BPV that is defined is 21 cylinders.

```
                        POOL STORAGE GROUP DEFINE          Page 2 of 2
Command ===> _____________________________________________________________

SCDS Name . . . . . : SYS1.STPPLEX.SCDS
Storage Group Name  : SGEAVT1

To DEFINE Storage Group, Specify:

 Allocation/migration Threshold :     High    85   (1-100)  Low . . 60   (0-99)
 Alloc/Migr Threshold Track-Managed:  High    85   (1-100)  Low . . 60   (0-99)
 Total Space Alert Threshold %  . . . . . 90       (0-99)
 Track-Managed Space Alert Threshold %  . 90       (0-99)
 Guaranteed Backup Frequency  . . . . . . 1_____  (1 to 9999 or NOLIMIT)
 BreakPointValue  . . . . . . . . . . . . 21___   (0-65520 or blank)
 Processing Priority  . . . . . . . . . . 50_     (1-100)
```

Figure 2-1 Sample Storage Group Definition

The ACS code was also updated to direct test data to SGEAVT1 storage group, as shown in Example 2-4.

Example 2-4 Updated ACS code to direct test data to SGEAVT1

```
SELECT
   WHEN (&DSN EQ TESTHLQ.EAVTEST.**)
      DO
         SET &STORGRP = 'SGEAVT1'
         EXIT
      END
  WHEN ....
```

During the tests, you can allocate data sets with different attributes, such as space (below and above BPV), recording format (Fixed, Variable, and Fixed Blocked), and types (such as VSAM, Sequential, and Partitioned), to check whether the BPV and allocation thresholds are working as expected.

Depending on your system settings, you might want to update ARCMDEXT exit code to convert ML1 migration to ML2 to prevent large data sets from using all DFSMShsm ML1 volumes. For more information about ARCMDEXT exit, see the IBM Knowledge Center, section z/OS DFSMS Installation Exits:

https://ibm.biz/BdrB4i

2.2.8 Optionally migrate data

With EAV volumes deployed to your systems, you can plan to move your large data sets to an EAV volume. If you are running z/OS V2R2 or later releases, you can use the **DFSMShsm MOVE** command to move selected data sets to EAV volumes. It is possible to move data sets by DSN, Volume, and Storage Group. By migrating data to EAV, you can reduce the number of extents and multivolume data sets. As a result, the empty space left by this migration can be used for allocations that cannot use cylinder-managed space.

2.3 Aiding migration with dynamic volume expansion

Dynamic volume expansion (DVE) reduces the complexity of migrating to larger volumes. Previously, IBM clients used migration utilities that required more volume for each volume that was being expanded and required data to be moved. DVE expands a volume beyond the 62,520 cylinders without moving data or application outage. However, the volume should be defined as 3390 model A at DS8000. Active Copy Service (such as FlashCopy®) relationships must be removed. The volume grows in chunks of an integral multiple of 1113 cylinders. This increase is equivalent to a 1 GB segment. It is important to realize that the capacity in the same extent pool must be available.

A volume can be grown dynamically by using one of the following methods:

- Command-line interface (DSCLI)
- Web browser GUI

3

Implementation tips

Other factors can effect your extended address volume (EAV) implementation's effectiveness. The operational considerations that are suggested in this chapter are designed to help increase the success of your implementation.

This chapter includes the following topics:

- 3.1, "Introduction to DFSMShsm" on page 22
- 3.2, "Implementation considerations" on page 23
- 3.3, "Implementing EAV" on page 26

3.1 Introduction to DFSMShsm

DFSMShsm is a functional component of the Data Facility Storage Management Subsystem (DFSMS) family, which provides facilities for managing your storage devices. DFSMShsm ensures that space is available on your direct access storage device (DASD) volumes, so you can extend existing data sets and allocate new ones.

DFSMShsm also ensures that backup copies of your data sets are always available if your working copies are lost or corrupted. It relieves you from manual storage management tasks and improves DASD use by automatically managing space and data availability in a storage hierarchy.

DFSMShsm is one of five components that, when combined, create a single, integrated software package for all of your installation's storage management needs. DFSMS consists of the following five components:

- DFSMSdfp: Provides storage, data, program, and device management functions through the storage management subsystem (SMS).
- DFSMSrmm: Provides tape management functions for removable media, such as virtual and physical tape cartridges.
- DFSMSdss: Provides data movement, copy, backup, and space management functions in a batch environment.
- DFSMSoam: Provides tape hardware management, such as cartridge entry, eject, and tape configuration database (TCDB) management.
- DFSMShsm: Provides backup, recovery, migration, and space management functions with optimum automation capabilities.

DFSMShsm is a policy-driven solution to storage management, which removes the requirement for batch jobs to perform backup, migration, or space retrieval functions. DFSMShsm works by rules that you can apply to manage your storage. These rules are also dynamically adjustable to allow the flexibility that is required in today's constantly changing environments.

With its flexibility, you can manage your storage at the data set level, device level, or even device pool level. DFSMShsm provides the means to manage every data set from the time of its creation until the time that its last backup is no longer required.

For more information about DFSMShsm, see *IBM z/OS DFSMShsm Primer*, SG24-5272, which is available at this website:

http://www.redbooks.ibm.com/abstracts/sg245272.html?Open

3.2 Implementation considerations

In this section, we describe implementation considerations.

3.2.1 EAVs and DFSMShsm

DASD space management can be performed by using the volume threshold and track-management threshold. Track-management threshold is set by using the following methods:

- SMS (specifically, Storage Group)
- Non-SMS (specifically, the **ADDVOL** command)

The track-managed threshold was added to ensure that track-managed space is managed even when overall volume does not exceed the threshold value (see Table 3-1).

Table 3-1 Threshold value comparison

Track-managed threshold exceeded	Volume threshold exceeded	Data set selection
Yes	Yes	All data sets
No	Yes	All data sets
Yes	No	Only data sets with one or more of the first three extents in track-managed space
No	No	None

3.2.2 Migration subtasking in z/OS V2R1

By processing data sets in the following migration subtasks for a level 0 volume migration task, the aggregate throughput of all the migration tasks is improved:

- `MIGRATIONSUBTASKS(YES | NO)` parameter on the **SETSYS** command

  Allows DFSMShsm to run multiple subtasks concurrently under each migration task for primary space management, on-demand migration, and interval migration.

- `ADDITIONALSUBTASKS(nn)` subparameter

  Allows you to dynamically change the number of extra subtasks that the system can use running under each migration task.

These subtasks add to the number of subtasks that the system uses when the `MIGRATIONSUBTASKS` parameter is specified.

3.2.3 DFSMS storage tiers enhancements

Today, moving data to newly defined disk volumes within a storage environment can be manual intensive and cumbersome. In z/OS V2.2, the DFSMS Class Transition function moves data laterally to the same tier of storage, in addition to moving data to different storage tiers. Also, in z/OS V2.2, the `MOVE` keyword was added to the **MIGRATE DSNAME**, **VOLUME**, and **STORAGEGROUP** commands. The `MOVE` keyword added the following capabilities:

- Every data set is processed, regardless of management class policy or threshold and Advanced Copy Services (ACS) routines are started to determine the new storage class and storage group.

- You can move DB2 data from smaller volumes to the larger, newly defined EAVs.
- Simplifies the task of migrating data to newly defined disk volumes.

3.2.4 EAV and Catalog/IDCAMS

Basic catalog structure (BCS) data sets are EAS-eligible. The DEFINE UCAT statement or the MODEL – ICFCATALOG and VOLCATALOG statements have the following values for EATTR:

- NO: Cannot include the extended attribute DSCBs or optionally be in EAS.
- OPT: Can include the extended attribute DSCBs and optionally be in EAS with the following considerations:
  - Both recorded in the VVDS for the catalog objects and in the DSCBs that get created in the VTOC
  - Catalog object is restricted to track-managed space when EATTR is not specified. The action that is taken by the system today for catalog defines.
  - No EATTR option in pre z/OS V1R12 systems.
  - EXPORT/IMPORT preserves EATTR value across systems.

VSAM volume data sets (VVDS) data sets are EAS-eligible. Use the DEFINE CLUSTER for VVDS object. EATTR features the following settings:

- NO: Cannot include the extended attribute DSCBs or optionally be in EAS.
- OPT: Can include the extended attribute DSCBs and optionally be in EAS with the following considerations:
  - Both recorded in the DSCBs that get created in the VTOC.
  - VVDS objects are restricted to track-managed space when EATTR is not specified. This action is taken when VVDS is allocated by the system.
  - No EATTR option in pre-z/OS V1R12 systems.
  - Data class or model does not apply to a DEFINE of a VVDS.
  - Only way to perform this action is with the EATTR keyword.

3.2.5 EAV and DFSORT

DFSORT adds the ability to quickly and easily sort, merge, copy, report, and analyze your business information. It also adds versatile data handling at the record, fixed position, and length or variable position and length field, and bit level. It also can perform the following functions:

- Sort, merge, or copy data sets while including, excluding, or reformatting records.
- Collate and compare data according to your national and cultural needs by using the National Language Support of DFSORT.
- Add timestamps to records, perform date field conversions and arithmetic, include or omit records based on today's date, a past date or a future date, and convert SMF, TOD, and ETOD date and time values to readable form.
- Join and match records from different data sets in various ways.

DFSORT data sets that can be allocated in EAS are available in DFSORT R12 supporting these data sets:

- SORTIN, SORTOUT, and OUTFIL data sets with a maximum size of large format sequential supported.
- SORTWK data sets with the following limitations:
  - Basic sequential format with a limitation of 65,534 tracks
  - Sequential data sets up to 16 million blocks (tracks)
  - Larger data sets can be allocated, but excess is not used
  - Less in a resource-constrained environment

3.2.6 EAV and Data Set Services

Data Set Services (DFSMSdss) is a component of DFSMS and is used to perform the following tasks:

- Move and replicate data

  DFSMSdss offers powerful, user-friendly functions with which you can move or copy data between volumes of like and unlike device types. It can also copy backed up data.

- Manage storage space efficiently

  DFSMSdss can increase performance by reducing or eliminating DASD free-space fragmentation.

- Back up and recover data

  DFSMSdss provides you with host system backup and recovery functions at the data set and volume levels with encryption capabilities. It also includes an enhanced Stand-alone Restore Program with which you can restore vital system packs during disaster recovery without a host operating system.

- Convert data sets and volumes

  DFSMSdss can convert your data sets and volumes to system-managed storage, or return your data to a non-system-managed state as part of a recovery procedure without data movement.

The **`DFSMSdss Consolidate`** command performs data set extent consolidation and space reduction, which reduces the number of extents by attempting to relocate multiple non-contiguous data set extents into contiguous space within each managed space.

The **`Consolidate`** command supports the same type of data set filtering as COPY, DUMP, and RESTORE functions. You can specify which data sets are to be included.

Note: There is no need to DEFRAG a volume to consolidate extents.

3.2.7 Other considerations

Consider the following points:

- Although EAV volumes tend to have more data sets allocated to it because of its larger size, improvements in DFSMS and VTOC reads might reduce volume contention and the use of HyperPav can reduce or eliminate IOS Queuing time.

- Because there is extra space for allocation on the volume, data sets are likely to reduce the number of volumes to which they expand. This issue can reduce the parallelism in certain circumstances.
- In addition, the use of EAV volumes includes the following advantages:
  - Fewer unit control blocks (UCBs)
  - Fewer volume statistics to manage
  - Less multi-volume data sets
  - Reduces fragmentation
  - Could reduce extends as over-allocated space becomes usable

3.3 Implementing EAV

You should run the Generic Tracker Facility for EAV on target workloads. After running the tool, start capturing and analyzing error instances. Next, follow up on the informational and warning instances as applicable.

Use the TRKADDR macro for comparison and manipulation of track addresses. Review System Modification Program Extended (SMP/E) fix category, checking `IBM.Function.EAV` for support levels. For more information, see the software developers support table for EAV support at website:

`http://www-03.ibm.com/systems/z/os/zos/software/isv113.html`

Note: Before z/OS V2.2, this URL link took a user to a page where they selected “Vendor software products for z/OS” for z/OS R10 and above. Currently, this link is valid for V2.2 and displays only business partners that support z/OS V2.2.

To expand a volume on DS8000 with dynamic volume expansion (DVE) to become an EAV volume, use DS8000 Storage Manager or the DSCLI. To use DVE, you must perform the following tasks:

- Convert the volume to 3390 model A
- Identify available space in the same extent pool
- Expand to any size
- Understand that copy services relationships (for example FlashCopy) must first be withdrawn.

A volume remains online and accessible to applications without moving data, which reduces the complexity of moving data to larger volumes. There is no need for the target volume to be available.

Use ICKDSF REFORMAT REFVTOC, which updates the VTOC and VTOC index to access new cylinders since it is nondisruptive.

Also, consider the use of Dynamic VTOC Refresh, which is enabled by using F DEVMAN,ENABLE(REFVTOC) and can be queried with F DEVMAN,REPORT.

With DVE, the system detects the change in size and automatically starts the **`reformat`** command. The VTOC or index is rebuilt and refreshed to use the newly acquired cylinders. All systems in the sysplex are updated with the new size that is based on an ENF signal. DFSMShsm listens for the signal to update its structures and Common VTOC Access Facility (CVAF) calls SMS directly to update its structures.

Determine the BPV, which directs the preferred placement of EAS-eligible data sets on EAV. This placement is set by storage group or **SMS PARMLIB SETSMS** command to define system default (system default is 10 cylinders) with the following values:

- 0 says any space amount, prefers EAS
- 65520 says no space amount, prefers EAS

Set the value large enough to minimize your average over allocation percentage. A larger value drives the per cent over allocation smaller. Set the value small enough to have a large percentage of allocations prefer EAS. A smaller value drives per cent of over allocations larger. The specific application should drive the BPV values of 21 and 100 are used often.

To ensure that a VTOC has a place to expand into, you can perform one of the following actions:

- Allocate dummy data sets that are next to the VTOC before allocating the index data set.
- Allocate an index data set much larger than normal that is next to the VTOC.
- Extend your VTOC by running an **ICKDSF REFORMAT NEWVTOC** command. ICKDSF allocates your new VTOC in another extent within the volume.
- Create a VTOC that is much larger than normal, which allows it to have the capacity in the future to support a larger number of data sets.

You should standardize volume initialization (ICKDSF) values by performing the following actions:

- Initialize the volume with a VTOC index.
- Define larger VTOCs to allow for growth.
- Allocate dummy data set next to VTOC to reserve space as an expansion area. Allow for future nondisruptive ICKDSF EXPAND VTOC.
- Do not specify index size parameters as ICKDSF determines the needed size based on the VTOC size.

You should consider the use of HyperPAV (Parallel Access Volumes - Aliases) because as a volume size grows, more data is addressable through the device's subchannel, which in turn can cause contention. Parallel Access Volumes (PAVs) is often the approach that is used to mitigate this contention. As the volume of data on a device grows, the demand for more alias devices grows as well.

The HyperPAV architecture is included the concept of Alias Management Groups (AMG). Within an AMG, an alias device can be used for input/output (I/O) to any base PAV device in the AMG. The initial HyperPAV implementation defined an AMG for each logical storage subsystem (LSS) to monitor the current state use RMF HyperPAV Information.

When migrating non-VSAM data sets to EAV and having the data sets eligible to be allocated in the EAS, you must override EATTR=NO or its default. Use dynamic patch available in DSS, which overrides source EATTR to OPT for non-VSAM data sets during logical COPY (OA42848).

Figure 3-1 shows an EATTR override.

```
//STEPT006 EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SET PATCH 5B=FF                /* enable source eattr override    */
  COPY                           /* move data set to EAV            */ -
    DS(INCL(PATCHX5B.MIG.EAV)) /* data set to be moved            */ -
    FR(PREF)                     /* prefer fast replication         */ -
    FCTOPPRCPRIMARY(PMR)         /* ensure mirror stays full duplex */ -
    DEBUG(FRMSG(DTL))            /* detailed fast replication msgs  */ -
    STORCLAS(SCMIXTGT)           /* target storage class            */ -
    ADMIN                        /* bypass authorization checking   */ -
    DELETE                       /* delete source when finished     */
  SET PATCH 5B=00                /* disable source eattr override   */
/*
```

H is 4-bit track number (0-14)

cccCCCC is 28-bit cylinder number in a contiguous (normal) form

- The colon shows that it is a normalized address

Figure 3-1 EATTR override example

A

System macros and more

Various items and macros that pertain to EAV are described in this appendix, which includes the following topics:

- “Manipulating track changes” on page 30
- “OBTAIN, CAMLST, and CAVFDIR macros” on page 31
- “DEVTYPE (SVC 24) macro” on page 33
- “EAS-eligible - Finding affected programs” on page 33
- “LSPACE macro” on page 34
- “DATATYPE keyword” on page 34
- “DADSM compatibility issues” on page 35

Manipulating track changes

EAV volumes use 28-bit cylinder track addresses instead of the current 16-bit cylinder track addresses to determine where the data set is in a volume. Therefore, track addresses must be manipulated before they are compared to ensure that the correct comparison is performed.

Example A-1 shows the results of a comparison in which the program is trying to compare a 16-bit cylinder track address against a 28-bit cylinder track address. Performing a simple comparison returns the first value, which is higher than the second value.

Example A-1 Results of 16-bit cylinder to 28-bit cylinder track address

```
x'FFF0000E'     -     Cylinder 65,520 track 14
x'0000001E'      -     Cylinder 65,536 track 14
```

To manipulate a track address, you can use a new TRKADDR macro. Programs should not need to perform 28-bit manipulation. Use this macro for all track address computations, even those computations that are not directly affected by this support for VSAM data sets that use EAS.

TRKADDR is an assembler macro that performs conversion and compare operations on direct access storage devices (DASD) track addresses in the form `CCCCcccH`, where `CCCC` is the 16 low-order bits of the cylinder number and `ccc` is the 12 high-order bits of the cylinder number. This track address is referred to as a *28-bit cylinder address*.

TRKADDR works equally well with track addresses that contain a cylinder number less than or greater than 16 bits. It works with all DASD types that are supported by z/OS and includes the following functions:

- Calculate the relative track number on the volume.
- Compare two track addresses.
- Extract the 28-bit cylinder number.
- Extract the 4-bit track number.
- Increment the track address by one track and increments the cylinder number, if necessary.
- Normalize the cylinder number to permit comparing one `cchh` against another.
- Convert a relative track number to a 28-bit cylinder address.
- Set the cylinder number in a 28-bit track address.
- Convert a normalized track address into an absolute 28-bit track address.

Unless otherwise stated, you can specify any registers 0 - 15, except that register 0 cannot be used to address storage. TRKADDR does not use any other registers, even register 13. You can start TRKADDR in 24-bit, 31-bit, or 64-bit mode.

If you use the SYSSTATE macro with AMODE64=YES in an earlier source code statement, TRKADDR might generate more efficient code. You can use TRKADDR in the following ways:

- Calculate the relative track number on the volume (TRKADDR ABSTOREL)
- Compare two track addresses (TRKADDR COMPARE)
- Extract 28-bit cylinder number (TRKADDR EXTRACTCYL)
- Extract 4-bit track number (TRKADDR EXTRACTTRK)

- Increment track address (TRKADDR NEXTTRACK)
- Normalize cylinder number (TRKADDR NORMALIZE)
- Convert a relative track number to a 28-bit cylinder address (TRKADDR RELTOABS)
- Set cylinder number from register (TRKADDR SETCYL)
- Convert normalized track address into an absolute 28-bit track address (TRKADDR NORMTOABS)

IECTRKAD macro

The macro IECTRKAD can be used with programs that are written in a high-level language, such as C, C++, Cobol, or PL/I. This macro features the same functions that are available on TRKADDR.

IECTRKAD is a callable service to perform conversions and comparisons of 28-bit cylinder addresses. The track addresses are in the form `CCCCcccH`, where `CCCC` is the 16 low-order bits of the cylinder number and `ccc` is the 12 high-order bits.

OBTAIN, CAMLST, and CAVFDIR macros

If you specify a data set name by using OBTAIN and the CAMLST SEARCH option, the OBTAIN routine reads the 96-byte data portion of the format-1 DSCB and the absolute track address of the DSCB into virtual storage. The absolute track address is a 5-byte field in the form CCHHR that contains zeros virtual input/output (VIO) data sets.

Access to the volume table of contents (VTOC) records was enhanced to allow a program to read the logical Data Set Control Block (DSCB) chain for a data set in one invocation of a service. The following enhancements were made:

- OBTAIN and CAMLST include an option to specify the number of DSCBs you want to read.
- CVAFDIR provides the keyword MULTIPLEDSCB=YES to indicate to CVAFDIR processing to use the multiple buffers that are passed in the buffer list.

OBTAIN search processing stores another 2 bytes in the callers return area immediately after the 101 bytes that is set by OBTAIN on prior releases. Ensure that your program provides the minimum 140-byte return area. These 2 bytes are set to the total number of consecutive 140-bytes areas that are needed to read all the DSCBs for the data set, as shown in the following example:

```
OBTAIN NUMBERDSCB=number_dscbs
```

NUMBERDSCB specifies a value 0 - 255 that designates the number of consecutive 140-byte return areas that are provided in `wkarea_relexp`. Currently, the system does not support a chain of more than 12 DSCBs for one data set, but you can provide an area that is longer than currently needed.

The system verifies that the provided area is valid. When you provide an area that is long enough to contain more than one DSCB, obtain processing returns DSCBs for the requested data set name in logical VTOC order until all the 140-byte return areas are used.

The logical VTOC order is a format-1 DSCB, followed by zero or more format-3 DSCBs or a format-8 DSCB, followed by one or more format-9 DSCBs, followed by zero or more format-3 DSCBs. No absolute maximum number of DSCBs for a data set should be assumed. The actual number of DSCBs are returned in a field that is in the first 140-byte return area.

If you code the NUMBERDSCB parameter on OBTAIN, the macro execution stores the value in the CAMLST area. You cannot code a register on the CAMLST macro.

Note: For programs that are run on a pre-z/OS R10 system that do not support this keyword, the NUMBERDSCB value is treated as if it were 1.

CVAFDIR macro

For an indexed or non-indexed VTOC, you can use the CVAFDIR macro to perform the following functions:

- Read or write one or more DSCBs by specifying the name of the data set they represent.
- Read or write one or more DSCBs by specifying their addresses.

In addition, you can use the CVAFDIR macro to perform the following functions for an indexed VTOC:

- Read or write VTOC index records. (This function allows calling programs to modify the VTOC index.)
- Read and retain in virtual storage the first high-level VIER, and VIERs that are used during an index search.
- Read and retain in virtual storage the space map VTOC index records (VIRs).
- Free VIRs that are retained in virtual storage.
- Specify whether multiple DSCBs should be processed

CVAFDIR with MULTIPLEDSCBS NO|YES feature the following definitions:

- MULTIPLEDSCBS=NO

  This specification indicates that the calling program requests that only one DSCB should be processed. This specification is the default for MF=L and MF=I forms of the CVAFDIR macro. When the MULTIPLEDSCBS keyword is not specified on the MF=E form, the CV4MULTD setting is left unchanged. When MULTIPLEDSCBS=NO is specified or defaulted, only the first available buffer list entry is processed.

- MULTIPLEDSCBS=YES

  This specification indicates that the calling program requests to read/write multiple DSCBs to or from a buffer list that contains more than one buffer list entry. This parameter causes an indicator in the CVPL, CV4MULTD to be set on. Multiple DSCB processing for reads and writes is requested by specifying the MULTIPLEDSCBS=YES keyword and providing a buffer list that contains more than one buffer list entry (BFLHNOE>1).

DEVTYPE (SVC 24) macro

You can use the DEVTYPE macro to obtain device characteristics information about I/O devices. The DEVTYPE macro that is issued with the INFOLIST parameter (INFO=DASD) returns a different format of the device characteristics information. This parameter includes a 4-byte value for the number of cylinders and is mapped to field DVAICYL in shipped mapping macro, IHADVA.

This macro obtains the following device characteristic information about I/O devices:

- DEVTYPE non-INFOLIST calls:
  - Returns a 2-byte value for the number of cylinders
  - Not valid for an EAV
- DEVTYPE INFO=DASD (INFOLIST calls):
  - Existing call
  - Returns a 4-byte value for the number of cylinders
  - In addition, this macro returns:
    - Multi-cylinder unit (MCU) value
    - First cylinder address where cylinder-managed space begins
    - Cylinder-managed space supported indicator
    - Extended attribute DSCBs supported indicator
    - Block size of index data set

EAS-eligible - Finding affected programs

A product or program can be affected by EAV. Be aware of the following items to support processing a data set that is EAS-eligible:

- Issuing an OBTAIN macro to read a DSCB for an EAS-eligible non-VSAM data set or issuing a macro whose name begins with CVAF for an EAS-eligible non-VSAM data set.
- You must code a new EADSCB=OK option. For performance reasons, you might want to use a new option to read all of the DSCBs for a specified data set with one call.
- Almost any channel program that is issued for an EAS-eligible non-VSAM data set is affected, including building and monitoring.
- Use track address for an EAS-eligible non-VSAM data set, such as those extent entries in IOSEEK in the IOSB, IOBSEEK in the input/output block (IOB) or DS1EXT1 in the DSCB. The track address contains 28-bit cylinder numbers. You can use the TRKADDR macro to assist with these manipulations.
- Use the BBCCHHR field in the I/O error text that is returned by the SYNADAF macro. This result is in EBCDIC form and is similar to text that is returned by VSAM.
- The DASDCALC macro returns information about the space for a data set. It requires new options to return values in larger fields.
- Calculating the size of an EAS-eligible VSAM or non-VSAM data set from the cylinder and track numbers of its extents. These cylinder and track numbers might be from a VTOC, DEB, or from an access method internal control block. The TRKADDR macro can assist with this calculation.
- Examine programs that read VTOCs or DSCBs.

- Programs can read VTOCs or individual DSCBs with BSAM, QSAM, EXCP, OBTAIN, CVAFDIR, and CVAFFILT. Consider the following points:
  - If your program opens an EAV VTOC, it must specify EADSCB=OK on the DCBE macro.
  - Examine these VTOC-reading programs to see whether they might be affected by seeing a format-8 DSCB when they expected a format-1 DSCB. The programs also might be affected by seeing a format-9 DSCB when expecting a format-3 DSCB.
- If your program issues the CVAFDIR or CVAFFILT macro for a data set that has a format-8 DSCB, the macro must have EADSCB=OK.
- Although there are no intended programming interfaces for channel programs with extended format data sets, any such programs must take the 28-bit cylinder numbers into account. The 28-bit cylinder numbers might be in a DSCB, IOB, or channel program. The TRKADDR macro is available to assist with manipulating track addresses.

LSPACE macro

LSPACE provides information about volume size, free space on the volume, free space on the VTOC and INDEX, volume fragmentation, and VTOC status. Also provided is information about the size of the track-managed space and its free space statistics.

The LSPACE macro returns status information (such as LSPACE subfunction, return code, and reason code) in the parameter list. The LSPACE macro also returns the return code in register 15. The following keywords were added:

- EXPMSG

  For volumes that are configured with more than 9999 cylinders, you can use the EXPMSG option to create an expanded message return area that the LSPACE macro needs. For volumes that are configured with cylinder-managed space, you can use the XEXPMSG option to create an extended expanded message return area that the LSPACE macro needs. The use of XEXPMSG is recommended for all requests to return message data.
- EXPDATA

  The expanded data return area (EXPDATA) returns binary data of free space and total volume space information for volumes. For volumes with cylinder-managed space, this status is returned as free space for the entire volume and free space for the track-managed space. The two sets of free space data are the same for a volume that does not have cylinder-managed space.

  The use of EXPDATA is recommended for all requests to return binary data. You can have LSPACE return more information, such as the format 4 DSCB and the total number of free extents on the volume or the fragmentation index.

DATATYPE keyword

The DATATYPE keyword is allowed only when the DATA or EXPDATA keyword is specified. Only the specified information is returned to the caller. DATATYPE is valid for non-EAV and EAV. This keyword eliminates unnecessary I/O that is required to retrieve free space information that is not required by the caller.

If you specify the type, the function returns 1 if the string matches the type; otherwise, 0 is returned. If the string is null, the function returns 0 (except when the type is X, which returns 1 for a null string).

The following new DATATYPE keywords are available:

- ALL: Returns all of the following information
- VOLUME: Returns free space for volume
- VTOC: Returns free space for VTOC
- INDEX: Returns free space for index
- FRAGINDEX: Returns the fragmentation index

The new PLISTVER keyword is used to manage the use of the longer LSPACE parameter list.

DADSM compatibility issues

Applications that have awareness of the DSCBs often cannot access EAS-eligible data sets without code changes. The EADSCB=OK keyword was added to the following services:

- OBTAIN (CAMLST macro)
- CVAFDIR
- CVAFFILT
- CVAFDSM
- CVAFSEQ
- OPEN (DCBE macro) – opening VTOC or VSAM data set with EXCP access.

Not specifying EADSCB=OK causes these services to fail if they are issued to a data set that supports extended attribute DSCBs or a volume that supports cylinder-managed space (CVAFDSM and OPEN).

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Some publications that are referenced in this list might be available in softcopy only:

- *IBM z/OS V2R2: Storage Management and Utilities,* SG24-8289
- *IBM z/OS V2.1 DFSMS Technical Update*, SG24-8190
- *z/OS Version 1 Release 13 Implementation*, SG24-7946
- *z/OS V1.12 DFSMS Technical Update*, SG24-7895
- *ABCs of z/OS System Programming: Volume 4*, SG24-6984
- *z/OS V1.11 DFSMS Release Guide*, SG24-7768
- *DFSMS V1.10 and EAV Technical Guide*, SG24-7617

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and other materials at this website:

ibm.com/redbooks

Other publications

The following publications also are relevant as further information sources:

- *z/OS DFSMS Using Data Sets*, SC26-7410
- *DFSMS Access Method Services for Catalogs*, SC26-73942
- *z/OS MVS System Management Facilities (SMF)*, SA22-7630
- *MVS System Command Manual*, SA22-7627
- *DFSMSdfp Diagnosis*, GY27-7618
- *IBM Health Checker for z/OS V1R12.0 User's Guide*, SA22-7994
- *TSO/E Customization*, SA22-7783
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592

Online resources

The following websites also are relevant as further information sources:

- z/OS V2.2 DFSMS Knowledge Center:

  https://ibm.biz/Bdrcux
- z/OS V2.1 Information Center:

  http://pic.dhe.ibm.com/infocenter/zos/v2r1/index.jsp
- z/OS Hot Topics:

  http://www-03.ibm.com/systems/z/os/zos/library/hot-topics/hot-topics.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5364-00

ISBN 0738455423

Printed in U.S.A.