Redbooks
ibm.com/redbooks

DFSMS: From Storage Tears to Storage Tiers

Keith Winnard

Jose Gilberto Biondo Jr

Glenn Wilcock

z Systems

IBM

Redpaper

TOC
Redbooks
Creating Hybrid Clouds with IBM Bluemix Integration Services
David Kwock
Rahul Gupta
Vasfi Gucer
Cloud
Analytics
IBM.
Solution Guide

It's good
to be
noticed.

THIS PAGE INTENTIONALLY LEFT BLANK

From Storage Tears to Storage Tiers

This IBM® Redpaper™ publication describes how class transitions differ from the classic migration processes and how they can influence application performance levels.

Performance is focused on the following goals:

- Reduce the number of I/Os required to manage the storage
- Optimize data locations to the appropriate device type during its lifecycle

Note: This paper applies to the IBM z/OS® V2.2 level.

This Redpaper publication includes the following topics:

- Understanding the challenges
- Classic migration and recall versus class transition

Understanding the challenges

Business applications and system software depend on input/output (I/O) operations to bring data from external storage into memory for processing, and (where applicable) to write data to one or more external storage peripherals.

The typical make up of external storage is a control unit and an I/O device, such as disk or tape. Multiple devices can be connected to a single or multiple control units for availability and performance reasons.

The channel subsystem (CSS) is a collection of entities that manage and sustain the I/O operation. Although the CSS is an important part of availability and performance, it is out of the scope of this Redpaper publication.

Storage devices vary based on the following characteristics:

- Performance: The speed at which the control unit and device data can be accessed.
- Data retention: Determine whether to retain permanent data or cater for temporary data.
- Capacity: How much data can be held on the device.
- Data set type: Affects the storage that it can be held on for direct access.
- Attributes: The volume can be static or portable.
- Functionality: The control unit might have independent functionality separate from the platform instances to which it is connected.

The data that is stored and accessed also has varying characteristics, including the following typical types:

- System critical: Required for the continuous and smooth running of the system.
- Business application critical: Required to process orders and complete transactions.
- Control Data: Data that is needed to start events that are based on timers or events, such as a job schedule calendar.
- Value: Data goes through a lifecycle depending on its worth, such as:
  - Immediate transaction data
  - Data that is passed between programs
  - Archived data
  - Sensitive data
  - Copied data
  - Regulatory data

The application program cannot process the data while it is moving from external storage to the memory for the processor unit. This issue results in a delay to the application.

Systems software and business applications are prioritized. The systems software features a hierarchy of tasks that include relative priorities that must run to keep the system available. Business applications also feature this configuration; the higher priority tasks must process the data as quickly as possible to meet customer and business Service Level Agreements. This paper focuses on application data for the remainder of this paper but the principles are the same for operational and infrastructure-related system data.

Data occupying storage must be available directly to the application and delivered to it in the shortest time possible to prevent or minimize delays in access times. The application task waits until the data arrives.

The combination of storage and data characteristics present a challenge to the storage manager to define the following policies:

- Consider the storage and data characteristics
- Manage the data’s location
- Minimize the time to access the data
- Are sensitive to the data’s current position within its lifecycle
- Reduce intermediate staging of data
- Recognize the priority of the data
- Assist in providing the appropriate response times
- Provide the appropriate storage management return on investment (ROI)

Next, we review the aspects of classic storage management strategy and techniques and then see how these aspects can be changed to help achieve better performance levels.

Classic migration and recall versus class transition

In this section, the classic migration approach is reviewed and the transition class is introduced.

Classic migration and recall

Here, we review the classic migration approach, starting with the storage tiers.

Basic tiers

The following traditional tiers are available:

- Primary storage level 0 hierarchy
- Migration level 1 hierarchy
- Migration level 2 hierarchy

The primary storage level contains data that is directly accessed by programs, as shown in Figure 1. At this level, the DFSMShsm manages the data.

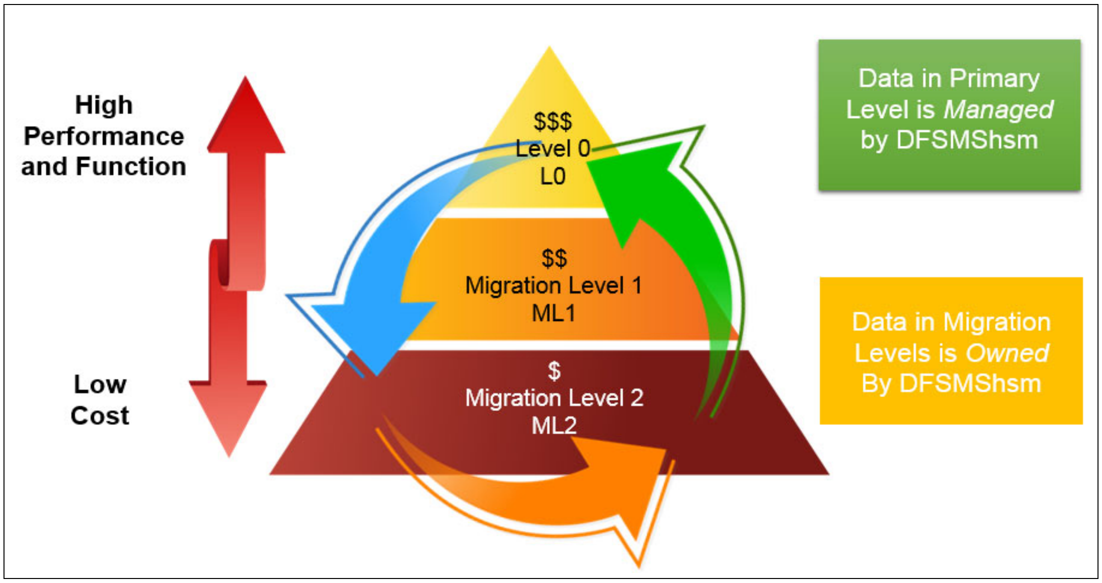

High Performance and Function
Low Cost
$$$
Level 0
L0
$$
Migration Level 1
ML1
$
Migration Level 2
ML2
Data in Primary Level is *Managed* by DFSMShsm
Data in Migration Levels is *Owned* By DFSMShsm

Figure 1 Classic migration approach

The classic DFSMS storage hierarchy was available and in use by customers around the world for over 35 years and consists of three levels of data management with different performance and cost: Primary volumes (L0), Migration Level 1 (ML1), and Migration Level 2 (ML2).

Data is moved down in the hierarchy by policy-based space management. It is moved to a migration level that converts the data to a proprietary format that cannot be directly accessed by users or applications. Before this data can be accessed, it must be returned to its original format and moved back to the Level 0 tier.

The recall process might be acceptable for batch and non-critical data; however, applications that require a higher performance can retain data on high-performance solid-state drive (SSD) disks.

Devices and tiers

There often are five types of devices that can be managed in the following tiers:

- Tier 1: Flash cards and flash drives that are shown in Figure 2 on page 4 by SSDs
- Tier 2: SAS 10k or 15 K RPM disk drives
- Tier 3: Nearline 7.2 K RPM disk drives

Although the primary storage hierarchy can contain multiple device types (as shown in Figure 2), it is still a single tier.

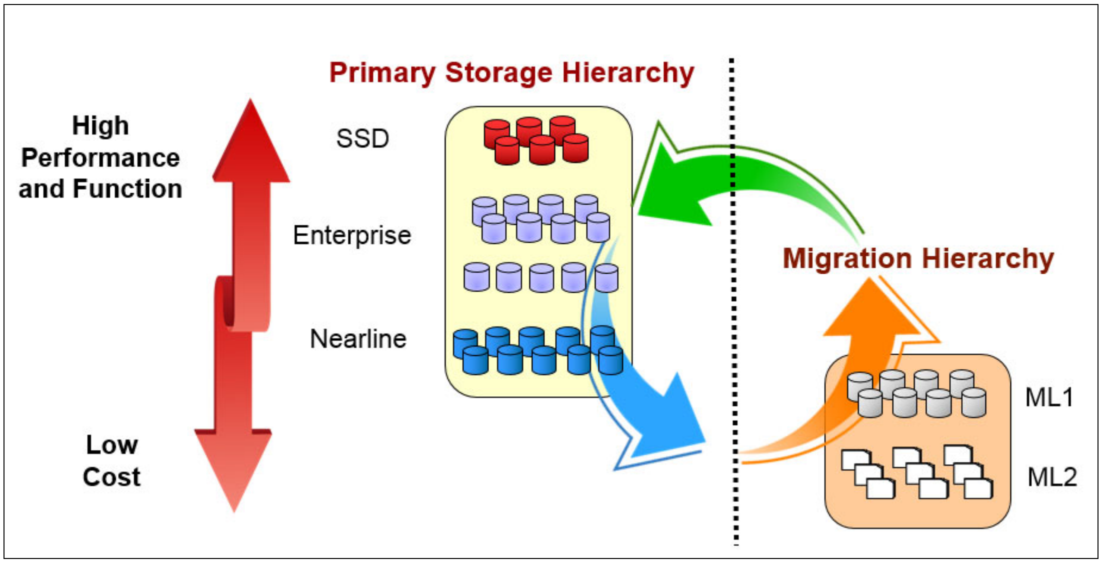


Figure 2   Storage devices within the primary hierarchy

The drives within a tier must be homogeneous. If 10 K and 15 K RPM disk drives are in the same extent pool, they are managed as a single tier.

Although SSD is high-speed storage that reduces I/O times by providing a faster read/write response time than Enterprise and nearline disks, it attracts a higher cost. Some applications can function with a lower response time.

Other solutions, such as IBM Easy Tier®, address this problem to some extent by moving 1 GB extents between SSD and Enterprise volumes, according to a heat map. In this configuration, extents that are more frequently accessed are most likely to be moved over to SSD.

Because Storage Tier is managed at the hardware level, it is not data set oriented, and situations where critical data is not moved to SSD because other data in the same physical extent is not frequently accessed.

Figure 3 on page 5 shows a sample heat map that is used by Storage Tiers. In this map, some hot data might be allocated on Nearline and cold data is allocated on SSDs based on the number of accesses the 1 GB extent received within a specific time. At specific intervals, Easy Tier transfers the hot data from the Nearline extent to an SSD extent and the cold data from the SSD extent to a Nearline extent.

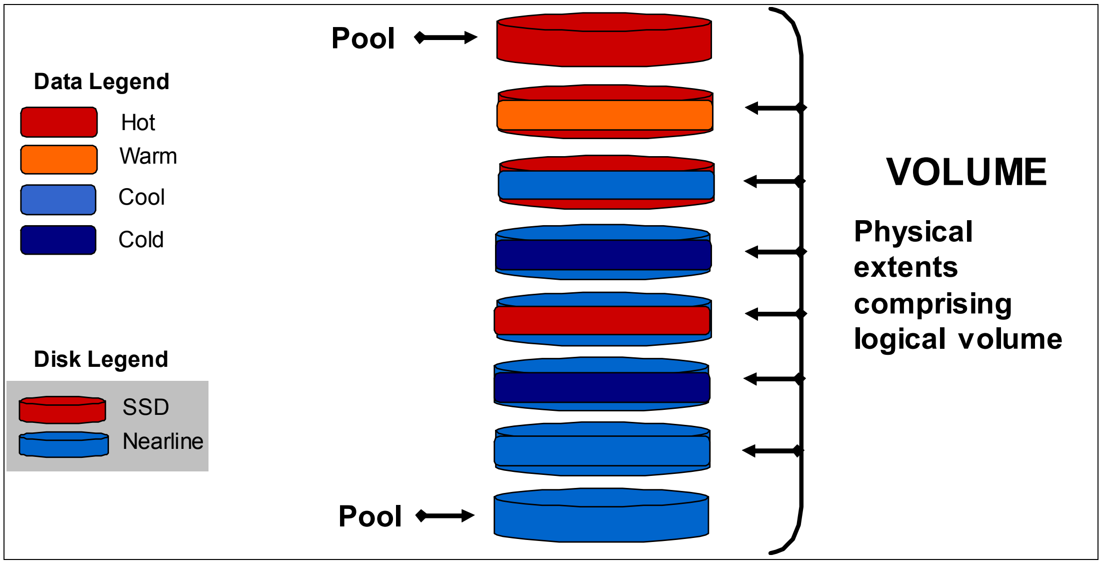


Figure 3 Storage tier heat map of a volume

Class transitions

The function of DFSMShsm Space Management processing is to use policy-based automation to ensure that volumes within the Primary Storage Hierarchy have enough free space for new data. It also ensures that data is stored at the lowest acceptable tier in the Storage Hierarchy.

The Class transition introduces an automated, policy-based space management that moves data from tier to tier *within* the Primary (Level 0) Hierarchy. This data movement can be performed automatically on DFSMShsm space management functions or user-initiated on z/OS V2R2 and later releases.

When Space Management (Primary, Interval, or on-demand) starts, it selects the data sets for transition based on Management Class policies to decide whether the data set is eligible for transition. If so, DFSMShsm starts ACS routines to assign a new Storage Class, Management Class, or Storage Group. If the Storage Group changes, DFSMShsm attempts to move the data set to the new pool.

By using Class transition, you can create several Storage Groups with different storage devices, allocate the data to the correct pool based on required response time, and relocate it to other pools as its criticality decreases. This solution can be used with Easy Tier to provide an even higher level of data management from a software and hardware perspective.

Figure 4 on page 6 shows a sample storage environment set-up with a smart tier 0 (high-performance disk), a smart tier 1 (lower-performance and cost efficient), and migration level 2. By using class transition, you can move your data between tier 0 and tier 1 while keeping the data available for use by applications and on a device that is suited to the application’s performance needs.

Smart Tier 1 uses Easy Tier to transparently move data between SSDs and Enterprise disks based on the heat map. Inactive data never is stored on a tier that is lower than an Enterprise disk.

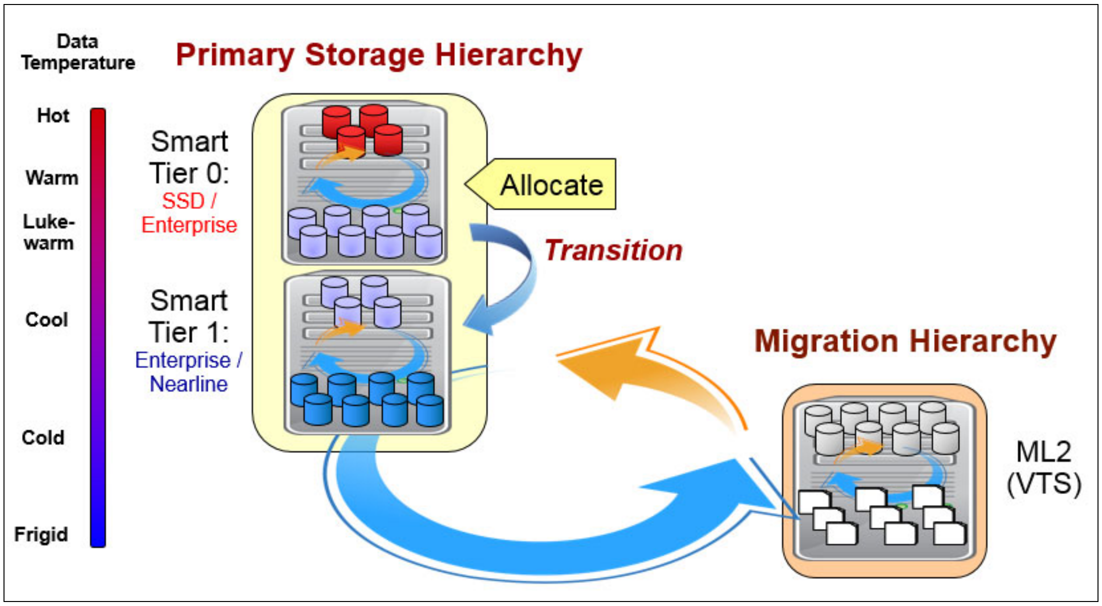


Figure 4 Smart tiers within the primary hierarchy

On the left side of Figure 4 is the data temperature, which is a measure of how often the data is being accessed. Hot data is data that is frequently referenced. This example shows an example environment in which IBM Easy Tier devices are the primary tier in the hierarchy. Data is allocated to an Easy Tier device and the storage controller transparently moves data up and down within the storage controller based on the temperature of the data.

After 90 days from creation, there is a subset of data that includes decreasing business value; for example, online customer reports. It is expected that these reports are not referenced often, and even if they are, it is acceptable for the access times to be slower.

In addition, it is not wanted that a frequently accessed historical report is moved to an SSD within the Easy Tier subsystem. These data sets are assigned a policy such that they are transitioned out of an Easy Tier device after 90 days. This configuration ensures that the reports are on the lowest cost storage, but still enables the reports to be kept online and avoid migration. They are transitioned to Smart Tier 1, which is comprised mostly of Nearline devices, but also of Enterprise devices.

Easy Tier moves the hot data up to Enterprise drives and the cold data down to Nearline drives. Because the data in this Smart Tier features lower business value, it cannot move to SSDs because it is not defined within the Smart Tier 1. It also frees storage within the Easy Tier devices for new allocations.

If a report is not accessed for 1 year and 1 day, it is eligible to be migrated to ML2, where it is expected to remain.

Activity levels

You must be aware of the following distinction when migrate or recall and class transitions are used:

- When a data set is recalled, it is returned to the storage class as directed by the ACS routines, which often are higher than where a data set is stored after transition.
- When a data set transitions to a lower class of storage, it remains there until it is transitioned again or until it is migrated.

> **Note:** For an IBM FlashCopy® to be used for a transition, the movement must be within the same storage controller. This requirement might be difficult to achieve, depending on your configuration.

Migration, recall, and transition

Figure 5 shows a data set temperature since its creation until it is no longer used. Initially, the data sets can be frequently accessed; therefore, it remains available to the application. However, as the data set ages, the frequency of requests to access the data set can fall, and the migration and recall activities are increased until it is no longer used. Migrate and recall activities increase higher CPU utilization and delays in response times while the data sets are being recalled for use by the application.

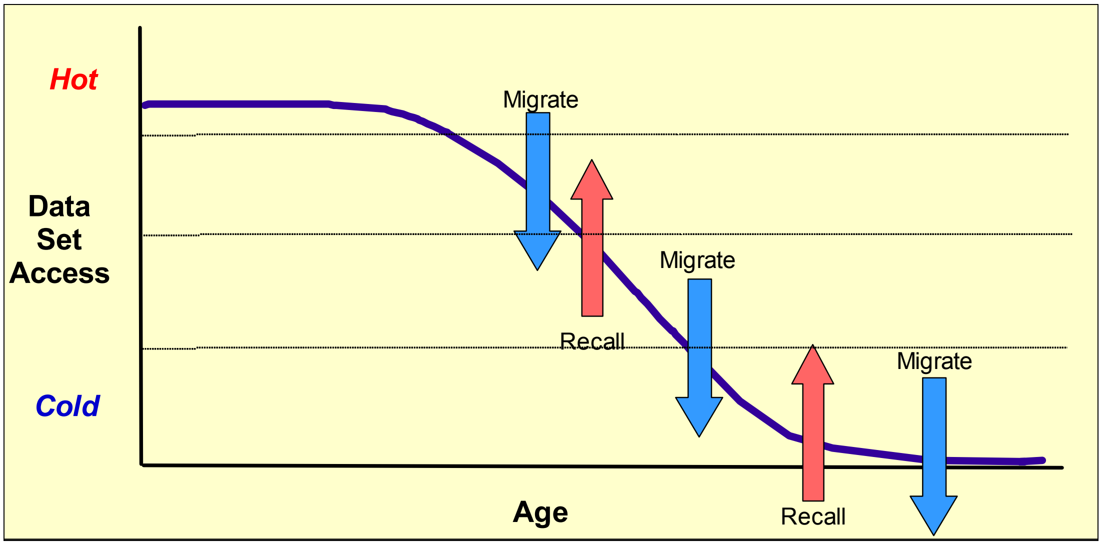


Figure 5 Migration and recall activity levels

In contrast, Figure 6 shows how multiple migration and recall cycles can be replaced with a single class transition.

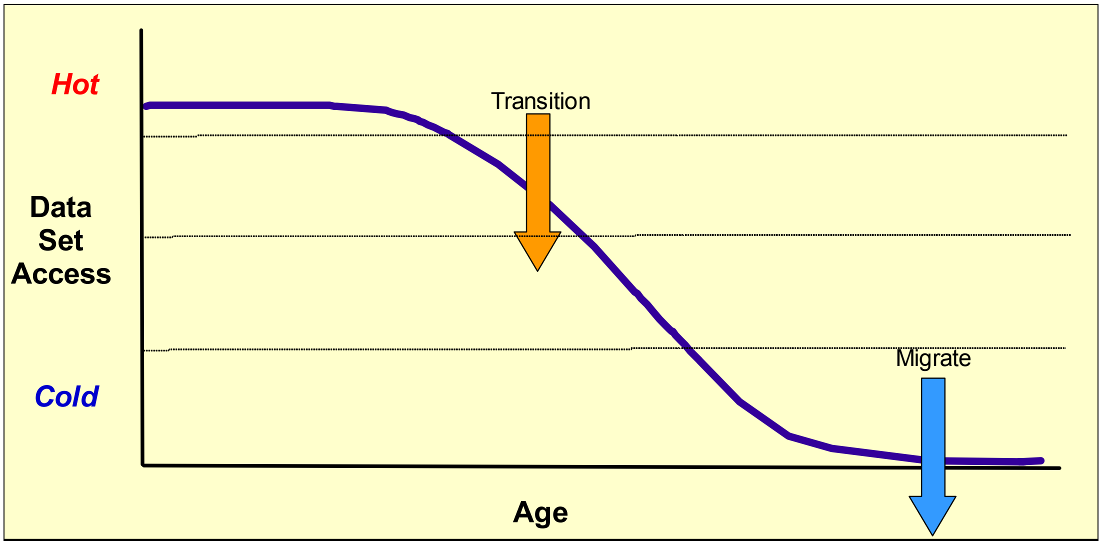


Figure 6 Transition activity levels

The transition moves the data set to a lower tier within the primary hierarchy, which maintains immediate access to the data set. A potentially single migration to ML2 might be effected after a specified period.

Figure 7 shows a comparison of space management processing between a data set that is going through the classic approach of migrate and recall, and the transition approach with a migrate option.

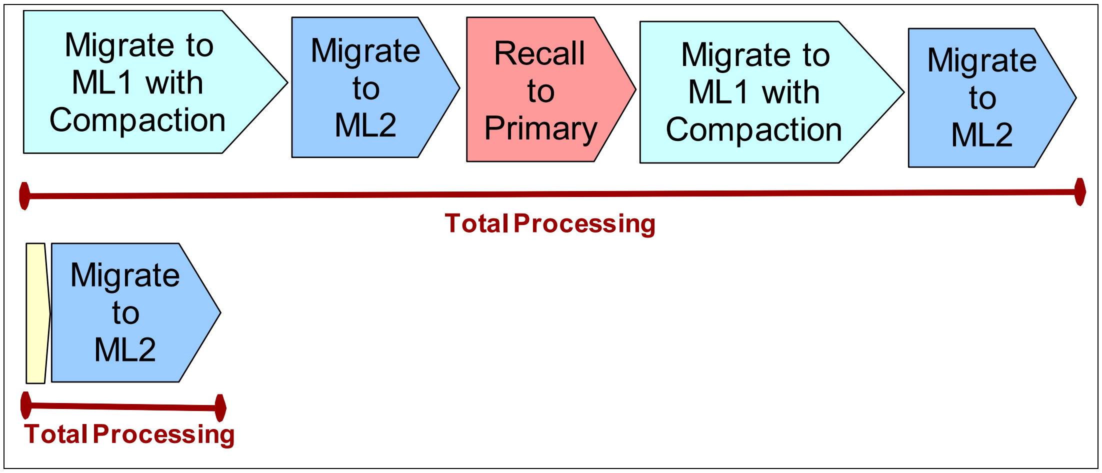


Figure 7 Migrate, recall, and transition scenario

Implementing and using class transition

Planning and preparation helps you achieve a smooth implementation. A phased approach is wiser than a single instance conversion.

Preparation

Before implementing class transition in your environment, consider the following points that might affect how transitions are performed in your systems:

- Consider setting up two Storage Groups that contain high-performance volumes into two specific Storage Groups that are used by applications that can benefit from the high-performance volumes.
- Set up more Storage Groups with Enterprise and Nearline volumes to be used as target Storage Groups as the critical data ages and can be transitioned from the higher performing Storage Groups.
- An option is available to set up a few Storage Groups with volumes that are managed by Easy Tier. This configuration gives you extended flexibility to move your data between SSDs and HDDs until the data is eligible for transition.

Sample set-up

In this section, we provide you with more information about how you might approach implementing the class transitions. Because your configuration is unique to you, your implementation plan and execution is suited to your environment.

The following sample activities are used:

- In this sample we start by setting up the following Storage Groups:
  - TIER0: High performance
  - TIER1: Low performance
- If TIER0 and target TIER1 Storage Groups are defined on the same DASD controller, DFSMShsm can perform the data movement by using Flashcopy technology. If the FlashCopy is not available or cannot be used by the time transition occurs, DFSMShsm uses standard I/O, which uses CP time and takes longer.
- The ACS routines are updated with appropriate class transition logic. The environment variable value `SPMGCLTR` identifies that the ACS routine is being called for a class transition. During class transition processing, the ACS Storage Class, Management Class and Storage Group routines are called to assign the most appropriate constructs. Example 1 shows the Logic for updating the Storage Class for a data set during class transition.

Example 1 Sample ACS code to handle transitions

```
SELECT

   WHEN &ACSENVIR = 'SPMGCLTR' AND &STORCLAS = 'SCTIER0' DO

         SET &STORCLAS = 'SCTIER1'

         EXIT

   END

      OTHERWISE  DO

           SET &STORCLAS = &STORCLAS

          EXIT

     END

END
```

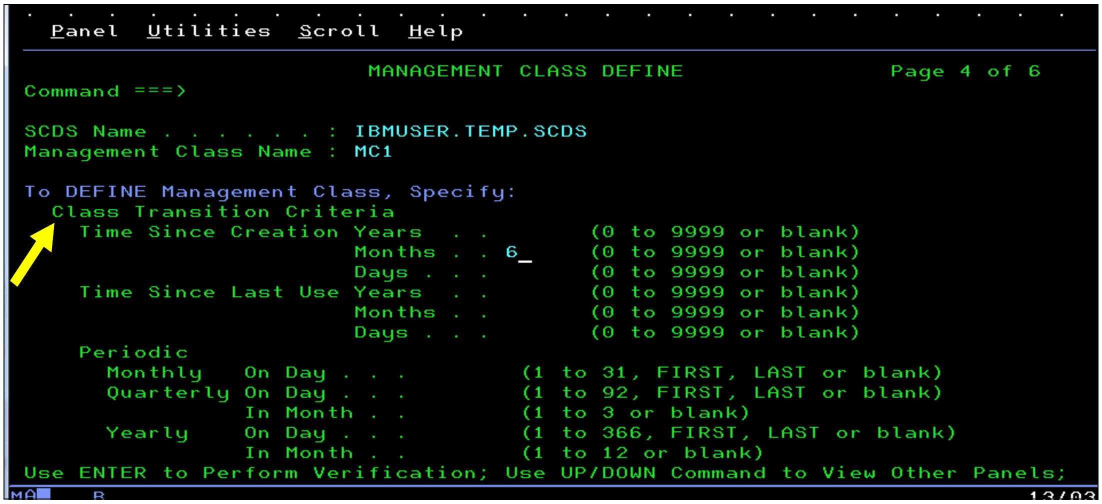
- We update management class constructs with transition information.

> **Note:** Plan to change only a few Management Classes each time to avoid flooding DFSMShsm with transitions.

- Within the Management Class, the Class Transition Criteria specifies whether and when a data set is transitioned. The following options are available for use:
  - Default: Class transitions are not performed.
  - Time since Creation: The data set can transition on or after this time. This setting is a subjective setting. It indicates that regardless of the usage of the data set, it must be transitioned.

- Time since Last Use: The data set can transition on or after this time. This setting is an objective setting. It indicates that a data set must not be transitioned until it is not referenced for a certain period.
- Periodic: The data set can transition on a specific date. This setting is a subjective setting. It indicates that regardless of the usage of the data set, it must be transitioned.

Figure 8 shows the Class transition criteria option when defining a new Management Class.



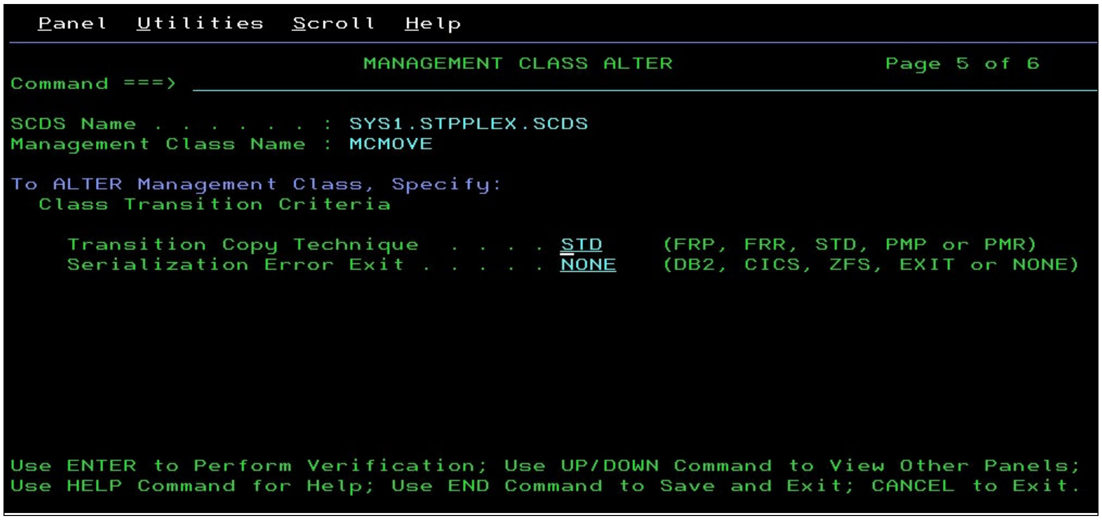
```
  Panel  Utilities  Scroll  Help

                         MANAGEMENT CLASS DEFINE              Page 4 of 6
Command ===>

SCDS Name . . . . . . . : IBMUSER.TEMP.SCDS
Management Class Name : MC1

To DEFINE Management Class, Specify:
  Class Transition Criteria
    Time Since Creation Years  . .       (0 to 9999 or blank)
                        Months . . 6_    (0 to 9999 or blank)
                        Days . . .       (0 to 9999 or blank)
    Time Since Last Use Years  . .       (0 to 9999 or blank)
                        Months . .       (0 to 9999 or blank)
                        Days . . .       (0 to 9999 or blank)
    Periodic
      Monthly   On Day . . .        (1 to 31, FIRST, LAST or blank)
      Quarterly On Day . . .        (1 to 92, FIRST, LAST or blank)
                In Month . .        (1 to 3 or blank)
      Yearly    On Day . . .        (1 to 366, FIRST, LAST or blank)
                In Month . .        (1 to 12 or blank)
Use ENTER to Perform Verification; Use UP/DOWN Command to View Other Panels;
```

Figure 8 Class transaction criteria

- ► Time Since Creation and Periodic can specify that a transition occur when a data set is being accessed. There is a new field called “Serialization User Exit” That is used to decide what action to take if a data set cannot be exclusively serialized for data movement.
- ► An application’s database might always be open and special processing is required to transition the data at any time. When a data set is selected for transition is open, DFSMShsm checks for the Serialization User Exit attribute that is defined in the Management Class. The following options can be used to define how to handle this situation:
  - NONE: (Default) If a data set cannot be exclusively serialized, the data set is not transitioned.
  - IBM DB2®: Data that is assigned to this Management Class is DB2 objects. Start DB2 to close and deallocate the object. If this process is successful, the object is serialized and moved and DB2 is started to reopen the object.
  - IBM CICS®: Data that is assigned to this Management Class is CICS objects. Start CICS to take the object offline. If this process is successful, the object is serialized and moved and CICS is started to reopen the object.
  - zFS: Data that is assigned to this Management Class is zFS data sets. Start zFS to unmount the data set. If this process is successful, the data set is serialized and moved and zFS is started to remount the data set.
  - EXIT: User exit is started to preprocess and post-process the data set. It enables users and ISVs to provide an exit that is started before and after transitioning an allocated data set. The data set is transitioned only if serialization is obtained after the first invocation of the exit.

- You can also define the Transition copy technique to be used for each Management Class. The following copy techniques are available:
  - Standard: (Default) Use standard I/O.
  - Fast Replication Preferred: Prefer Fast Replication. If it cannot be used, use standard I/O.
  - Fast Replication Required: Require Fast Replication: If it cannot be used, fail the data movement. This technique requires the target volume to be in the same storage controller.
  - Preserve Mirror Preferred: Prefer Preserve Mirror: This technique indicates that a Metro Mirror primary volume can become a FlashCopy target volume. If Preserve Mirror cannot be used, FlashCopy or standard I/O can be used.
  - Preserve Mirror Required: Require Preserve Mirror: The transition is performed only if the Metro Mirror primary target volume does not move to duplex pending. This parameter has no effect if the target volume is not a Metro Mirror primary volume.

  Figure 9 shows the Transition copy technique and Serialization Error Exit options when a Management Class is defined or altered.



```
  Panel  Utilities  Scroll  Help
                          MANAGEMENT CLASS ALTER                 Page 5 of 6
Command ===> 

SCDS Name . . . . . . : SYS1.STPPLEX.SCDS
Management Class Name : MCMOVE

To ALTER Management Class, Specify:
  Class Transition Criteria

    Transition Copy Technique  . . . . STD     (FRP, FRR, STD, PMP or PMR)
    Serialization Error Exit . . . . . NONE    (DB2, CICS, ZFS, EXIT or NONE)








Use ENTER to Perform Verification; Use UP/DOWN Command to View Other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 9 Transition copy techniques and serialization exit options

- When a copy technique other than Standard is specified, a valid backup copy must be available and the DS Change Indicator is set to `OFF` before the data set is transitioned.
- If you use a product other than DFSMShsm to create backup copies of your data sets, you can patch DFSMShsm to not check whether a valid backup copy is available before transition by using the following patch:

  ```
  PATCH.MGCB.+111 BITS(......1.)
  ```

- In z/OS V2R2 and later releases, the use of Preserve Mirror Required allows you to transition data sets even if the data set change indicator is set to `ON`. To enable this feature in previous releases, you can use the following patch:

  ```
  PATCH.MGCB.+111 BITS(.......1)
  ```

- A new DFSMShsm SETSYS attribute is available to control whether class transitions must be performed by event-driven migration (On-demand and Interval migration). You can also define whether serialization user exit should be performed if a data set that is selected for transition is open by an application. To enable class transition for on-demand and serialization user exit, use the following command:

```
SETSYS CLASSTRANSITIONS(EVENTDRIVENMIGRATION(Y|N SERIALIZATIONEXIT(Y|N)))
```

> **Note:** The data set migration exit ARCMDEXT (MD) was also updated to override transition with migration.

Scenarios

This section presents two scenarios that show the benefits of implementing class transition.

Scenario 1: Application and historical data

The first scenario is based on the following requirements:

- Monthly customer statements are allocated on standard enterprise disk.
- After 45 days, the statements remain online, but are transitioned to Nearline disk to reduce the cost of storing the statements online.
- As the statements age, they are not likely to be updated.
- Only one backup version is required.

To meet these requirements, the following Storage Groups and Management Classes are defined:

- SGHIGHP: A high-performance group that uses SSD and Enterprise volumes in the backend and are controlled by Storage Tier.
- SGHIST: Holds the historical data and is assigned to Nearline volumes.
- MCSTMT45: Keeps two backup copies of the data set and enables it for transitions after 45 days since it was created.
- MCHSTRC: Manages the historic data with one backup version and no migration.

The Management Class definitions are listed in Table 1.

Table 1 Scenario 1 Management Class definitions

Definition	MCSTMT45	MCHSTRC
Command or Auto Migrate	NONE	NONE
Number of Backup Versions	2	1
Class Transition Criteria Time Since Creation Days	45	N/A
Transition Copy Technique	FRP	N/A
Serialization Error Exit	NONE	N/A
Expire After Days Non-Usage	N/A	NOLIMIT

The ACS routines require updates to handle the class transition processing. Example 2 shows the ACS routine statements for setting up MCHSTRC Management Class.

Example 2 Scenario 1 ACS routine for MCHSTRC

```
IF &ACSENVIR = 'SPMGCLTR' AND &MGMTCLAS = 'MCSTMT45' THEN

   DO

         SET &MGMTCLAS = 'MCHSTRC'

         EXIT

   END
```

Example 3 shows the ACS routine statements for setting up the SGHIST Storage Group selection. Because the Management class was reassigned, the new value is passed on.

Example 3 Scenario 1 ACS routine for SGHIST

```
IF &ACSENVIR = 'SPMGCLTR' AND &MGMTCLAS = 'MCSTMT45' THEN

   DO

         SET &STORGRP = 'SGHIST'

         EXIT

   END
```

During data set allocation, the Management Class MCSTMT45 and Storage Group SGHIGHP are selected based on ACS selection. For the next 45 days, the data set is managed by DFSMShsm by using the Management Class MCSTMT45, which includes two backup copies. The hardware Storage Tier moves the data between SSD and Enterprise volumes based on the heat map, as shown in Figure 10.

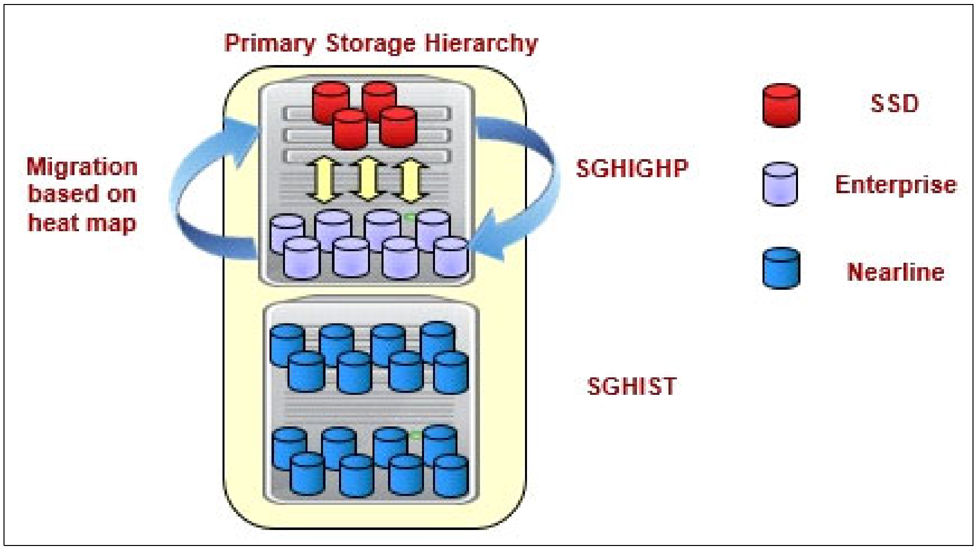


Figure 10 Management Class MCSTMT45 and Storage Group SGHIGHP relationship

After 45 days pass since the data set was last used, it is eligible for transition if all conditions are met (the DSCI indicator is OFF and a valid backup is available). The DFSMShsm calls ACS routines to assign the new constructs and move the data set, as shown in Figure 11.

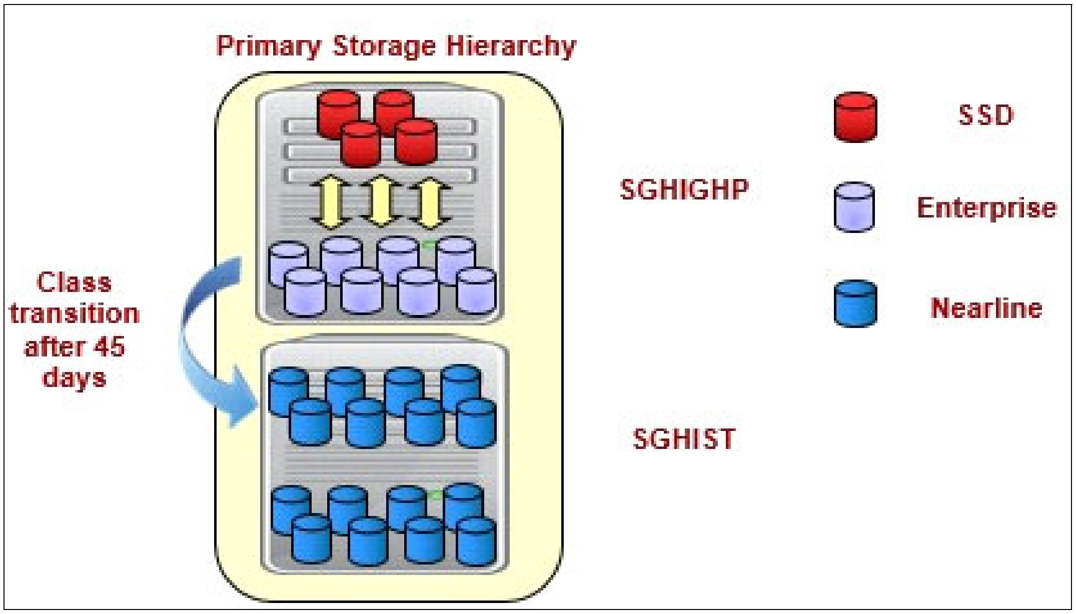


Figure 11 Class transition to Storage Group SGHIST and Management Class MCHSTRC

Scenario 2: DB2 Tables

The second scenario is based on the following requirements:

- New DB2 tables are created for a temporary project.
- The tables require high performance volumes.
- After six months, the project is completed.
- At project completion, the tables can be moved to Enterprise volumes.

To meet these requirements, the following Storage Groups and Management Classes are defined:

- SGSSD: A high-performance group that uses SSD volumes.
- SGENTP: Holds the project’s DB2 Tables on Enterprise volumes.
- MCDBTMP: Keeps the project’s DB2 tables on SSD volumes and enables them for transitions after six months from the creation date.
- MCDBHIST: Manages the project’s DB2 tables after the project development phase completes.

Two new Management Classes are defined to accommodate the data sets: MCDBTMP and MCDBHIST.

The Management Class definitions are listed in Table 2.

Table 2 Scenario 2 Management Class definitions

Definition	MCDBTMP	MCDBHIST
Command or Auto Migrate	NONE	NONE
Number of Backup Versions	2	1
Class Transition Criteria Time Since Creation	6	N/A
Transition Copy Technique	PMR	N/A
Serialization Error Exit	DB2	N/A
Expire After Days Non-Usage	N/A	NOLIMIT

The ACS routines require updates to handle the class transition processing. Example 4 shows the ACS routine statements for setting up MCDBTMP Management Class.

Example 4 Scenario 2 ACS routine for MCDBTMP

```
IF &ACSENVIR = 'SPMGCLTR' AND &MGMTCLAS = 'MCDBTMP' THEN

   DO

         SET &MGMTCLAS = 'MCDBHIST'

         EXIT

   END
```

Example 5 shows the ACS routine statements for setting up the SGHIST Storage Group selection. Because the Management class was reassigned, the new value is passed on.

Example 5 Scenario 2 ACS routine for MCDBHIST

```
IF &ACSENVIR = 'SPMGCLTR' AND &MGMTCLAS = 'MCDBHIST' THEN
   DO
         SET &STORGRP = 'SGENTP'
         EXIT
   END
```

The tables are allocated on SGSSD and remain there until the six months pass, as shown in Figure 12.

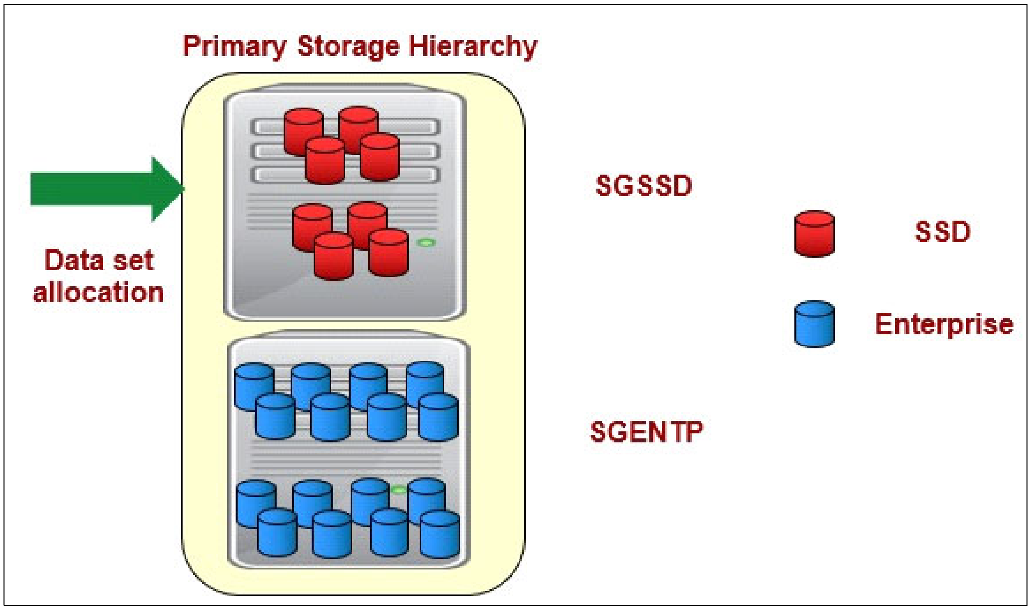


Figure 12 Initial allocation to high performance SSD

Because we set the Serialization User Exit to DB2, DFSMShsm calls DB2 to deallocate the data set, perform class transition, and call DB2 to allocate the data set again if the data set is open. If the unallocation fails, no transition is performed, as shown in Figure 13.

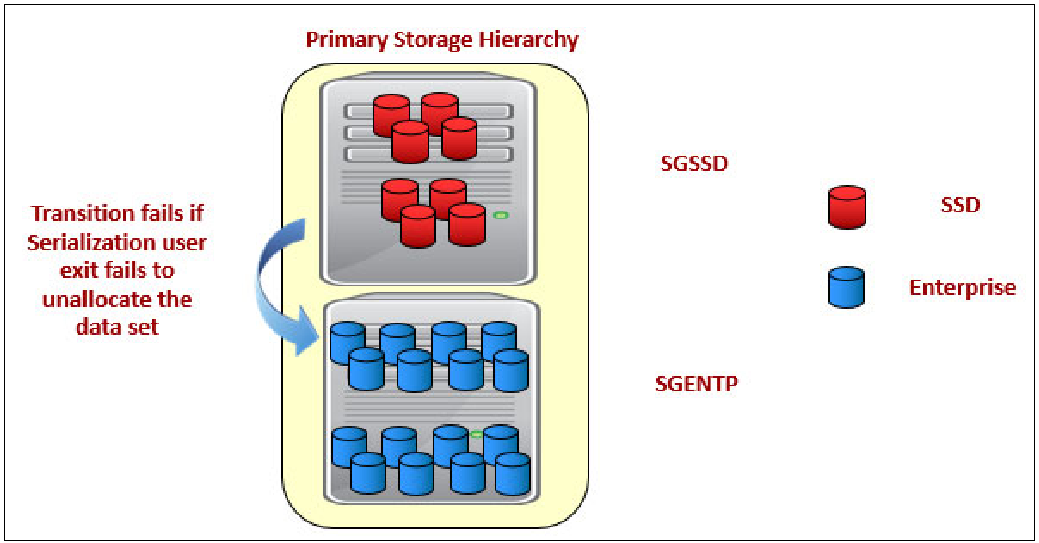

Primary Storage Hierarchy
Transition fails if Serialization user exit fails to unallocate the data set
SGSSD
SGENTP
SSD
Enterprise

Figure 13 Effect of Serialization User Exit

Summary

Class transitions can help to achieve, sustain, and manage application performance levels through the lifecycle of the application’s data. The key benefit is that the application data can have a greater direct availability to the application rather than have to be processed via an intermediary to stage the data.

The data positioning can be managed by using a combination of Storage Groups, Management Classes, ACS routines, and the Storage Tiers.

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 z/OS Project Leader at the International Technical Support Organization, 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 mainframes.

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

Glenn Wilcock is a Senior Technical Staff Member with the IBM Systems Unit in Tucson, Arizona. He is the lead architect for z/OS DFSMShsm, which provides automated, policy-driven space management and backup and recovery for enterprise systems data. Glenn has over 25 years of experience developing storage software solutions.

Thanks to **Barbara McDonald** (DFSMS Product Strategist, IBM Boulder) for direction and guidance throughout the creation of this IBM Redbooks publication.

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

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

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®
DB2®
Easy Tier®
FlashCopy®
IBM®
Redbooks®
Redpaper™
Redbooks (logo) ®
z/OS®

REDP-5341-00

ISBN 0738455377

Printed in U.S.A.