IBM Systems

DFSMShsm best practices – Part 2

Legal disclaimer

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice.

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

References in this document to IBM products, programs, or services does not imply that IBM intends to make such such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

DFDSS
DFHSM
DFSMSdfp
DFSMShsm
DFSMSdss

IBM®
System/390®
SNAP/SHOT®
Enterprise Storage Server®
FlashCopy®
OS/390®
z/OS®

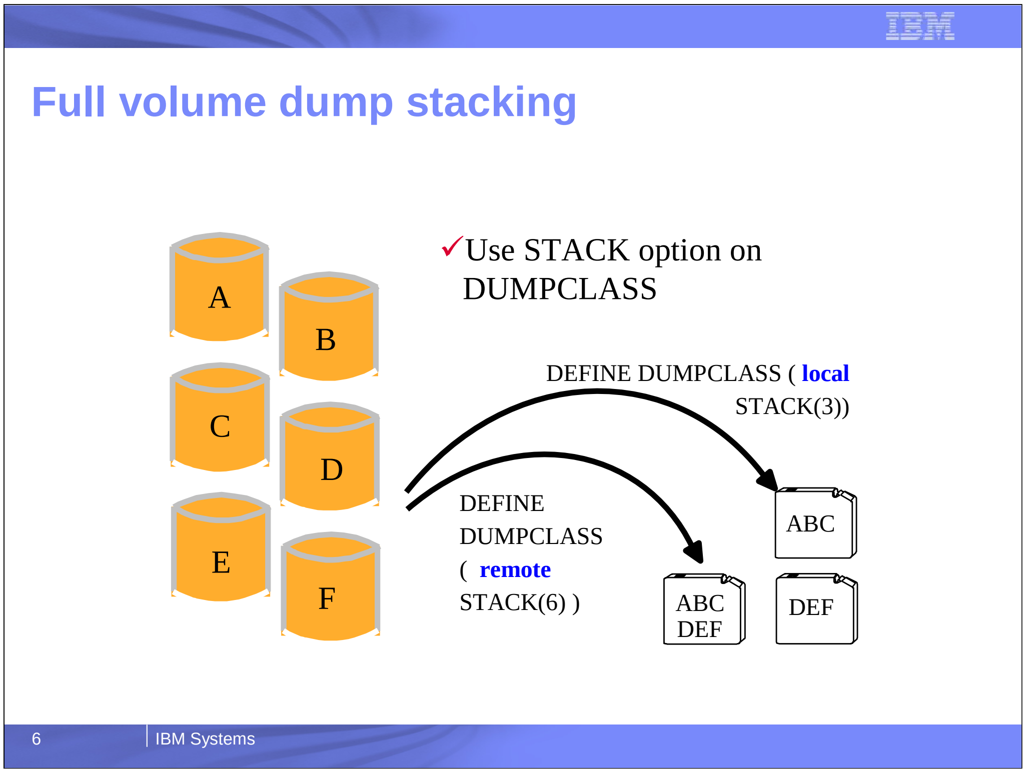
Agenda

- Managing and tuning the **HSM tape** environment
  - Full volume dump stacking
  - ABARS file stacking
  - Collocate ABARS ML2 data
  - Reduce tape spanning
  - Recycle TAPELIST and SELECT
  - Recycle performance

- Miscellaneous tips and hints
  - Mash
  - SMF consolidation
  - Avoid log overhead
  - Use of the ONLYIF command in PARMLIB member

Agenda: managing and tuning the HSM tape environment

- **Managing and tuning the HSM tape environment**
- Miscellaneous tips and hints



Full volume dump stacking

✓Use STACK option on DUMPCLASS

DEFINE DUMPCLASS (**local** STACK(3))

DEFINE DUMPCLASS (**remote** STACK(6))

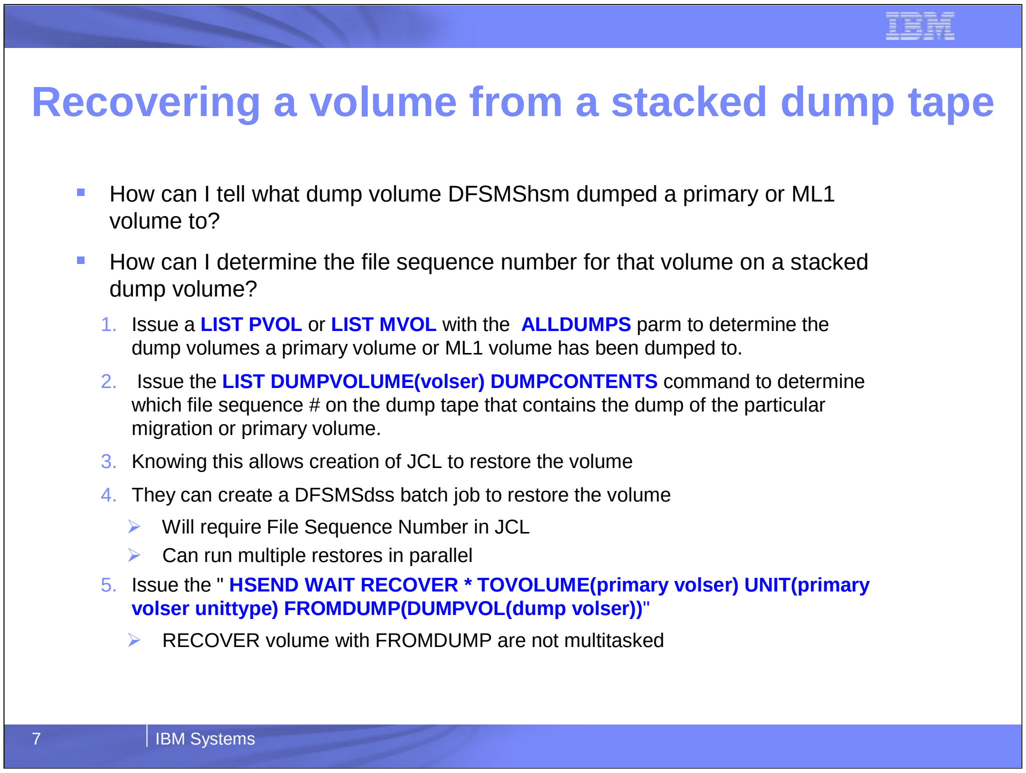
Installations can dump multiple volumes on a single dump tape by using the STACK option in the DUMPCLASS definition

Installations can also create up to 5 dump copies of a volume in parallel by dumping to multiple dump classes simultaneously.

In this example the dump is performed to 2 dumpclasses in parallel, one to be kept onsite and another to be kept in a remote location.

In this case you might want a higher stack value for the remote dumplcass to minimize the number of dump tapes that have to be stored offsite.

Additionally, you may wish to have a lower stack value for those dumps being kept locally to allow for greater parallelism during restore processing.



Recovering a volume from a stacked dump tape

- How can I tell what dump volume DFSMShsm dumped a primary or ML1 volume to?
- How can I determine the file sequence number for that volume on a stacked dump volume?

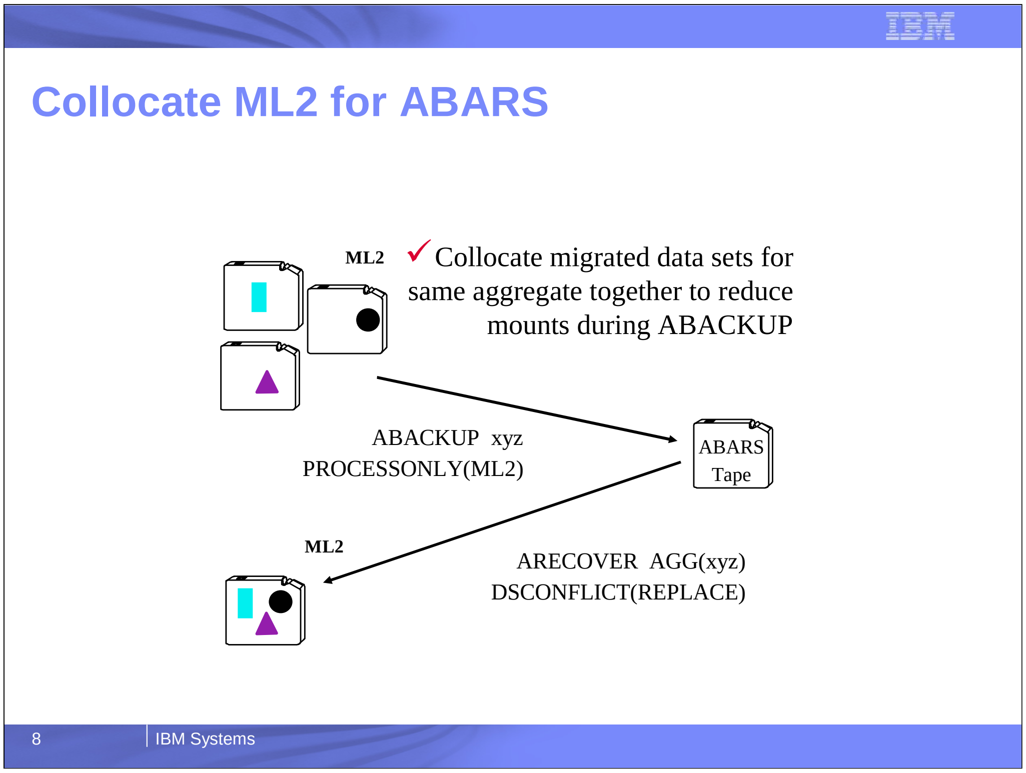
1. Issue a **LIST PVOL** or **LIST MVOL** with the **ALLDUMPS** parm to determine the dump volumes a primary volume or ML1 volume has been dumped to.
2. Issue the **LIST DUMPVOLUME(volser) DUMPCONTENTS** command to determine which file sequence # on the dump tape that contains the dump of the particular migration or primary volume.
3. Knowing this allows creation of JCL to restore the volume
4. They can create a DFSMSdss batch job to restore the volume
   - ➢ Will require File Sequence Number in JCL
   - ➢ Can run multiple restores in parallel
5. Issue the " **HSEND WAIT RECOVER * TOVOLUME(primary volser) UNIT(primary volser unittype) FROMDUMP(DUMPVOL(dump volser))**"
   - ➢ RECOVER volume with FROMDUMP are not multitasked

This slide explains how installations can determine what dump tape volume contains the dump copy of a particular primary or ML1 volume and what is its dump copy file sequence number on that tape.

Having this information allows installations to create recovery jobs for a particular primary or ML1 volume.

This information could be used to create a DSS batch restore job or could be used to create an HSM RECOVER command.

The advantage of creating a DSS batch job is that you can process multiple restores in parallel whereas the HSM volume recovery processing is single threaded.



Collocate ML2 for ABARS

ML2

✓Collocate migrated data sets for same aggregate together to reduce mounts during ABACKUP

ABACKUP xyz
PROCESSONLY(ML2)

ABARS
Tape

ML2

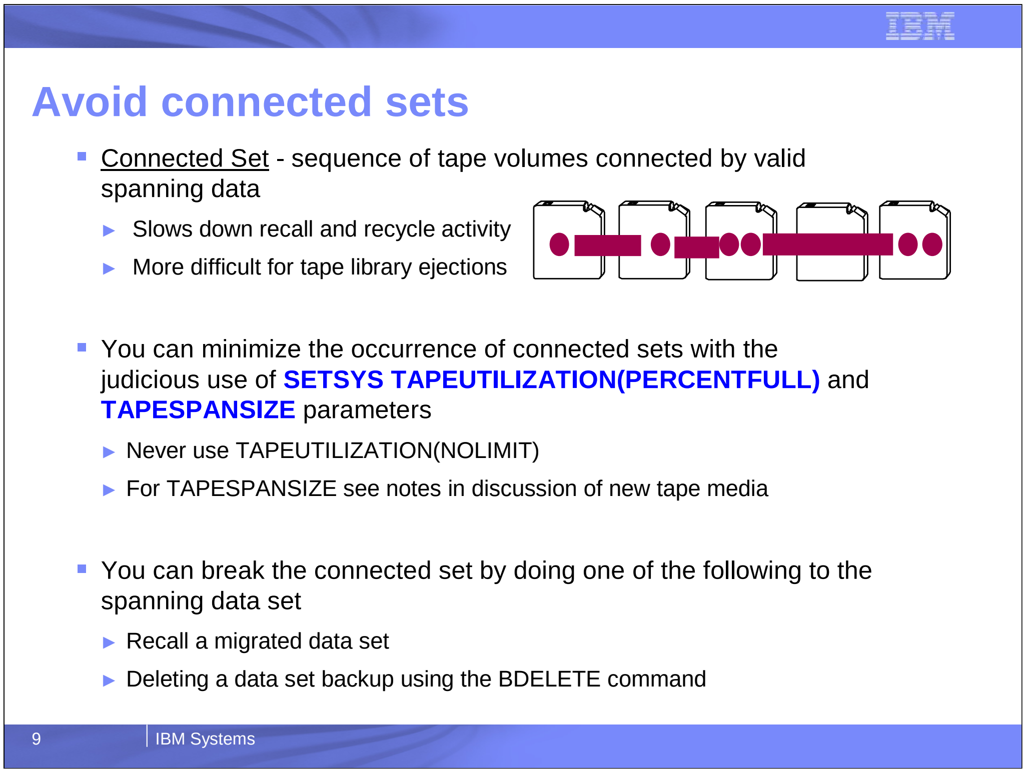
ARECOVER AGG(xyz)
DSCONFLICT(REPLACE)

If installations are using ABARS to backup ML2 tape data sets, over time it can take more and more ML2 tape mounts to backup all ML2 data sets defined to an aggregate.

This slide shows how installations can use the ABACKUP command with the PROCESSONLY(ML2) parameter to backup only those data sets in the aggregate that are on ML2 tape.

They can then turn around on the same system in which they performed the backup and issue an ARECOVER command with the DSCONFLICT(REPLACE) option.

This will cause all ML2 tape data sets to be restored to a minimum number of ML2 tape cartridges, thus collocating the ML2 data for a particular aggregate to a minimum number of cartridges.
The old ML2 data sets will be deleted and the MCDS updated to reflect the new location of the ML2 data sets being restored
The next time a normal ABACKUP is run for that aggregate, all the ML2 data sets will be in a single set of cartridges and perhaps all on 1 cartridge



Avoid connected sets

- Connected Set - sequence of tape volumes connected by valid spanning data
  - Slows down recall and recycle activity
  - More difficult for tape library ejections
- You can minimize the occurrence of connected sets with the judicious use of **SETSYS TAPEUTILIZATION(PERCENTFULL)** and **TAPESPANSIZE** parameters
  - Never use TAPEUTILIZATION(NOLIMIT)
  - For TAPESPANSIZE see notes in discussion of new tape media
- You can break the connected set by doing one of the following to the spanning data set
  - Recall a migrated data set
  - Deleting a data set backup using the BDELETE command

This slide indicates to installations that they should try to avoid what is called "connected sets"

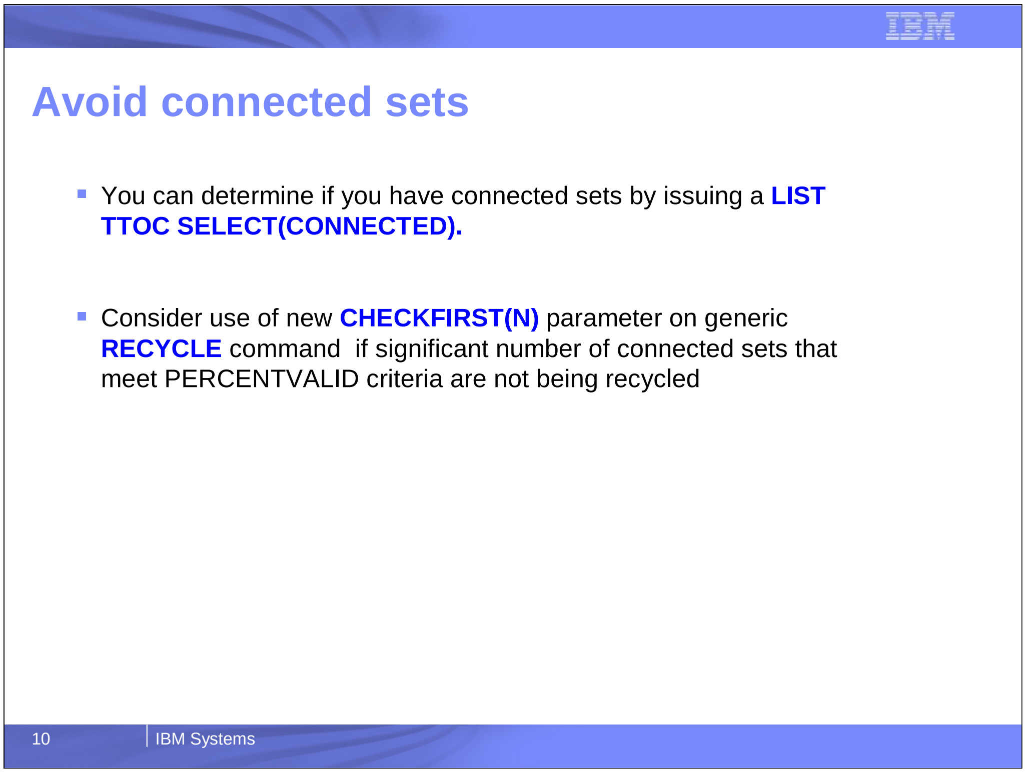
Connected sets are created when a data set spans from 1 tape to another. This is not really a problem for data set sizes which are large compared to the capacity of a tape but it is not good to have small data sets span tapes.

When DFSMShsm generic recycle processing takes place it computes the average percentage of valid data on the connected set rather than just a single volume. It is only when the average of the connected set meets the percent valid criteria that the volumes are recycled.

Also recalling or recovering data that spans tapes requires mounting of multiple tapes

Judicious use of SETSYS TAPEUTILIZATION(PERCENTFULL) and TAPESPANSIZE can help in reducing the occurrence of connected sets
Never use PERCENTFULL of NOLIMIT - this will cause HSM to not perform any type of tape limiting and could result in long chains of connected sets.

Set TAPESPANSIZE based on the media capacity. The larger the overall media capacity the larger this value should be.
This foil also shows how existing connected sets can be broken



Avoid connected sets

- You can determine if you have connected sets by issuing a **LIST TTOC SELECT(CONNECTED).**
- Consider use of new **CHECKFIRST(N)** parameter on generic **RECYCLE** command if significant number of connected sets that meet PERCENTVALID criteria are not being recycled

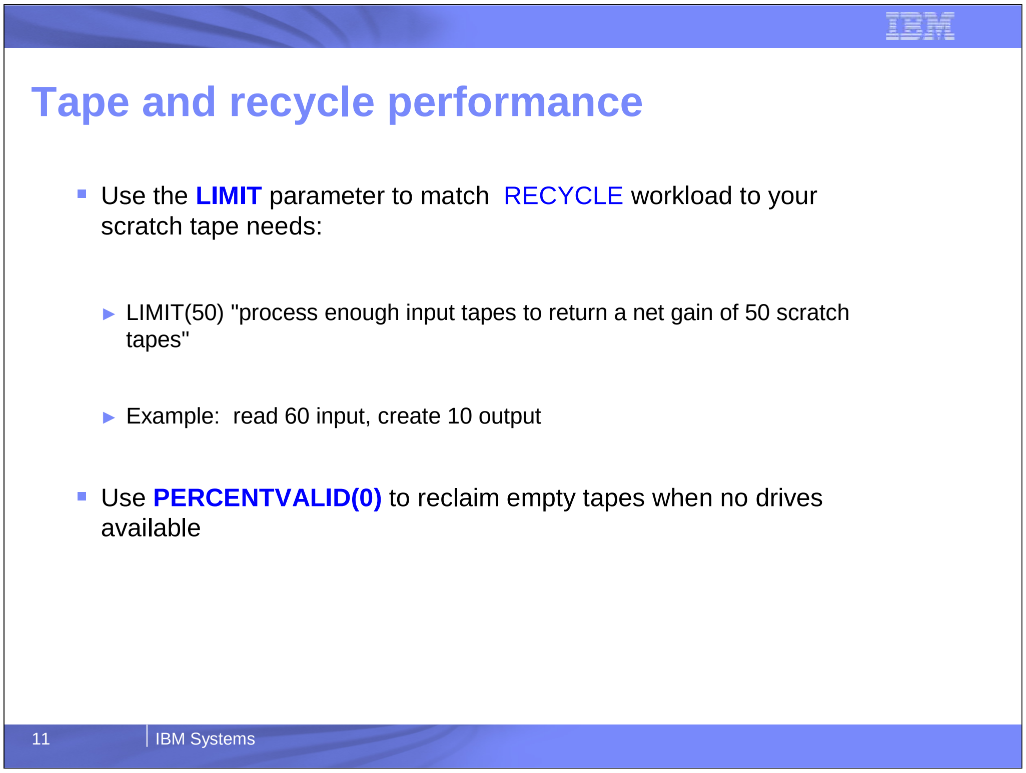
Installations can retrieve a list of their connected sets by performing a LIST TTOC SELECT(CONNECTED)

Prior to DFSMShsm V1R7 DFSMShsm would recycle a connected set only when
The average utilization of all the volumes in the connected set met the percent valid specified on the generic recycle command AND

The first volume in the connected set meets the percent valid criteria

This could leave lots of connected sets not recycled if they experienced high utilization of the first volume in the connected set even though the average utilization of the connected set was low

As of DFSMShsm V1R7, installations can specify CHECKFIRST(N) on the generic recycle command to indicate just to check to see if the average utilization of the connected set meets the percent valid criteria and not require that the first volume meet the criteria
This should result in more volumes being returned to scratch sooner



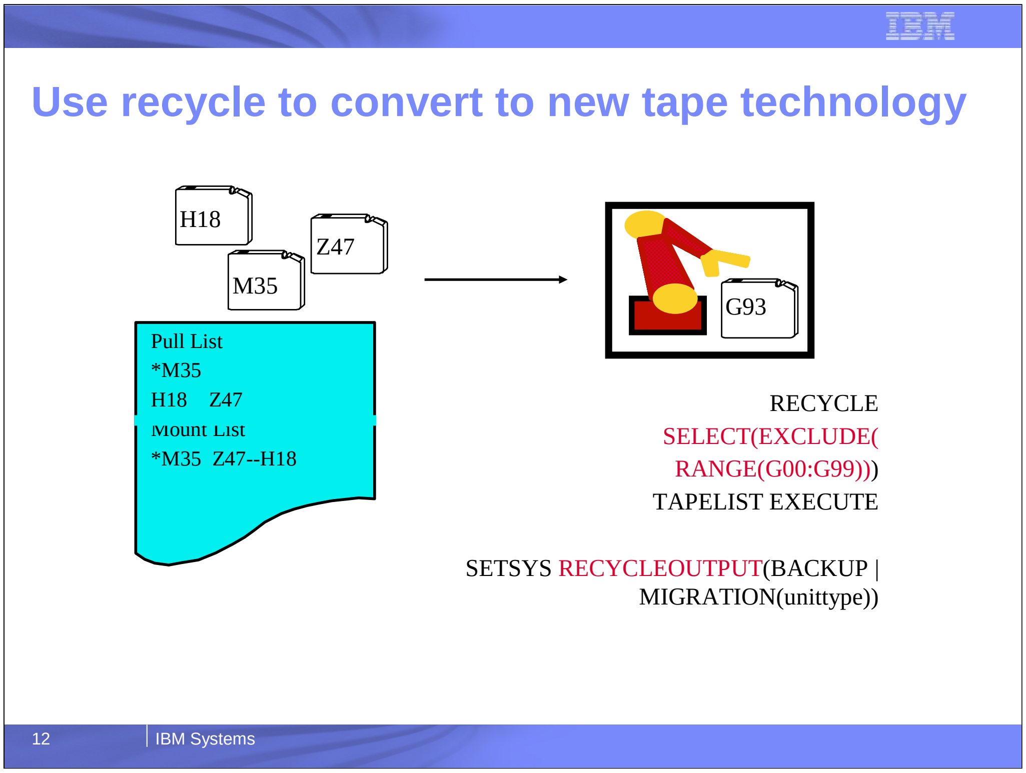
Tape and recycle performance

- Use the **LIMIT** parameter to match RECYCLE workload to your scratch tape needs:
  - LIMIT(50) "process enough input tapes to return a net gain of 50 scratch tapes"
  - Example: read 60 input, create 10 output
- Use **PERCENTVALID(0)** to reclaim empty tapes when no drives available

If an installation knows how many tapes are required to be returned to scratch per invocation of a generic recycle to meet service level agreements it can tell RECYCLE to quit processing once the net number of tapes returned to scratch reaches that limit

This can be accomplished by using the LIMIT parameter on the generic RECYCLE command

Also, if you have a case where you know you have tapes that have no valid data, you can issue the RECYCLE command with PERCENTVALID(0) specified and those tapes will be recycled and there will be no tape mounts required.



Use recycle to convert to new tape technology

H18
Z47
M35
G93

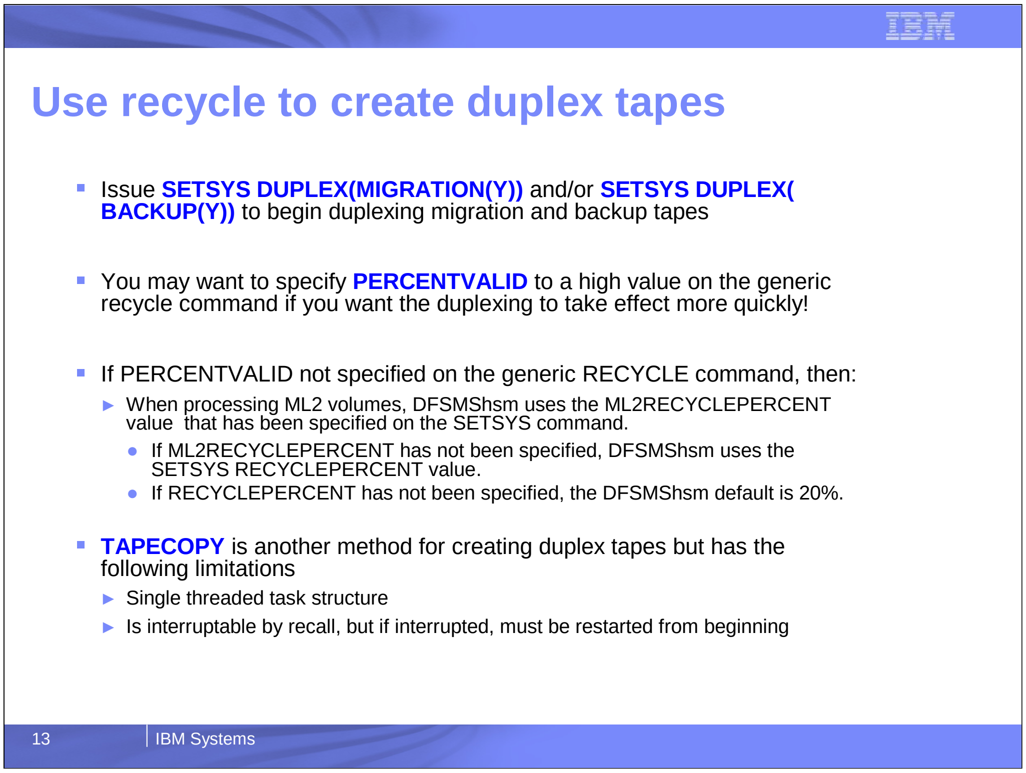
Pull List
*M35
H18 Z47
Mount List
*M35 Z47--H18

```
RECYCLE
SELECT(EXCLUDE(
RANGE(G00:G99)))
TAPELIST EXECUTE
```

```
SETSYS RECYCLEOUTPUT(BACKUP |
MIGRATION(unittype))
```

RECYCLE can be an effective tool to convert from older technology tapes to newer technology tapes for DFSMShsm migration and backup tapes

This can be accomplished by excluding newer technology tapes from recycle input
Use the SELECT(EXCLUDE(RANGE parameters if the RECYCLE command to exclude the range of volsers that are used for the newer technology tapes
Then, installations should specify the newer technology tape unit type in the SETSYS RECYCLEOUTPUT command
When the RECYCLE command is issued, ti will then only select the old technology tape volumes as input and will recycle those volumes to the new technology unit typesi

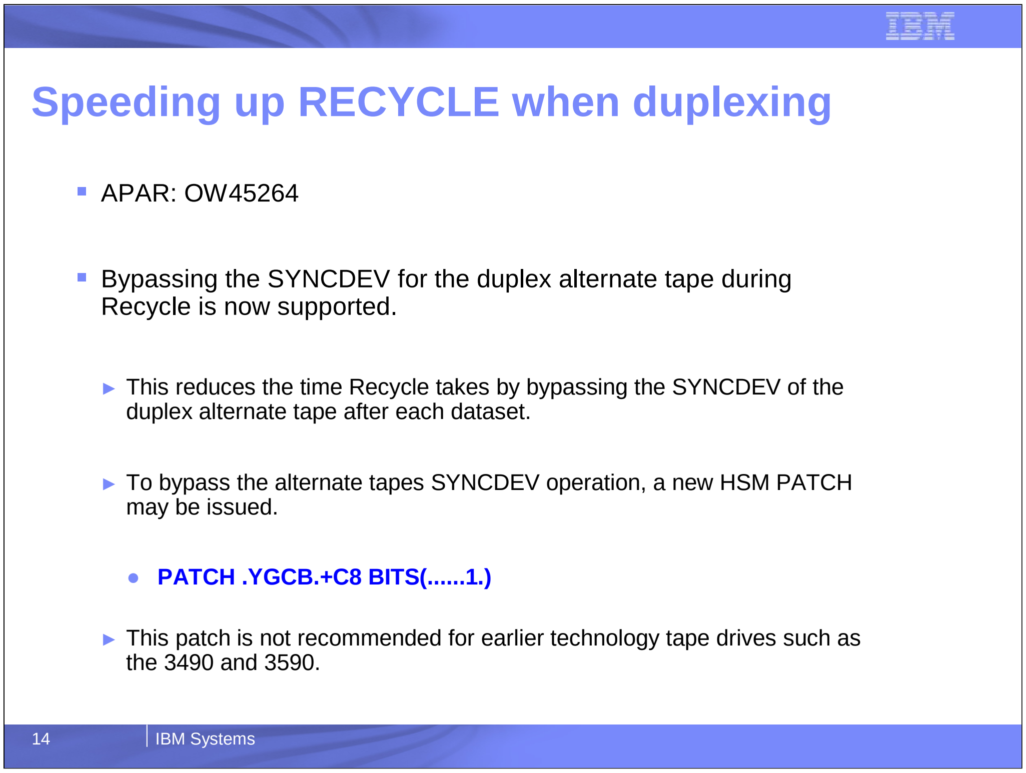


Use recycle to create duplex tapes

- Issue **SETSYS DUPLEX(MIGRATION(Y))** and/or **SETSYS DUPLEX(BACKUP(Y))** to begin duplexing migration and backup tapes
- You may want to specify **PERCENTVALID** to a high value on the generic recycle command if you want the duplexing to take effect more quickly!
- If PERCENTVALID not specified on the generic RECYCLE command, then:
  - When processing ML2 volumes, DFSMShsm uses the ML2RECYCLEPERCENT value that has been specified on the SETSYS command.
    - If ML2RECYCLEPERCENT has not been specified, DFSMShsm uses the SETSYS RECYCLEPERCENT value.
    - If RECYCLEPERCENT has not been specified, the DFSMShsm default is 20%.
- **TAPECOPY** is another method for creating duplex tapes but has the following limitations
  - Single threaded task structure
  - Is interruptable by recall, but if interrupted, must be restarted from beginning

If installations decide they want to use DFSMShsm tape duplexing and they want to duplex all their existing DFSMShsm tapes they can accomplish this by issuing the SETSYS DUPLEX(MIGRATION(Y)) or SETSYS DUPLEX BACKUP(Y)) commands and then recycle all their existing DFSMShsm migration and backup volumes.

In order to recycle all the tape volumes the PERCENTVALID parameter on the RECYCLE command should be set to 100.
If installations wish to do this in smaller increments they can use a smaller value for PERCENTVALID



Speeding up RECYCLE when duplexing

- APAR: OW45264
- Bypassing the SYNCDEV for the duplex alternate tape during Recycle is now supported.
  - This reduces the time Recycle takes by bypassing the SYNCDEV of the duplex alternate tape after each dataset.
  - To bypass the alternate tapes SYNCDEV operation, a new HSM PATCH may be issued.
    - **PATCH .YGCB.+C8 BITS(......1.)**
  - This patch is not recommended for earlier technology tape drives such as the 3490 and 3590.

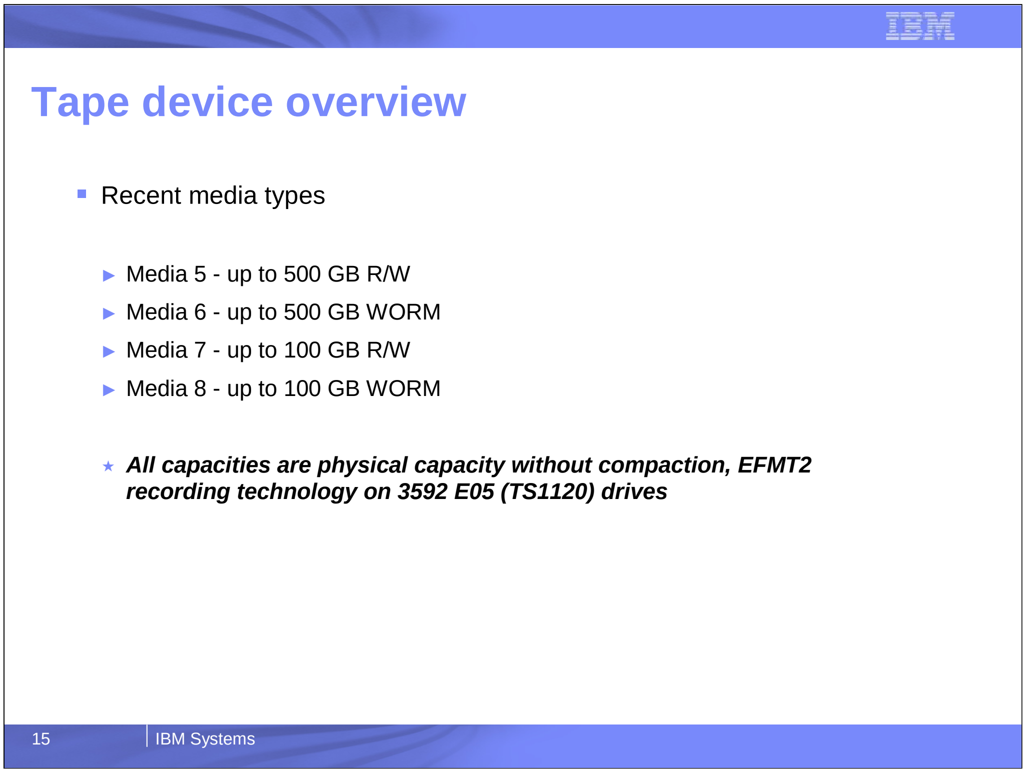
Migration elapsed times rise dramatically when syncs are done to two 3592-J tape drives simultaneously. The effect is most noticeable when migrating small data sets.

Customers can now suppress SYNCDEV on the alternate tape during duplex migration. Syncs for the migrationalternate tape can be turned off via:

PATCH .MCVT.+196 BITS(..1.....)
and back on via:
PATCH .MCVT.+196 BITS(..0.....)

The default is to do syncs on the alternate tapes.

This patch is not recommended for earlier technology tape drives such as the 3490 and 3590.



Tape device overview

- Recent media types
  - Media 5 - up to 500 GB R/W
  - Media 6 - up to 500 GB WORM
  - Media 7 - up to 100 GB R/W
  - Media 8 - up to 100 GB WORM

★ ***All capacities are physical capacity without compaction, EFMT2 recording technology on 3592 E05 (TS1120) drives***

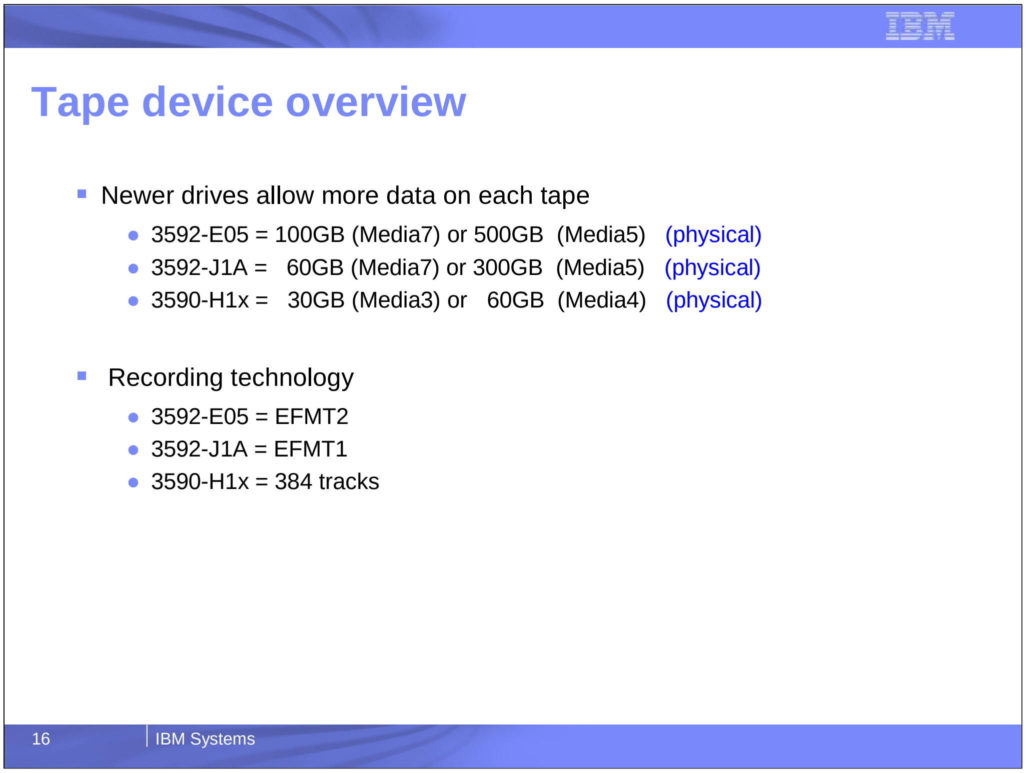
This slides lists the most recent Media types supported by IBM tape technology.

Media 5 and 6 represent 500 GB uncompressed capacity.
With 3:1 compression these capacities can exceed 1 TB
Media 7 and 8 represent shorter length tapes that can be used when increased performance is required when retrieving data from these tapes

WORM - indicates Write Once Read Many

R/W - indicated Read Write

These capacities are what you can get using EFMT2 recording technology on 3592 E05 drives

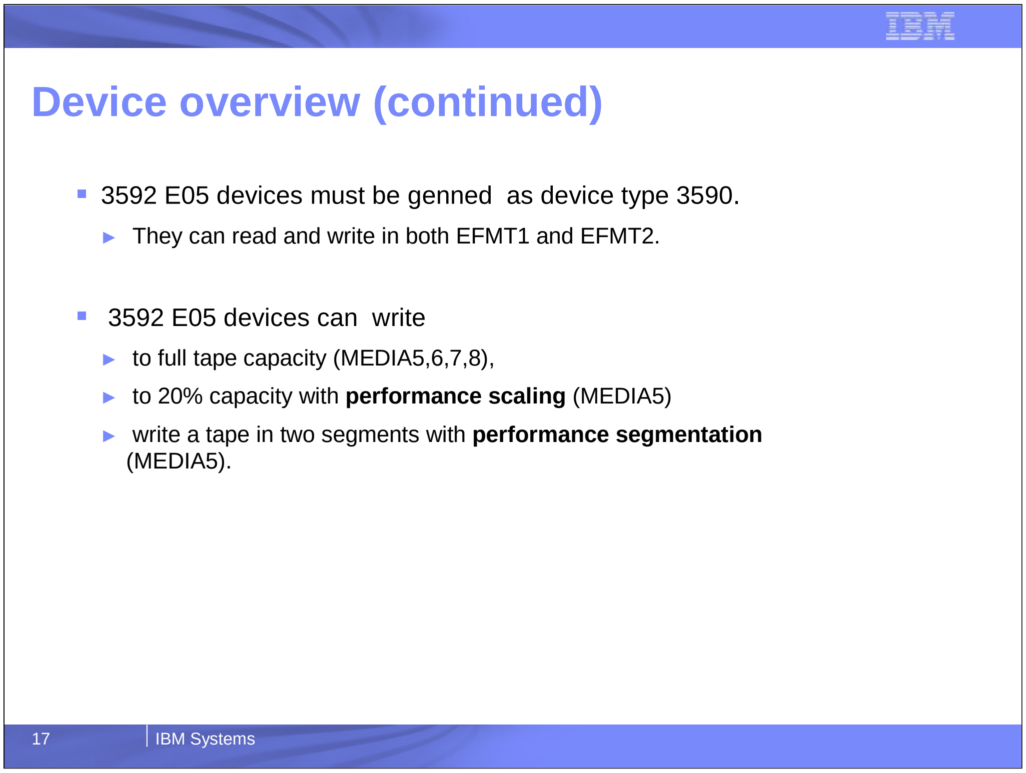


Tape device overview

- Newer drives allow more data on each tape
  - 3592-E05 = 100GB (Media7) or 500GB (Media5) (physical)
  - 3592-J1A = 60GB (Media7) or 300GB (Media5) (physical)
  - 3590-H1x = 30GB (Media3) or 60GB (Media4) (physical)
- Recording technology
  - 3592-E05 = EFMT2
  - 3592-J1A = EFMT1
  - 3590-H1x = 384 tracks

This slide is pretty self explanatory. It shows the various capacities of the newer media based on the tape drive technology

This also lists the various recording technologies available on the various tape drives.



Device overview (continued)

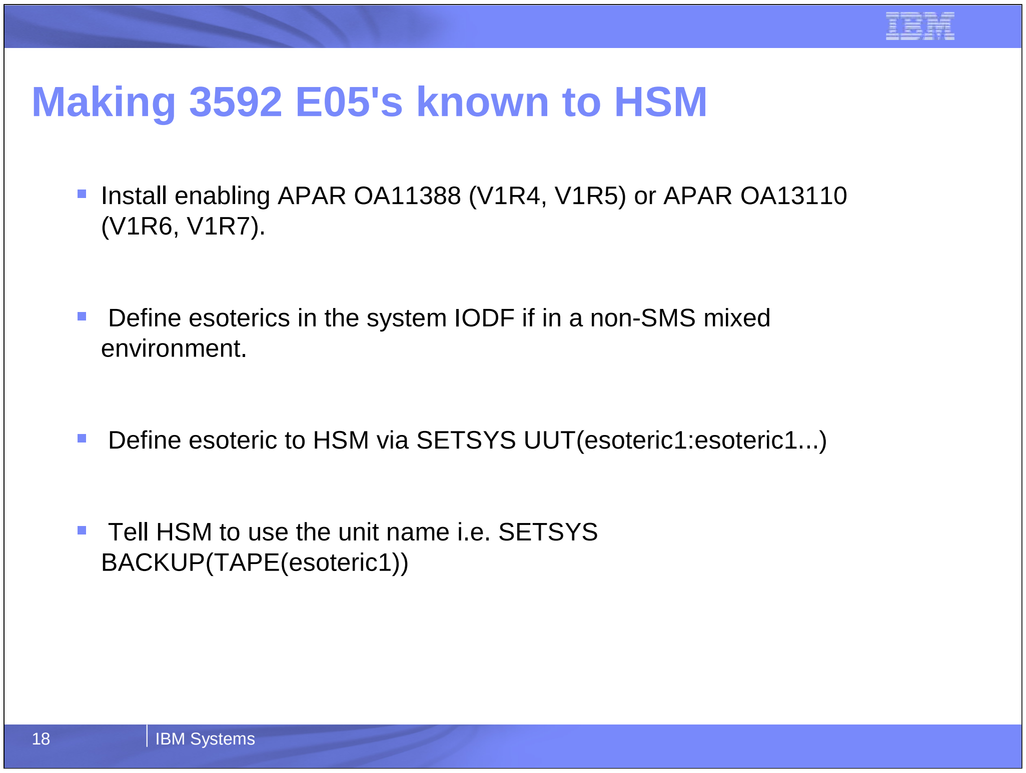
- 3592 E05 devices must be genned as device type 3590.
  - They can read and write in both EFMT1 and EFMT2.

- 3592 E05 devices can write
  - to full tape capacity (MEDIA5,6,7,8),
  - to 20% capacity with **performance scaling** (MEDIA5)
  - write a tape in two segments with **performance segmentation** (MEDIA5).

IBM 3592 E05 tape drives must be genned to the system as device type 3590

The E05 drives can read and write in both EFMT1 and EFMT2 recording technologies

The IBM 3592 E05 drive supports performance scaling for Media 5, which means that only 20% of the tape capacity is used
This also supports performance segmentation which allows the tapes to be segmented into 2 parts.

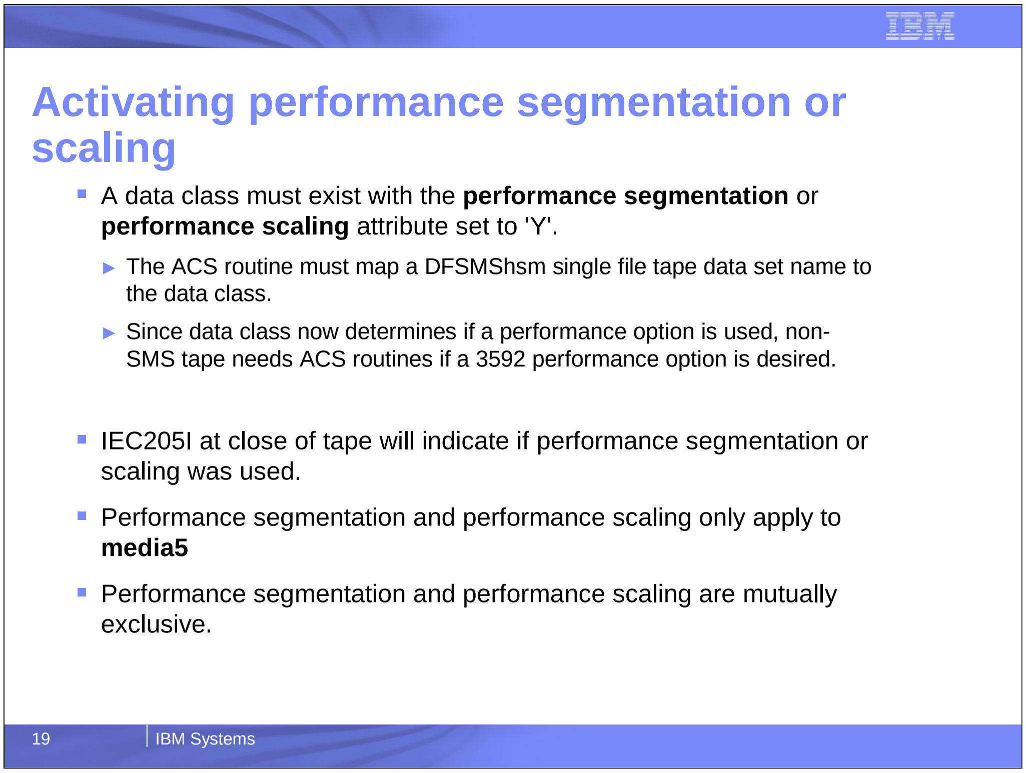


Making 3592 E05's known to HSM

- Install enabling APAR OA11388 (V1R4, V1R5) or APAR OA13110 (V1R6, V1R7).
- Define esoterics in the system IODF if in a non-SMS mixed environment.
- Define esoteric to HSM via SETSYS UUT(esoteric1:esoteric1...)
- Tell HSM to use the unit name i.e. SETSYS BACKUP(TAPE(esoteric1))

In order to take advantage of the new IBM 3592 E05 tape drives in DFSMShsm, installations need to install the enabling APAR that matches their release level

Define tape esoteric unit names in the system IODF in a non-SMS mixed tape environment
This esoteric then needs to be defined to DFSMShsm in its User Unit Table
This is accomplished by specifying SETSYS UUT using the new esoteric unit name
Once defined to the UUT, then the esoteric can be used as the unit name in other DFSMShsm commands



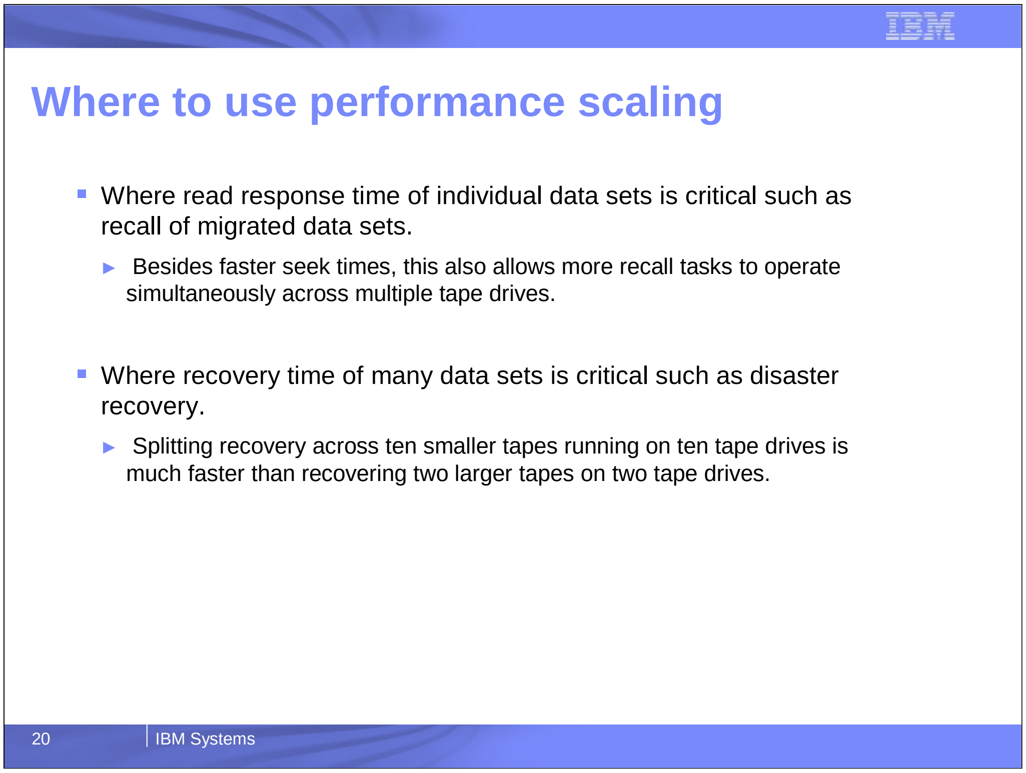
Activating performance segmentation or scaling

- A data class must exist with the **performance segmentation** or **performance scaling** attribute set to 'Y'.
  - The ACS routine must map a DFSMShsm single file tape data set name to the data class.
  - Since data class now determines if a performance option is used, non-SMS tape needs ACS routines if a 3592 performance option is desired.
- IEC205I at close of tape will indicate if performance segmentation or scaling was used.
- Performance segmentation and performance scaling only apply to **media5**
- Performance segmentation and performance scaling are mutually exclusive.

Installations can request that tapes take advantage of performance segmentation or performance scaling by setting these attributes to "Y in the data class assigned when the device is first allocated

All DFSMShsm tapes have a tape data set name that can be used in the data class ACS routines to direct the tape data set to a specific data class
The data class can be used for SMS and non-SMS tapes. If installations want to take advantage of the performance options, they need to assign non-SMS tapes to an appropriate data class
An IEC205I message will be issued when a tape is closed that will indicate if performance scaling or segmentation was used

Performance segmentation and scaling only apply to Media5 tapes
These settings are also mutually exclusive

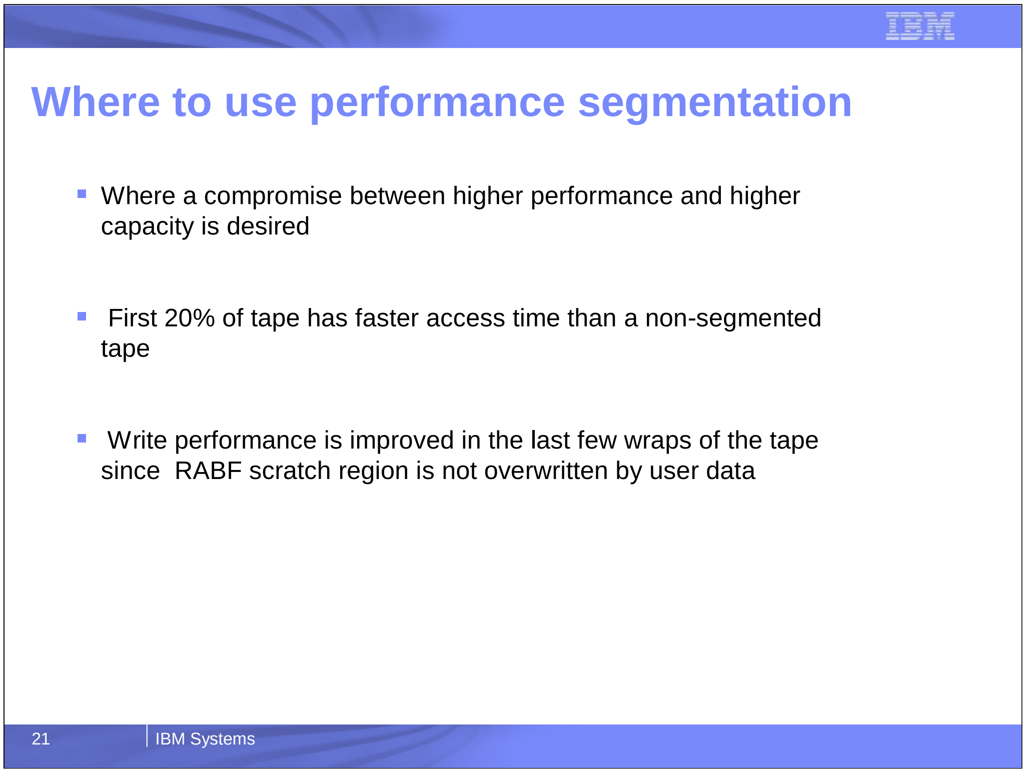


Where to use performance scaling

- Where read response time of individual data sets is critical such as recall of migrated data sets.
  - Besides faster seek times, this also allows more recall tasks to operate simultaneously across multiple tape drives.
- Where recovery time of many data sets is critical such as disaster recovery.
  - Splitting recovery across ten smaller tapes running on ten tape drives is much faster than recovering two larger tapes on two tape drives.

Performance scaling should be used whenever faster read access to data on the tape is required.
This can be effective in speeding up recall time and disaster recovery times.

Because there is less data on a performance scaled tape the migration/backup data can be spread over more tapes thus allowing for greater parallelism during recall or recovery



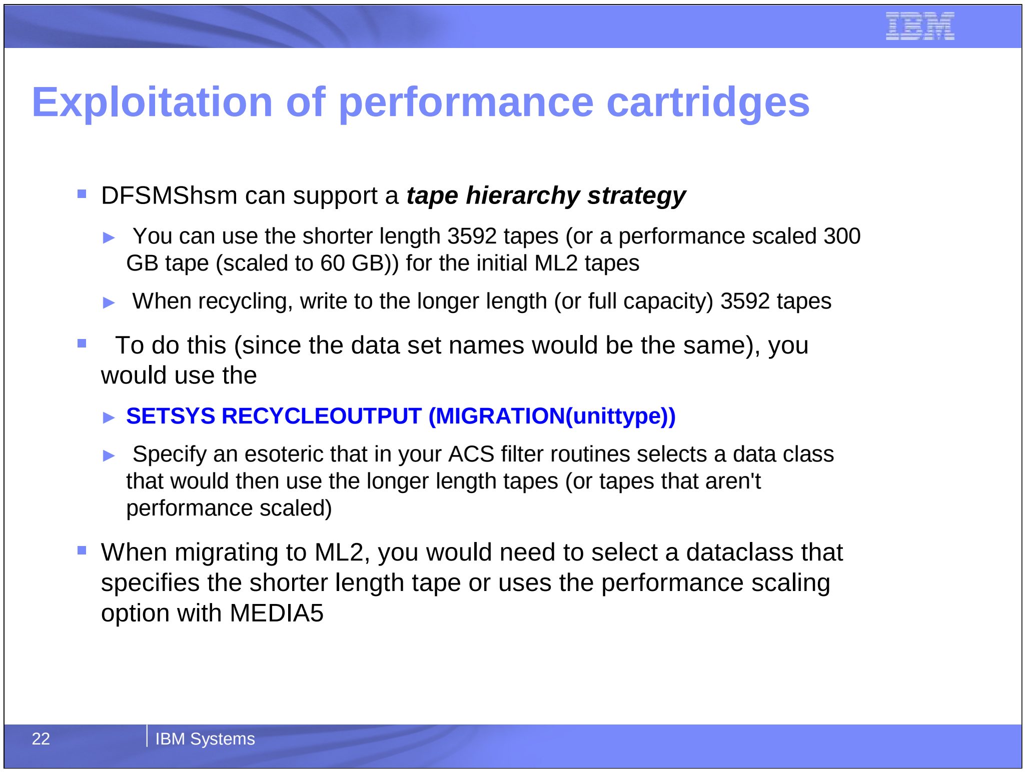
Where to use performance segmentation

- Where a compromise between higher performance and higher capacity is desired
- First 20% of tape has faster access time than a non-segmented tape
- Write performance is improved in the last few wraps of the tape since RABF scratch region is not overwritten by user data

Performance Segmentation can be used where a compromise between higher performance and higher capacity is desired
The first 20% of the tape has faster access time than non-segmented tape, but about the same access time for the last 80% of the tape.

You cannot specifically direct data on a performance segmented tape to the high performance segment.

Once the tape fills beyond 20% of capacity, the remaining data on the tape is automatically written to the lower performance segment



Exploitation of performance cartridges

- DFSMShsm can support a ***tape hierarchy strategy***
  - You can use the shorter length 3592 tapes (or a performance scaled 300 GB tape (scaled to 60 GB)) for the initial ML2 tapes
  - When recycling, write to the longer length (or full capacity) 3592 tapes
- To do this (since the data set names would be the same), you would use the
  - **SETSYS RECYCLEOUTPUT (MIGRATION(unittype))**
  - Specify an esoteric that in your ACS filter routines selects a data class that would then use the longer length tapes (or tapes that aren't performance scaled)
- When migrating to ML2, you would need to select a dataclass that specifies the shorter length tape or uses the performance scaling option with MEDIA5

This slide explains how installations can create a tape hierarchy for migration. They would initially migrate to a performance scaled tape or a shorter length tape.

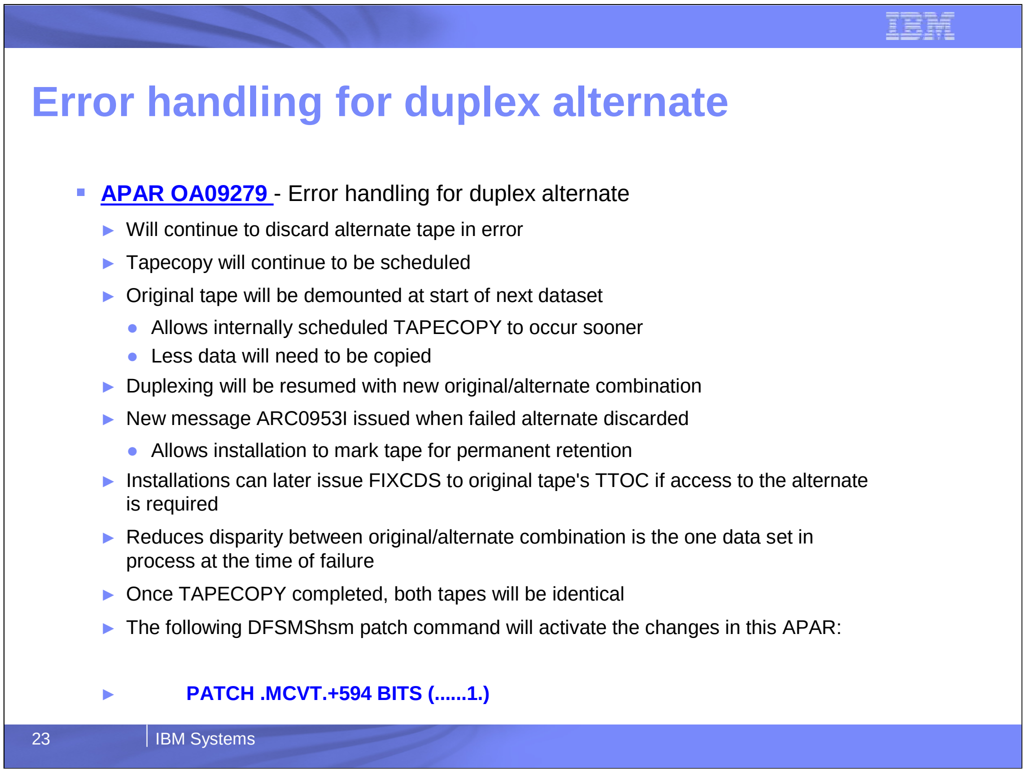
After the data has aged and the original tape is eligible for recycle, then the migration data could to moved to a non-performance scaled tape or a higher capacity cartridge.

The thought process is that as data ages the likelihood of being recalled would decrease over time, so speed of recall would not be as important, however if the data does get recalled in the short term it could be retrieved from the performance scaled or shorter tape more quickly.

To support such a strategy

The ML2 tape data set name should be in the ACS selection routines to direct the ML2 tape to a performance scaled or shorter length tape

Installations should specify the unittype in the SETSYS RECYCLEOUTPUT(MIGRATION(unittype)) to longer length tape. The unit name will be passed to the data class ACS routine which could use a combination of data set name and unit name to direct these allocations to longer length tapes



Error handling for duplex alternate

- APAR OA09279 - Error handling for duplex alternate
  - Will continue to discard alternate tape in error
  - Tapecopy will continue to be scheduled
  - Original tape will be demounted at start of next dataset
    - Allows internally scheduled TAPECOPY to occur sooner
    - Less data will need to be copied
  - Duplexing will be resumed with new original/alternate combination
  - New message ARC0953I issued when failed alternate discarded
    - Allows installation to mark tape for permanent retention
  - Installations can later issue FIXCDS to original tape's TTOC if access to the alternate is required
  - Reduces disparity between original/alternate combination is the one data set in process at the time of failure
  - Once TAPECOPY completed, both tapes will be identical
  - The following DFSMShsm patch command will activate the changes in this APAR:
  - **PATCH .MCVT.+594 BITS (......1.)**

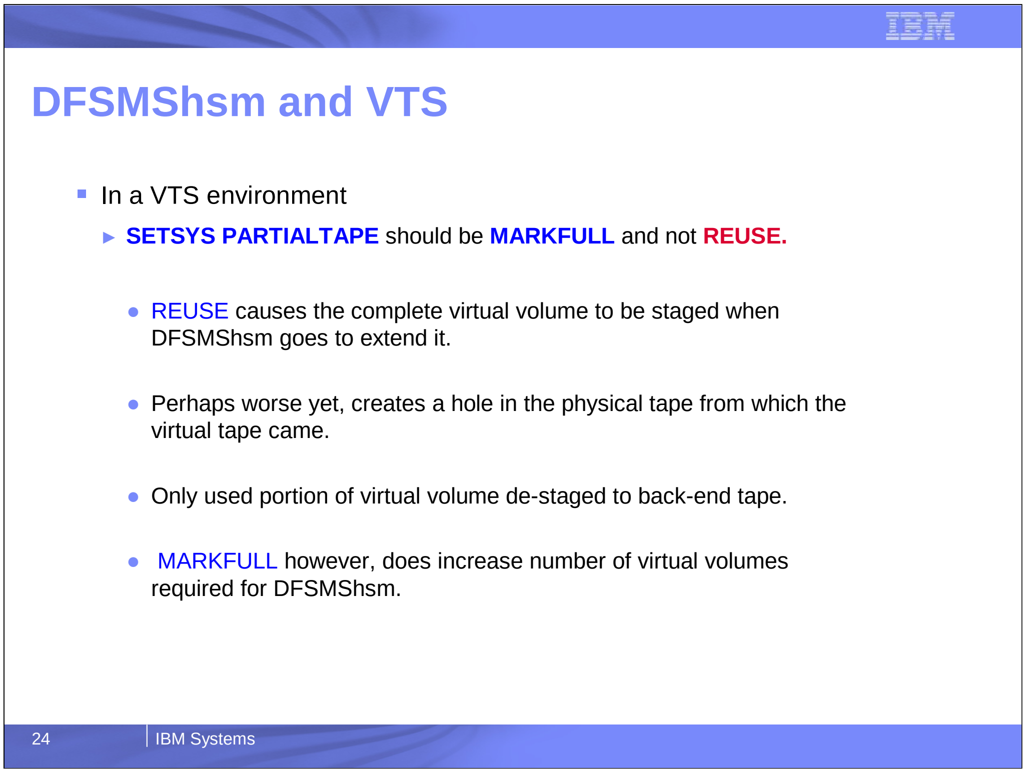
This APAR provides relief for the single-copy exposure on ML2 tapes during the time between the suspension of duplexing and the completion of an internal TAPECOPY.

These changes require invocation via a DFSMShsm PATCH command to limit the changes to those customers for whom maintaining a DUPLEX relationship for DFSMShsm ML2 tapes is of a time-critical nature. The following DFSMShsm patch command will activate the changes in this APAR:

```
patch .mcvt.+594 bits(......1.)
```

The following changes will reduce the lag time between the occurrence of the error and the issuance of the internal TAPECOPY command, as well as reducing the amount of data that must be copied.

For ML2 tapes, after an error causes duplexing to stop and the original tape continues, DFSMShsm will stop using the original tape at the completion of the current dataset. Before continuing, a new pair of output tapes will be selected to maintain the duplex relationship. As before, the duplex tape that received the error will be released and an internal TAPECOPY scheduled to copy the original tape.



DFSMShsm and VTS

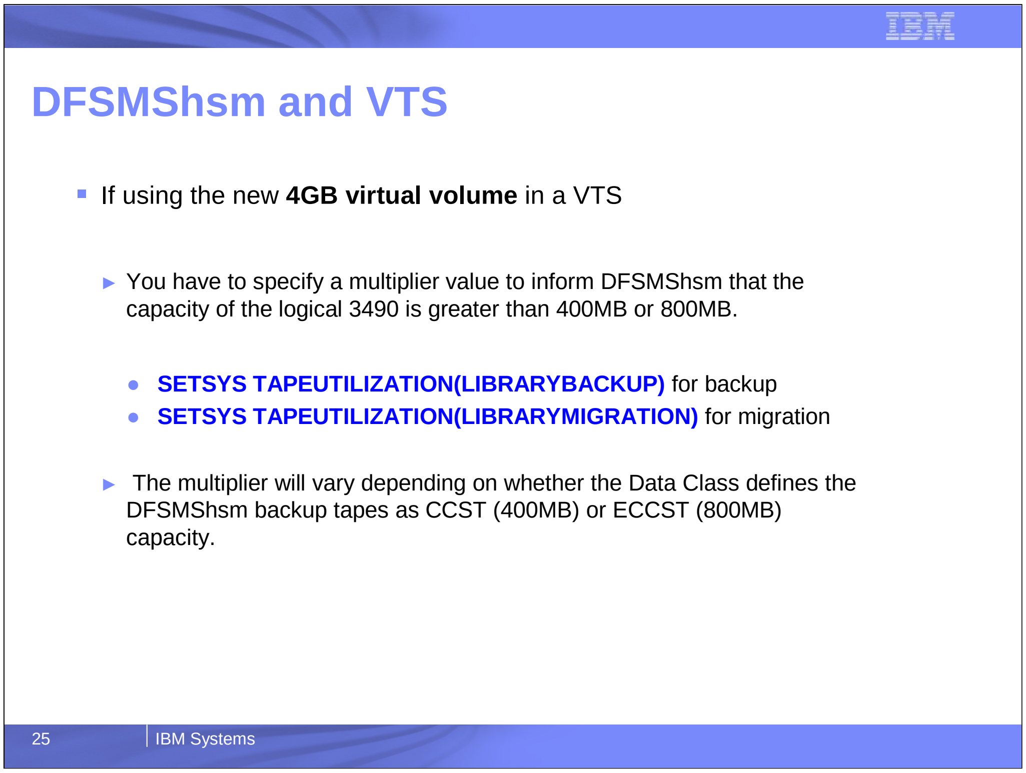
- In a VTS environment
  - **SETSYS PARTIALTAPE** should be **MARKFULL** and not **REUSE.**
    - REUSE causes the complete virtual volume to be staged when DFSMShsm goes to extend it.
    - Perhaps worse yet, creates a hole in the physical tape from which the virtual tape came.
    - Only used portion of virtual volume de-staged to back-end tape.
    - MARKFULL however, does increase number of virtual volumes required for DFSMShsm.

When using DFSMShsm in a VTS environment, installations should use the MARKFULL option of SETSYS PARTIALTAPE rather than the REUSE option. If installations use the REUSE attribute it means that if the virtual volume is not in the VTS cache when HSM attempts to extend the tape, then the VTS must stage the entire volume from the backend container volume.

When staged, this creates a hole in the backend container volume, which causes more frequent VTS reclamations to occur.

If installations use the MARKFULL option of SETSYS PARTIALTAPE, only the used portion of the virtual volume will get de-staged from VTS cache to the back-end tape.
Less data will have to be recalled into VTS cache if there is a recall request for a data set on a back-end volume.

MARKFULL will increase the number of virtual volumes required for DFSMShsm



DFSMShsm and VTS

- If using the new **4GB virtual volume** in a VTS
  - You have to specify a multiplier value to inform DFSMShsm that the capacity of the logical 3490 is greater than 400MB or 800MB.
    - **SETSYS TAPEUTILIZATION(LIBRARYBACKUP)** for backup
    - **SETSYS TAPEUTILIZATION(LIBRARYMIGRATION)** for migration
  - The multiplier will vary depending on whether the Data Class defines the DFSMShsm backup tapes as CCST (400MB) or ECCST (800MB) capacity.

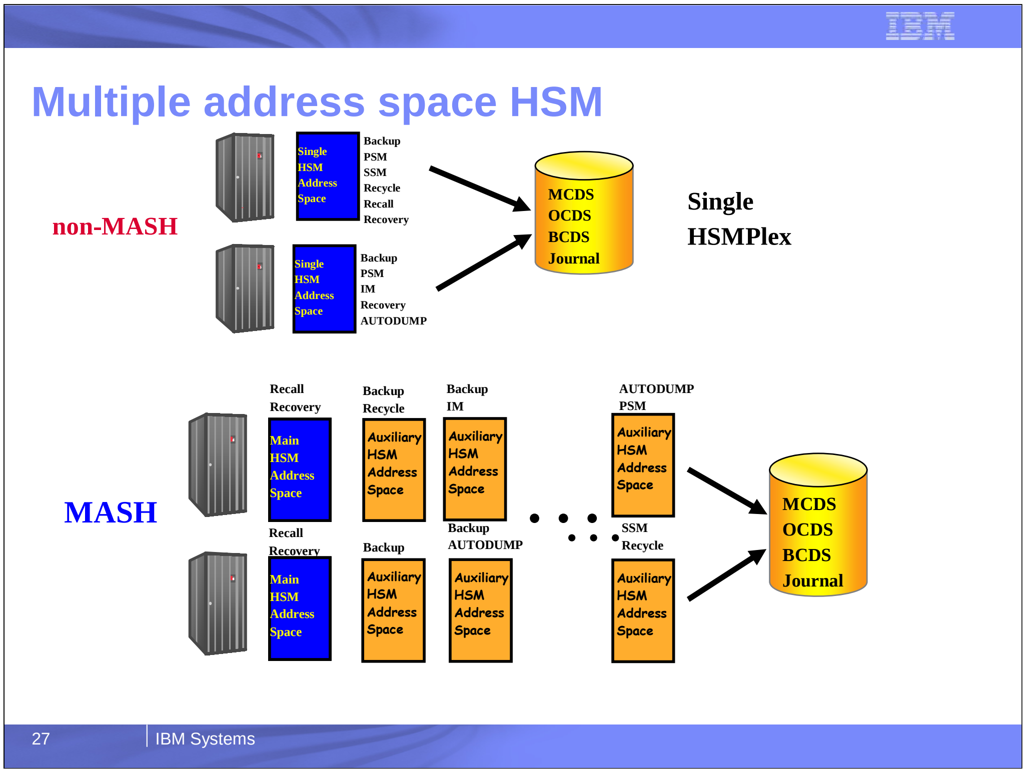
If an installation wishes to take advantage of the newer 4GB virtual volumes in a VTS they must set their SETSYS TAPEUTILIZATION settings with a multiplier

For instance if they defined the data class to allocate the backup tapes as Media1 (400MB) capacity, they would set the TAPEUTILIZATION to 1000 to get to 4GB.
1000 would actually fill the tape to capacity and would be prone to tape spanning, we would really recommend a value such as 997.

For Media2 (800MB), the suggested tape utilization would be something in the neighborhood of 497-499).

Agenda: Miscellaneous tips and hints

- Managing and tuning the HSM tape environment
- **Miscellaneous tips and hints**



Multiple address space HSM

non-MASH

Single HSM Address Space — Backup, PSM, SSM, Recycle, Recall, Recovery

Single HSM Address Space — Backup, PSM, IM, Recovery, AUTODUMP

MCDS OCDS BCDS Journal

Single HSMPlex

MASH

Recall Recovery — Main HSM Address Space

Backup Recycle — Auxiliary HSM Address Space

Backup IM — Auxiliary HSM Address Space

AUTODUMP PSM — Auxiliary HSM Address Space

Recall Recovery — Main HSM Address Space

Backup — Auxiliary HSM Address Space

Backup AUTODUMP — Auxiliary HSM Address Space

SSM Recycle — Auxiliary HSM Address Space

MCDS OCDS BCDS Journal

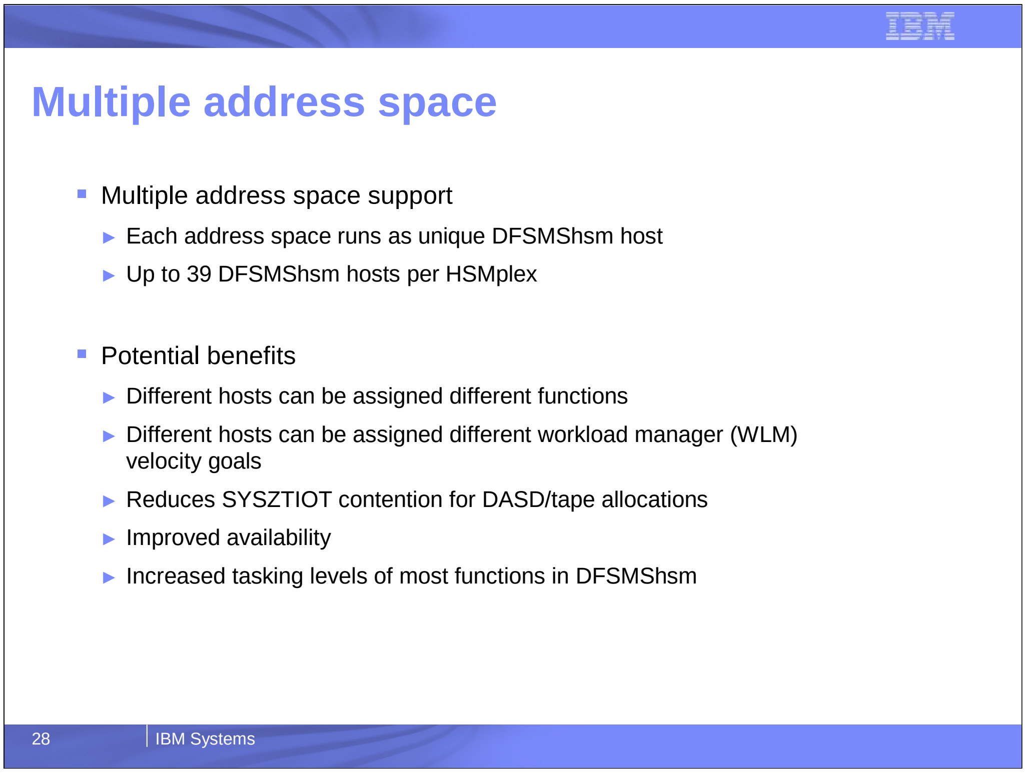
This slide depicts DFSMShsm in a non-MASH environment as compared to DFSMShsm in a MASH environment

Without MASH, there is a limit of 1 DFSMShsm primary address space per z/OS image

MASH stands for **M**ulti-**A**ddress **S**pace **H**SM

MASH allows multiple instances of DFSMShsm in a single z/OS image, up to 39.

All DFSMShsm's in a MASH environment are part of a single HSMplex sharing a single set of DFSMShsm CDSs and Journal



Multiple address space

- Multiple address space support
  - Each address space runs as unique DFSMShsm host
  - Up to 39 DFSMShsm hosts per HSMplex
- Potential benefits
  - Different hosts can be assigned different functions
  - Different hosts can be assigned different workload manager (WLM) velocity goals
  - Reduces SYSZTIOT contention for DASD/tape allocations
  - Improved availability
  - Increased tasking levels of most functions in DFSMShsm

This slide is mostly self explanatory. MASH allows for up to 39 HSM address spaces to be active on a single instance of z/OS. All instances of HSM must be within a single HSMplex.

There are a number of potential benefits to MASH including Allowing different HSMs to be assigned specific functions

Setting different WLM Velocity goals for different DFSMShsm hosts

Because there is a single SYSZTIOT resource per address space, MASH can provide some SYSZTIOT constraint relief

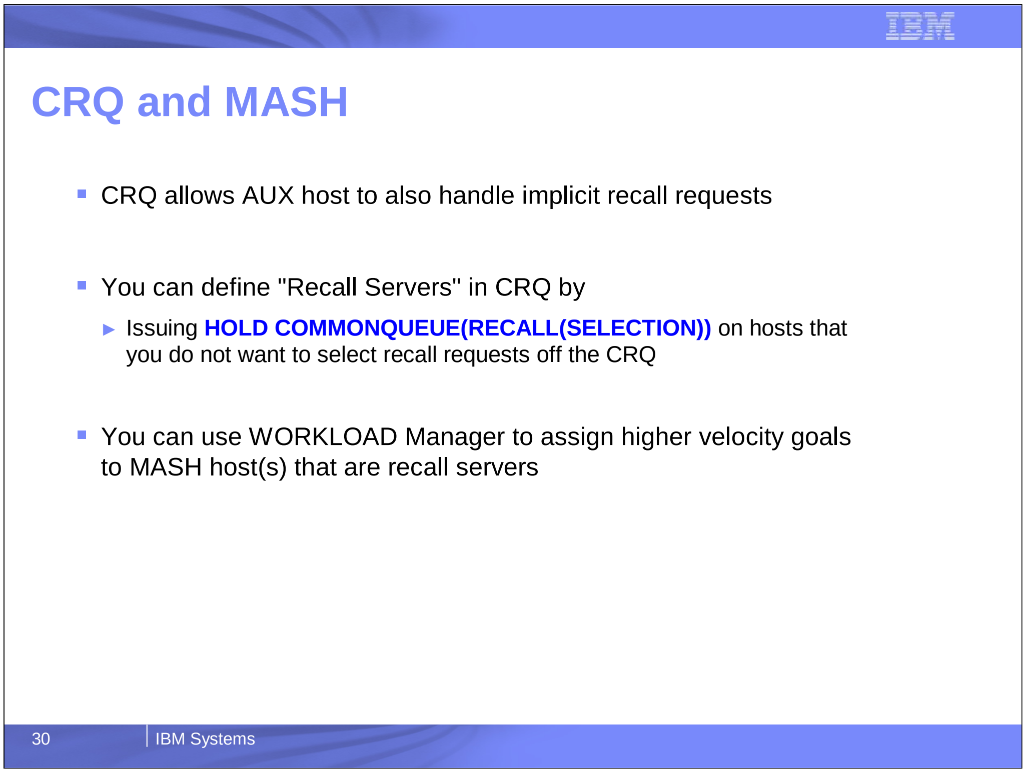
Improved availability, especially when combined with CRQ

Allows HSM to scale much more massively, since each host can be set to the maximum number of tasks per function, multiplied by up to 39 hosts.

MAIN versus AUX hosts

- HOSTMODE=**MAIN**
  - One per system image
  - Handles implicit recalls, HSEND commands, batch and TSO end user requests
  - Manages ABARS secondary address spaces
  - This is the default HOSTMODE

- HOSTMODE=**AUX**
  - Multiple per system image
  - Scheduled Automatic Functions
    - PSM, SSM, Interval
    - Autobackup
    - AUTODUMP
  - Explicit requests using Console Commands, Netview or EMCS

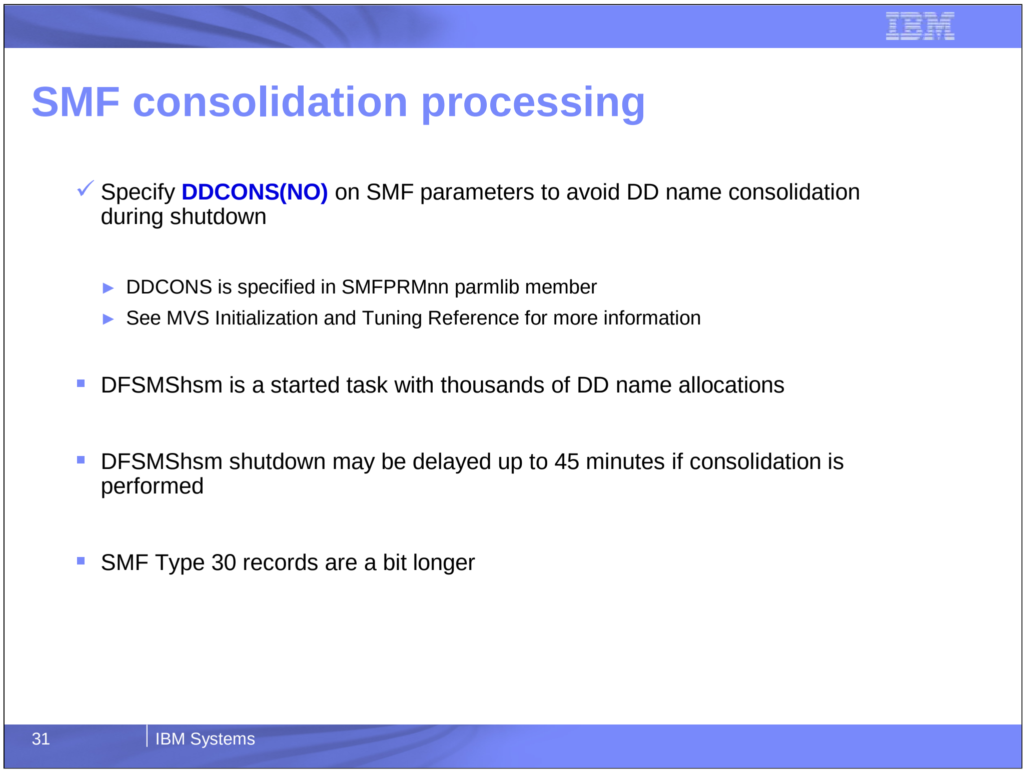


CRQ and MASH

- CRQ allows AUX host to also handle implicit recall requests
- You can define "Recall Servers" in CRQ by
  - Issuing **HOLD COMMONQUEUE(RECALL(SELECTION))** on hosts that you do not want to select recall requests off the CRQ
- You can use WORKLOAD Manager to assign higher velocity goals to MASH host(s) that are recall servers

The slide details some of the synergy between CRQ and MASH
One of the limitations of an AUX host is that they do not handle implicit recall requests, where a TSO user or batch job references a migrated data set causing a recall. This occurs because there is no method to direct these requests to an AUX host. However in a CRQ environment all recall requests are placed in the coupling facility structures which allow all participating hosts to see and select these recall requests.

You can establish recall servers in a CRQ/MASH environment so that all hosts can place recall requests on the CRQ but only certain hosts process recall requests
This can be accomplished by specifying HOLD COMMONQUEUE(RECALL(SELECTION))



SMF consolidation processing

✓ Specify **DDCONS(NO)** on SMF parameters to avoid DD name consolidation during shutdown

- ▸ DDCONS is specified in SMFPRMnn parmlib member
- ▸ See MVS Initialization and Tuning Reference for more information

- DFSMShsm is a started task with thousands of DD name allocations
- DFSMShsm shutdown may be delayed up to 45 minutes if consolidation is performed
- SMF Type 30 records are a bit longer

Installations have observed long delays in DFSMShsm shutting down after they issue a DFSMShsm STOP command.

One of the reasons the shutdown can take a long period of time is because SMF is performing consolidation processing for SMF type 30 records

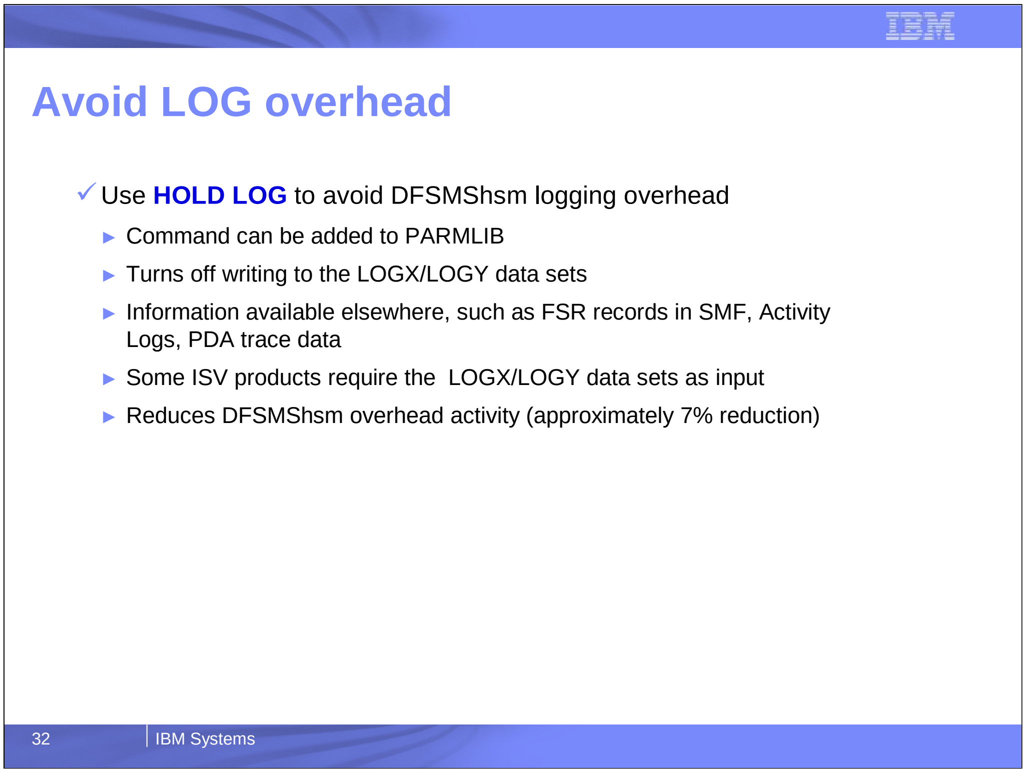
Specifying DDCONS(N) in SMFPRMnn will prevent this

DDCONS {(YES)} {(NO) }
Specifies whether duplicate EXCP entries for type 30 SMF records are to be consolidated.

When DDCONS(YES) is specified, SMF merges the EXCP count for these duplicate entries into one entry if the following information is the same: ddname, Device class, Unit type, Channel address and Unit address

Long-running jobs might take a long time to end in this case, because of the building of the SMF type 30 records for a long-running job.

DDCONS(NO) requests that this consolidation function be bypassed, which results in a reduction in the amount of processing required to build the records, and thus a reduction in the amount of time required to complete the job.



Avoid LOG overhead

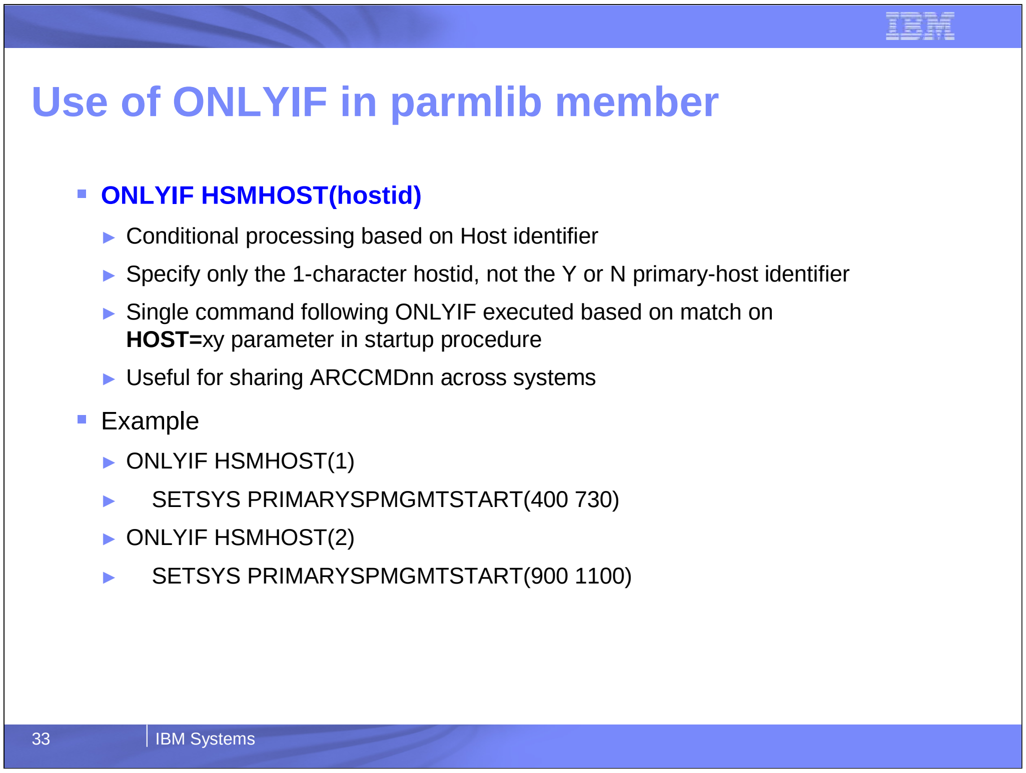
- ✓ Use **HOLD LOG** to avoid DFSMShsm logging overhead
  - ▸ Command can be added to PARMLIB
  - ▸ Turns off writing to the LOGX/LOGY data sets
  - ▸ Information available elsewhere, such as FSR records in SMF, Activity Logs, PDA trace data
  - ▸ Some ISV products require the LOGX/LOGY data sets as input
  - ▸ Reduces DFSMShsm overhead activity (approximately 7% reduction)

Studies have shown that DFSMShsm logging can add as much as a 7% overhead to DFSMShsm processing.

In recent years, DFSMShsm development has not been adding new log records for newer functions

It is recommended that if logging is not required, installations can issue the HOLD LOG command to stop logging.

There are certain ISV products that are dependent on information in the DFSMShsm log. You should inquire with any ISVs you might have that have DFSMShsm add-on products, whether or not they utilize information in the DFSMShsm logs. If yes, you should not HOLD logging.



Use of ONLYIF in parmlib member

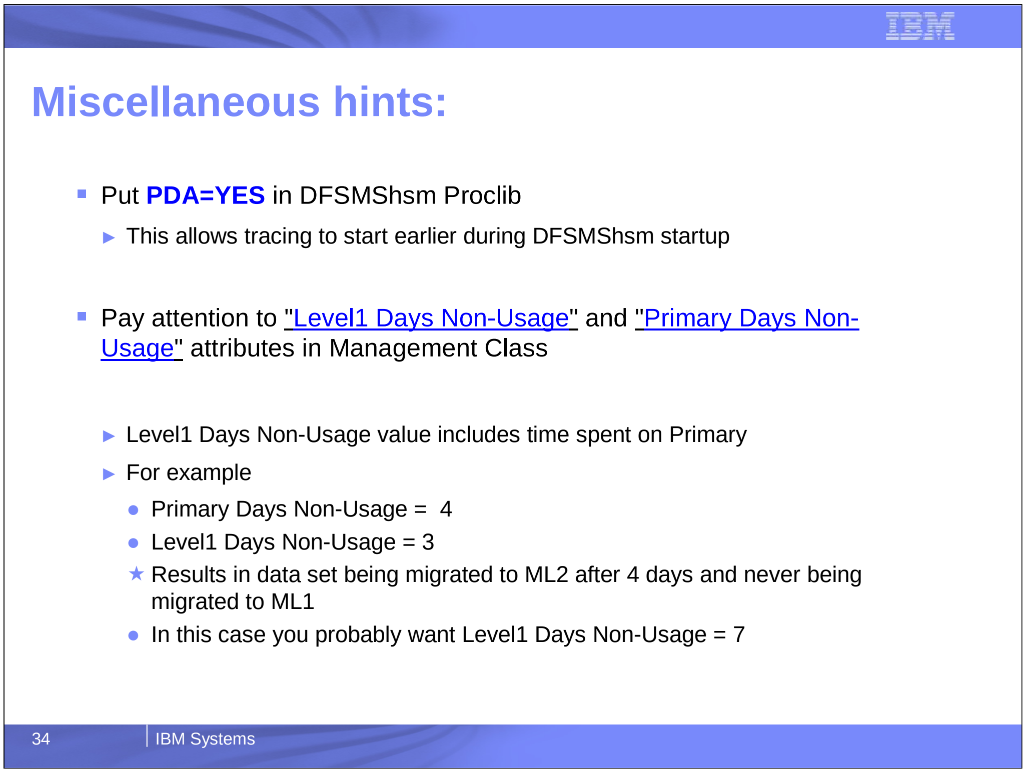
- **ONLYIF HSMHOST(hostid)**
  - Conditional processing based on Host identifier
  - Specify only the 1-character hostid, not the Y or N primary-host identifier
  - Single command following ONLYIF executed based on match on **HOST=**xy parameter in startup procedure
  - Useful for sharing ARCCMDnn across systems
- Example
  - ONLYIF HSMHOST(1)
  -     SETSYS PRIMARYSPMGMTSTART(400 730)
  - ONLYIF HSMHOST(2)
  -     SETSYS PRIMARYSPMGMTSTART(900 1100)

Many installations have unique ARCCMDnn parmlib members for each instance of DFSMShsm.
This can cause a change control headache because changes do not get reflected in all instances of the DFSMShsm parmlib

Installations can consolidate parmlib members and use the ONLYIF command to apply parmlib commands to a specific host

There is no DO/END logic so each command that you want directed to a specific host must be preceded by an ONLYIF statement

The HSMHOST corresponds to the HOSTID parameter in the startup procedure parameters



Miscellaneous hints:

- Put **PDA=YES** in DFSMShsm Proclib
  - This allows tracing to start earlier during DFSMShsm startup

- Pay attention to "Level1 Days Non-Usage" and "Primary Days Non-Usage" attributes in Management Class
  - Level1 Days Non-Usage value includes time spent on Primary
  - For example
    - Primary Days Non-Usage = 4
    - Level1 Days Non-Usage = 3
    - ★ Results in data set being migrated to ML2 after 4 days and never being migrated to ML1
    - In this case you probably want Level1 Days Non-Usage = 7

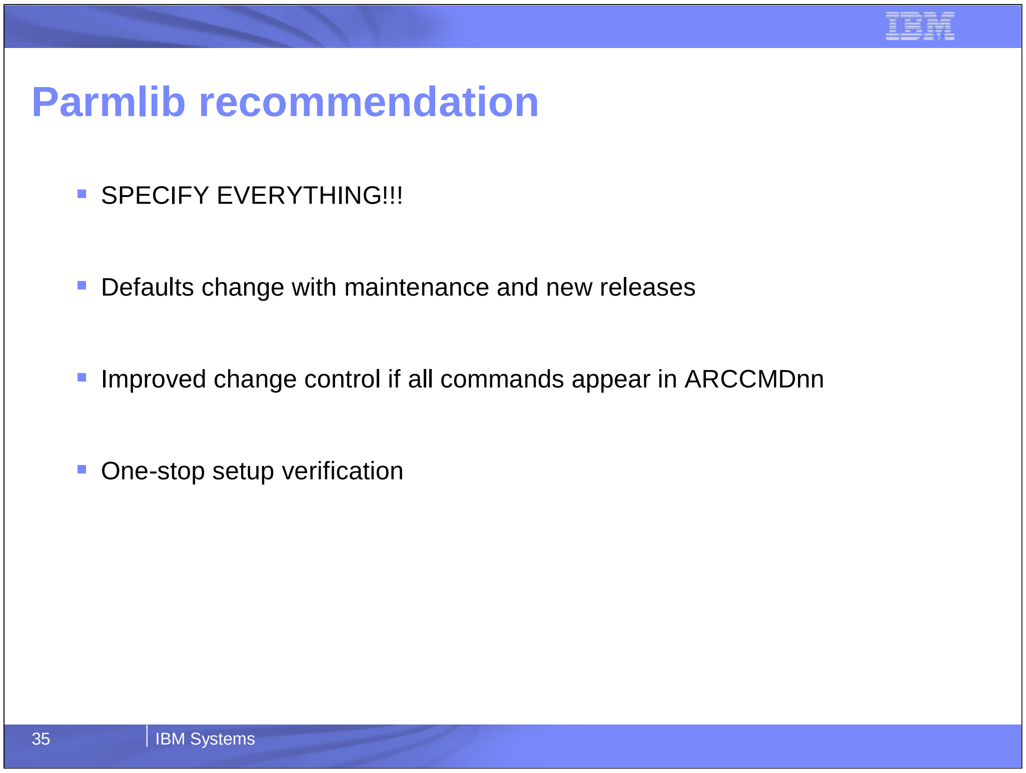
Installations should specify the DFSMShsm startup procedure parameter PDA=YES, rather than just having the parameter in the ARCCMDnn parmlib member.

This is to ensure any DFSMShsm processing that takes place between startup parm processing and ARCCMDnn scan, parse and processing are traced in the PDA output.

We have found a number of installations that are confused about the LEVEL1 DAYS NON-USAGE attribute in the Management Class resulting in premature migration to ML2 tape

Keep in mind that this attribute includes both the days the data set was resident on the primay volume as well as the days spent on the ML1 volume

If Level1 Days Non-Usage is lower than Primary Days Non-Usage the data set will get migrated directly to ML2 without ever being migrated to ML1



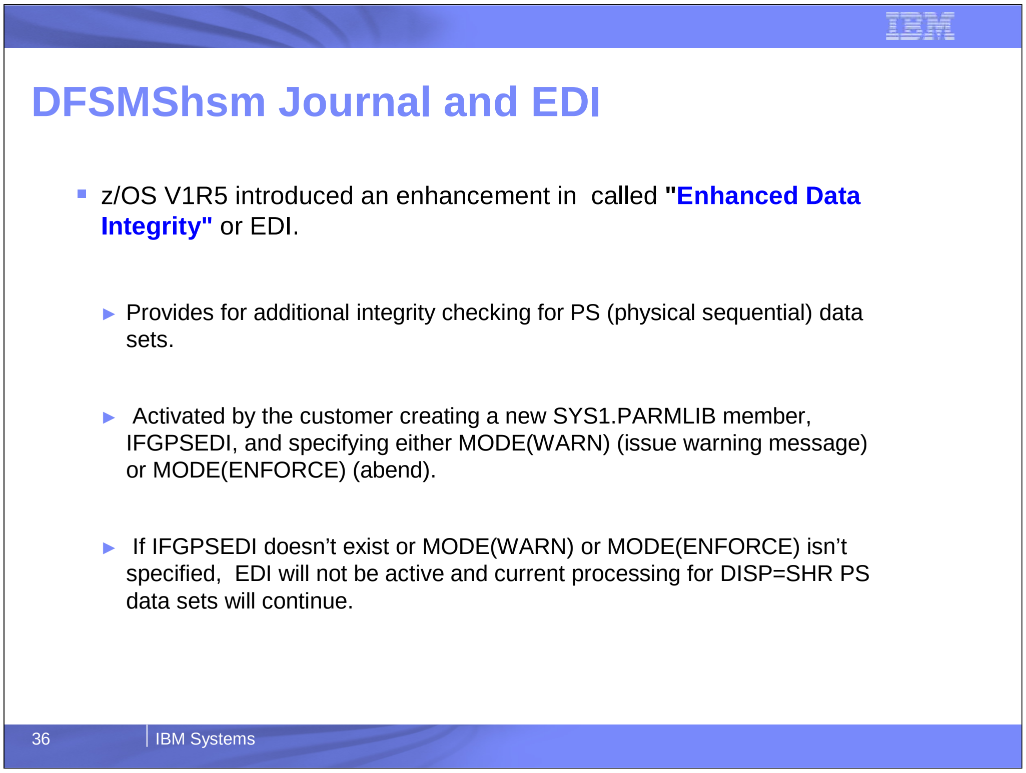
Parmlib recommendation

- SPECIFY EVERYTHING!!!
- Defaults change with maintenance and new releases
- Improved change control if all commands appear in ARCCMDnn
- One-stop setup verification

We recommend that you specify all required parameters/subparameters in your DFSMShsm parmlib member.

Do not allow command parameters to default

Do not get surprised by default settings that you did not expect
Although not common, defaults can actually change between releases



DFSMShsm Journal and EDI

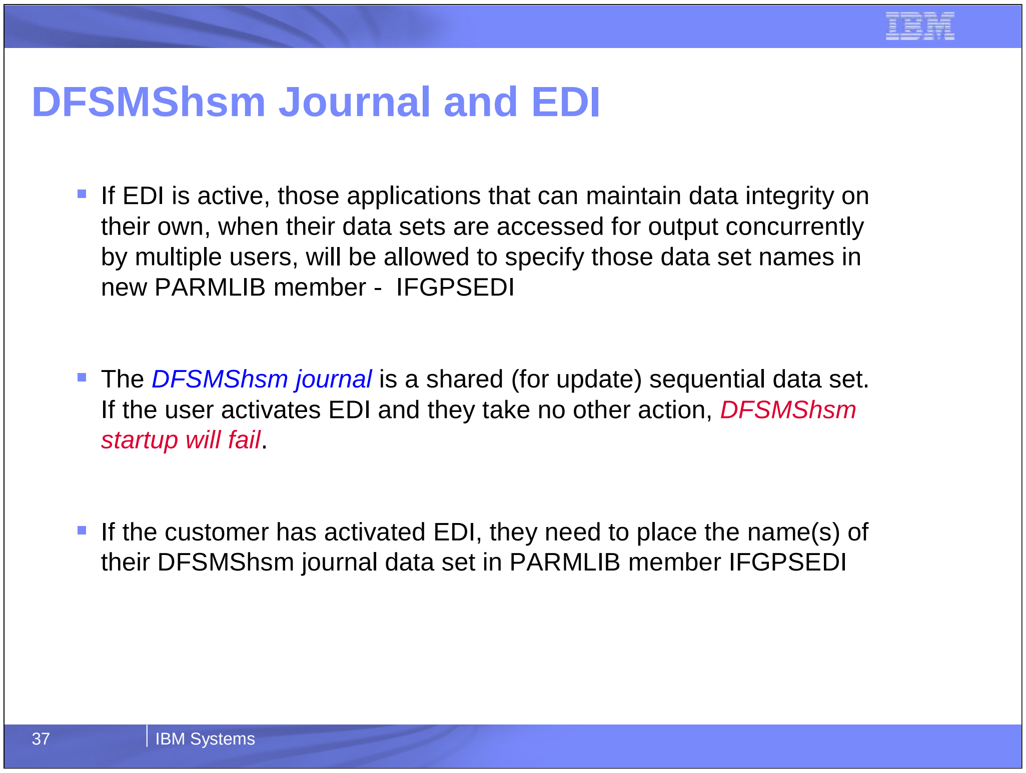
- z/OS V1R5 introduced an enhancement in called **"Enhanced Data Integrity"** or EDI.
  - Provides for additional integrity checking for PS (physical sequential) data sets.
  - Activated by the customer creating a new SYS1.PARMLIB member, IFGPSEDI, and specifying either MODE(WARN) (issue warning message) or MODE(ENFORCE) (abend).
  - If IFGPSEDI doesn't exist or MODE(WARN) or MODE(ENFORCE) isn't specified, EDI will not be active and current processing for DISP=SHR PS data sets will continue.

Enhanced data integrity can prevent users from concurrently accessing a shared sequential data set on DASD for output or update processing, thus avoiding any resulting loss of data

You can activate enhanced data integrity by creating an IFGPSEDI parmlib member.
When you activate enhanced data integrity, you can request that multiple users no longer have concurrent output or update access to a sequential data set on DASD

Enhanced data integrity applies only to sequential data sets. Authorized applications can exclude their sequential data sets from enhanced data integrity protection.
For information about how to exclude sequential data sets from enhanced data integrity, see z/OS DFSMS Using Data Sets.

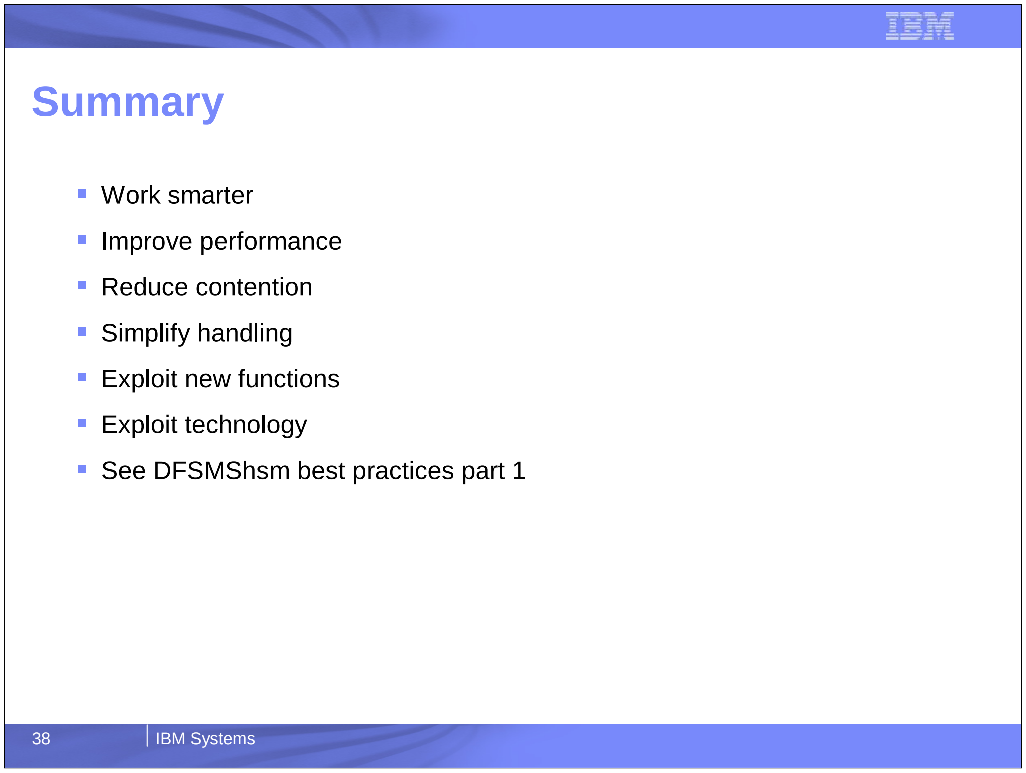
If you do not need enhanced data integrity, do not create the IFGPSEDI member of SYS1.PARMLIB.



DFSMShsm Journal and EDI

- If EDI is active, those applications that can maintain data integrity on their own, when their data sets are accessed for output concurrently by multiple users, will be allowed to specify those data set names in new PARMLIB member - IFGPSEDI
- The *DFSMShsm journal* is a shared (for update) sequential data set. If the user activates EDI and they take no other action, *DFSMShsm startup will fail*.
- If the customer has activated EDI, they need to place the name(s) of their DFSMShsm journal data set in PARMLIB member IFGPSEDI

If you have activated EDI you must add the DFSMS journal data set name to the IFGPSEDI parmlib member or DFSMShsm startup will fail



Summary

- Work smarter
- Improve performance
- Reduce contention
- Simplify handling
- Exploit new functions
- Exploit technology
- See DFSMShsm best practices part 1

This presentation covered a number of hints and tips to help you get the most out of your DFSMShsm environment

Please take time to read some of the DFSMShsm publications such as the DFSMShsm Storage Admin Ref and Storage Admin Guide so that you can get more detailed information on how to use DFSMShsm

If you are new to DFSMShsm , the DFSMShsm Primer Redbook (SG24-5272-01) provides an excellent overview of the functionality of DFSMShsm