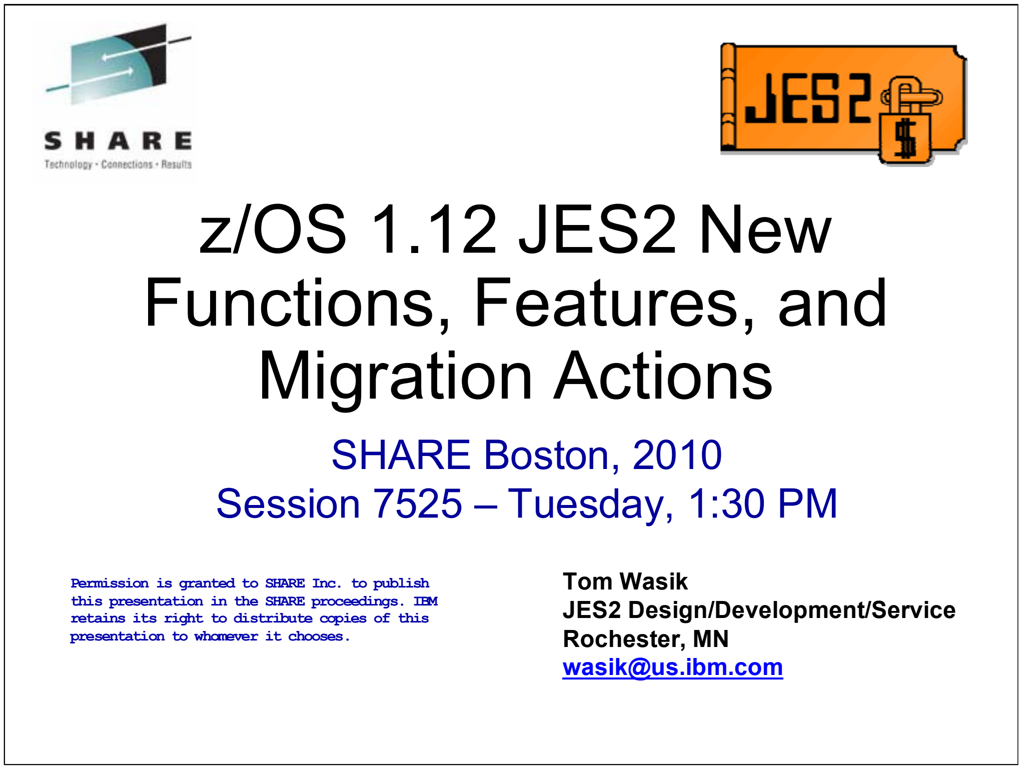
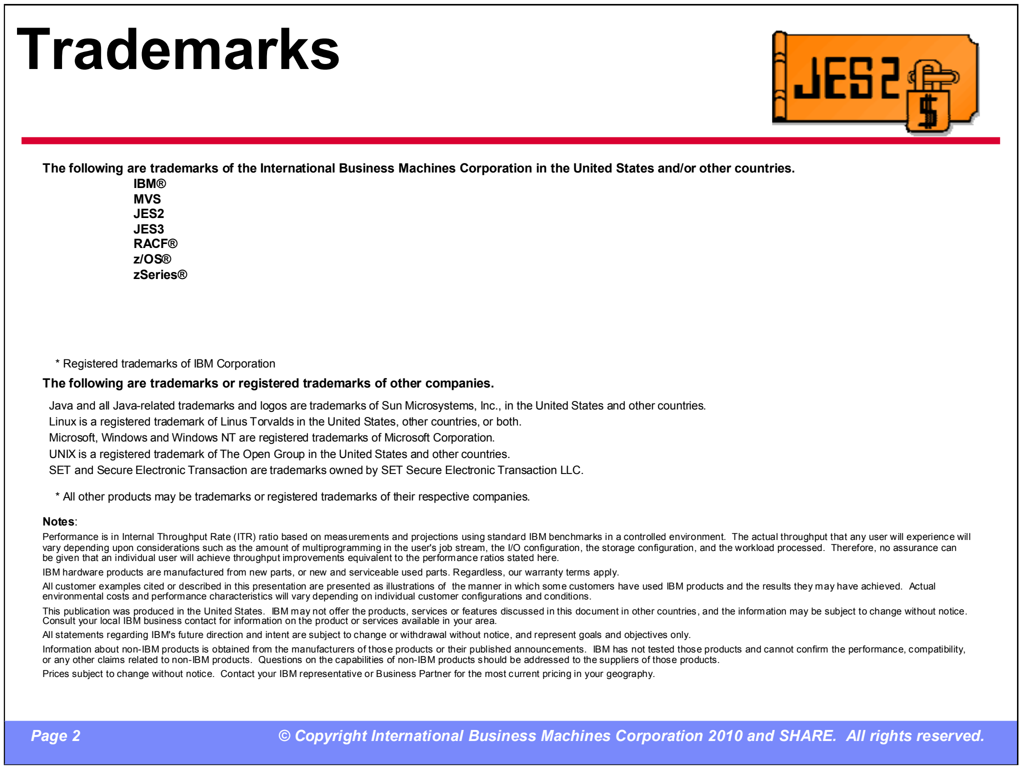


z/OS 1.12 JES2 New Functions, Features, and Migration Actions

SHARE Boston, 2010
Session 7525 – Tuesday, 1:30 PM

Tom Wasik
JES2 Design/Development/Service
Rochester, MN
wasik@us.ibm.com

This presentation includes technical details of the changes made in z/OS 1.12 JES2 including migration consideration. It is intended as a guide to installing the new release and exploiting the new functions. It also includes a review of the changes made in z/OS 1.11.



Trademarks

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

IBM®
MVS
JES2
JES3
RACF®
z/OS®
zSeries®

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

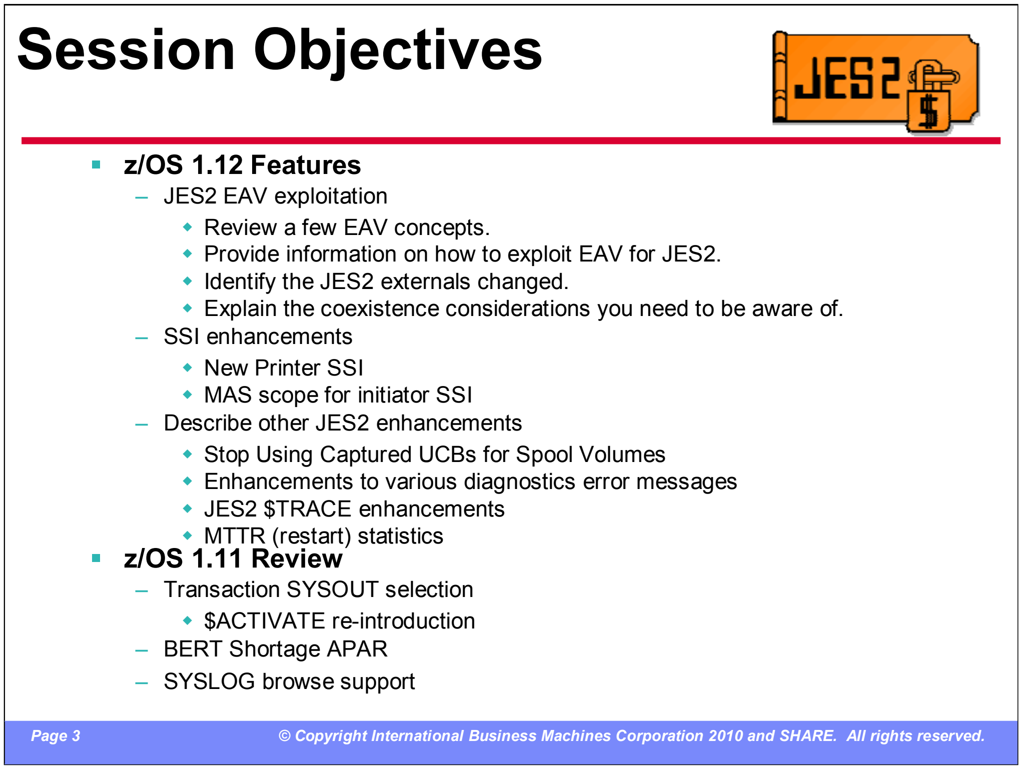
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Page 2

The required trademark page.....



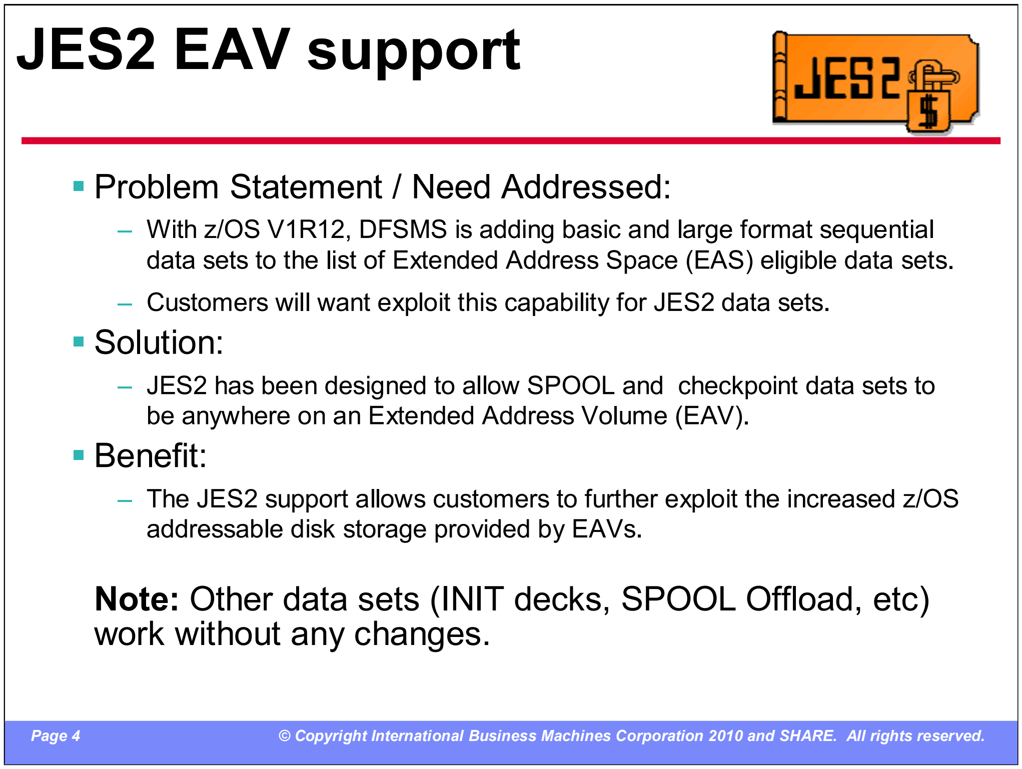
Session Objectives

- **z/OS 1.12 Features**
  - JES2 EAV exploitation
    - Review a few EAV concepts.
    - Provide information on how to exploit EAV for JES2.
    - Identify the JES2 externals changed.
    - Explain the coexistence considerations you need to be aware of.
  - SSI enhancements
    - New Printer SSI
    - MAS scope for initiator SSI
  - Describe other JES2 enhancements
    - Stop Using Captured UCBs for Spool Volumes
    - Enhancements to various diagnostics error messages
    - JES2 $TRACE enhancements
    - MTTR (restart) statistics
- **z/OS 1.11 Review**
  - Transaction SYSOUT selection
    - $ACTIVATE re-introduction
  - BERT Shortage APAR
  - SYSLOG browse support

Page 3

z/OS 1.12 JES2 added support to place the SPOOL and Checkpoint data sets in EAS storage (above the 64K cylinder line). It also added a new SSI to retrieve printer status information. There were also a number of serviceability (RAS) changes that will be discussed.

z/OS 1.11 JES2 added support for selecting SYSOUT in SAPI and extended status by transaction job names and job ids requiring a new $ACTIVATE level . There were also enhancements made to deal with BERT shortages, and support to browse SYSLOG data using the SPOOL data set browse interface.



JES2 EAV support

- Problem Statement / Need Addressed:
  - With z/OS V1R12, DFSMS is adding basic and large format sequential data sets to the list of Extended Address Space (EAS) eligible data sets.
  - Customers will want exploit this capability for JES2 data sets.
- Solution:
  - JES2 has been designed to allow SPOOL and checkpoint data sets to be anywhere on an Extended Address Volume (EAV).
- Benefit:
  - The JES2 support allows customers to further exploit the increased z/OS addressable disk storage provided by EAVs.

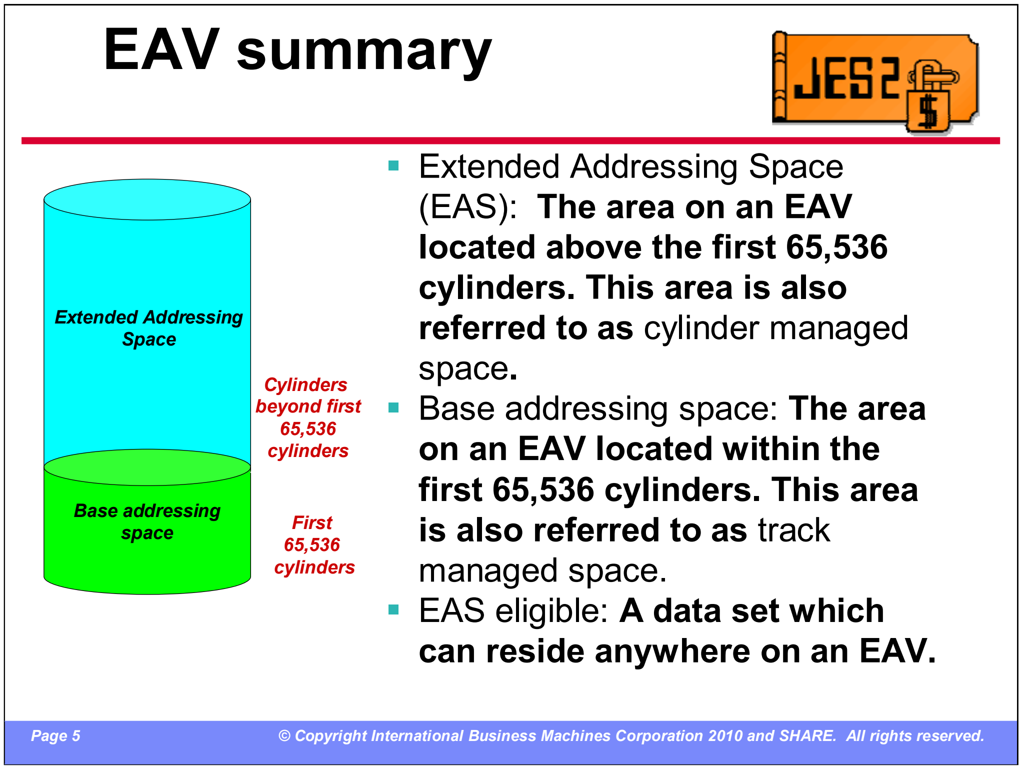
Note: Other data sets (INIT decks, SPOOL Offload, etc) work without any changes.

Page 4

With z/OS V1R10 and higher releases, z/OS has added support for DASD volumes having more than 65,520 cylinders. To expand the capacity of DASD storage volumes beyond 65,520 cylinders, z/OS had to extend the track address format. Hence the name **Extended Address Volume (EAV)** for a volume of more than 65,520 cylinders.

EAVs provide increased z/OS addressable disk storage. EAVs help you relieve storage constraints as well as simplify storage management by providing the ability to manage fewer, large volumes as opposed to many small volumes.

With z/OS V1R12, DFSMS added support for base and large format sequential data sets that can now be exploited for JES2 for SPOOL and checkpoint data sets.



EAV summary

Extended Addressing Space
Base addressing space
Cylinders beyond first 65,536 cylinders
First 65,536 cylinders

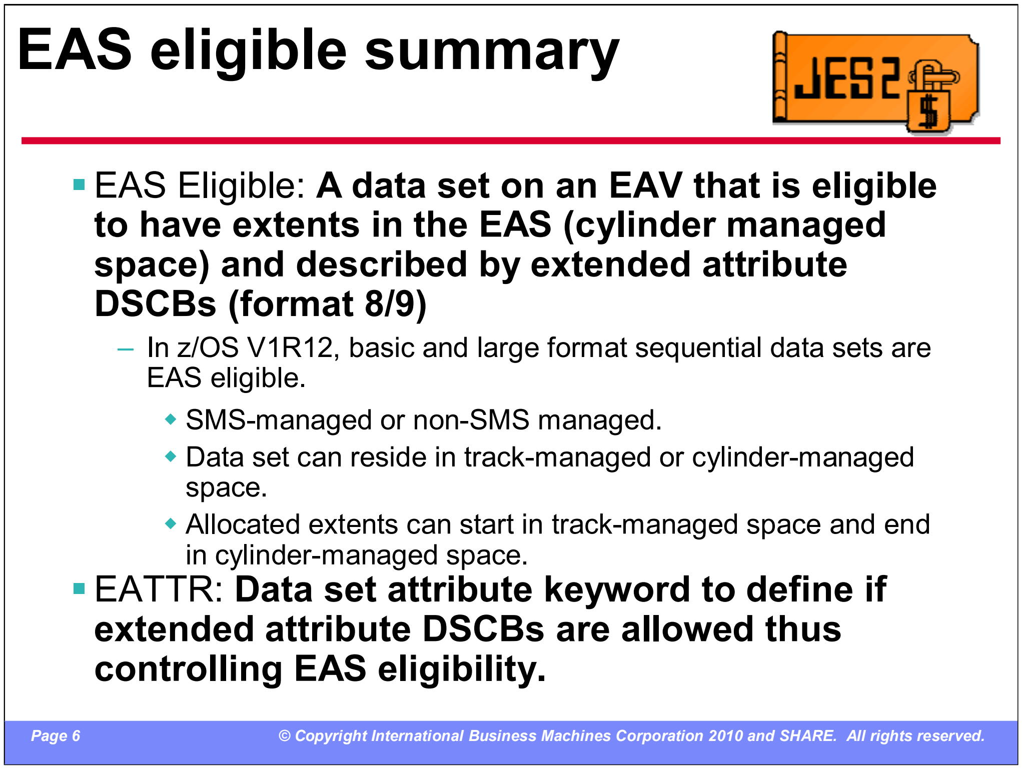
- Extended Addressing Space (EAS): **The area on an EAV located above the first 65,536 cylinders. This area is also referred to as** cylinder managed space**.**
- Base addressing space: **The area on an EAV located within the first 65,536 cylinders. This area is also referred to as** track managed space.
- EAS eligible: **A data set which can reside anywhere on an EAV.**

Page 5

An Extended Address Volume (EAV) is defined to be a volume with more than 65,520 cylinders.

The Extended Addressing Space (EAS) of an EAV refers to the cylinders of a volume beyond the first 65,535.

In this diagram of an EAV, cylinders up to but not including cylinder 65,536 are in the base addressing space of the EAV. Cylinders starting with cylinder 65,536 are in the EAS (cylinder managed space) of the EAV.



EAS eligible summary

- EAS Eligible: **A data set on an EAV that is eligible to have extents in the EAS (cylinder managed space) and described by extended attribute DSCBs (format 8/9)**
  - In z/OS V1R12, basic and large format sequential data sets are EAS eligible.
    - SMS-managed or non-SMS managed.
    - Data set can reside in track-managed or cylinder-managed space.
    - Allocated extents can start in track-managed space and end in cylinder-managed space.
- EATTR: **Data set attribute keyword to define if extended attribute DSCBs are allowed thus controlling EAS eligibility.**

Page 6

EAS eligible data sets are defined to be those that can be allocated in the Extended Addressing Space and have extended attributes. EAS is also referred to as cylinder-managed space. Below EAS is also referred to as track-managed space.

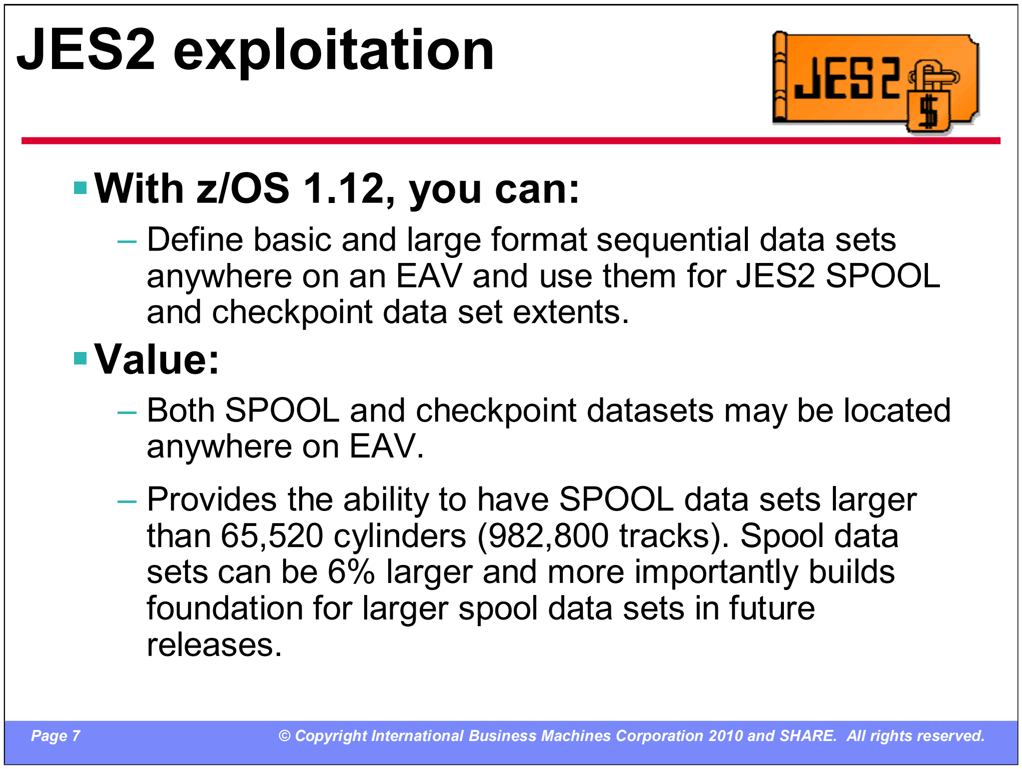
EAS eligible data sets added in z/OS V1R12 include basic and large format sequential data sets.

A new data set attribute, EATTR, first added in z/OS V1R11, allows a user to control whether a data set can have extended attribute DSCBs and thus control whether it can be allocated in EAS.

EATTR is used to control whether a data set can have extended attribute DSCBs and thus control whether it can be allocated in EAS.

• EATTR of NO indicates that the data set can not have extended attributes or reside in EAS. This is the default for non-VSAM data sets such as basic and large format sequential data sets.

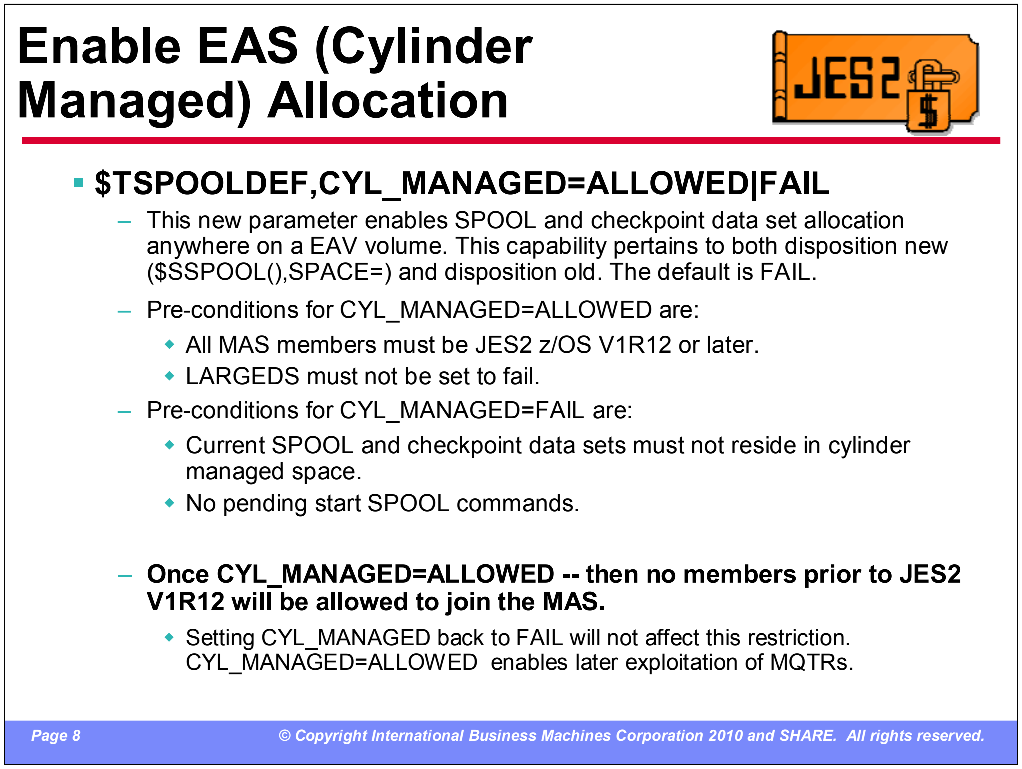
• EATTR of OPT indicates that the data set can have extended attributes and can optionally reside in EAS.



JES2 exploitation

- **With z/OS 1.12, you can:**
  - Define basic and large format sequential data sets anywhere on an EAV and use them for JES2 SPOOL and checkpoint data set extents.
- **Value:**
  - Both SPOOL and checkpoint datasets may be located anywhere on EAV.
  - Provides the ability to have SPOOL data sets larger than 65,520 cylinders (982,800 tracks). Spool data sets can be 6% larger and more importantly builds foundation for larger spool data sets in future releases.

JES2 exploitation of the EAV support added in z/OS V1R12 allows customers to define EAS eligible data sets for use as SPOOL extents and checkpoint data sets. The data sets can be placed anywhere on a EAV volume but they cannot be larger that the 1,048,575 track JES2 architectural limit in size.



Enable EAS (Cylinder Managed) Allocation

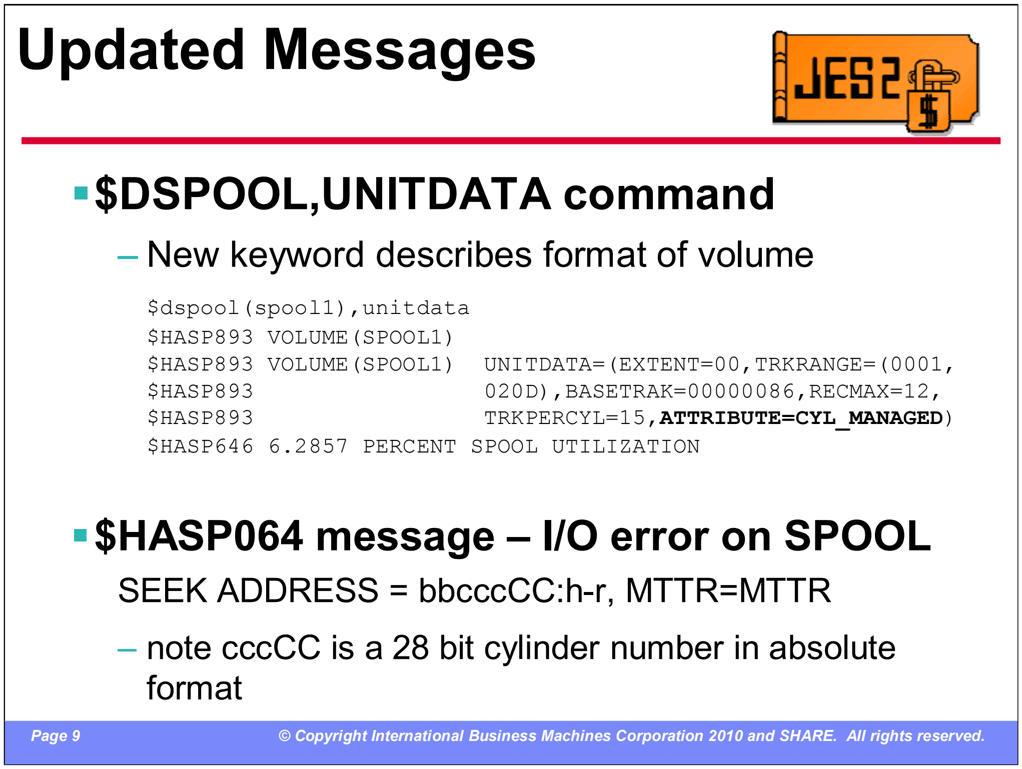
- **$TSPOOLDEF,CYL_MANAGED=ALLOWED|FAIL**
  - This new parameter enables SPOOL and checkpoint data set allocation anywhere on a EAV volume. This capability pertains to both disposition new ($SSPOOL(),SPACE=) and disposition old. The default is FAIL.
  - Pre-conditions for CYL_MANAGED=ALLOWED are:
    - All MAS members must be JES2 z/OS V1R12 or later.
    - LARGEDS must not be set to fail.
  - Pre-conditions for CYL_MANAGED=FAIL are:
    - Current SPOOL and checkpoint data sets must not reside in cylinder managed space.
    - No pending start SPOOL commands.
  - **Once CYL_MANAGED=ALLOWED -- then no members prior to JES2 V1R12 will be allowed to join the MAS.**
    - Setting CYL_MANAGED back to FAIL will not affect this restriction. CYL_MANAGED=ALLOWED enables later exploitation of MQTRs.

The ability to create or use a SPOOL and Checkpoint data set that is in EAS storage is controlled by the new CYL_MANAGED keyword on the SPOOLDEF statement. By default, CYL_MANAGED is set to FAIL. This allows down level members (prior to z/OS 1.12) to continue to access the SPOOL and checkpoint volumes. To activate the support for EAS data sets, you first must complete your migration to z/OS 1.12 (once you activate the support you cannot warm start down level members into the MAS), and ensure that LARGEDS on SPOOLDEF is not set to FAIL. Given those 2 requirements, you can then use an operator command to set SPOOLDEF CYL_MANAGED=ALLOWED. It is possible to reset CYL_MANAGED to FAIL, but only if there are no EAS volumes active and there are no pending start SPOOL commands.

Similar to SPOOLDEF LARGEDS=ALLOWED, once the new CYL_MANAGED=ALLOWED is set, then you cannot start pre-z/OS 1.12 JES2 members into the MAS. IBM Recommends stabilizing on z/OS 1.12 before considering setting CYL_MANAGED=ALLOWED.

Sample command and response

```
$TSPOOLDEF,CYL_MANAGED=ALLOWED
$HASP844  SPOOLDEF   BUFSIZE=3992,DSNAME=SYS1.HASPACE,
$HASP844             FENCE=(ACTIVE=NO,VOLUMES=1),GCRATE=NORMAL,
$HASP844             LASTSVAL=(2010.058,22:48:01),LARGEDS=ALLOWED,
$HASP844             SPOOLNUM=32,CYL_MANAGED=ALLOWED,TGSIZE=6,
$HASP844             TGSPACE=(MAX=16288,DEFINED=525,ACTIVE=525,
$HASP844             PERCENT=6.2857,FREE=492,WARN=80),TRKCELL=3,
$HASP844             VOLUME=SPOOL
```



Updated Messages

- **$DSPOOL,UNITDATA command**
  - New keyword describes format of volume

```
$dspool(spool1),unitdata
$HASP893 VOLUME(SPOOL1)
$HASP893 VOLUME(SPOOL1)   UNITDATA=(EXTENT=00,TRKRANGE=(0001,
$HASP893                  020D),BASETRAK=00000086,RECMAX=12,
$HASP893                  TRKPERCYL=15,ATTRIBUTE=CYL_MANAGED)
$HASP646 6.2857 PERCENT SPOOL UTILIZATION
```

- **$HASP064 message – I/O error on SPOOL**

  SEEK ADDRESS = bbcccCC:h-r, MTTR=MTTR

  - note cccCC is a 28 bit cylinder number in absolute format

Page 9

To better identify the format of a SPOOL volume, a new ATTRIBUTE keyword was added to the $D SPOOL(xxxxx),UNITDATA. The possible values are ABSOLUTE, RELATIVE, LARGEDS, or CYL_MANAGED.

Device addressing is different, but compatible, for non-EAV and EAV volumes. A device address is the form of *BBCCHH*. Only the *CCHH* is relevant for this discussion. The *CC* is the 16 bit-cylinder number and *HH* is the 16-bit track number in that cylinder. This is true for non-EAV volumes.

If the volume is an EAV, the cylinder number in these four *CCHH* bytes will be 28 bits and the track number will be four bits. For EAV volumes the absolute and normalized is as follows:

CCCCccch (Absolute address)

CCCC - 16 **low** order bits of cylinder number

ccc - 12 **high** order bits of cylinder number

h - 4 bit Track number (limit of 15 tracks per cylinder)

cccCCCCh (Normalized address)

Used to perform <> operations. Also used to perform arithmetic operations to cylinder and head values.

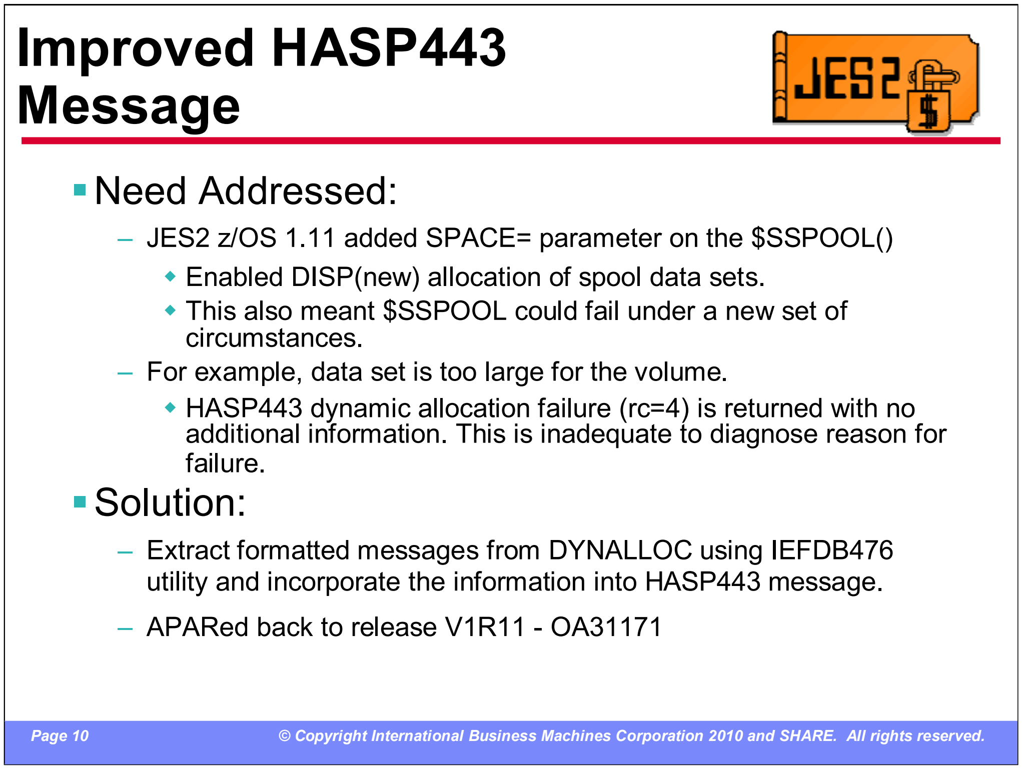
ccc – 12 high order bits of cylinder address

CCCC – 16 low order bits of cylinder number

h – 4 bit Track Number

Internally JES2 uses MTTR and MQTRs reference data set records.

Absolute addresses must be converted to MTTR/MQTRs and visa versa.



Improved HASP443 Message

- Need Addressed:
  - JES2 z/OS 1.11 added SPACE= parameter on the $SSPOOL()
    - Enabled DISP(new) allocation of spool data sets.
    - This also meant $SSPOOL could fail under a new set of circumstances.
  - For example, data set is too large for the volume.
    - HASP443 dynamic allocation failure (rc=4) is returned with no additional information. This is inadequate to diagnose reason for failure.
- Solution:
  - Extract formatted messages from DYNALLOC using IEFDB476 utility and incorporate the information into HASP443 message.
  - APARed back to release V1R11 - OA31171

Page 10

Currently, when a start SPOOL command fails because of dynamic allocation, all that is displayed is a return code that allocation failed. There is no indication as to why the allocation failed. To address this shortcoming, the HASP443 message was enhanced to display the messages that are returned by the DYNALLOC request. These are generally IKJ messages normally associated with TSO ALOC command failures.

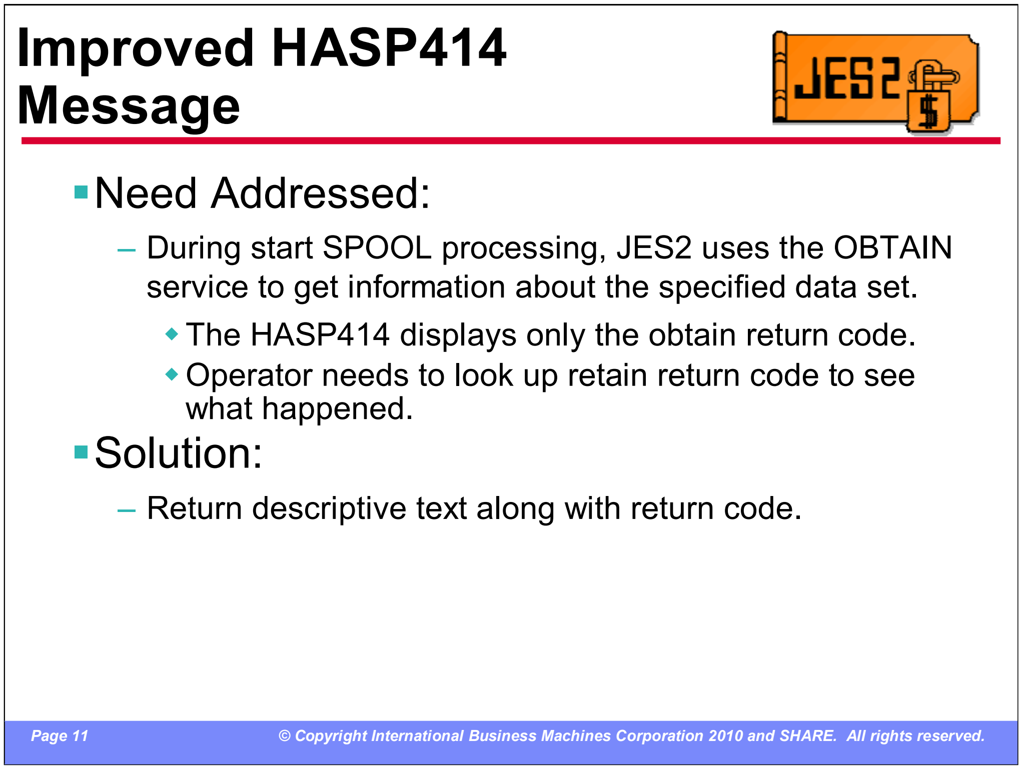
$S SPOOL example

Scenario: Starting a SPOOL volume and not enough space on volume.

```
      $sspl(spool2),space=(cyl,30000)

      $HASP893 VOLUME(SPOOL2) STATUS=INACTIVE,COMMAND=(START)
$HASP646 4.3809 PERCENT SPOOL UTILIZATION

      $HASP443 SPOOL2 NOT ALLOCATED
         DYNAMIC ALLOCATION FAILURE, RC=04
         IKJ56245I DATA SET SYS1.HASPACE NOT ALLOCATED, NOT
         ENOUGH SPACE ON VOLUME
         IKJ56245I USE DELETE COMMAND TO DELETE UNUSED
         DATA SETS
```



Improved HASP414 Message

- Need Addressed:
  - During start SPOOL processing, JES2 uses the OBTAIN service to get information about the specified data set.
    - The HASP414 displays only the obtain return code.
    - Operator needs to look up retain return code to see what happened.
- Solution:
  - Return descriptive text along with return code.

Page 11

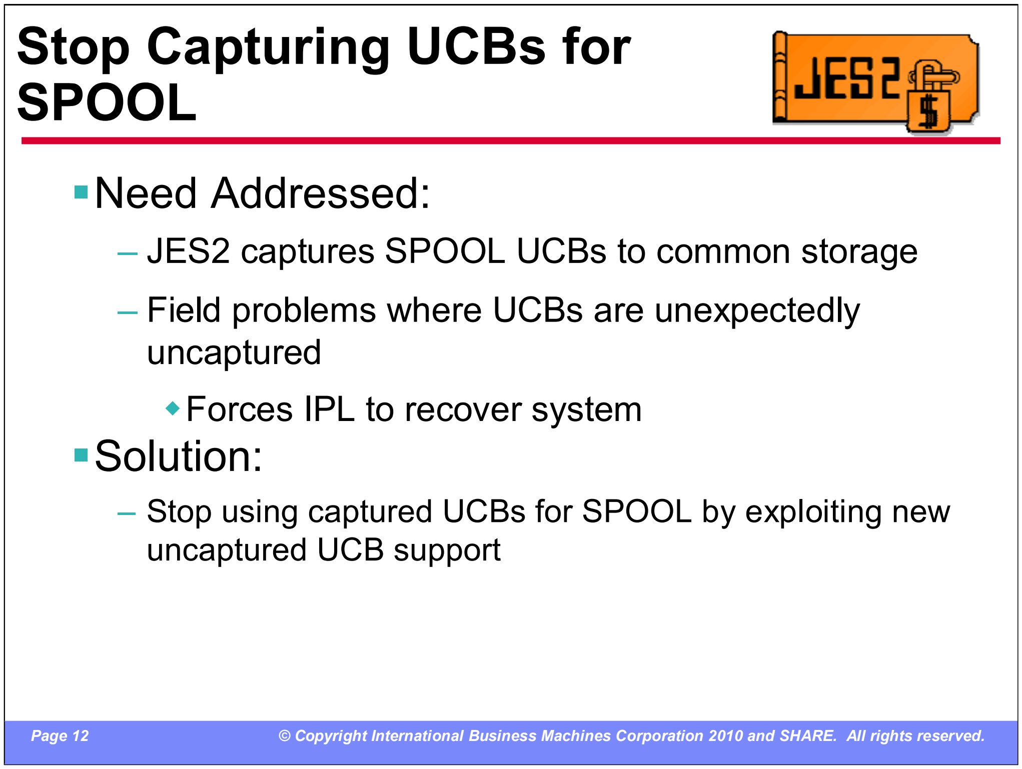
When preparing to allocate either a checkpoint or SPOOL data set -- JES2 uses the OBTAIN supervisor service to get information about the specified data set. When OBTAIN fails, JES2 issues the HASP414 with simply the obtain return code. The return code alone is not adequate to expediently diagnose the problem.

To address this, descriptive text was added to the HASP414 message to describe the meaning of the return code.

$S SPOOL example

Scenario: Starting a volume and corresponding data set resides in EAS (cylinder managed space) and SPOOLDEF parameter CYL_MANAGED is set to FAIL.

```
$sspl(spool3)
$HASP893 VOLUME(SPOOL3)  STATUS=INACTIVE,COMMAND=(START)
$HASP646 1.3333 PERCENT SPOOL UTILIZATION
$HASP414 MEMBER IBM1 -- OBTAIN FAILED FOR DATASET=SYS1.HASPACE ON
         VOLSER=SPOOL3 WITH CC 24 -- DATA SET RESIDES IN CYLINDER
         MANAGED STORAGE AND SPOOLDEF,CYL_MANAGED=FAIL
$HASP443 SPOOL3 NOT ALLOCATED
         OBTAIN FAILURE, RC=01
```



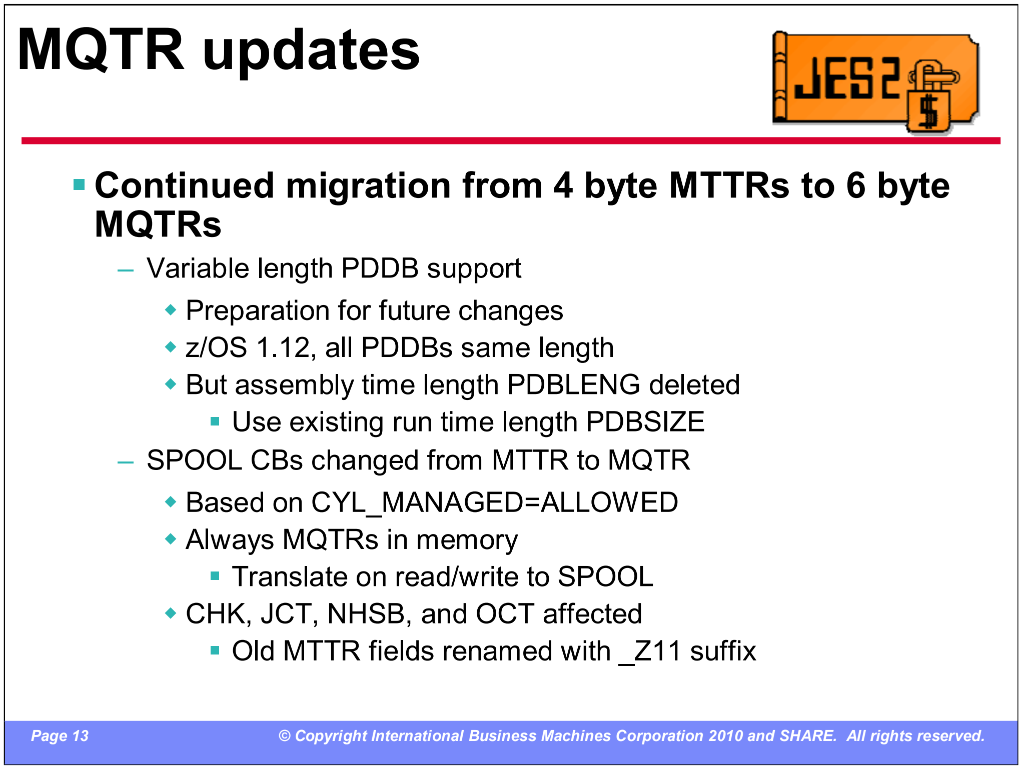
Stop Capturing UCBs for SPOOL

- Need Addressed:
  - JES2 captures SPOOL UCBs to common storage
  - Field problems where UCBs are unexpectedly uncaptured
    - Forces IPL to recover system
- Solution:
  - Stop using captured UCBs for SPOOL by exploiting new uncaptured UCB support

Page 12

JES2 currently recommends that customers not move the UCB for SPOOL volumes above the line. This is because JES2 will capture the UCB for the SPOOL volume to common storage for use in the DEB data area that JES2 builds to access SPOOL. However, if the customer does move the UCB above the line (many customers move all their UCBs above the line), they are exposed to an undiagnosed field problem with the captured UCB. The problem surfaces as an error trying to access a SPOOL volume on one system. This is because the UCB that the JES2 DEB points to has been unexpectedly uncaptured. What is suspected is the capture count for the page with the UCB has gone bad (too many uncaptures of a different UCB on the same page) and the page gets uncaptured prematurely. This is difficult to diagnose since the error could have occurred hours or days before the JES2 symptom.

Within the z/OS 1.12 DFSMS changes, there is a change to the DASD extension to the DEB that supports a 31 bit UCB address. JES2 will delete the code that captures the UCB for SPOOL and instead build the DASD DEB extension that supports 31 bit DEB addresses. This allows JES2 to do away with the UCB capture for SPOOL volumes.



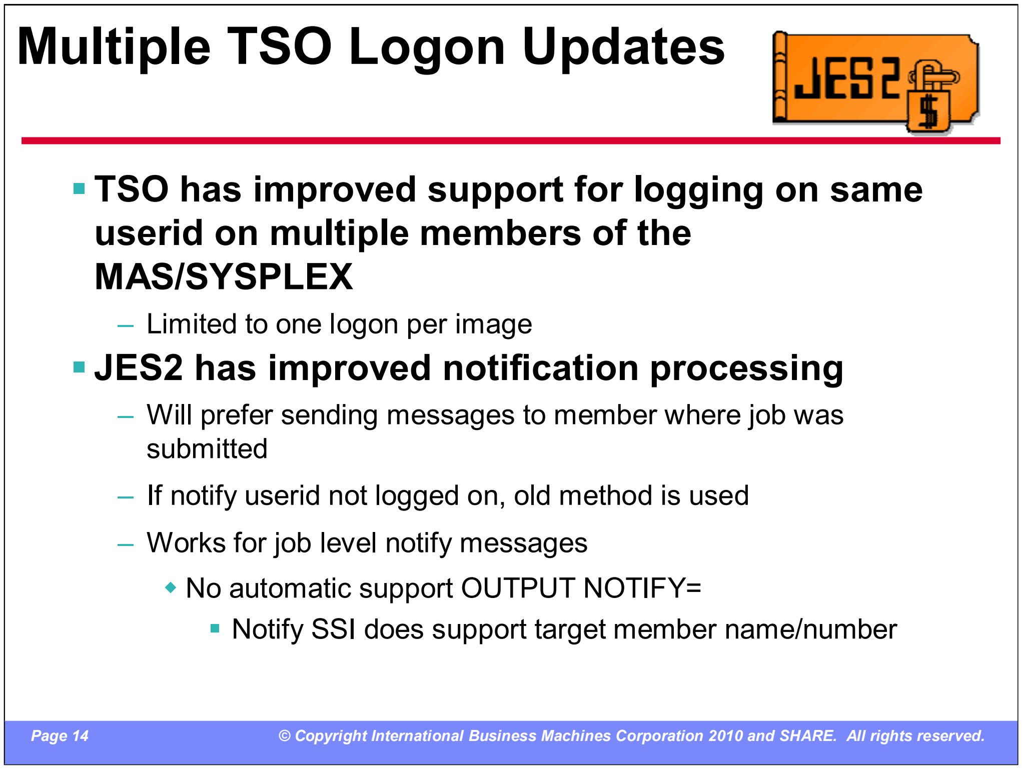
MQTR updates

- **Continued migration from 4 byte MTTRs to 6 byte MQTRs**
  - Variable length PDDB support
    - Preparation for future changes
    - z/OS 1.12, all PDDBs same length
    - But assembly time length PDBLENG deleted
      - Use existing run time length PDBSIZE
  - SPOOL CBs changed from MTTR to MQTR
    - Based on CYL_MANAGED=ALLOWED
    - Always MQTRs in memory
      - Translate on read/write to SPOOL
    - CHK, JCT, NHSB, and OCT affected
      - Old MTTR fields renamed with _Z11 suffix

Page 13

JES2 is continuing its migration from the 4 byte MTTR to a 6 byte MQTR. This will remove the architectural limit of 1M tracks per SPOOL data set. There are 2 major changes in this release in this area. The first is the completion of the support for variable length PDDBs. There has been a run time length field (PDBSIZE) in the PDDB for years. But the assembly time length PDBLENG has also existed. In z/OS 1.12, JES2 has completed the work needed to support variable size PDDBs. So at this time, we are deleting the compile time PDDB length field (PDBLENG). The size of the PDDB in z/OS 1.12 is constant, but IBM highly recommends not using the compile time PDDB length. You should be using services to access the data in the PDDB or services to step through the PDDB.

The second change is additional migration of pointers in SPOOL control blocks from MTTRs to MQTRs. The control blocks affected always have MQTRs in memory (they are translated when read/written from SPOOL by $CBIO). If you still access any of these SPOOL addresses directly, you will either need to migrate to the MQTR form of the SPOOL address or used other services to get the information you need.



Multiple TSO Logon Updates

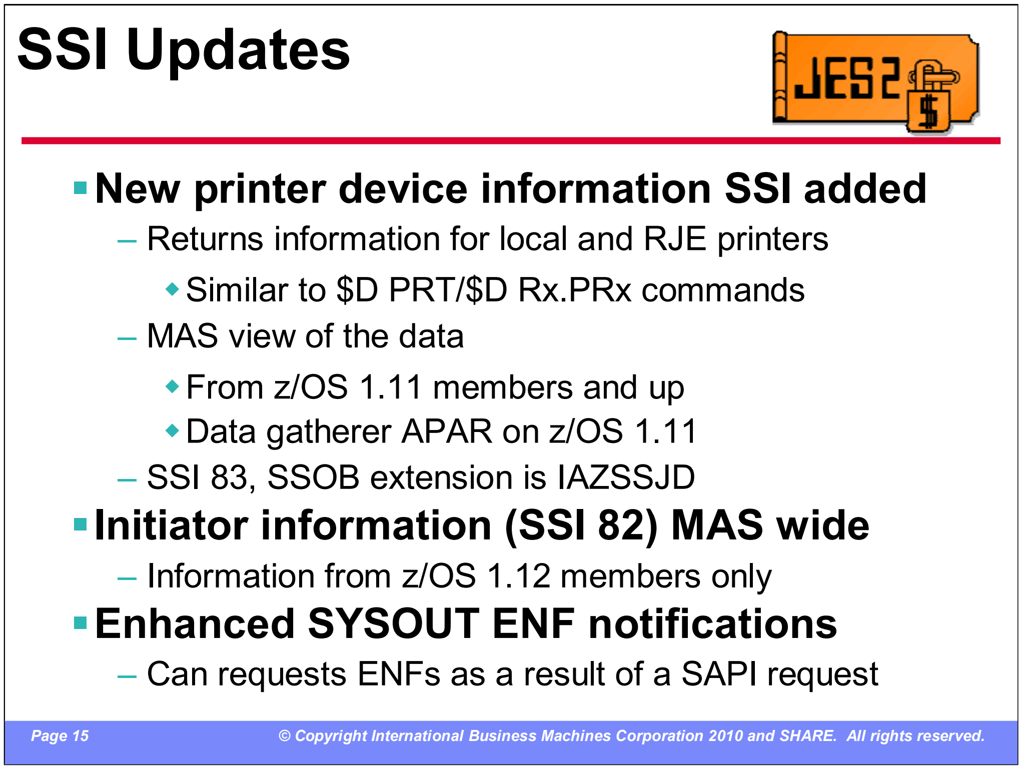
- TSO has improved support for logging on same userid on multiple members of the MAS/SYSPLEX
  - Limited to one logon per image
- JES2 has improved notification processing
  - Will prefer sending messages to member where job was submitted
  - If notify userid not logged on, old method is used
  - Works for job level notify messages
    - No automatic support OUTPUT NOTIFY=
      - Notify SSI does support target member name/number

Page 14

TSO has made some changes to better support logging into the same userid on multiple members of the MAS (once per MVS image). However, JES2 has also made some changes to how the job level notify messages are issued.

When deciding where to send a notify message, JES2 will prefer the JES2 member number where the job was submitted (went through input processing). If the notify userid is logged onto that member, then that is where JES2 will send the notify message. This applies even if the notify userid is NOT the userid that submitted the job. If the notify userid is not on the same NJE node as the submitter of the job, then the old processing is used.

There is no automatic support for NOTIFY= on the OUTPUT card. The notify userid in this case is passed to an application for processing. There is no direct mechanism for setting the member were the submitter was logged on. However, the notify SSI that actually sends the message does support specifying a member to notify and that member is used if provided.



SSI Updates

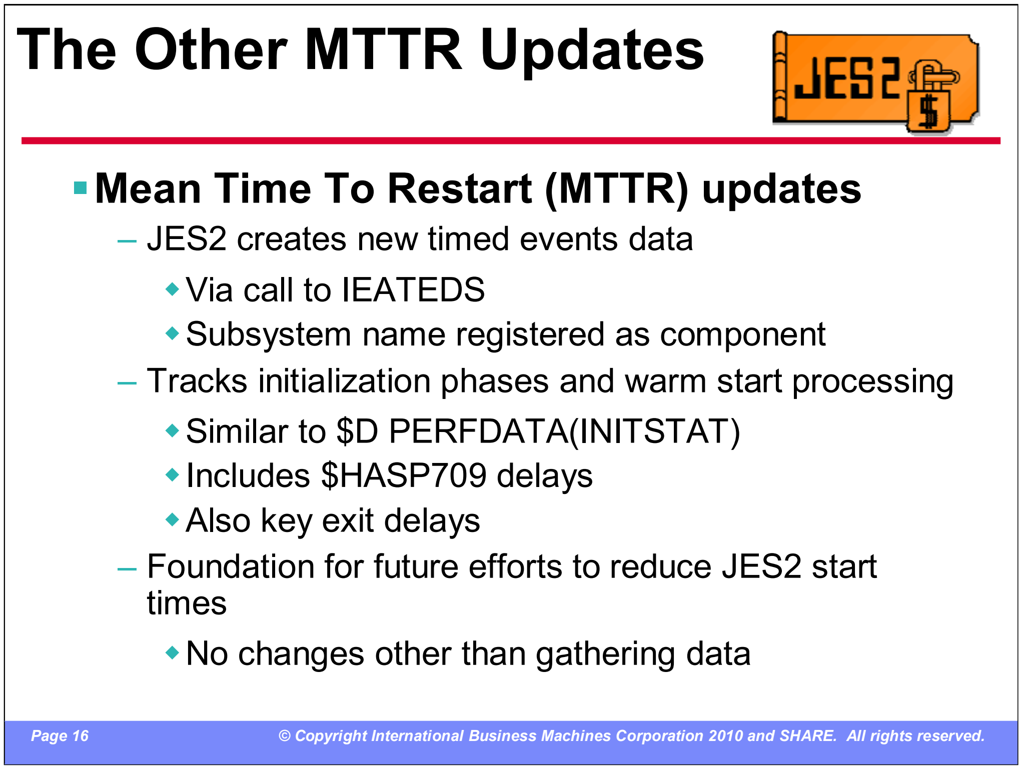
- New printer device information SSI added
  - Returns information for local and RJE printers
    - Similar to $D PRT/$D Rx.PRx commands
  - MAS view of the data
    - From z/OS 1.11 members and up
    - Data gatherer APAR on z/OS 1.11
  - SSI 83, SSOB extension is IAZSSJD
- Initiator information (SSI 82) MAS wide
  - Information from z/OS 1.12 members only
- Enhanced SYSOUT ENF notifications
  - Can requests ENFs as a result of a SAPI request

Page 15

New in this release is an SSI (83 with SSOB extension IAZSSJD) to obtain information on local and remote printers. This information is available on z/OS 1.12 from any z/OS 1.11 MAS member and later. Only local and remote printer are currently supported via the SSI (though the IAZSSJVD data area does define additional devices). The information is provided via a data gather code that was shipped in z/OS 1.11. To fully support requests from z/OS 1.12, you must apply the data gatherer APARs **OA31703** and **OA32712** on your z/OS 1.11 system. These APARs are a pre-req for the JES2 toleration APAR OA28532.

The initiator subfunction of the JES property SSI was enhanced to support requesting information from other members. This requires the target member to be running z/OS 1.12.

SYSOUT ENF processing was enhanced to support requesting ENFs for data sets select by SAPI. This is useful if a SAPI application is requesting SYSOUT information and returning it for later processing but still wants to track the status of the data set.



The Other MTTR Updates

- **Mean Time To Restart (MTTR) updates**
  - JES2 creates new timed events data
    - Via call to IEATEDS
    - Subsystem name registered as component
  - Tracks initialization phases and warm start processing
    - Similar to $D PERFDATA(INITSTAT)
    - Includes $HASP709 delays
    - Also key exit delays
  - Foundation for future efforts to reduce JES2 start times
    - No changes other than gathering data

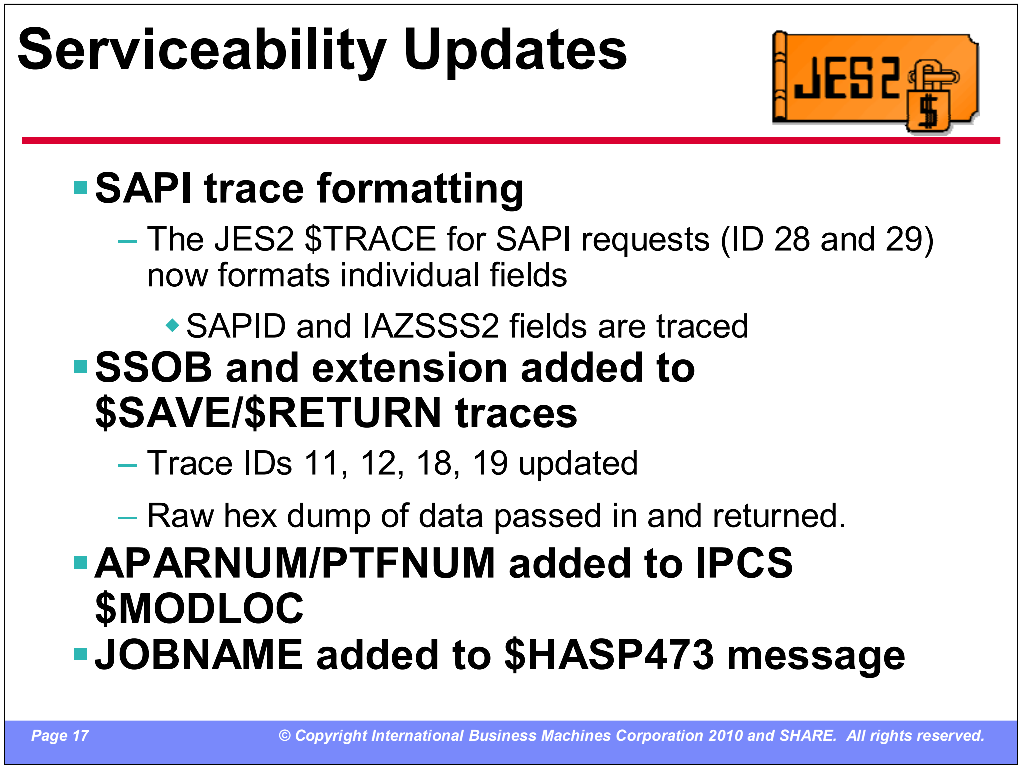
These updates are for the “other” MTTR, mean time to restart, not MTTR the SPOOL address scheme.

Note: MVS and HASP were using MTTR LONG before anyone else in the industry, so please, don’t ask me about why the acronym was reused. However, we are migrating away from MTTR to MQTR (just coincidence). Hopefully, MQTR will not be re-used any time soon.

JES2 has added support to collect the new timed event records via the IEATEDS service. These records can be used as part of an overall effort to understand and eventually reduce restart delays. JES2 registers with the service using our subsystem name (if you call your JES2 “HASP” then that is what you will see as the component in the records. If you run a secondary JES2, then that subsystem name will be the component for its records.

JES2 tracks initialization phase times (similar to $D PERFDATA(INITSTAT)), delays associated with HASP709 messages (IEFAUINT etc), key exit delays (Exit 0, etc), and the time to complete warm start.

At this point, the only change made was to gather data. No actual changes to JES2 restart processing have been made.



Serviceability Updates

- **SAPI trace formatting**
  - The JES2 $TRACE for SAPI requests (ID 28 and 29) now formats individual fields
    - SAPID and IAZSSS2 fields are traced
- **SSOB and extension added to $SAVE/$RETURN traces**
  - Trace IDs 11, 12, 18, 19 updated
  - Raw hex dump of data passed in and returned.
- **APARNUM/PTFNUM added to IPCS $MODLOC**
- **JOBNAME added to $HASP473 message**

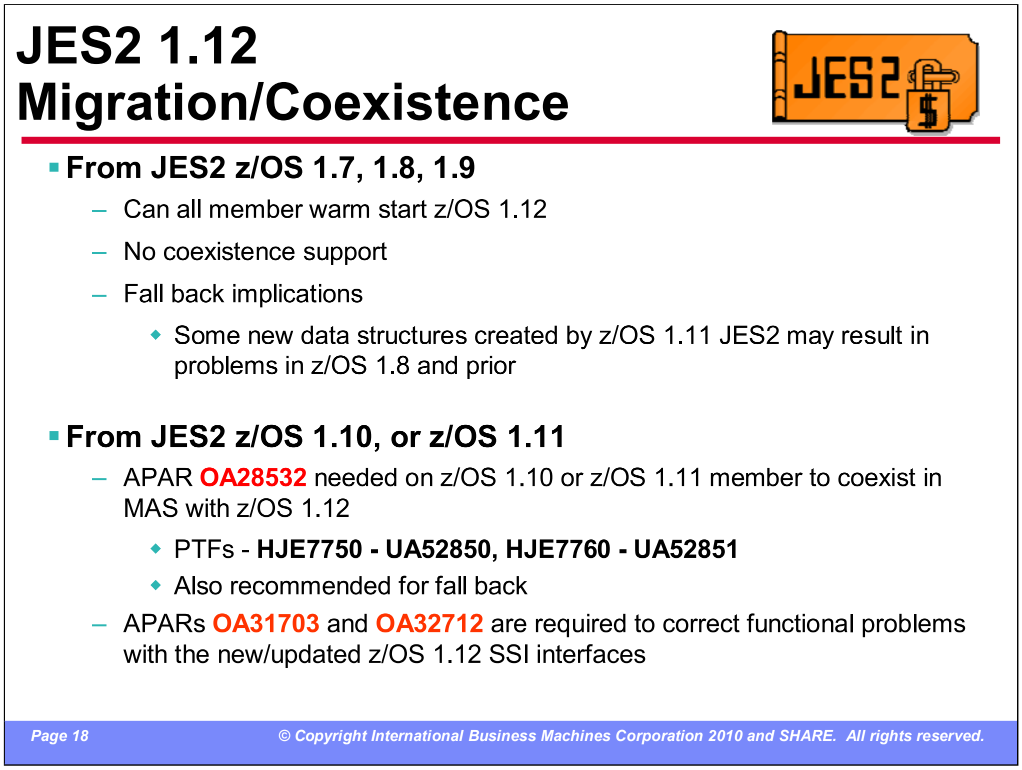
A number of serviceability updates were made in this release. This include:

•Improved formatting of the existing SAPI JES2 $TRACEs. The current traces dump pure hex data. To locate a specific field, you must know its offset. This makes searching a trace much more difficult. With these updates, all fields are formatted as field=value. This significantly increases the size of the output but makes processing the records easier.

•The standard JES2 $SAVE/$RETURN trace IDs 11, 12, 18 and 19 have been updated to include the SSOB and extension when tracing entry and exit to any SSI routine. The data is currently dumped in raw hex.

•The IPCS formatter $MODLOC (locates what JES2 module a passed address is in) has been updated to format the last APAR/PTF applied to the module.

•The $HASP473 message ($CPOOL Build failure) has been updated to include the JOBNAME that was trying to build the $CPOOL.



JES2 1.12 Migration/Coexistence

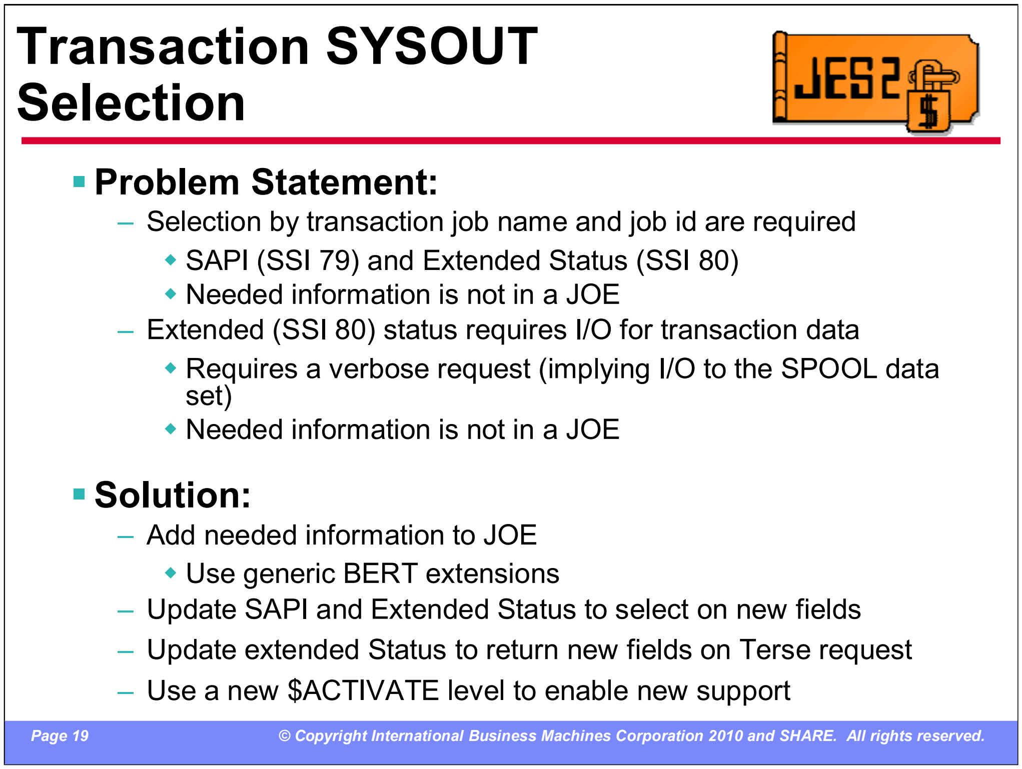
- **From JES2 z/OS 1.7, 1.8, 1.9**
  - Can all member warm start z/OS 1.12
  - No coexistence support
  - Fall back implications
    - Some new data structures created by z/OS 1.11 JES2 may result in problems in z/OS 1.8 and prior
- **From JES2 z/OS 1.10, or z/OS 1.11**
  - APAR **OA28532** needed on z/OS 1.10 or z/OS 1.11 member to coexist in MAS with z/OS 1.12
    - PTFs - **HJE7750 - UA52850, HJE7760 - UA52851**
    - Also recommended for fall back
  - APARs **OA31703** and **OA32712** are required to correct functional problems with the new/updated z/OS 1.12 SSI interfaces

Page 18

The recommended migration path to z/OS 1.12 is from z/OS 1.10 or z/OS 1.11. APAR OA28532 must be installed for MAS coexistence with z/OS 1.12 JES2. It is also highly recommended on down level releases for fallback support.

If you are migrating from z/OS 1.8 or 1.9, it is possible to do an all-member warm start to z/OS 1.12 However there is no coexistence support and fallback will result in problems with SYSLOG data sets printing on JES mode or RJE printers (due to changes in made in z/OS 1.11)

A direct migration from releases prior to z/OS 1.7 is not recommended.



Transaction SYSOUT Selection

- **Problem Statement:**
  - Selection by transaction job name and job id are required
    - SAPI (SSI 79) and Extended Status (SSI 80)
    - Needed information is not in a JOE
  - Extended (SSI 80) status requires I/O for transaction data
    - Requires a verbose request (implying I/O to the SPOOL data set)
    - Needed information is not in a JOE
- **Solution:**
  - Add needed information to JOE
    - Use generic BERT extensions
  - Update SAPI and Extended Status to select on new fields
  - Update extended Status to return new fields on Terse request
  - Use a new $ACTIVATE level to enable new support

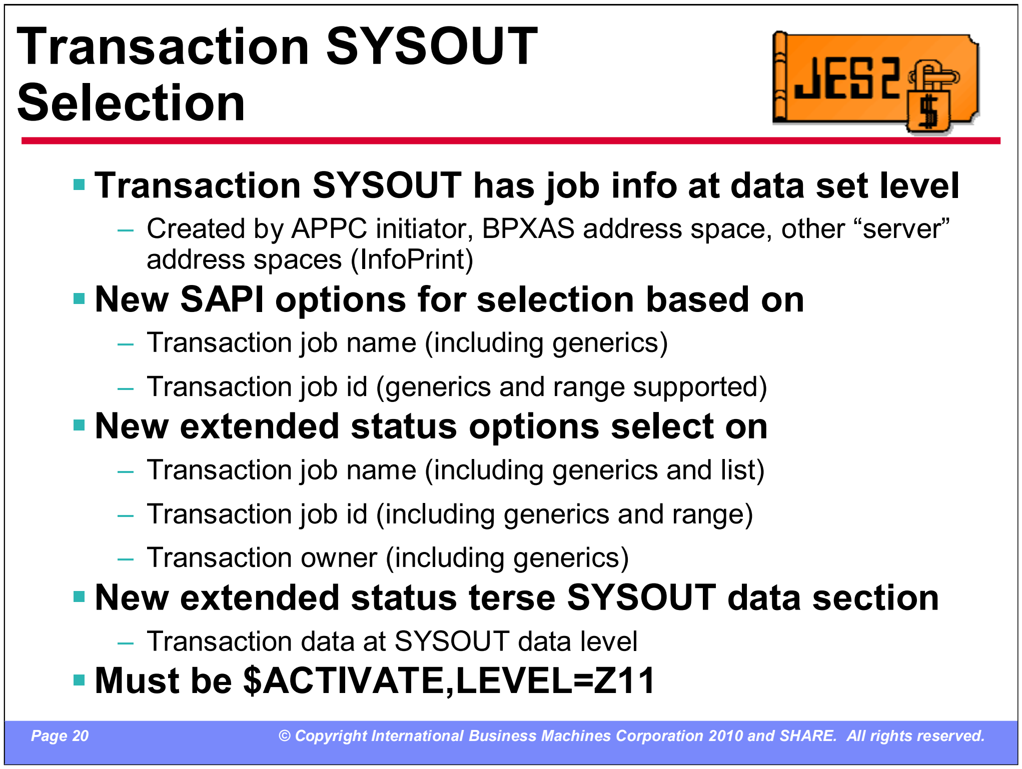
Currently, SAPI and extended status do not support selection by APPC (transaction) job name or job id. A number of vendors have requested this function. As more print work moves to z/OS and OMVS (BPXAS) application increase, the problem of selecting work by transaction identifier instead of address space name becomes greater. Also, the transaction information is only available in extended status if a verbose request is made. This implies SPOOL I/O for every SYSOUT element returned.

Fundamentally the problem is that the needed information is not in the checkpoint (associated with the JOE). It cannot be added to the JOE because the current structure has no space to place the needed data in. The JOE needs to be extended so that the transaction data can be added (and so that future requirements can be address).

To solve these problems, the JES2 $JOE data area has been enhanced to support BERT extensions much like what was done for the $JQE data area in OS/390 release 4. A new $DOGJOE service and macro was created to encapsulate the expanded JOE. This requires code which updates a JOE must first check the JOE out, update the checked out copy, and check it back in. This drove changes to many services that manipulate JOEs including some changes to basic interfaces to the service. This could have implications for some exits and user mods.

Since all members must be at the z/OS 1.11 level of JES2 for these services to be available, a new checkpoint level was created to allow installation to decide when the new processing is enabled. Similar to previous releases, the $ACTIVATE command has been re-introduced in this release. The new checkpoint level is for z11 mode (the current checkpoint level is known as z2 mode). $ACTIVATE can be used to do from z2 to z11 mode and back. The $D ACTIVATE command will inform you of the current checkpoint level and if it can be changed.

With the new $ACTIVATE level, the JOE fields needed by both SAPI and Extended Status to do filtering on transaction data are now available. Extended status can also now return transaction data in terse sections. At this point, JES2 has the flexibility to easily add additional fields to the $JOE.



Transaction SYSOUT Selection

- **Transaction SYSOUT has job info at data set level**
  - Created by APPC initiator, BPXAS address space, other “server” address spaces (InfoPrint)
- **New SAPI options for selection based on**
  - Transaction job name (including generics)
  - Transaction job id (generics and range supported)
- **New extended status options select on**
  - Transaction job name (including generics and list)
  - Transaction job id (including generics and range)
  - Transaction owner (including generics)
- **New extended status terse SYSOUT data section**
  - Transaction data at SYSOUT data level
- **Must be $ACTIVATE,LEVEL=Z11**

Page 20

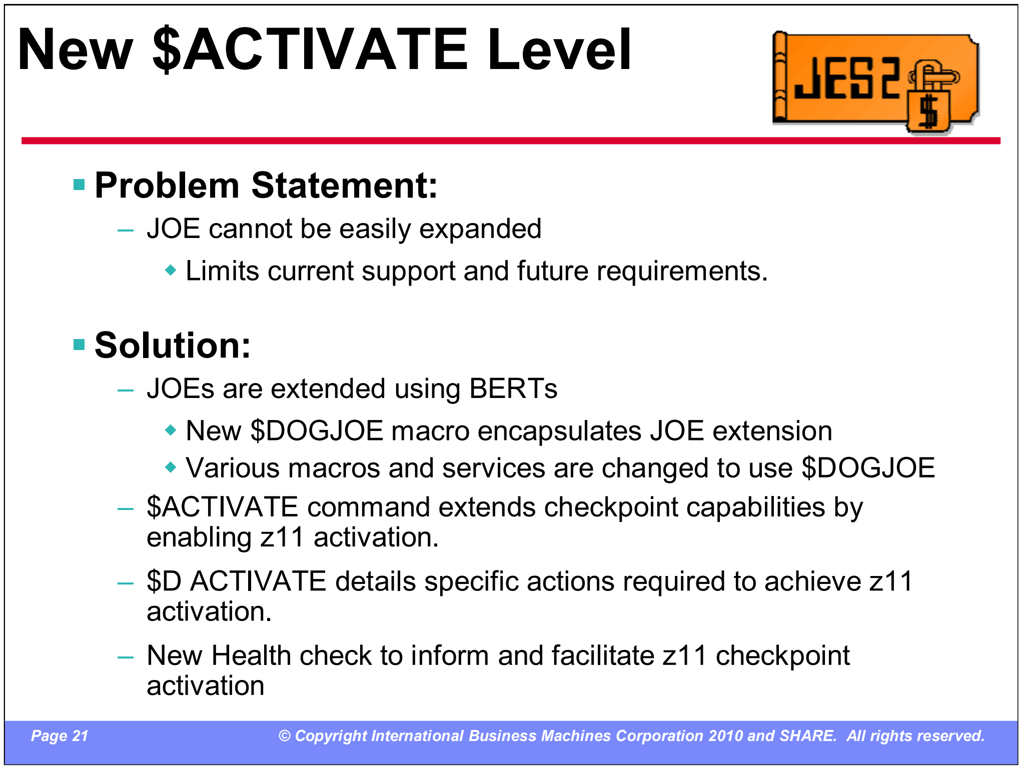
Transaction data is any SYSOUT data that has job level information associated with a SYSOUT data set. These are always SPIN data sets. The original creator of these data sets is APPC initiators. Later BPXAS address spaces exploited the same support. The support was generalized so that any address space can create transaction output by creating a task level JSAB using the IAZXJSAB macro. This is how “server” address spaces such as InfoPrint can create transaction output.

The SYSOUT level job name and job id can now be used to select SAPI output by setting the SSS2STPN and SSS2STPI selection bits. Job name supports generics and job id supports both generics and ranges.

Similarly, SYSOUT level job name and job id can be used to select SYSOUT using extended status. STATSTPN enables transaction job name, STATSTPI for transaction job id, and STATSTPU for transaction owner. The transaction owner allows you to select all jobs and transaction output owned by the same userid. The job name supports generics and the job name list. Transaction job id supports generics and a range. Transaction owner supports generics.

The SYSOUT terse data returned from extended status now includes a new transaction data section. If the SYSOUT has transaction data, this section (STSATERS section type) contains the transaction job name and job id.

All the transaction data is only available if the JESPLEX is in z11 checkpoint mode (z11 activation level).



New $ACTIVATE Level

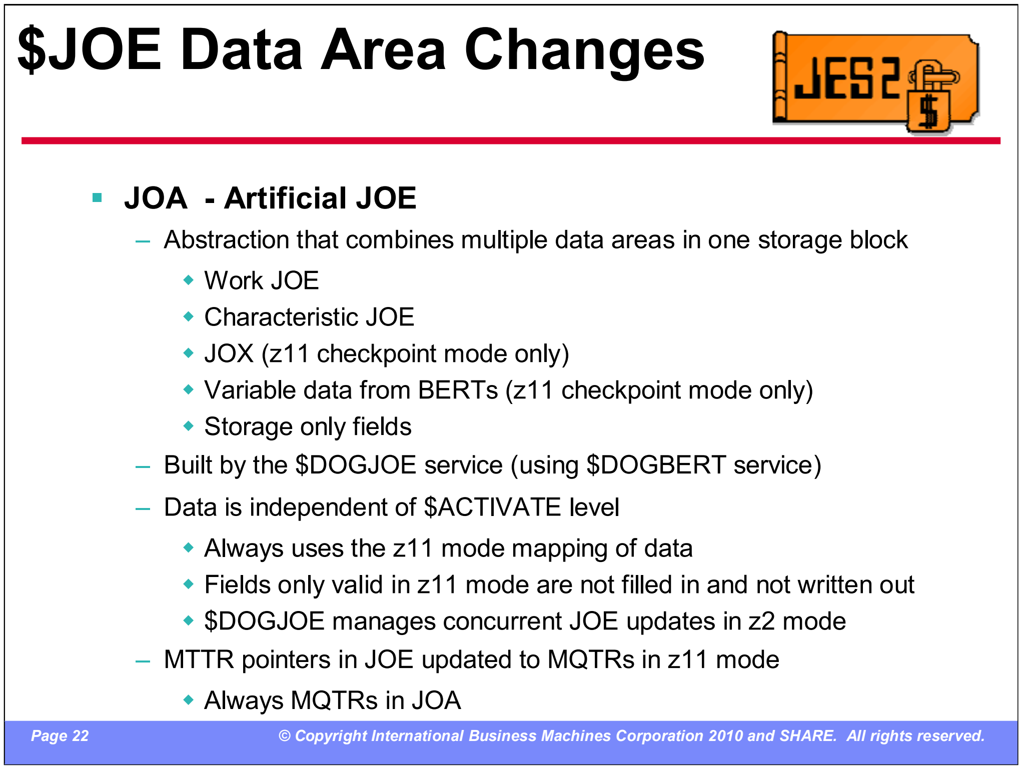
- **Problem Statement:**
  - JOE cannot be easily expanded
    - Limits current support and future requirements.
- **Solution:**
  - JOEs are extended using BERTs
    - New $DOGJOE macro encapsulates JOE extension
    - Various macros and services are changed to use $DOGJOE
  - $ACTIVATE command extends checkpoint capabilities by enabling z11 activation.
  - $D ACTIVATE details specific actions required to achieve z11 activation.
  - New Health check to inform and facilitate z11 checkpoint activation

Page 21

The current structure of the $JOE data area (represents a group of SYSOUT that is ready to print) can not be easily expanded without a cold start. This limits what can be done now and into the future. A solution is needed that allows additional data to be easily added to the checkpoint.

To solve these problems, the JES2 $JOE data area has been enhanced to support BERT extensions much like what was done for the $JQE data area in OS/390 release 4. A new $DOGJOE service and macro was created to encapsulate the expanded JOE. Code which updates a JOE must first check the JOE out, update the checked out copy, and check it back in. This drove changes to many services that manipulate JOEs including some changes to basic interfaces to the service. This could have implications for some exits and user mods.

Since all members must be at the z/OS 1.11 level of JES2 for these services to be available, a new checkpoint level was created to allow installation to decide when the new processing is enabled. Similar to previous releases, the $ACTIVATE command has been re-introduced in this release. The new checkpoint level is for z11 mode (the current checkpoint level is known as z2 mode). $ACTIVATE can be used to go from z2 to z11 mode and back. The $D ACTIVATE command will inform you of the current checkpoint level and if it can be changed. A new health check was added to display if you have met all the requirements for z11 mode and it will nudge the customer to $ACTIVATE.

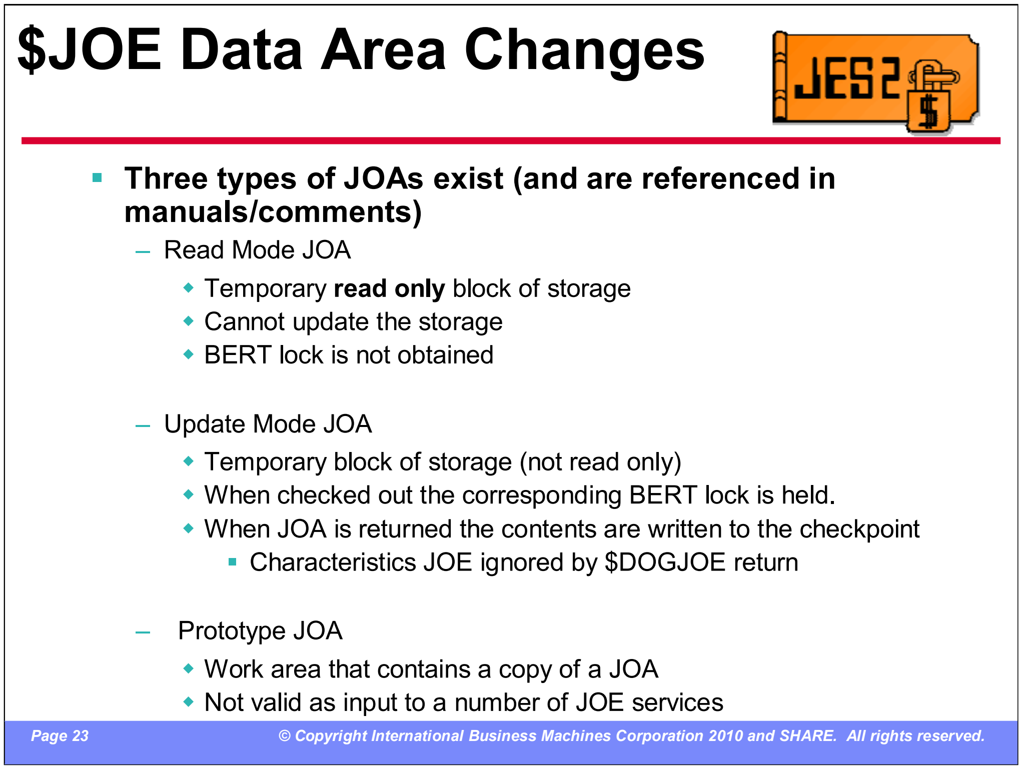


$JOE Data Area Changes

- **JOA - Artificial JOE**
  - Abstraction that combines multiple data areas in one storage block
    - Work JOE
    - Characteristic JOE
    - JOX (z11 checkpoint mode only)
    - Variable data from BERTs (z11 checkpoint mode only)
    - Storage only fields
  - Built by the $DOGJOE service (using $DOGBERT service)
  - Data is independent of $ACTIVATE level
    - Always uses the z11 mode mapping of data
    - Fields only valid in z11 mode are not filled in and not written out
    - $DOGJOE manages concurrent JOE updates in z2 mode
  - MTTR pointers in JOE updated to MQTRs in z11 mode
    - Always MQTRs in JOA

Page 22

New with this release is the concept of an artificial JOE. The is an abstraction that combines multiple data areas into a single structure for ease of reference. It contains copies of the work and characteristics JOE, the JOE extension (JOX), information from BERTs (obtained by the $DOGBERT service) and some storage only fields. The JOA is built and managed by the $DOGJOE service (we wanted to have a $DOG# - dollar dog pound – but could not get the name to work). This abstraction is independent of the current JES2 checkpoint mode. This implies that when building a JOA in z2 mode, fields that are only meaningful in z11 mode are not filled in and when returning a JOA in z2 mode, fields only valid in z11 mode are ignored. Serialization of JOE updates in z2 mode is handled by the $DOGJOE service.

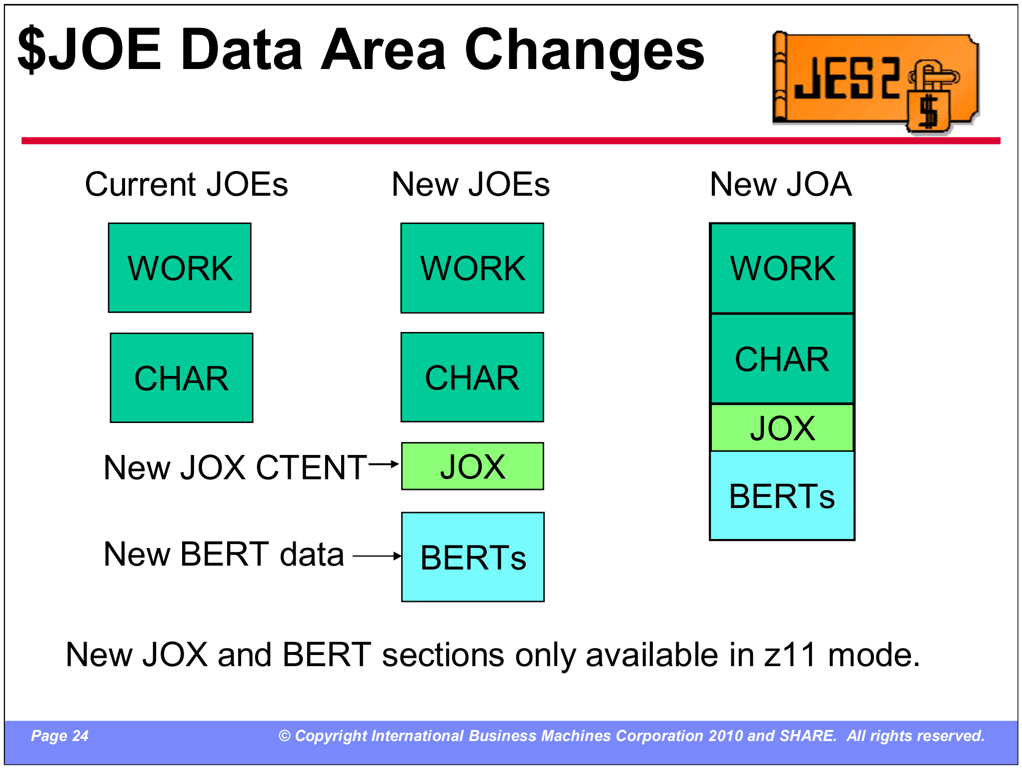


$JOE Data Area Changes

- **Three types of JOAs exist (and are referenced in manuals/comments)**
  - Read Mode JOA
    - Temporary **read only** block of storage
    - Cannot update the storage
    - BERT lock is not obtained
  - Update Mode JOA
    - Temporary block of storage (not read only)
    - When checked out the corresponding BERT lock is held.
    - When JOA is returned the contents are written to the checkpoint
      - Characteristics JOE ignored by $DOGJOE return
  - Prototype JOA
    - Work area that contains a copy of a JOA
    - Not valid as input to a number of JOE services

There are 3 types of artificial JOEs (JOAs) that are referred to in the product. They all contain the same data, but are used in different ways. The first two are created by the $DOGJOE service directly. A read mode JOA is a read only (actually storage key 0) version of a JOA built by $DOGJOE (directly or via the $#JOE service). Read mode JOAs are not serialized (they are not locked). Read mode JOAs are used when updating of the JOA is not needed such as display commands or while a JOE is printing. Update mode JOAs are updateable versions of a JOA built by $DOGJOE. The corresponding BERT lock is held by the PCE that checks out the JOA (or the lock is available and the PCE will not $WAIT). When the JOA is returned to $DOGJOE, the changes made are copied into the real work JOE, the JOX, and the BERT. Characteristic JOE information is available in the JOA and can be updated. But if the characteristics JOE changes need to be reflected in the checkpoint, a $#MOD must be done ($DOGJOE ignores the characteristic JOE information on a return request.

Finally, there is a prototype JOA. These are not built by $DOGJOE and cannot be passed into the $DOGJOE service. They are often imbedded in other data areas and may not contain all the BERT data that a regular JOA has. They can be used as input to services like $#ADD, but in general only exist to save JOA information where a read mode JOA is not practical.

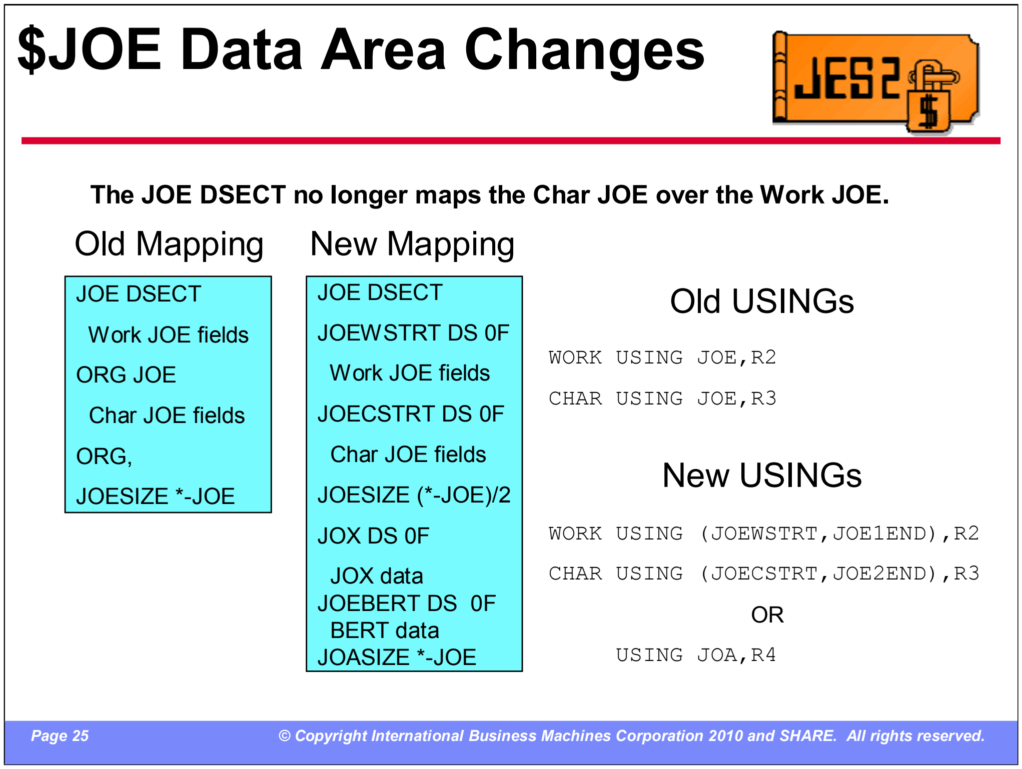


$JOE Data Area Changes

New JOX and BERT sections only available in z11 mode.

The existing work and characteristics JOE still exist in this release. The additional data that needed to be associated with the work JOE has been added to a new type of JOE BERT. To make room for the larger MQTR pointers, a new section has been added to the CKPT, the JOX. There is one JOX for every JOE, but only the JOX for the work JOE is currently used. MTTR fields in the work JOE have been moved to MQTR fields in the JOX. All these fields are then mapped into a single data structure, the artificial JOA by the $DOGJOE service.

In z2 mode, the JOEs look the same as they have in previous releases. There is no JOX nor data in BERTs. In z11 mode, the 3 MTTR fields in the work JOE are moved into new MQTR fields in the JOX, one of the old MTTR fields is re-used as the BERT token, and new data is placed in the BERTs. Regardless of mode, the JOA always appears to be in z11 mode (ie MTTR fields moved to MQTR field), This insulates code from the current $ACTIVATE level.



$JOE Data Area Changes

The JOE DSECT no longer maps the Char JOE over the Work JOE.

Old Mapping

```
JOE DSECT
  Work JOE fields
ORG JOE
  Char JOE fields
ORG,
JOESIZE *-JOE
```

New Mapping

```
JOE DSECT
JOEWSTRT DS 0F
  Work JOE fields
JOECSTRT DS 0F
  Char JOE fields
JOESIZE (*-JOE)/2
JOX DS 0F
  JOX data
JOEBERT DS  0F
  BERT data
JOASIZE *-JOE
```

Old USINGs

```
WORK USING JOE,R2
CHAR USING JOE,R3
```

New USINGs

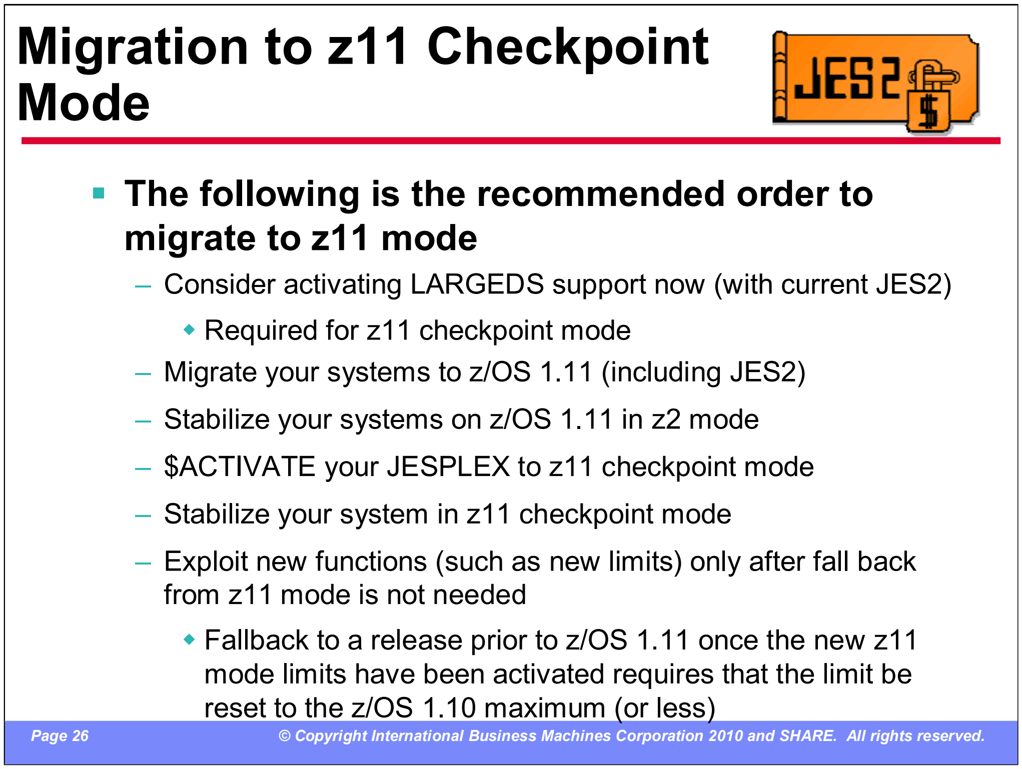
```
WORK USING (JOEWSTRT,JOE1END),R2
CHAR USING (JOECSTRT,JOE2END),R3
```

OR

```
     USING JOA,R4
```

Page 25

The mapping of the JOE DSECT has changed with this release. In previous releases, the Char JOE was mapped over the Work JOE in the DSECT. This has lead to many errors over the years when codes incorrectly accesses work JOE fields in a char JOE or visa versa. Because of this, and to simplify accessing the new JOA structure, the $JOE data area has been re-mapped. The char JOE now follows the work JOE in the mapping. New JOX and BERT fields follow those areas. This requires a change to any USINGs that are used to address the characteristics JOE. The JOE (or JOA) label should in general only be used to address an artificial JOA. The JOEWSTRT and JOECSTRT should be used to address the work and characteristics JOEs. It is strongly suggested that the using specify the proper limits so that they are not used accidentally to access data beyond the end of the particular JOE you are pointing to.



Migration to z11 Checkpoint Mode

- **The following is the recommended order to migrate to z11 mode**
  - Consider activating LARGEDS support now (with current JES2)
    - Required for z11 checkpoint mode
  - Migrate your systems to z/OS 1.11 (including JES2)
  - Stabilize your systems on z/OS 1.11 in z2 mode
  - $ACTIVATE your JESPLEX to z11 checkpoint mode
  - Stabilize your system in z11 checkpoint mode
  - Exploit new functions (such as new limits) only after fall back from z11 mode is not needed
    - Fallback to a release prior to z/OS 1.11 once the new z11 mode limits have been activated requires that the limit be reset to the z/OS 1.10 maximum (or less)

Page 26

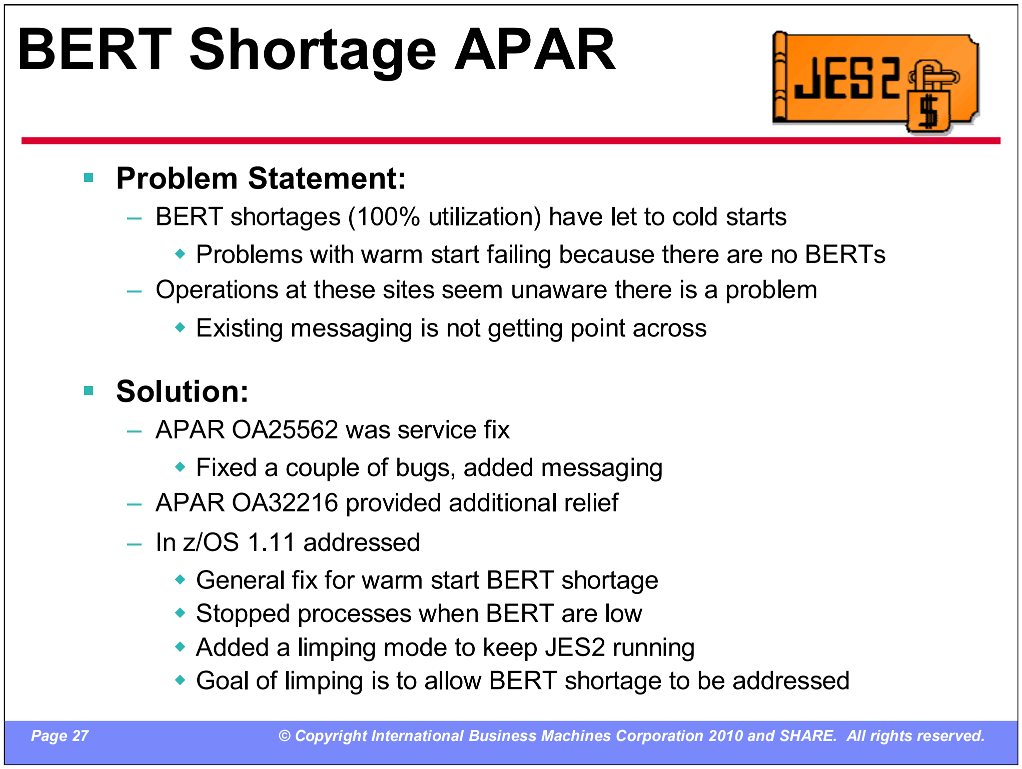
Fully exploiting z11 checkpoint mode requires 3 basic steps.

•Activate LARGEDS support (can be done before or after migrating to z/OS 1.11 JES2)

•Migrate to z/OS 1.11 JES2

•$ACTIVATE z11 mode

At each step along the way, you should ensure that your system is stable before moving to the next step. The code is designed to make it easy to fall back one step. But having to fall back two steps may require additional effort. That is why it is important to ensure your system is stable before moving to the next step.

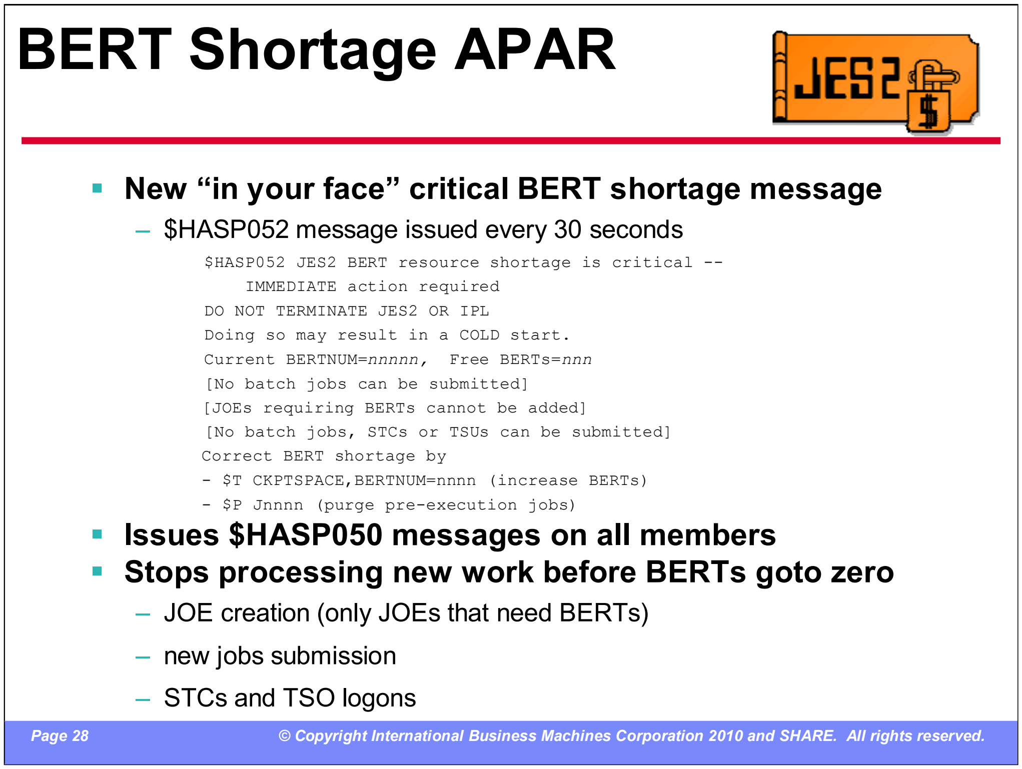


BERT Shortage APAR

- **Problem Statement:**
  - BERT shortages (100% utilization) have let to cold starts
    - Problems with warm start failing because there are no BERTs
  - Operations at these sites seem unaware there is a problem
    - Existing messaging is not getting point across
- **Solution:**
  - APAR OA25562 was service fix
    - Fixed a couple of bugs, added messaging
  - APAR OA32216 provided additional relief
  - In z/OS 1.11 addressed
    - General fix for warm start BERT shortage
    - Stopped processes when BERT are low
    - Added a limping mode to keep JES2 running
    - Goal of limping is to allow BERT shortage to be addressed

When BERTs were introduced in OS/390 release 4, there were a number of problems that occurred when we ran out of BERTs. The most severe of these problems was a shortage of BERTs during warm start processing. There were processes that needed BERTs during warm start, but if there were no BERTs, available, warm start could not continue. This would force a cold start. This problem was addressed back in OS/390 by fixing each problem area we encountered. But the problem has crept back.

OA25562 has fixed the major problems that occur when there are no BERTs available on a warm start and has added some additional messaging. But z/OS 1.11 has gone further. It has introduced a limping mode into JES2. If JES2 is in this mode, it has given up on some process that needed a BERT and continued without completing that process. The goal is to get JES2 running so that the installation can address the BERT shortage problem. Once the problem has been addressed, JES2 must be restarted (hot start works).



BERT Shortage APAR

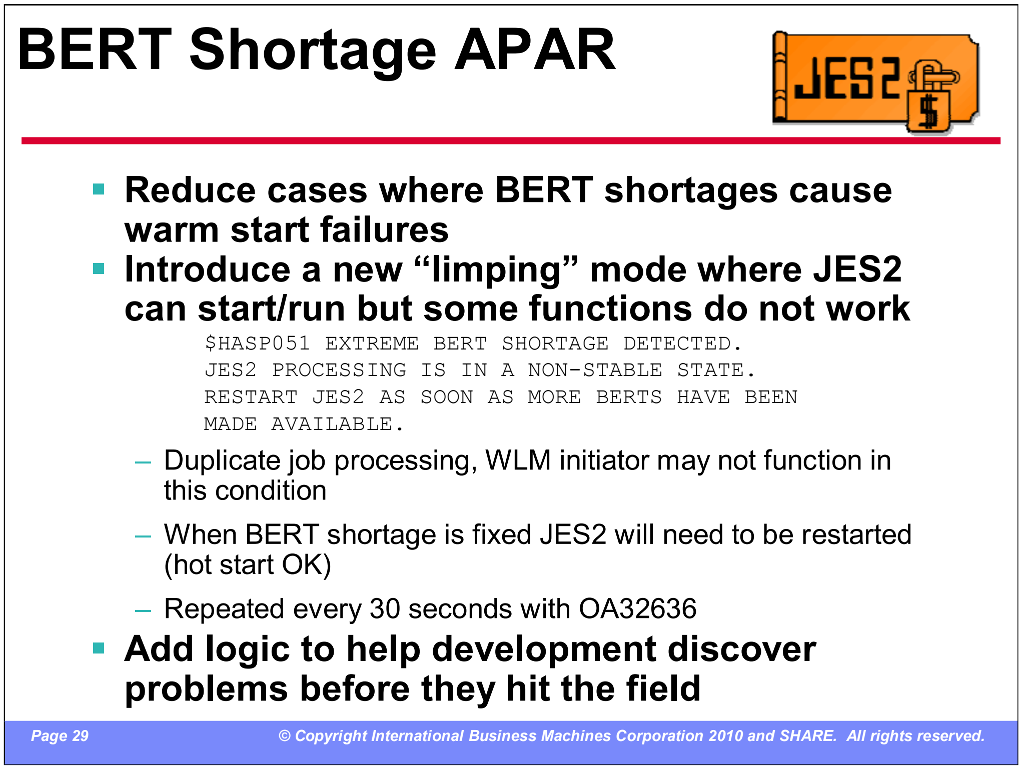
- **New "in your face" critical BERT shortage message**
  - $HASP052 message issued every 30 seconds

```
$HASP052 JES2 BERT resource shortage is critical --
    IMMEDIATE action required
DO NOT TERMINATE JES2 OR IPL
Doing so may result in a COLD start.
Current BERTNUM=nnnnn,  Free BERTs=nnn
[No batch jobs can be submitted]
[JOEs requiring BERTs cannot be added]
[No batch jobs, STCs or TSUs can be submitted]
Correct BERT shortage by
- $T CKPTSPACE,BERTNUM=nnnn (increase BERTs)
- $P Jnnnn (purge pre-execution jobs)
```

- **Issues $HASP050 messages on all members**
- **Stops processing new work before BERTs goto zero**
  - JOE creation (only JOEs that need BERTs)
  - new jobs submission
  - STCs and TSO logons

Page 28

One thing that came up in a review of customer situations where a BERT shortage resulted in a cold start, was a total disregard for the existing $HASP050 message that warned of the problem. In one case, where the system programmer was aware of the HASP050 messages, there was no appreciation of what damage a BERT shortage could cause. To address this, a new message that is explicit about the implications of a BERT shortage and is issued when BERT utilization drops below 256 free BERTs. The HASP050 message is now issued on all members of the MAS (for other MAS level limits, it is only issued on one member of the MAS). Another change was to suspend JOE creation, input processing for JOBs and input processing STCs/TSUs as the number of free BERTs drops. Originally, the limits were 32, 16, and 8 respectively. APAR OA32216 increased this upper limit to 2x maximim BERTs for JQA plus 2x maximum for CAT plus maximum for WSC This is 48 for z/OS 1.9 an 1.10, 50 for z/OS 1.11 and 1.12. This should slow down consumption of BERTs at a time when the limits are critical.



BERT Shortage APAR

- **Reduce cases where BERT shortages cause warm start failures**
- **Introduce a new “limping” mode where JES2 can start/run but some functions do not work**

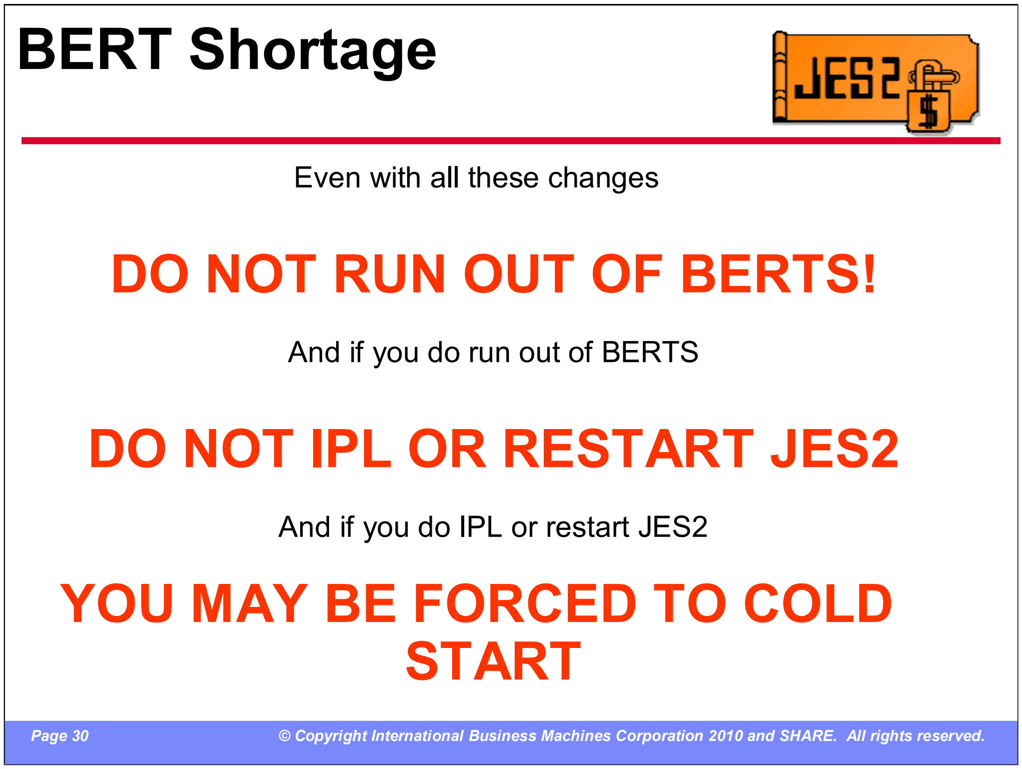
```
$HASP051 EXTREME BERT SHORTAGE DETECTED.
JES2 PROCESSING IS IN A NON-STABLE STATE.
RESTART JES2 AS SOON AS MORE BERTS HAVE BEEN
MADE AVAILABLE.
```

  - Duplicate job processing, WLM initiator may not function in this condition
  - When BERT shortage is fixed JES2 will need to be restarted (hot start OK)
  - Repeated every 30 seconds with OA32636
- **Add logic to help development discover problems before they hit the field**

Page 29

A more extensive fix has been added to z/OS 1.11 JES2. Code has been added to internally fail processing that could not get the needed BERTs during warm start processing BUT to continue to allow JES2 to initialize. This implies that some data structures may not be updated correctly or may not be created. Some JES2 functions may not work correctly due to this. However, because we are allowing JES2 to initialize, it is possible to free sufficient BERTs to allow a normal JES2 start to continue. In this state, JES2 is essentially broken. It is “limping” but must be restarted as soon as the BERT shortage has been relieved. JES2 issues the $HASP051 message to inform the operator that we are in this state. The only way to restore normal operations is to restart JES2 (a hot start is all that is required). In this state, jobs with duplicate jobs names may run at the same time. WLM initiators may not work. There may be problems with submitting jobs, creating output, logging on a TSO user.

Diagnostic code was added to help us discover BERT shortage related problems earlier in the development phase. This is an attempt to avoid getting into this limping mode of operations.



BERT Shortage

Even with all these changes

DO NOT RUN OUT OF BERTS!

And if you do run out of BERTS

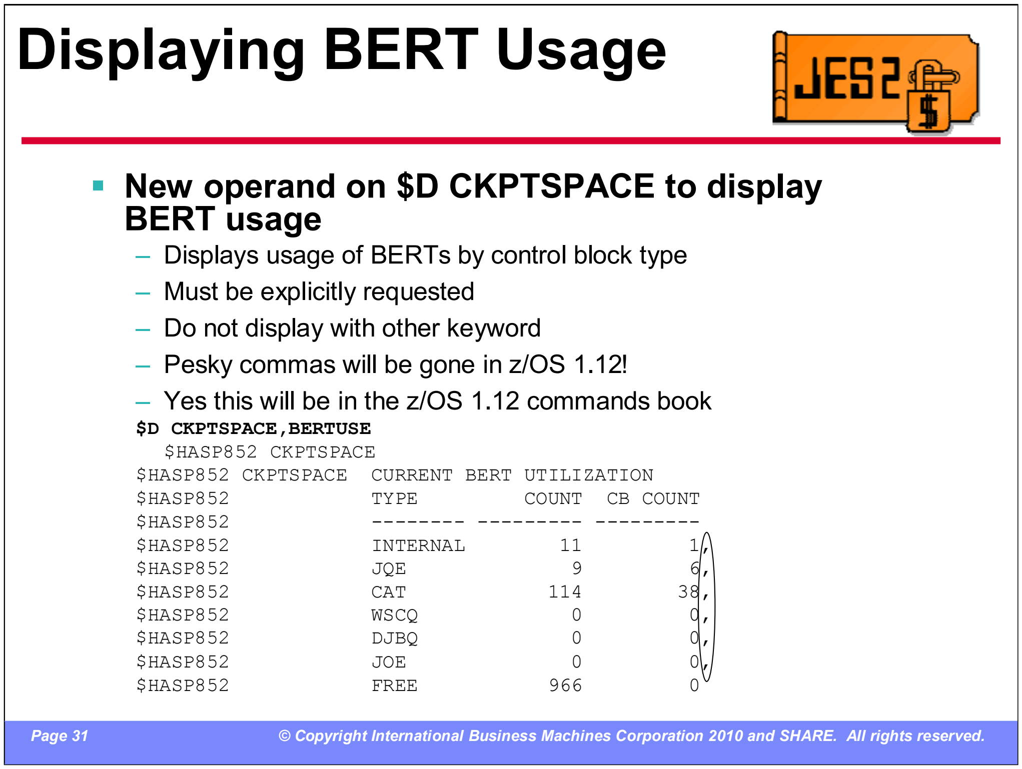
DO NOT IPL OR RESTART JES2

And if you do IPL or restart JES2

YOU MAY BE FORCED TO COLD START

z/OS 1.11 has added more cases when we use BERTs. We believe that we have addressed all the issues with BERT shortages in this release. We believe that we can start JES2 when there is a BERT shortage. However, we have given you the proverbial loaded gun and told you the safety is on. You could point it at your foot and pull the trigger. You would be assuming you trust the safety (and that we have set it properly). But the best advice is to NEVER trust the safety and never point a gun (loaded or otherwise) at your foot (or anything other appendage you may be fond of). If you NEVER run out of BERTs, then you will never have to test our safety net.

To display what BERTs are being used for, issue the command $D CKPTSPACE,BERTUSE.



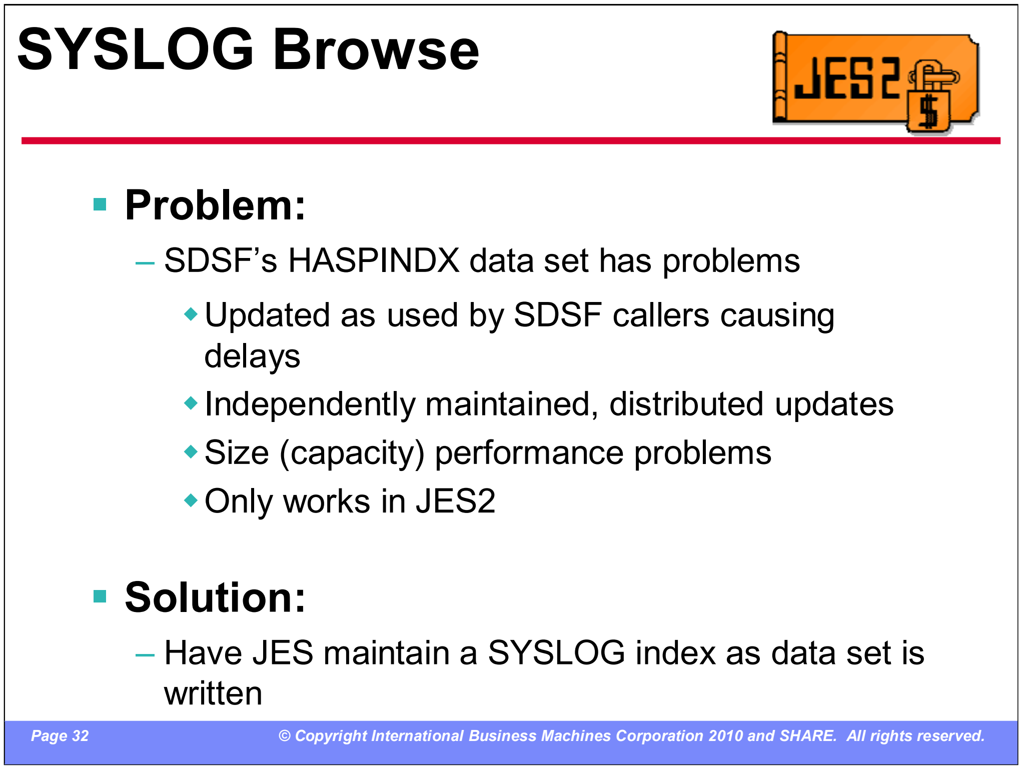
Displaying BERT Usage

- **New operand on $D CKPTSPACE to display BERT usage**
  - Displays usage of BERTs by control block type
  - Must be explicitly requested
  - Do not display with other keyword
  - Pesky commas will be gone in z/OS 1.12!
  - Yes this will be in the z/OS 1.12 commands book

```
$D CKPTSPACE,BERTUSE
  $HASP852 CKPTSPACE
$HASP852 CKPTSPACE  CURRENT BERT UTILIZATION
$HASP852            TYPE         COUNT  CB COUNT
$HASP852            -------- --------- ---------
$HASP852            INTERNAL        11         1,
$HASP852            JQE              9         6,
$HASP852            CAT            114        38,
$HASP852            WSCQ             0         0,
$HASP852            DJBQ             0         0,
$HASP852            JOE              0         0,
$HASP852            FREE           966         0
```

To assist in determining what control blocks are using BERT, a new operand BERTUSE was added to the $D CKPTSPACE command. This operand must be explicitly specified (does not appear in a normal $D CKPTSPACE command). For each control block type, it displays the current number of BERTs assigned to that control block type and how many control blocks are represented in that count. In the example given, there are 9 BERTs being used by 6 JQEs.

Please note the pesky commas are gone in z/OS 1.12. You can stop sending us your cards and letters (and PMRs). Please tolerate them until then.

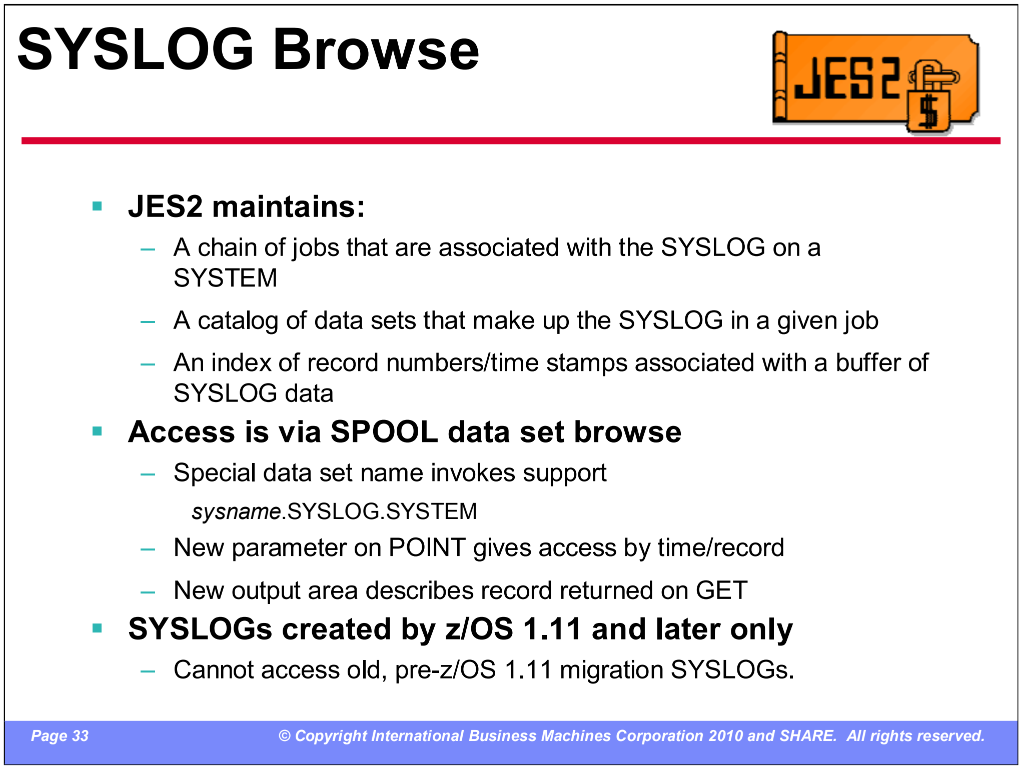


SYSLOG Browse

- **Problem:**
  - SDSF’s HASPINDX data set has problems
    - Updated as used by SDSF callers causing delays
    - Independently maintained, distributed updates
    - Size (capacity) performance problems
    - Only works in JES2
- **Solution:**
  - Have JES maintain a SYSLOG index as data set is written

Currently, the SDSF SYSLOG function depends on a single SYSPLEX scope data set, HASPINDX, to access SYSLOG data sets across the MAS. This data set is maintained by SDSF running in various TSO address spaces as users look at the SYSLOG data set. This causes a number of problems. Serialization and data corruption is one problem (each updater of the data set must serialize the update, but there can be data corruption if a user ABENDs while accessing the data set). Another problem is performance. In the morning, the first person who accesses SYSLOG can experience a significant delay waiting for the HASPINDX data set to be updated. There are also performance issued related to the size of the SYSLOG data sets. These problems are well known among SDSF users. Additionally, the HASPINDX data set only supports JES2. SDSF would need to re-create the function to support JES3.

To address this, JES (both JES2 and JES3) will now maintain an index for the SYSLOG data set as it is created. The index will track both the jobs/data sets that make up the logical SYSLOG data set and the individual records in the data set. The index allows rapid access to records in the SYSLOG by record number and by time. The SPOOL data set browse interface was updated to allow browsing of the logical SYSLOG data set for a system using this index.



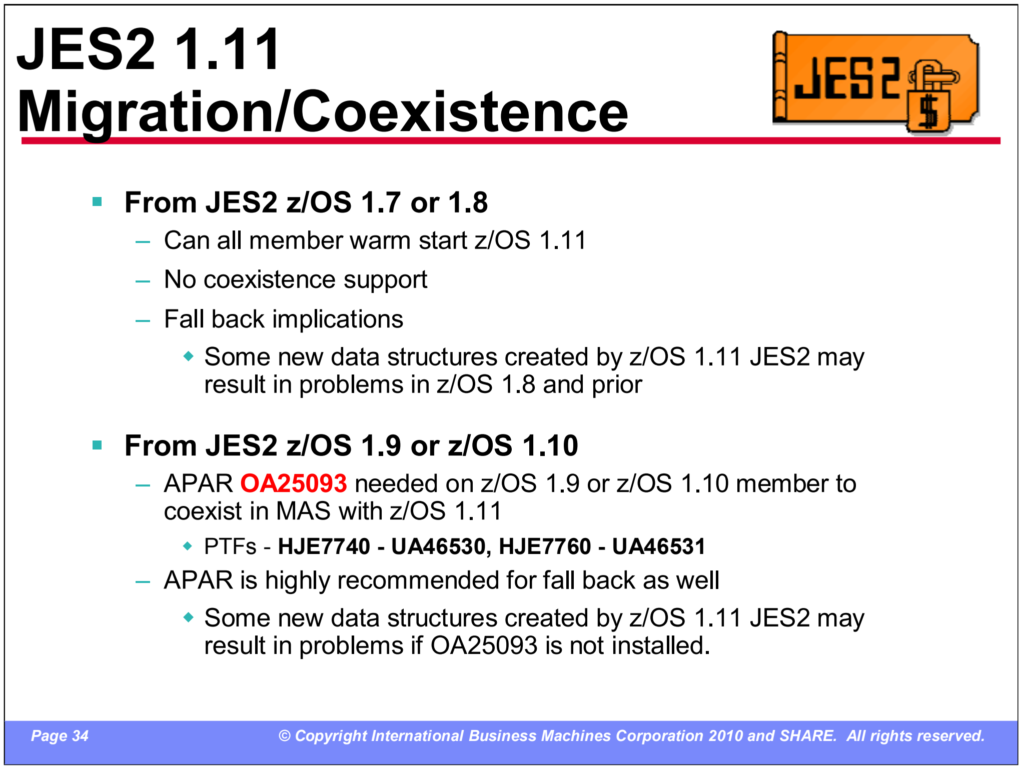
SYSLOG Browse

- **JES2 maintains:**
  - A chain of jobs that are associated with the SYSLOG on a SYSTEM
  - A catalog of data sets that make up the SYSLOG in a given job
  - An index of record numbers/time stamps associated with a buffer of SYSLOG data
- **Access is via SPOOL data set browse**
  - Special data set name invokes support

    sysname.SYSLOG.SYSTEM
  - New parameter on POINT gives access by time/record
  - New output area describes record returned on GET
- **SYSLOGs created by z/OS 1.11 and later only**
  - Cannot access old, pre-z/OS 1.11 migration SYSLOGs.

The HASPINDX support that currently exists in SDSF has been replaced with a new mechanism within JES2. As part of this support, JES2 maintains a chain of the SYSLOG JQEs that exists for each SYSTEM in the MAS. There are 2 new index structures on SPOOL that track the SPIN data sets that the SPOOL job creates, and the buffers with each data set. These indexes are maintained as the SYSLOG data sets are created (and not built by the application as was done with the HASPINDX data set). The buffer index maps a record number and time stamp to a SPOOL data record.

These new structures are used when the SYSLOG is access via SPOOL data set browse. Allocation is the same as any other SPOOL data set however, a special data set name triggers the SYSLOG allocation. Once allocated and opened, the logical SYSLOG can be accessed using the existing SPOOL browse access methods. In addition, a new format for the argument to POINT is supported to locate a record by time or a specific record. Also, GET processing can return additional information about the record returned.



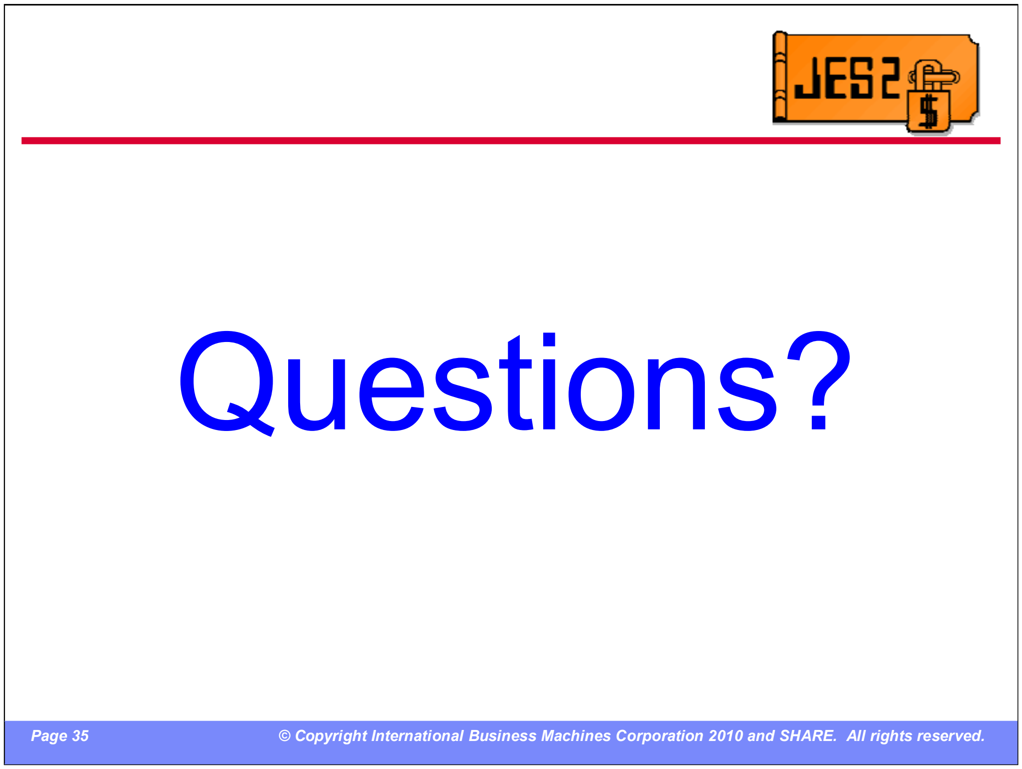
JES2 1.11 Migration/Coexistence

- **From JES2 z/OS 1.7 or 1.8**
  - Can all member warm start z/OS 1.11
  - No coexistence support
  - Fall back implications
    - Some new data structures created by z/OS 1.11 JES2 may result in problems in z/OS 1.8 and prior
- **From JES2 z/OS 1.9 or z/OS 1.10**
  - APAR **OA25093** needed on z/OS 1.9 or z/OS 1.10 member to coexist in MAS with z/OS 1.11
    - PTFs - **HJE7740 - UA46530, HJE7760 - UA46531**
  - APAR is highly recommended for fall back as well
    - Some new data structures created by z/OS 1.11 JES2 may result in problems if OA25093 is not installed.

The recommended migration path to z/OS 1.11 is from z/OS 1.9 or z/OS 1.10. APAR OA25093 must be installed for MAS coexistence with z/OS 1.11 JES2. It is also highly recommended on down level releases for fallback support. There are new data structures created in JES2 z/OS 1.11 related to the SYSLOG indexing function that cause problems with printing a SYSLOG data set on a JES mode or RJE printer.

If you are migrating from z/OS 1.7 or 1.8, it is possible to do an all-member warm start to z/OS 1.11 However there is no coexistence support and fallback will result in problems with SYSLOG data sets printing on JES mode or RJE printers.

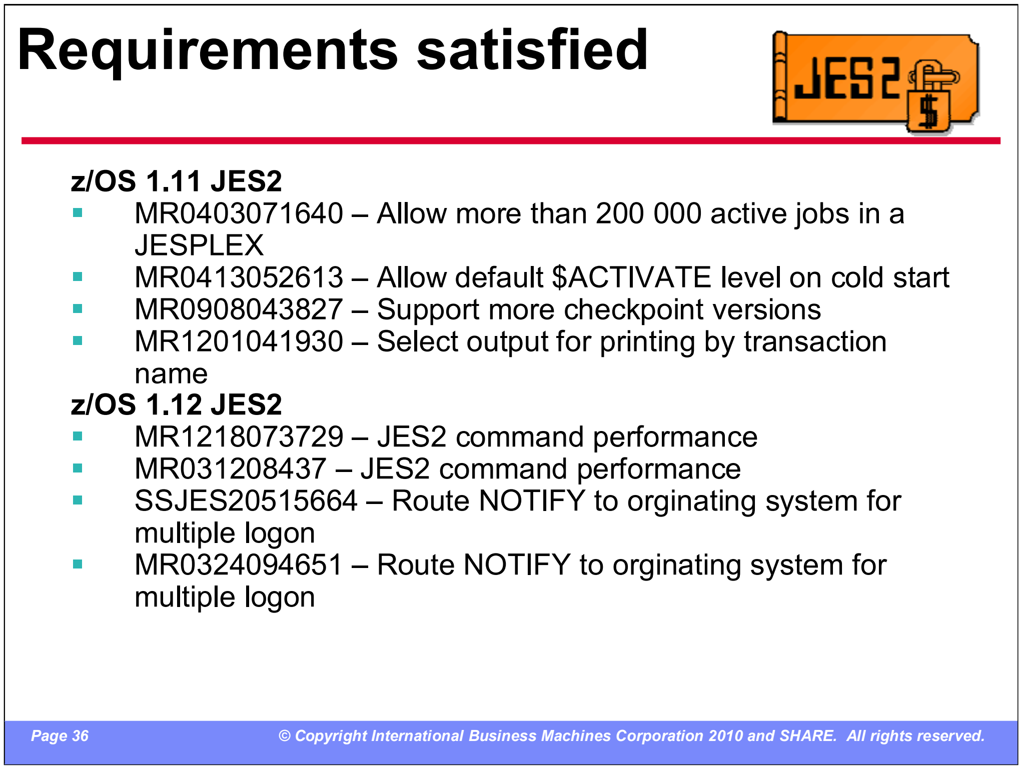
A direct migration from releases prior to z/OS 1.7 is not recommended.



Questions?

Page 35

Any questions???



Requirements satisfied

z/OS 1.11 JES2

- MR0403071640 – Allow more than 200 000 active jobs in a JESPLEX
- MR0413052613 – Allow default $ACTIVATE level on cold start
- MR0908043827 – Support more checkpoint versions
- MR1201041930 – Select output for printing by transaction name

z/OS 1.12 JES2

- MR1218073729 – JES2 command performance
- MR031208437 – JES2 command performance
- SSJES20515664 – Route NOTIFY to orginating system for multiple logon
- MR0324094651 – Route NOTIFY to orginating system for multiple logon