

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

SHARE Anaheim, 2011
Session 8688 – Monday, 4:30 PM
(LONG SESSION)

Permission is granted to SHARE Inc. to publish this presentation in the SHARE proceedings. IBM retains its right to distribute copies of this presentation to whomever it chooses.

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 preview of z/OS 1.13 JES2.

Because the JES2 latest status and new release presentations were combined this year, the material traditionally in the JES2 latest status presentation is also included and will be presented during the “long” part of this session.

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.

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 1.13 Preview**
 - Batch Modernization
 - ◆ Instream data in PROCs/Includes
 - ◆ Controlling job return codes
 - ◆ Spin any spin data set
 - ◆ Remove job from execution and restart on a step boundary
- **JES2 Latest Status**

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 1.13 has made a number of updates to improve BATCH job processing. These items include support for instream data in PROCs and included, a mechanism for controlling job return codes, the ability to spin any spin data set by time, size, or operator command, and a feature that allows jobs to be removed from execution and restarted on a step boundary.

This presentation will also include JES2 latest status information.

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

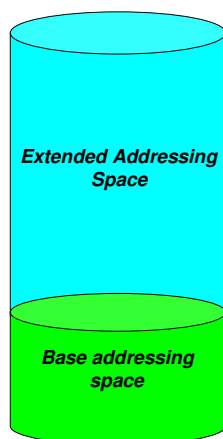
© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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



*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.**

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

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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 than 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.

Page 8

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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.

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.

\$\$ 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

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

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

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

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

Page 16

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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
 - ◆ SAPIID and IAZSS2 fields are traced
- **SSOB and extension added to \$SAVE and \$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**

Page 17

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

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

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.

z/OS 1.13 Preview



■ Instream data in PROCs and INCLUDEs

- Simplifies writing JCL PROCs
 - ◆ No need for separate control data set
- Support DD * and DD DATA in full in PROCs and INCLUDEs
 - ◆ Works with instream PROCs
 - ◆ No automatic generation of SYSIN DD * like JCL
- Works for all users of PROC (batch and started tasks)
 - ◆ Job must run under JES2 (no MSTR subsystem)
- Must convert on a z/OS 1.13 member
 - ◆ Can run on any level member

Page 19

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

z/OS 1.13 added support for instream data sets in JCL PROCs and INCLUDEs. This allows JCL coders to combine the JCL and control data sets in one PROC member. The DD * and DD DATA JCL cards and all their operands can now be placed in a JCL PROC followed by the instream data. During conversion processing, this instream data will be stripped out and placed into a JES2 instream data set. When the job runs, it can then access this data set like it would an instream data set included in the JCL. The only difference between this support and standard instream support is JES2 would generate a //SYSIN DD * card whenever it encountered a non-JCL card in the job stream. With this support, if a non-JCL card is encountered in a PROC or INLCUDE outside a DD * or DD DATA, it will continue to be flagged as an error.

This support works for all users of PROCs started tasks and batch jobs. However it only works for jobs running under a JES2 subsystem (does not work if the job is run under the master subsystem). Once z/OS 1.13 is installed, all that is required is that he job convert on a z/OS 1.13 member. It can later execute on any level member.

This instream data sets are not part of the submitted JCL so they are not included when looking at the original JCL (eg via SDSF SJ command) and are not sent over NJE to other nodes. They do appear in the data set list (eg via SDSF ? – JDS command).

z/OS 1.13 Preview



▪ Instream data in PROC example

```
//HELLO   PROC
//STEP1  EXEC ASMHCLG
//C.SYSIN DD *
TEST     CSECT ,
          STM 14,12,12(13)
          BALR 12,0
          USING *,12
          ST 13,SAVAREA+4
          LA 13,SAVAREA
          SPACE 1
          WTO 'Hello world!'
          SPACE 1
          L 13,SAVAREA+4
          LM 14,12,12(13)
          SR 15,15
          BR 14
          SPACE 1
SAVAREA  DC 18F'0'
          END
//L.TEST DD DUMMY
//L.SYSXX DD *
//       PEND
```

Page 20

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

Here is a simple example of a PROC that calls another nested PROC and has an instream input. This can be run in a batch job or by just doing a S HELLO.

```
s hello
$HASP100 HELLO      ON STCINRDR
$HASP373 HELLO      STARTED
+Hello world!
$HASP395 HELLO      ENDED
```

z/OS 1.13 Preview



- **New job card operand to control job RC**
 - JOBRC= MAXRC | LASTRC | (STEP,*name.name*)
 - ♦ MAXRC is existing processing (default)
 - ♦ LASTRC is return code of last step
 - ♦ (STEP,*name.name*) is return code of identified step
 - If step not executed, defaults to MAXRC
- **Affects return code seen in**
 - Extended status (eg SDSF)
 - ENF 70
 - HASP165 message
 - \$DJ,CC= command
- **JOBCLASS JOBRC= MAXRC|LASTRC to affect processing for all jobs in the job class**
- **Two additional error case return codes defined**
 - CONVERTER ERROR – Conversion processing ABENDED processing the job
 - SYSTEM FAILURE – System crashed while job was running and job could not be restarted.

Page 21

© Copyright International Business Machines Corporation 2011 and SHARE. All rights reserved.

The success or failure of a JCL stream is not always determined by the highest completion code of any step. Sometimes it is the completion code of the last step or sometimes it is the completion code of a specific step. The new JOBRC= operand on the JOB card now gives the JCL writer the ability to control what completion code will be presented for the job.

There are 3 values for JOBRC, MAXRC (current and default processing), LASTRC (use the return code of the last step) or (STEP,*name.name*) to specify a specific step. If a specific step is specified and that step does not run, then the processing goes back to the maximum return code.

The return code for the job is presented in extended status (SDSF), ENF 70, the \$HASP165 message and the \$DJ CC= command.

An installation can elect to change the default processing for JOBRC on a job class basis by using the new JOBRC= setting on the JOBCLASS JES2 parameter.

Two new conditions for job failure were also added:

- CONVERTER ERROR – for when conversion processing for the job ABENDs
- SYSTEM FAILURE – for when the job was running when the system crashed and the job was not restartable.

z/OS 1.13 Preview



- **Added function to spin any spin data set**
 - Similar to what was done for JESLOG
 - Applies to any data set allocated as SPIN
 - ◆ No application code/JCL change needed
 - Spin based on size, time, operator command
- **Update to SPIN= DD operand**
 - SPIN=(UNALLOC,*option*)
 - ◆ '*hh:mm*' - Spin at specific time
 - ◆ '+*hh:mm*' - Spin every hh:mm interval
 - ◆ *nnn*, *nnnK*, *nnnM* - Spin every nnn lines
 - ◆ NOCMND - Cannot be spun by command
 - ◆ CMNDONLY - Can be spun via operator command (default if no interval)
- **\$TJn,SPIN,DDNAME=*name* command added**

JESLOG SPIN processing was added in z/OS 1.2 to allow long running jobs to spin their job log and system messages data sets. However, many jobs have additional log data sets that could consume large amounts of SPOOL space. z/OS 1.13 extended the JESLOG support to any SPIN data set the job may allocate. The SPIN= operand on the DD statement was enhanced with an optional value indicating when a SPIN data set should be spun off. The spin option can be a time of day, and interval, or a size. In addition, the function can be disabled by specifying NOCMND. By default spin data sets can be spun by operator command. The DDNAME= operand was added to the \$TJ.SPIN command to spin a specific data set.

z/OS 1.13 Preview



▪ Remove job on step boundary

- New STEP operand on \$EJ command
 - ◆ Causes job to exit execution at end of current step
 - ◆ Optional HOLD operand makes job held
 - ◆ Job is queued for execution
- Job must be journaling (JOURNAL=YES on JOBCLASS)
- Uses existing continue restart function of z/OS
 - ◆ Previously used to restart jobs after an IPL
- Full syntax \$EJxxx,STEP [,HOLD]
 - ◆ Full cross member support

Long running jobs can be an issue when an installation is trying to shutdown a system. If the job has multiple long running steps, this enhancement can help the situation. A new STEP operand was added to the \$E J command (restart job). When specified, the job will be removed from execution at the next step boundary. The job must be restartable for the command to be accepted (must have a JES journal data set). This support utilizes the existing continue restart function on z/OS to restart the job from the next step.

Once removed from execution, the job is placed back in awaiting execution state and optionally held. If the job is not held, it is assumed that the appropriate classes or initiators were drained so the job will not resume execution on the member where it was executing.

Current JES2 Releases



FMIDs, Birthdays & Obituaries

JES2 Rel.#	FMID	First Available	No Longer Available	End of Service
z/OS 1.4	HJE7707	9/02	9/04	3/07
z/OS 1.5	HJE7708	3/04	9/04	9/07
z/OS 1.6	HJE7708	9/04	10/05	9/07
z/OS 1.7	HJE7720	9/05	10/06	9/08
z/OS 1.8	HJE7730	9/06	10/07	9/09
z/OS 1.9	HJE7740	9/07	10/08	9/10
z/OS 1.10	HJE7750	9/08	10/09	9/11*
z/OS 1.11	HJE7760	9/09	9/10	9/12*
z/OS 1.12	HJE7770	9/09	9/11*	
z/OS 1.13	HJE7780	9/11*		

See http://www.ibm.com/systems/z/os/zos/support/zos_eos_dates.html

* = projected...

JES2/MVS Compatibility



JES2 Release:

z/OS Release	JES2 z/OS R.5 HJE7708	JES2 z/OS R.7 HJE7720	JES2 z/OS R.8 HJE7730	JES2 z/OS R.9 HJE7740	JES2 z/OS R.10 HJE7750	JES2 z/OS R.11 HJE7760	JES2 z/OS R.12 HJE7770
z/OS R5	X						
z/OS R6	X						
z/OS R7	X	X					
z/OS R8	X	X	X				
z/OS R9		X	X	X			
z/OS R10			X	X	X		
z/OS R11				X	X	X	
z/OS R12					X	X	X

IBM recommends the same level of z/OS & JES2 throughout your plex.

-JES levels supported by a given z/OS release are same as the JES levels that can coexist in a MAS, See "z/OS V1R11.0 Planning for Installation" Ch. 4 (GA22-7504)

at [http://publibz.boulder.ibm.com/cgi-bin/bookmgr OS390/BOOKS/E0Z2B1A1/4.5.1](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/E0Z2B1A1/4.5.1)

Survey Questions



What is your JES2 Release level (are you \$ACTIVATED) ?

JES2 Rel.#	SHARE 116 2/2011	SHARE 115 8/2010	SHARE 114 3/2010	SHARE 113 8/2009	SHARE 112 3/2009	SHARE 111 8/2008	SHARE 110 2/2008	SHARE 109 8/2007	SHARE 108 3/2007	SHARE 107 8/2006	SHARE 105 8/2005	SHARE 104 3/2005	SHARE 103 8/2004	SHARE 102 2/2004
OS/390R10											1 (1)	5	4	9(9)
z/OS R2											1	2	2	5(4)
z/OS R4	1						1	4(4)	4(4)	6(4)	22(17)	28(22)	17	23(7)
z/OS R5		1						6(6)	5(5)	14(10)	13	8	4	
z/OS R7				2	1	2	12	19	10	6	-			
z/OS R8	1			3	4	14	13	18	6					
z/OS R9	5	5	19	31	14	16	5							
z/OS R10	8	9	12	20	5									
z/OS R11	26(7)	12(5)	9(2)	3(0)										
z/OS R12	13													

Survey Questions



Anaheim

1. Do you migrate your JES2 and z/OS at the same time?

Yes

No

2

2. If not, is there something IBM could do to make it easier to migrate together?

Yes*

No

*If yes, please see me afterwards or send me a note about what we can do to make migrating to a new level of JES2 easier.

wasik@us.ibm.com

3. How old is your oldest job on SPOOL?

Days?

13

Weeks?

10

Months?

9

Years?

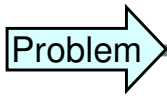
2

JES2 Service Information

■ HiPer APARs (Hi Impact, or Pervasive)

<u>yy/mm/dd</u>	<u>APAR</u>	<u>COMMENTS</u>
11/01/26	OA35313	Security APAR (JESPOOL)
11/01/24	OA35010	VFREPRG DISTERR due to NJE buffer mismanagement
11/01/12	OA35165	JOE name of 00000001 in SDSF JDS display
11/01/07	OA34987	High CPU with many idle NJE lines
10/11/17	OA34796	Jobs don't run after WLM service definition change
10/11/11	OA34611	Security APAR (SECLABELs)
10/11/04	OA34128	JOE queue error after failed \$O command without "PROTECTED"
10/10/22	OA34412	Job output altered on \$TO command even when not matching filter
10/10/14	OA33718	\$K24 ABEND with CKPT RECONFIG while members initializing
10/09/28	OA34293	Repeated DAS rebuild message \$HASP896 on warm start
10/09/23	OA34019	Looping browsing an old (pre-z/OS 1.9) created SYSOUT data set
10/09/09	OA32551	Errors using SPOOL browse in a mixed level MAS
10/09/09	OA33359	WLM JOB not starting even with available INITS
10/08/13	OA32799	ABEND068 RC108 HASPCKVR checkpoint versions subtask terminates
10/08/13	OA32292	SPOOLS used mask not updated properly at hot start
10/08/06	OA33074	NETVIEW ASID hung waiting on \$SJBLOCK in HASCJBTR
10/08/06	OA33032	SDSF prefix command does not show transaction output
10/07/30	OA33063	Higher cpu usage in SDSF starting with z/OS 1.9
10/07/01	OA33407	High CPU usage for JES2 z11 on devices or SAPI PCE
10/06/04	OA32761	HASP895 JOB/OUTPUT ERROR DETECTED RC=82 & HASP003 RC=94
10/06/04	OA32216	JES2 hangs during initialization if BERT shortage
10/05/28	OA31813	\$HASP9213 message after issuing command(s) against a large amount of jobs
10/03/26	OA31565	SDSF panel response slow due to linkage stack overhead
10/02/26	OA31557	ABEND \$IV8 retroactivating to z2 mode
10/02/26	OA31121	ABEND0C4 because PITSJB points to free SJB
10/02/19	OA31671	ROUTECD= filter always returns a not found
10/02/12	OA31674	JESNEWS data set not passed to FSS
10/02/05	OA31416	WLM jobs not selected for execution after policy change
10/01/29	OA31480	\$HASP096 after held SPIN data set altered by PSO
10/01/29	OA31479	ABEND \$Q03 after held SPIN data set altered by PSO
10/01/29	OA31091	ABEND0C4 or \$DISTERR during warm start

SDSF Performance APARs



Performance problems with extended status SSI

- ▶ Could affect any user of SSI 80



- OA34347 – Improve SYSOUT class and owner filtering (OPEN)
- OA34521 – PE fix for OA33063 (OC4 with multi-tasking user of extended status)
- OA33725 – PE fix for OA33063 (Extended status requests under initiator causes ABEND)
- OA33063 – High CPU for terse and verbose requests and with large numbers of DESTIDs/NODEs
- OA31565/OA31571 – High CPU due to linkage stack and extra queue scan
- OA28856/OA28918 – Problems when filters are active
- OA27885 – PE fix for OA24615
- OA26013 – Job position value
- OA24615 – Excessive CPU returning WLM queue Position
- OA25498 – Poor performance returning information for a large number of jobs/SYSOUT

SDSF Performance APARs



Problem → **SDSF fixes related to performance**

- Fixes** →
- **PM20221 – Fragmented storage for ULOG**
 - **PM18941 – PE fix for PM01352 (exclude list)**
 - **PM01352 – OUTPUT panel performance**
 - **PK87391 – DA panel performance problems**
 - **PK86058 – STATUS filtering performance**
 - **PK82936 – Purging jobs with lots of output**
 - **PK81154 – HELD panel delays reading CHKs**
 - **PK52910 – HELD panel performance with APPC on**

Security APARs



- **OA34611 and OA35313**
 - ▶ Obscure problems in special environments
 - ▶ No relationship between 2 problems
 - SECLABEL issue
 - JESSPOOL issue
 - ▶ Put them on

Other Performance APARs



- **OA34987 – Problems with many idle NJE SYSOUT or job transmitters (z/OS 1.11, 1.12)**
- **OA33407 – Excessive \$#POSTing for SYSOUT processors (SAPI), POST even if dest does not match (z/OS 1.11, 1.12)**

OA35165 – JOENAME displays as 0000001



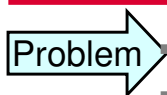
Problem → **Extended status data set list returns a JOE name of 0000001**

- ▶ Only for data sets that do not have JOEs
- ▶ Shows up in SDSF JDS display off ST and DA panel
- ▶ Meaningless clutter

Change → **Area set to blanks if no JOE name**

Fix → **JES2 z/OS 1.10, 1.11, 1.12 – F101**

OA34796 – Jobs don't run after WLM service definition change

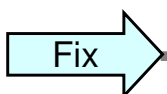


Jobs in a MODE=WLM job class do not run

- **Incorrect processing if job assigned new service class**
 - ▶ Service class is changed but job not requeues
 - ▶ Not on WLM queue so cannot run in WLM init
 - ▶ Class is WLM so cannot run on JES2 init

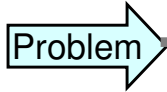


WLM service definition changes that alter job's service class



JES2 z/OS 1.11, 1.12 – F012

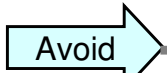
OA32551 – SPOOL Browse Problems



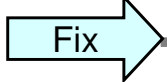
Problems processing instorage buffers

- ▶ Browsed job submitted on z/OS 1.10 viewed on z/OS 1.11
- ▶ Problems with data sets that get ENDREQed
 - Timing window

- **Can cause 24 bit private storage overlays (key 0)**
- **Has resulted in IPL when captured UCB overlaid**

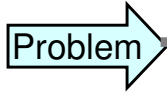


Running mixed level MAS for long periods



JES2 z/OS 1.9, 1.10, 1.11, 1.12 – F009

OA32799 – \$ACTIVATE \$K05 ABEND

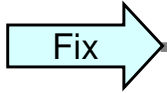


\$ACTIVATING from z11 to z2 mode and then back to z11 mode can result in errors

- ▶ DEBUG options cause \$K05 ABEND
- ▶ W/O DEBUG, bad data, ABENDs or Loops in extended status SSIs
- ▶ ABEND on member that did not \$ACTIVATE

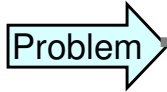


Issuing \$ACTIVATEs from z11 to z2 to z11 in a MAS



JES2 z/OS 1.11– F008

OA32292 – Jobs SPOOL Mask Incorrect After a Hot Start

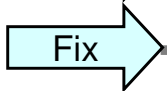


Various problems result in a jobs SPOOL usage mask missing entries on a hot start

- ▶ Mask is important to ensuring no jobs using a SPOOL volume being drained or halted
- ▶ Could result in premature draining of a SPOOL volume

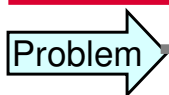


Draining SPOOLS after a hot start



JES2 z/OS 1.9, 1.10 – F008

OA33407 – SAPI/Printer Performance Problem

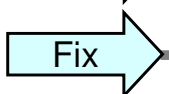


High CPU when printers/SAPI select on user routecode

- ▶ SAPI/Printer get posted to select work when nothing to select
- ▶ Device does a selection and then goes to sleep
- ▶ Happens when select on user destination (eg FRED)



Selecting on user destinations without fix



JES2 z/OS 1.11, 1.12 - F007

OA32636 – Improved HASP050



Problem → **\$HASP050 message issued when over threshold and percent usage changes**

- ▶ When at 100% usage, percentage does not change
- ▶ Message is missed when usage is very high and not changing

Change → **\$HASP050 message issued every 30 second when 99% utilization or greater even if percent usage does not change**

Fix → **JES2 z/OS 1.9, 1.10, 1.11, 1.12 – F006**

OA23689 - Unable to Warmstart - no \$BERTs

Problem → After JES2 runs out of \$BERTs, JES2 restarts but ABENDs with \$DP3.

- ▶ Customer had to COLD Start after trying to WARM/HOT START...
- ▶ There must be a way for JES2 to come up so that \$BERTs can be added to avoid Cold Start.

Avoid! → **Moral: Don't run out of \$BERTs!**

- JES2 - all current releases . . .
 - ▶ (Fixed in future release)
- See also OA25562 (F808)
 - ▶ (Will highlight shortage messages.)
- **Again: Don't run out of \$BERTs!**

OA32216 - Unable to Warmstart - no \$BERTs**Problem**

MISC processor waiting for BERTs to update CAT and WSC counts

- ABEND \$QG1 due to wait for BERTs in exit 49
- Unable to warm start in z/OS 1.9 and 1.10

Change

BERT threshold added in OA23689 increased

- 2x max for JQA plus 2x max for CAT plus max for WSC
- 48 for z/OS 1.9 and 1.10, 50 for z/OS 1.11 and 1.12

Avoid!

Again: Don't run out of \$BERTs!

Fix

JES2 z/OS 1.9, 1.10, 1.11, 1.12 – F006

OA32333 – Problems when OUTTIME=UPDATE



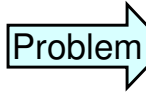
Problem → **SAPI tracking of “don’t show” attribute for
JOE incorrect if OUTDEF
OUTTIME=UPDATE**

- ▶ JOEs re-presented to application inappropriately
- ▶ SAPI application may have unpredictable error

Avoid! → **Using OUTDEF OUTTIME=UPDATE**

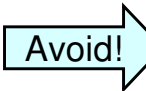
Fix → **JES2 z/OS 1.11 and 1.12 – F006**

OA33032 – SDSF PREFIX Filtering with Transaction Output Incorrect

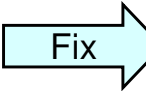


SDSF PREFIX filtering (O and H) not matching transaction output in z2 CKPT mode

- ▶ Extended status not returning transaction jobs as requested
- ▶ Some performance problems may be noticed in z11 mode

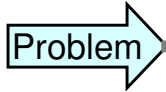


Transaction filtering in SDSF in z2 mode



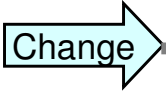
JES2 z/OS 1.11 and 1.12 - F008

OA32089 – New Message for Automatic/Init Command



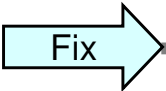
Commands from Init Deck and automatic commands are not echoed to the console

- ▶ If there is a syntax error, it is hard to determine the source
- ▶ Command responses without command echo in SYSLOG can be confusing



Issue HASP249 message to echo command

- ▶ Indicates source is “initialization” or “auto command id=xxx”



JES2 z/OS 1.9, 1.10, 1.11, 1.12 - F007

OA31837 – SPOOL Read SSI



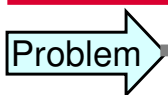
Problem → **SPOOL read SSI call fails to fetch data**

- ▶ Asking for instorage buffers
- ▶ Request data from non-local member then local
- ▶ Does not release storage between requests
- ▶ Local request set to last non-local member

Avoid! → **Mixing local and non-local SPOOL read requests without cleaning up storage**

Fix → **JES2 z/OS 1.11 – F002**

OA31813 – Command Performance



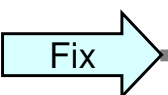
Command performance with multiple filters

- ▶ Filter optimization not working with more than one filter
- ▶ Problem introduced in z/OS 1.5
- ▶ More filters cause more problems
- ▶ Put filters most likely not to match first



Using multiple filters against large job range

- ▶ Split commands to have less jobs/JOEs per command



JES2 z/OS 1.9, 1.10, 1.11 – F005

OA31671 – R= command filter



Problem → Output command filters fail to find match

- ▶ ROUTECDE=/DEST= filters
 - \$CO, \$DO, \$O, \$PO, and \$TO
- ▶ Filter contains generics
- ▶ Command does not do what is requested

Avoid! → Commands that use generic routecde

- ▶ Eg. \$PO JQ,Q=T,R=MMA250Y*,PROT

Fix → JES2 z/OS 1.11 – F002

OA31557 – ABEND on retroactivate



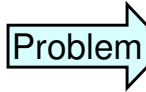
Problem → **ABEND \$IV8 \$ACTIVATE LEVEL=Z2**

- ▶ Highest JQE defined is in use
- ▶ JQE is not properly reset
 - \$ACTIVATE notices problem and ABENDs
- ▶ Impacts ability to fall back to z2 mode

Avoid! → **\$ACTIVATE LEVEL=z11 without fix**

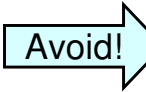
Fix → **JES2 z/OS 1.11 – F002**

OA31479/OA31480 – ABENDs after PSO request

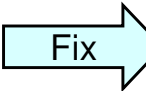


PSO request to alter held SPIN JOE

- ▶ TSO OUTPUT command
- ▶ Modify user portion of DEST, writer name, forms
- ▶ Various ABENDs can occur
 - Depending on \$ACTIVATE level



Altering SPIN JOEs with PSO (TSO OUTPUT)



JES2 z/OS 1.11 – F001

OA31330 – SNA RJE ABEND 0C4



Problem → **ABEND 0C4 AUTOLOG a SNA RJE**

- ▶ Must be last/highest RJE defined
- ▶ RJE table ends on a page boundary

Avoid! → **Reducing number of RJEs defined**

- ▶ TPDEF RMTNUM
- **Define extra RJE if last RJE is AUTOLOG**

Fix → **JES2 z/OS 1.9, 1.10, 1.11 – F001**

Trial and tribulations of \$HASP150



- **OA10772 – MSGHASP150 will only display 3 digits of the JOEID1 value**
 - ▶ \$HASP150 CICSPBR2 OUTGRP=48593.123.1 ON R6155.PRI 6 (6) RECORDS
- **Real “fix” incorporated into z/OS 1.8**
 - And then....
- **OA22397 – Wrong JOBID placed in \$HASP150 message**
- **OA23776 – MSGHASP150 incorrectly sent to PSF JOBLOG**
- **OA26451 – HASP003 RC=(28),ERROR BUILDING MESSAGE HASP150P on JES mode 3800 printers.**
- **OA29284 – \$HASP150 Message to remote RJE console contains bad text**
- **OA34006 - \$HASP704 JOEID1 truncated**



Questions?

Any questions???

Requirements satisfied



z/OS 1.12 JES2

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