



z/OS 2.2 JES2 Product Update and Latest Status

SHARE Orlando, Session 17421
Tuesday, August 11, 2015

Tom Wasik

JES2 Development
Rochester, MN

wasik@us.ibm.com



#SHAREorg



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.



Trademarks

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

- IBM®
- MVS™
- Redbooks®
- RETAIN®
- z/OS®
- zSeries®

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.

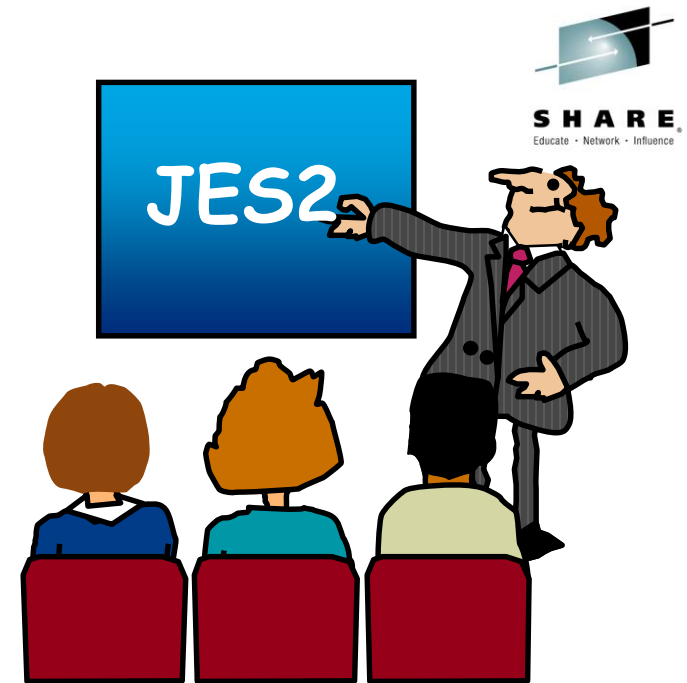
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products 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.



Agenda...

- JES2
 - Current JES2 Releases
 - Migration & Implementation Tips
 - Recent Service
- Other Presentation
 - SDSF Product Update for z/OS 2.2
 - Monday 4:30 PM-5:30 PM - Europe 2
 - What's New in z/OS 2.2 JES2: Job Execution Controls
 - Tuesday 1:45 PM-2:45 PM - Europe 2
 - What's new in z/OS 2.2 JES2: Other Changes
 - Tuesday 3:15 PM-4:15 PM - Europe 2



Current JES2 Releases

FMIDs, Birthdays & Obituaries

JES2 Rel.#	FMID	First Available	No Longer Available	End of Service
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	10/10	9/12
z/OS 1.12	HJE7770	9/10	10/11	9/14
z/OS 1.13	HJE7780	9/11	01/14	9/16
z/OS 2.1	HJE7790	9/13	01/16	9/18*
z/OS 2.2	HJE77A0	9/15		9/20*

See http://www-03.ibm.com/systems/z/os/zos/support/zos_eos_dates.html

*** = projected...**

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



SHARE
in Orlando **2015**



JES2/MVS Compatibility

- Statement of Direction
- z/OS 1.13 is the last z/OS release that supports running a down level JES.
- As of z/OS 2.1, IBM only **supports** running the 2.1 JES on the 2.1 MVS (2.2 JES only supported on 2.2 MVS)
 - z/OS 1.13 and earlier will not be **supported** running on a z/OS 2.1 or z/OS 2.2
- This does NOT affect what levels can co-exist in a MAS
 - Just the mixing and matching of JES and MVS levels.
- Warning messages during initialization:
 - WTO - JES2 level (*HJExxxx*) is not supported running on *mvs_level*
 - WTOR - Reply "CONTINUE" to initialize JES2 in this unsupported environment, "TERM" to shut down JES2

z2 Checkpoint Mode

- Statement of Direction
- z/OS 2.1 is planned to be the last release to support z2 checkpoint mode
 - \$ACTIVATE LEVEL=Z2
- z11 checkpoint mode was introduced in z/OS 1.11
 - \$ACTIVATE LEVEL=Z11
- Migrate to z11 mode if you have not already done so
 - Required to be done before warm starting z/OS 2.2
 - See z/OS 1.11 presentation for information on impacts of z11 mode

JES2 z/OS 2.2 Overview

- Growth line items
 - New \$ACTIVATE level (z22)
 - Increase limits for Jobs and JOEs
 - Checkpoint processing enhancements
 - Automatically set HOLD and DORMANCY
 - Additional SYSOUT work selection performance enhancements
- Job Execution Controls
 - Job group structure for sequencing jobs
 - Hold job until specified time
 - Start job by specified time
 - Run job with another job (by name)
- Step completion code data set
 - Information on each step execution
 - Select SMF records for the job
- PROCLIB Information SSI
- JES3 JECL toleration

JES2 Job Growth Changes

- New \$ACTIVATE level (z22) to support new functions
 - Level z2 is not supported in this release
- SYSOUT work selection performance extended to local devices
 - New option to activate support
- Updated limits for major checkpoint items
 - JOBNUM limit now 1,000,000
 - JOENUM limit now 2,500,000
 - BERTNUM limit now 2,500,000
- Increased BLOB size
 - Cache of available track groups increased from 256 to 1024
- New command to display duplicate job families
 - \$D DUPJOB

JES2 Growth Changes – New \$ACTIVATE Level

- `$ACTIVATE, LEVEL=z22`
 - Requires `SPOOLDEF CYL_MANAGED=ALLOWED` be set
 - Enables a number of z22 function
 - Larger limits
 - Job execution zones (dependent job control)
 - Larger SPOOL BLOB size (256 updated to 1024)
 - Checkpoint extend/alter processing
- Same management options same as when z11 mode was introduced
 - `$ACTIVATE` supports `LEVEL=z11` or `LEVEL=z22`
 - `OPTSDEF COLD_START_MODE= Z11` or `Z22`
 - `S JES2,PARM=(UNACT)` warm starts in z11 mode
 - `$D ACTIVATE` displays current mode/what is needed for other mode
- Checkpoint size increases for BLOB size and new ZJC data area
 - About 6 tracks depending on settings

JES2 Growth Changes – CKPT commands

- New options on the CKPTDEF command
 - \$T CKPTDEF, CKPT n =(DSNAME|VOLSER|STRNAME, INUSE)
 - Runs fast path reconfiguration if needed
 - Not needed for DSNAME/VOLSER/STRNAME if INUSE=NO
 - Only one change at a time allowed
 - CKPT1 or CKPT2
 - DSN/VOL or STRNAME or INUSE
- New information on \$D CKPTSPACE command
 - Number of tracks in a DASD data set (TRACKS=xxxx)
 - Current and maximum size of a CF structure (SIZE=(*cur,max*))
 - Size is in 1K blocks
- New options to increase the size of the checkpoint
 - SPACE=(TRK|CYL|MAX,*nnnn*) for CKPT on DASD
 - SIZE=*nnnn* for CKPT on CF (ALLOWALTER is active for checkpoint)
- CF structure deleted when changing INUSE to NO

JES2 Growth Changes – CKPT Auto Tuning

- New function – automatic checkpoint cycle management
 - Adjust HOLD= and DORMANCY= based on workload
 - Can display, but not set, current values
 - SDSF MAS display shows what members are using
 - Adjusts as member join and leaves MAS
- To enable automatic checkpoint cycle management:
`$T MASDEF,CYCLEMGT=AUTO`
 - This setting has MAS scope and persists until it is explicitly disabled.
- The function can also be enabled via MASDEF initialization statement
- The function can be turned off at any time via:
`$TMASDEF,CYCLEMGT=MANUAL`
 - All members will revert to HOLD and DORMANCY settings they had before automatic management had been turned on.

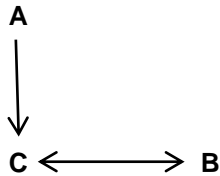
JES2 Growth Changes – Work Selection Performance

- JES2 SYSOUT work selection process for the SAPI applications extended to other devices (local printers and punches and offload)
 - SYSOUT work selection is the process of selecting SYSOUT output groups (\$#GET)
- These enhancements address both:
 - active work selection – assigning work to a device (GET)
 - passive work selection – waking up the device when work with appropriate characteristics becomes available (POST)
 - New OUTDEF option LDEV_OPT=YES|NO
- Active SYSOUT work selection uses JOE index added in V2R1
- Improvements have been made to the JOE index management.
 - Unused JOE index nodes periodically removed.

Job Execution Controls

- Execute jobs in a particular sequence
 - Goal to increase parallelism
 - Break multi-step job into multiple jobs
 - Allow independent steps to run in parallel
 - Allow dependent steps to run simultaneously
 - Pipe data between jobs
 - JCL based syntax
 - NOT processed by the converter (MVS)
 - Processed by JES2 only
 - Submitted in entirety before jobs submitted
 - Available “out of the box” with pre-defined syntax
- Job scheduling enhancements
 - Hold job until specified time
 - Start job by specified time
 - Run job with another job (by name)

Job Execution Controls – JOBGROUP example



```
//G      JOBGROUP
//A      GJOB
//B      GJOB
//C      GJOB
//      AFTER  NAME=A
//      CONCURRENT  NAME=B
//G      ENDGROUP
//*      Schedule statements below
//*
//A      JOB  TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=B
//      SCHEDULE JOBGROUP=G
//STEP1  EXEC  PGM=IEFBR14
//*
//B      JOB  TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//      SCHEDULE JOBGROUP=G
//STEP1  EXEC  PGM=IEFBR14
//*
//C      JOB  TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//      SCHEDULE JOBGROUP=G
//STEP1  EXEC  PGM=IEFBR14
```

Job Execution Controls – JOBGROUP

- New JCL statements to define a job group structure
 - JOBGROUP – Defines set of jobs to run together
 - ENDGROUP – Ends a JOBGROUP
 - GJOB – Defines a job within a job group
 - BEFORE – Defines a before relationship between jobs
 - AFTER – Defines an after relationship between jobs
 - CONCURRENT – Defines a set of jobs that execute together
 - JOBSET – Defines a set of jobs that have identical dependencies
 - ENDSET – Ends a JOBSET
 - SJOB – Defines a job within a job set
- New JCL statement in a batch job
 - SCHEDULE – Defines scheduling information within a job

Job Execution Controls – Registering Jobs

- Jobs are associated with (registered to) a job group using the SCHEDULE JCL statement

```
//          SCHEDULE JOBGROUP=grpname
```

- Security check when jobs register
 - Jobs can register to a job group if
 - Userid associated with job group and batch job the same
 - Owner of batch job has read access to JESJOBS profile
`GROUPREG.nodename.grpname.userid`
nodename – local NJE node name
grpname – name of the job group
userid – userid associated with job group

Job Execution Controls – Logging Job

- A job group logging job is created for each job group.
- The logging job has the same name and identifier as a job group.
 - Job id is Gnnnnnnn (G for Group)
- The logging job has two data sets:
 - JCL statements used to define the job group (JESJCLIN)
 - Job log data set (JESMSG LG)
- The logging job is a central place to collect messages related to important events in the life of the job group and its constituent jobs.
 - Jobs – executed/skipped, return codes, etc
- This job group log can be browsed from SDSF as any other job log.
- Life span of a job group logging job and data structures:
 - Remains in JES2 until all jobs in the group are purged
 - For TYPE=SCAN until the logging job itself is purged

Job Execution Controls – GRPDEF Statement



- Use GRPDEF to define and monitor key aspects of job groups.
 - Available as an initialization statement
 - Available in \$D and \$T commands
- Keywords on GRPDEF
 - ZJCNUM – Total number of ZJC objects defined
 - Range is 1 – 500000, default is 1000
 - ZJCWARN – Usage warning level (percent)
 - Range is 0 – 100, default is 80%
 - JOBGROUP_JOB_MAX – Maximum number of jobs in a job group
 - Range is 10 – 2000, default is 2000
 - CONCURRENT_MAX – Maximum number of jobs in a concurrent job set
 - Range is 0 – 200, default is 0



Job Execution Controls – Job Group Commands

- **\$AG** Release the job group. Jobs defined to the job group are eligible to run.
- **\$CG** Cancel all the jobs in a job group and the job group
- **\$DG** Display job group. Various views are provided.
- **\$HG** Hold the job groups. Prevents any jobs defined to the job group from starting execution.
- **\$PG** Purge all the jobs in the job group. When the last job is purged the job group will be scheduled to be purged.
- **\$TG** Modify some of the key attributes defined on the JOBGROUP statement

Job Execution Controls – Job Group Commands

- Operands added or updated on \$DJ
 - AFTER – list jobs in the job group that are dependent on this job
 - The job in the \$DJ command is the “parent” of the dependent jobs listed in the display
 - BEFORE – list of jobs in the job group that are a parent to this job
 - The job in the \$DJ command is a “child” or “dependent job” of the parent job(s) listed in the display
 - JOBGROUP – name of the job group this job is registered to
 - DELAY – updates made to reflect delays due to job group settings (HOLD, SYSAFF), concurrent sets, etc.

Job Execution Controls – Job Group Commands

- Keywords on \$DG
 - \$DG,JOBS – The jobs defined in a job group are displayed (column type display)
 - \$DG,JOBFull – displays additional values associated with the job defined in a job group (keyword display)
 - \$DG,DEPList – list all the dependencies defined in the job group (column type display)
 - \$DG,DEPFull – displays additional values for dependencies (keyword display)
 - \$DG,CONjobs – displays concurrent jobs in a job group
 - \$DG,INERROR – displays jobs marked in error
 - \$DG,SUMMARY – displays everything in above displays (the non full format)

Job scheduling enhancements – HOLDUNTL

- To hold a job until a specified time:
`// SCHEDULE HOLDUNTL=<time>`
- <time> can be specified in several ways:
 - `HOLDUNTL=' +hh:mm '`
 - A delta time from when the job entered the system.
 - This time is not subject to time offset changes.
 - `HOLDUNTL=(' hh:mm' , mm/dd/yyyy)`
or
`HOLDUNTL=(' hh:mm' , yyyy/ddd)`
 - A specific time in a future, when job should be released.
 - Date is optional.
 - This is local system time and is subject to time offset changes.

Job scheduling enhancements – STARTBY

- To specify a target time for job to start:
`// SCHEDULE STARTBY=<time>`
- <time> can be specified in several ways:
 - `STARTBY= ' +hh:mm'`
 - A delta time from when the job entered the system.
 - This time is not subject to time offset changes.
 - `STARTBY= (' hh:mm' , mm/dd/yyyy)`
or
`STARTBY= (' hh:mm' , yyyy/ddd)`
 - A specific time in a future for a job to start.
 - Date is optional.
 - This is local system time and is subject to time offset changes.

Job scheduling enhancements – WITH

- To specify that job must be executed on the same system where another reference job is currently active:

```
// SCHEDULE WITH=<jobname>
```
- WITH specification is an additional limitation on where a job can run.
 - If WITH is specified, the job will not be eligible for execution until the reference job is active.
 - In addition, the job can only be executed on the same system where the reference job is active.
- Job having a WITH specification can be submitted before or after the reference job becomes active or even submitted.
 - It is recommended to submit a job after the reference job becomes active.
 - Additional processing if job with WITH submitted first.
- Cannot do circular WITH (jobs A with B and B with A)

Step Completion Code

- The new JES2 spool data set EVENTLOG
 - Allocated automatically for batch jobs, started tasks and TSO users
- EVENTLOG contains machine readable records
 - Not intended to be human readable
 - EVENTLOG is non-printable, non-spinnable
 - Not SYSOUT, cannot be accessed by SAPI/PSO/FSS
 - Viewed via spool data set browse

Step Completion Code – EVENTLOG contents

- EVENTLOG will contain key SMF Type 30 subtype records
 - Subtype 1. Job start or start of other work unit.
 - Subtype 4. Step Total.
 - Subtype 5. Job termination or termination of other work unit.
- The user controls writing SMF records to EVENTLOG by using the SUP_EVENTLOG_SMF keyword on the JOBDEF command
 - Default is SUP_EVENTLOG_SMF=NO, indicating SMF records are not suppressed.
 - \$T JOBDEF,SUP_EVENTLOG_SMF=YES will keep SMF records from being written to EVENTLOG
 - The value of SUP_EVENTLOG_SMF is captured at the time the job enters the system and effects the job's EVENTLOG for the life of the job

Step Completion Code – EVENTLOG contents

- EVENTLOG records can be read using SPOOL Data Set Browse
- Read all EVENTLOG records using the fully qualified data set name
 - *userid.jobname.jobID.D0000008.EVENTLOG*
- Or, use the logical data set name
 - *userid.jobname.jobID.EVENTLOG*
- Use other logical data set names to see specific record types
 - *userid.jobname.jobID.EVENTLOG.STEPDATA* reads only STEPDATA records
 - *userid.jobname.jobID.EVENTLOG.SMF* reads only SMF records
 - *userid.jobname.jobID.EVENTLOG.SMFSTEP* reads only SMF Type 30 subtype 4 records
 - *userid.jobname.jobID.EVENTLOG.RESTART* reads only RESTART records

Step Completion Code – EVENTLOG and NJE



- EVENTLOG data set is eligible for transmitting/receiving via NJE
- Sender and receiver must support non-print SYSOUT
 - New signon “feature” supported by JES2
- EVENTLOG data set sent with JESMSGGLG data set



JES3 JECL Toleration

- New function to process JES3 JECL in JES2
- JES3JECL on INPUTDEF controls how JES2 treats JES3 JECL
`$T INPUTDEF, JES3JECL=PROCESS | IGNORE`
 - This setting has a member scope and is only active until this member is restarted.
 - JES3JECL=IGNORE ignores JES3 JECL statements
 - This is the default
 - JES3JECL=PROCESS enables the processing of JES3 JECL
 - Syntactically, JES3 JECL is not treated as a comment
 - Impacts what is passed to exits 4 and 54
- Note: Processing of individual JES3 JECL statements is controlled via JECLDEF (on the next chart)

JES3 JECL Toleration

- New parameters to control JES3 JECL processing

```
JECLDEF  JES3= (  
  MAIN          =      PROCESS | IGNORE | WARN | FAIL  
  FORMAT         =      IGNORE | WARN | FAIL  
  ROUTE          =      IGNORE | WARN | FAIL  
  OPERATOR       =      IGNORE | WARN | FAIL  
  DATASET        =      IGNORE | WARN | FAIL  
  ENDDATASET     =      IGNORE | WARN | FAIL  
  PROCESS        =      IGNORE | WARN | FAIL  
  ENDPROCESS     =      IGNORE | WARN | FAIL  
  NET            =      IGNORE | WARN | FAIL  
  NETACCT        =      IGNORE | WARN | FAIL  
  PAUSE          =      IGNORE | WARN | FAIL  
  )
```

JES3 JECL Toleration

- Similar new parameters to control JES2 JECL processing

```
JECLDEF  JES2= (  
    JOBPARM      =      PROCESS | IGNORE | WARN | FAIL  
    MESSAGE      =      PROCESS | IGNORE | WARN | FAIL  
    NETACCT      =      PROCESS | IGNORE | WARN | FAIL  
    NOTIFY       =      PROCESS | IGNORE | WARN | FAIL  
    OUTPUT       =      PROCESS | IGNORE | WARN | FAIL  
    PRIORITY     =      PROCESS | IGNORE | WARN | FAIL  
    ROUTE        =      PROCESS | IGNORE | WARN | FAIL  
    SETUP        =      PROCESS | IGNORE | WARN | FAIL  
    XEQ          =      PROCESS | IGNORE | WARN | FAIL  
    XMIT         =      PROCESS | IGNORE | WARN | FAIL  
    )
```

JES3 Null JCL Statement Compatibility

- JES3 ignores any records following a null JCL statement
 - “//” followed by all blanks
- JES2 ignores null JCL statement
 - Processing ends with /*EOF, //JOB, etc
 - Converter stops at “//” but JES2 JECL statements after “//” honored
- NULLJCL on INPUTDEF controls how JES2 treats a null JCL statement:

```
$TINPUTDEF, NULLJCL=IGNORE | EOF
```

- This setting has a member scope and is only active until this member is restarted.
- NULLJCL=IGNORE maintains traditional JES2 behavior
 - Default value
- NULLJCL=EOF treats null JCL statement as a logical end of file
 - This matches what JES3 does.

PROCLIB Information SSI

- PROCLIB data set information not available to applications
 - Needed to understand where PROCs are coming from
- New options on JES Property SSI (82)
 - SSJPPROD (44) – Obtain PROCLIB concatenation
 - SSJPPRRS (48) – Return PROCLIB storage
- New interface macro
 - IAZJPROC – Maps parameter list and output area
- Can request by
 - PROCLIB name
 - Job class (including STC/TSU)
 - PROCLIB used for the current address space
 - May differ if converted on other member

Jobs Queued to the \$SETUP Queue

- The \$SETUP queue in JES2 has existed for decades
- JES2 has not actually queued jobs to the \$SETUP queue
- In this release \$SETUP queue is used for
 - Logging job for a JOBGROUP
 - Jobs within a job group that have not had their dependencies met
 - Concurrent execution jobs that are waiting to get into execution
- \$SETUP can be considered a queue between conversion and execution
 - Except for logging jobs that go from input to setup to output
- Some jobs can move from setup to execution and back to setup queue
 - Concurrent jobs that have unmatched affinity
 - Concurrent jobs that WLM is delaying execution

New JOBID format (G0nnnnnn)

- Logging jobs associated with job groups have a new JOBID format
 - They start with a G instead of a J, S, or T
 - For example G0000123
- Commands that work with JOBIDs are impacted
 - JQ or JOBQ only impacts J, S, and T type jobs
 - No syntax that implies ALL job types
- \$PJQ(*),AGE=4 does NOT impact logging jobs
 - \$PG(*),AGE=4 is needed for G jobs
 - Same for \$PO and other commands
- Internally, looks like a batch job with an extra bit on

JQEFLAG3	DS	BL1	SOME MORE JOB QUEUE FLAGS
JQE3JOB	EQU	B'00000011'	BATCH JOB TYPE (WHEN BITS ZERO)
JQE3STC	EQU	B'00000001'	STC JOB TYPE
JQE3TSU	EQU	B'00000010'	TSU JOB TYPE
JQE3DFJG	EQU	B'00100000'	Job represents a JOBGROUP that is being defined z/2.2 ckpt mode & above



JES2 JCL Processing Changes

- JES2 JCL/JECL parsing was updated to deal with longer statements
 - CONCURRENT for example is greater than 8 characters
 - This JCL is NOT passed to MVS converter
 - Thus allowing longer statements and keywords
- Exits 2, 4, 52, and 54 have new 12 byte fields in the \$XPL
 - Traditional 8 byte fields still exist and have first 8 bytes
- \$STMTTAB KEYWORD= was also expanded to 12 bytes
 - Though there are no long keywords in this release
- New “job type” can be processed by exits 2, 4, 52, and 54
 - JOBGROUP functions like a job in these exits
 - JCL in a job group is unique to JOBGROUP processing
 - Exits will see new JCL and need to act accordingly
 - Bits JQE (JQE3DFJG), JCT (JCT6DFJG) and JRW (JRW1GROP)

JES3 JECL changes

- If activated JES3 JECL can be recognized by JES2 parsing
 - Treats things like `//*MAIN` as JECL
 - Passed to exits 4 and 54 as JECL with statement name MAIN
 - Operands can be processed by exit
- Can control processing on a job by job basis
 - Exit 2 and 52 can turn on or off
 - JES2 JECL parsing
 - JES3 JECL parsing
 - Both/neither
 - Currently only select keywords on `//*MAIN` processed

Other Exit Implications

- Concurrent execution:
 - Places multiple jobs in execution at the same time
 - Path to execution is round about:
 - One job in set is “master” job, not predictable which one
 - Master job placed on execution queue when ALL jobs have had their dependencies satisfied
 - When master job gets to head of queue, WLM called to see where set should execute
 - Master job returns to SETUP queue
 - WLM selects jobs by number (similar to \$SJ)
 - Jobs move to execution queue when selected
 - Exits cannot reject one job of concurrent set
 - Limits what exits like 32 (job select) can do
 - Can also impact exit 49 processing
 - Exit 51 may see different order of processing
 - Setup → Exec → Setup → Exec - for master job

ENF Changes

- Setup queue changes introduce new ENF 70s
 - Jobs move on and off SETUP queue
 - ENF70_Q_VOLWT phase equate
- ENF 78 not impacted
 - Issued once job moves beyond execution
 - Can never go back

Migration & Coexistence Considerations

- Migrating from JES2 z/OS 1.11 and 1.12 is via an all member warm start
 - No coexistence support
 - No fallback (other than spool offload and cold start)
 - Must \$ACTIVATE to z11 mode prior to warm start
- Migrating from JES2 z/OS 1.13 or z/OS 2.1
 - Must \$ACTIVATE to z11 mode prior to starting JES2 z/OS 2.2
 - APAR OA41740 needed on z/OS 1.13, or z/OS 2.1 member to coexist in MAS with z/OS 2.2
 - UA76769 – HJE7780
 - UA76770 – HJE7790
 - APAR is required for fall back as well
 - Some new data structures created by z/OS 2.2 JES2 will result in problems for prior releases if OA41740 is not installed.

Requirements Satisfied

- z/OS 2.2
 - MR1221051610 – Add ALTER support to the JES2 CKPT Coupling Facility Structure
 - MR0526042441 – JES2 Automatic Checkpoint Tuning
 - MR0710076832 – JES2 Checkpoint Initialization may cause IRA400E PAGEABLE STORAGE SHORTAGE
- APAR rolled into z/OS 2.2
 - MR032905515 – Reduce number of seemingly duplicate JES2 monitor alert messages (OA45349)
 - MR1118116734 – ENF70 Routing Options (OA41698)
- Partially satisfied in z/OS 2.2
 - SSJES204200 – JES2 dependent job control (did not address the CNTL requirement)

JES2 Service Information

HiPer APARs (Hi Impact, or Pervasive)

<u>yy/mm/dd</u>	<u>APAR</u>	<u>COMMENTS</u>
15/03/12	OA47006	Various conversion errors after \$T PROCLIB
15/03/13	OA47043	0C4 in HASPCKPT when CKPTDEF MODE=DUAL (bad AR15)
15/07/16	OA47952	High CPU in user address space because TCBNOIRB left on
15/05/20	OA47892	Security APAR
15/07/16	OA48109	Security APAR
15/07/23	OA48306	Security APAR (JES Common)
15/07/24	OA48307	Security APAR

OA48307 – NJE configuration parameters

- Added new configuration option for NETSERVs
NETSERV SECURE=REQUIRED|OPTIONAL|**USE_SOCKET**
 - Can limit NETSERV to ONLY allow AT-TLS connections
 - USE_SOCKET (default) gets setting from NETSERV's socket
 - SECURE=YES cause NETSERV SECURE=REQUIRED
 - SECURE=NO causes NETSERV SECURE=OPTIONAL
 - SOCKET SECURE=YES & NETSERV SECURE=OPTIONAL is same behavior as pre-APAR
- Affects inbound and outbound connections
- Requires JES Common APAR OA48306

z/OS Network Job Entry Security Best Practices

- Abstract:
 - Most installations have moved from BSC and SNA for NJE to TCP/IP, however the impact of this switch on security may have been overlooked. SNA and BSC are essentially a closed network whereas TCP/IP is an open network. The impact of this openness on concepts such as NJE node authentication and securing trusted nodes (nodes that can submit jobs without passwords) may not have been considered.
 - This document talks about best practices for helping secure an NJE network and managing the risk of trusted nodes, particularly when NJE is implemented over a TCP/IP network.
- **z/OS Network Job Entry Security Best Practices**
 - Published August 3rd, 2015
 - Available at one of the following links:

<http://www.ibm.com/partnerworld/page/ZSW03287USEN>

<http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=ZSW03287USEN&attachment=ZSW03287USEN.PDF>

General Security Best Practice

- NJE networks should be secure
 - Consider who can access your NJE connection
 - Could some windows computer on your network access your mainframe?
 - If so, is there a password and AT-TLS profile?
- What about your data sets on SPOOL?
 - JESSPOOL class protects data sets on SPOOL
 - It is analogous to DATASET class
 - If JESSPOOL is not active, your data set on SPOOL are not protected

Performance APARs

- OA47827 – Change JES2 ENFs from ASYNC to SYNCH
 - ENFs can be processed in parallel
 - No queuing of ENFs in ECSA to be processed serially
- OA48406 – Improve SYSOUT work selection optimization
 - WS_OPT=YES did not optimize if ONLY selecting on DEST
 - Needed at least DEST and OUTDISP
 - APAR allows optimization in this case
 - FYI – Have seen customer JES2 CPU drop over 50% with WS_OPT=YES
 - Are you optimizing your SYSOUT selection performance?

OA48262 – ECSA Storage Leak

Problem

- Cross member job commands leak ECSA (SP228)
- Source can be operator command or SSI 85
 - Command is against job active on other member
 - \$C, \$EJ, \$TJ SPIN, \$TJ SRVCLASS

Avoid

- Cross member command
- Or monitor ECSA growth and schedule IPLs

Fix

Apply PTF

OA48476 – \$K20 ABEND Draining SPOOL

Problem

- \$K20 \$ERROR ABEND processing change log
- Checkpoint mode DUAL
 - Track group map being \$CKPTed
 - Alignment of change log entries just right
 - Requires SYSPLEX (JESPLEX) IPL to recover

Avoid

- DUAL checkpoint mode
- Draining or starting SPOOL volumes

Fix

APAR is open

OA47043 – HASPCKPT 0C4 with MODE=DUAL

Problem

- BERT processing can ABEND with bad AR15
- Only occurs in CKPT MODE=DUAL
 - Code depended on AR15 over branch entry GETMAIN (documented as unpredictable)

Avoid

DUAL checkpoint mode

Fix

Apply APAR

OA47006 – Converter Errors After \$TPROCLIB

Problem

\$TPROCLIB can cause job to re-convert

- Timing problem between main task and subtask
- Various error messages
 - HASP363, HASP364, HASP096

Avoid

\$TPROCLIB on busy system

Fix

Apply APAR

Questions?

Questions?



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



8/21/2015

SHARE
in Orlando **2015**

