z/OS 3.2 IBM Education Assistant

Solution Name: Foundation for Intelligent JES2 Job Selection

Solution Element(s): JES2

July 2025

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Objectives

- In this presentation, we will describe the changes made to JES2 in z/OS 3.2 to support the intelligent job selection function
  - The changes needed to support the job selection function are complete
- The following related functions will be described in this presentation
  - A new $ACTIVATE level with additional checkpoint fields
  - A new data repository with an index in the CKPT and data stored on spool
  - A new job history function that stores a history of past job execution
  - Enhancements to job selection ($QGET) to improve performance
    - This will impact exits 14 (QGET) and 49 (Veto)
- Additionally, some miscellaneous changes will be described
  - A new $VERIFY command
  - Ability to access Spool Data Set Browse function via DD SUBSYS= keyword

Overview

- Who (Audience)
  - Users of WLM batch initiators
- What (Solution)
  - An enhancement to the JES2 batch job selection process to ensure that WLM initiators are given the best job to execute based on the capacity of the system to execute that job
    - Versus the current FIFO logic that selects the job based on queue age
- Wow (Benefit / Value, Need Addressed)
  - In a JESPlex, reduces overloading of one member with batch work when there is more capacity on another member

Usage & Invocation

- New $ACTIVATE LEVEL=Z32 command to create needed data areas in the CKPT
  - Full set of commands and options to manage the new $ACTIVATE level
- New DATADEF specification to define and manage the size of the data repository
  - Can also be used to enable and disable the repository
- Options to enable/disable job history and enhanced job selection
  - Currently not available

- New $VERIFY command to request verification (and optional rebuild of)
  - Job queue
  - Output queue
  - BERTs
  - Data repository
- Support for the existing SUBSYS= JCL keyword to access JES2 managed data sets on SPOOL

New $ACTIVATE level (1)

- $ACTIVATE,LEVEL=z32
  - Requires SPOOLDEF ADVANCED_FORMAT=ENABLED be set (ADVF=ENABLED)
  - Requires creation of “optional” CTENTs that may have been skipped from prior releases
    - ESQ (Email), CDI/CDT (policies), RGD (Resource Groups)
  - Enables a number of z32 function
    - JES2 data repository (new DRX and DRTM CTENTs)
    - New CTENTs for job level data (JQY and JQS CTENTs)
    - Additional data available in the CKPT (for SSI calls)
  - All new CTENTs are in 64 bit storage
- Same management options/commands as when z22 mode was introduced
  - $D ACTIVATE displays current mode/what is needed for other mode
  - $ACTIVATE supports LEVEL=z32 (activation) or LEVEL=z22 (retro-activation)
  - OPTSDEF COLD_START_MODE= Z22 or Z32
  - S JES2,PARM=(UNACT) warm starts in z22 mode (all member warm or hot start)
  - SSI 54 (Who-Am-I) CKPT_LEVEL= updated
  - Updated health check JES2_Upgrade_CKPT_Level
- Checkpoint size increases for new CTENTs (and optional CTENTs)

New $ACTIVATE level (2)

- Checkpoint size increase (depends on parms)
  - Max 4k pages goes from 218,019 (3.1) to 275,846 (3.2)
    - JQY – 128 bytes per JQE
    - JQS – 16 bytes per JQE (SPOOLNUM 32)
    - DRX – 64 bytes per index entry
    - DRTM – 20K based on 20,000 DRX limit
  - Typical sizes of new/added CTENTs
    - ESQ – 8K
    - CDI – 20K
    - CDT – 524K
    - RGD – 12K
    - JQY and JQS dependent on JOBNUM
    - DRX – 1.2M (20K index entries)
    - DRTM – 20K

New $ACTIVATE level (3)

- What's in the new $ACTIVATE?
  - JQY CTENT – fields added to JQA including
    - JCT track MQTR value (6 byte)
      - Replaces JQETRAK (MTTR – 4 byte)
      - Both JQETRAK_Z22 and JQYTRAK set
    - Submitting userid
    - Execution start/end time
    - Programmer name
    - Two 4 byte reserved fields for vendors
    - Repository index/sys_jobtoken
  - JQS CTENT – Job fields for SPOOL info
    - Spool used mask
      - **Eliminates SU mask in BERTs**
    - 4 byte TG count (current count is 23 bits)

- SSI 80 updated with:
  - STATJ2LV – $ACTIVATE level
  - STATNODE – Local NJE node name
  - STTRXTS – execution start time *
  - STTRXTE – execution end time *
  - STTRSUBU – Submitting userid *
  - STTRPNAM – Programmer's name *
  - STSCARIV – Job arrival time *
  - STSCQARV – Current queue time *

  * - New $ACTIVATE level only

New $ACTIVATE level (4)

```
 $d activate
          $HASP895 $DACTIVATE
$HASP895 JES2 CHECKPOINT MODE IS CURRENTLY Z22
$HASP895 THE CURRENT CHECKPOINT:
$HASP895  -- CONTAINS 650 BERTS AND BERT UTILIZATION IS 24
$HASP895     PERCENT.
$HASP895  -- CONTAINS 236 4K RECORDS.
$HASP895 z32 CHECKPOINT MODE ACTIVATION WILL:
$HASP895  -- EXPAND CHECKPOINT SIZE TO 688 4K RECORDS.
$HASP895 z32 ACTIVATION WILL FAIL IF ISSUED FROM THIS MEMBER.
$HASP895     THE FOLLOWING ISSUES PREVENT ACTIVATION:
$HASP895  -- ADVANCED_FORMAT MUST BE ENABLED.
$HASP895  -- CKPT1 IS TOO SMALL BY 338 4K RECORDS.
$HASP895  -- CKPT2 IS TOO SMALL BY 338 4K RECORDS.
```

JES2 Data Repository

- New general purpose data repository in JES2 to store information:
- The DRX CTENT (in the CKPT) stores the repository index entries
  - Number of index entries set using DATADEF INDEX=LIMIT=nnnnnn (max 1,000,000)
- Index uses the existing AVL Tree support to manage the index
  - Supports multiple trees for potential future expansion
- Bulk data is stored in records on SPOOL
  - Spool space associated with new system job $DATAREP
  - DRTM CTENT manages TGs used by repository (bit map of available records)
    - CTENT size is managed internally
  - Each index can have up to 255 spool tracks of data
- Requires $ACTIVATE to LEVEL=Z32 and setting DATADEF REPOSITORY=ENABLED

```
$ddatadef
          $HASP888 DATADEF
$HASP888 DATADEF  REPOSITORY=ENABLED,INDEX=(LIMIT=20000,WARN=80,
$HASP888          FREE=20000)
```

- Job history will be first user of the repository
  - Job history tree uses SYS_JOBTOKEN or JCLID1/2 (from JES2 3.1) as index into tree

Job History Data

- Information about past runs of a job
    - Stored in the JES2 repository
    - Used to project run time and CPU need for the job next time it is submitted
    - Expires (and is deleted) if not used for 30 days (current limit)
- Index for these entries is either:
    - JCLID1/JCLID2 – Hash value of JCL (SMF30JCLID1/SMF30JCLID2)
    - SYS_JOBTOKEN – value passed at job submission (SMF30JOBTOKEN)
        - SYS_JOBTOKEN can also be set by pre-execution policies or exits
- Currently, job history and enhanced job selection is disabled
    - Testing continues for this function
    - Enablement will be via APAR and switch once testing has completed

Enhancements to job selection ($QGET)

- As part of the Intelligent job selection process, $QGET is getting reworked
  - Changes in the WLM initiator selection pass (not other phases)
  - New local selectable work queue is being added
    - Only has jobs that can be locally selected (not active, held, limited, etc)
    - Maintained on each member
    - Can greatly reduce queue rattling when selecting work
    - New service (GTJBSEL) to check if a job can execute locally
- Changes will impact JES2 exits
  - Exit 14 ($QGET replacement exit)
    - New service GTJBSEL to determine if a job is eligible to run on the current member
    - Similar to GTSCREEN for SYSOUT
    - Current exit may continue to function
      - May want to update to take advantage of new service and side queue
  - Exit 49 ($QGET veto exit)
    - Installations need examine exit 49s to see if new structures/processing necessitates any changes
- Job selection not available at this point

$VERIFY command (1)

- Operator command to perform various verification processing
    - Similar to what is currently done during warm/hot start processing
        - Same services used by DEBUG QVERIFY=YES
    - Supports JOBQUEUE, OUTQUEUE, BERTS, and REPOSITORY

      ```
      $VERIFY JOBQUEUE
      ```
    - OPERCMDS class RACF check for *jesx*.VERIFY.*queue* with CONTROL access
        - *queue* – is one of JOBQ, OUTQ, BERTS, or REPO
    - Verifies one queue at a time
- Intended for use when problem is detected
    - Can result in high CPU overhead
    - Reports errors to console (with enhanced messaging)
- Sample usage:

```
1. $verify jobqueue
   $HASP1250 $VERIFYJOBQUEUE
   $HASP1250 Job queue verification completed - error detected
   $HASP1250 RC=x'12' - JQE in use count bad

2. $verify jobqueue
   $HASP1250 Job queue verification completed - no errors detected
```

$VERIFY command (2)

- Additional option REBUILD= YES|NO|COND to attempt a queue rebuild
  - COND does a rebuild only if verify detects an error
  - The default is NO
  - YES will do an unconditional rebuild
    - For cases where VERIFY does not detect error
- Should only use if advised by IBM support
  - High overhead
  - Rebuild action could remove active jobs/elements to correct error
- Sample usage:

```
$verify jobqueue,rebuild=cond
          $HASP1250 $VERIFYJOBQUEUE,REBUILD=
$HASP1250 Job queue verification completed - error detected
$HASP1250 RC=x'12' - JQE in use count bad
          $HASP517 JOB QUEUE REBUILD HAS COMPLETED.
```

SPOOL Data Set Browse enhancement (1)

- SPOOL data set browse allows a DYNALLOC to access any JES managed DS
  - Currently requires writing a program to do a dynamic allocation with a special TU
- Support added to allow specification of SUBSYS= on JCL DD statement
  - Simplifies accessing JES2 managed data sets from JCL (no program needed)
  - Specify JES2 subsystem where allocation is to be done
  - Option ACTIVE=YES|NO controls accessing instorage buffers
- Standard JESSPOOL RACF class protection applies
- DSN= naming conventions the same as SPOOL Data set browse

  Specifying the Data Set Name

  - Place data set name in apostrophes if using non-standard characters/formats

SPOOL Data Set Browse enhancement (2)

- A JESMSGLG (job log) data set (DSN copied from SDSF)

```
//SYSUT1   DD     DSN=IBMUSER.IBMUSERS.JOB00021.D0000002.JESMSGLG,
//      SUBSYS=JES2
```

- Same JESMSGLG (job log) data set with minimal DSN

```
//SYSUT1   DD     DSN='*.IBMUSERS.JOB00021.JESMSGLG',SUBSYS=JES2
```

- SMF data from EVENTLOG data set for an active job (SMF 30 records)

```
//SYSUT1   DD   DSN=IBMUSER.IBMUSER.TSU00020.EVENTLOG.SMF,
//         SUBSYS=(JES2,'ACTIVE=YES’)
```

- JCL used to allocate JCL a job and re-submit it

```
//IBMUSERB JOB   MSGLEVEL=(1,1),MSGCLASS=A,CLASS=A
//STEP1    EXEC   PGM=IEBGENER
//SYSPRINT DD     SYSOUT=A
//SYSUT2   DD     SYSOUT=(A,INTRDR)
//SYSIN    DD     DUMMY
//SYSUT1   DD     DSN='*.IBMUSERS.JOB00021.JCL',SUBSYS=JES2
```

- Current SYSLOG (on system SY1) – note SYSLOG always gets active buffers

```
//SYSUT1   DD   DSN='SY1.SYSLOG.SYSTEM',SUBSYS=JES2
```

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - General JES2 users will use these functions as needed

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level: YES
  - All members of the JESPlex (MAS) must be at z/OS 3.2 to $ACTIVATE to LEVEL=Z32
  - The $VERIFY function and DD SUBSYS= can be used at any time (no JESPlex considerations)
  - JES2 compatibility APAR OA65446 is required to tolerate coexistence in a JESPlex (MAS)
- List any toleration/coexistence APARs/PTFs.
  - JES2 compatibility APAR OA65446
- List anything that doesn't work the same anymore.
  - Exits 14 and 49
- Compatibility APAR OA65446 is also recommended for fallback to z/OS 3.1

Installation & Configuration

- No special installation is required.

Summary

- In this presentation we discussed the foundational enhancements required to lay the groundwork for intelligent job selection
  - $ACTIVATE
  - JES2 Data Repository
  - Job History
  - Enhancements to job selection (QGET)

- Additionally, some key operational enhancements were presented that aid/simplify system management
  - $VERIFY
  - SUBSYS= on the JCL DD statement

Appendix

- Related publications
  - z/OS 3.2 JES2 Commands
  - z/OS 3.2 JES2 Initialization and Tuning Guide
  - z/OS 3.2 JES2 Initialization and Tuning Reference
  - z/OS 3.2 JES2 Installation Exits
  - z/OS 3.2 JES2 Messages
  - z/OS 3.2 MVS Using the Subsystem Interface