#SHAREorg

SHARE
Technology · Connections · Results

# z/OS Basics:
# JES2/JES3 JCL/JECL Differences


Tom Wasik

IBM JES2 Development

David Jones

IBM JES3 Development




Friday, August 10, 2012
Session 11710: 9:30 AM – 10:30 AM

SHARE in Anaheim 2012

# 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.

# A VERY brief history of JCL/JECL (with a bit of poetic license)

- In the beginning, there was JCL
  - It described the basics of a job and it was good
- But then came the JESes, JES2 and JES3
  - They wanted their own specifications in JCL but were not type 1
  - So each created their own JCL statement prefixes
    - //* for JES3
    - /* for JES2 that was converted to //* before converter saw it
  - Each created their own solutions to their problems, and all was good
- But then, the customers wanted something that worked for ALL JESes
  - The OUTPUT card was born, keywords were added
  - The world was better, BUT not all differences were addressed….

# Objectives:

- Examine the main JCL/JECL of JES2 and JES3
  - Job level statements
  - NJE routing statements
  - SYSOUT property statements
- Understand how each JES accomplishes similar functions
- Relate the JECL statements to JES independent JCL statements
- Highlight the JES Unique aspects of the statements
- End goal is to help understand how to write JES independent JCL where possible
  - JECL is bad
    - Except when there is no other choice (the JECL made me do it)

Note: Neither JES has significantly updated JECL in Decades

# // JOB

JES2 /*JOBPARM

JES3 //*MAIN

# /*JOBPARM *v.* //*MAIN *v.* //JOB
# JOB Level Parameters

- In the days before SJF (Scheduler JCL Facility)
  - Adding JCL keywords was expensive ➔ never done
  - JES (remember field developed) wanted to provide function
  - Rather than wait for OS guys (Poughkeepsie), JES did their own thing
- JOBPARM and MAIN provided job level defaults
  - Everything from trivial (lines per page printed) to major (where job can run)
- Then came SJF to the rescue
  - Extendable way to define JCL with tables and common parsing
  - New keywords or JCL statements are now easy
  - SJF deals with parsing and validation
  - JES does the real work behind most keywords
- Keywords eventually start appearing on JCL cards

# //*MAIN *v.* /*JOBPARM *v.* // JOB Parameters compared

| //*MAIN | /*JOBPARM | // JOB | Usage notes |
|---|---|---|---|
| ACMAIN= | None | None | Archaic (system name for notify processing) |
| BYTES= | BYTES= | BYTES= | //*MAIN has warning frequency.<br>/*JOBPARM does not have action. |
| CARDS= | CARDS= | CARDS= | //*MAIN has warning frequency.<br>/*JOBPARM does not have action. |
| CLASS= | None | CLASS= | JES3 supports 8 characters for //*MAIN CLASS= |
| None | COPIES= | None | Replicates non-held, non-spin SYSOUT |
| DEADLINE= | None | None | JES3 Deadline Scheduling |
| EXPDTCHK= | None | None | JES3 tape expiration checking |
| FAILURE= | RESTART= | None | Must use FAILURE=RESTART if // JOB RESTART= is used. Subtle differences. |
| FETCH= | None | None | JES3 tape fetch message options |
| None | FORMS= | None | Equivalent to FORMS on JES3 default //*FORMAT |
| HOLD= | None | TYPRUN= | TYPRUN=HOLD. |

# //*MAIN *v.* /*JOBPARM *v.* // JOB Parameters compared (continued)

| //*MAIN | /*JOBPARM | // JOB | Usage notes |
|---|---|---|---|
| IORATE= | None | None | Archaic (used by selection processing) |
| JOURNAL= | None | None | JES3 job level control of journaling |
| None | LINECT= | None | JES2 lines per page count |
| LINES= | LINES= | LINES= | //*MAIN has warning frequency.<br>/*JOBPARM does not have action. |
| LREGION= | None | None | Archaic (used by selection processing) – In K bytes |
| None | NOLOG | None | JES2 suppress job log |
| ORG= | None | None | JES3 default routing (similar to //OUTPUT DEST) |
| PAGES= | PAGES= | PAGES= | //*MAIN has warning frequency.<br>/*JOBPARM does not have action. |
| PROC= | PROCLIB= | None | Ooops |
| RINGCHK= | None | None | Archaic (tapes had rings?) |
| None | ROOM= | None | Archaic (JES2 job default ROOM for SYSOUT) |
| SETUP= | None | None | JES3 setup capability. |

# //*MAIN *v.* // JOB
# Parameters compared (continued)

| //*MAIN | /*JOBPARM | // JOB | Usage notes |
|---|---|---|---|
| SPART= | None | None | JES3 SPOOL partition |
| SYSTEM= | SYSAFF= | None | Ooops |
| THWSSEP= | None | None | JES3 high water setup processing |
| None | TIME= | None | Archaic (JES2 wall clock time execution limit) |
| TRKGRPS= | None | None | Archaic (JES3 track groups for job) |
| TYPE= | None | None | Archaic (Type of operating system. VS2 or ?) |
| UPDATE= | None | None | JES3 scheduling function |
| USER= | None | USER= | ***These are not the same!*** **//*MAIN user Archaic** |

# Ooops?

- PROC/PROCLIB has no true JCL equivalent
  - Can use JCLLIB to specify specific PROC data sets for a job
  - No easy way to select system level default
- SYSAFF/SYSTEM also has no true JCL equivalent
  - Scheduling environment is the recommended solution (SCHENV=)
  - System decides where to run job based on where resources are
  - Deals with most if not all application level affinity questions
    - Especially since system names can change
    - Define a resource and a scheduling environment for each system
    - SCHENV=SY1 would then execute job on system SY1
  - Unfortunately does not control conversion processing
    - For the 1% where that matters

# // JOB USER= *v.* //*MAIN USER=

- // JOB USER=*userid* identifies to the system the person submitting the job.

- //*MAIN USER=*userid* identifies the job with the specified TSO/E user.
  - TSO/E *userid* can issue the TSO/E OUTPUT command to access sysout data sets from the job.
  - TSO/E *userid* can inquire about the status of the job or to cancel the job.
  - Requires use of user exits IATUX29, IATUX30 and disabling security authorization checking using JESSPOOL and JESJOBS classes!
  - i.e. This predates existing security checking!

# // XMIT

JES2 /*XMIT & /*ROUTE

JES3 //*ROUTE

# // XMIT *v.* /*XMIT or /*ROUTE XEQ or //*ROUTE XEQ

- All are used to transmit an input stream to a network node.
  - May be a job input stream which is then executed on the destination node.
  - May be other job definition statements recognized by the destination node.
- Differences exist with regard to which records from an input stream are transmitted.

# // XMIT (JCL like statement)

- Transmit a job input stream, or other job definition statements.
- Transmits all records following the XMIT JCL statement to:
  - Two character delimiter specified by DLM= parameter.
  - /* if DLM= not specified.
  - End of input stream.
- What follows the XMIT JCL depends on where the data is going
  - MVS system would expect a JOB card
  - VM system would expect VMBATCH control statements
  - Other operating system would have other expectations
- /*EOF and /*DEL are processed by the internal reader.
  - Occurrence after the XMIT JCL statement may result in errors.
  - Cannot be transmitted in JES2.
  - Can be transmitted in JES3 when SUBCHARS= is used to define a substitute for /* on the input of /*EOF and /*DEL.

# /*XMIT

- Transmit a job input stream or other job definition statements.
- Transmits all records following the /*XMIT JCL statement to:
  - Two character delimiter specified by DLM= parameter.
  - /* if DLM= not specified.
  - End of input stream.
- What following the /*XMIT JECL is same as //XMIT.
- /*EOF and /*DEL are processed by the internal reader.
  - Occurrence after the /*XMIT JCL statement may result in errors.
  - Cannot be transmitted using /*XMIT.

# /*ROUTE XEQ and /*XEQ

- Transmit the entire job input stream to execute at the destination.
- Transmits all records in the job.
  - /*ROUTE XEQ follows the JOB JCL statement.
- The preceding JOB JCL statement is transmitted with the job.
  - Must be valid on the sending and receiving system
- /*EOF and /*DEL are processed by the internal reader.
  - Cannot be transmitted using /*ROUTE XEQ.

# //*ROUTE XEQ

- Transmit a job input stream to be executed at the destination.
- Transmits all records following the //*ROUTE XEQ to:
  - A second JOB JCL statement following the //*ROUTE XEQ.
  - End of input stream including /*EOF.
  - No other delimiters can be specified.
- A JOB JCL statement must follow the //*ROUTE XEQ.
  - NJB should be used in place of JOB to avoid the job being submitted at the submitting node if //*ROUTE XEQ is in error.
  - NJB required in place of JOB for TSO/E submits.
  - NJB restored to JOB before transmit.
- /*EOF and /*DEL are processed by the internal reader.
  - Cannot be transmitted using //*ROUTE XEQ.

# // OUTPUT

JES2 /*OUTPUT et al

JES3 //*FORMAT

# Output characteristics

- JES2 output data set characteristics are based upon:
  - OUTCLASS(v) initialization statements (define SYSOUT classes)
  - // OUTPUT JCL statements
  - // DD JCL statement
  - /*OUTPUT JES2 control statement
  - /*ROUTE PRINT or PUNCH for DEST
  - /*JOBPARM for job level defaults of FORMS, LINECT, ROOM

- JES3 output data set characteristics are based upon:
  - JES3 defaults and OUTSERV initialization statement
  - SYSOUT initialization statements (define SYSOUT classes)
  - // OUTPUT JCL statements
  - // DD JCL statement
  - //*FORMAT JES3 control statement

# JES2 /*OUTPUT and // OUTPUT

- No generic /*OUTPUT concept
- Specific: /*OUTPUT ***code*** *parameters*
- DD statement points to ONE /*OUTPUT card
  - SYSOUT=(*class*,*writer*,***code***)

- Default: //*name* OUTPUT with DEFAULT=YES
- Direct: //*name* OUTPUT

- Parameters from /*OUTPUT and // OUTPUT never mixed.
  - When DD specifies OUTPUT= value
    or
    job has DEFAULT=YES OUTPUT JCL statements
  - Then SYSOUT ***code*** value is treated as a form name
    - Same as when no /*OUTPUT card *code* matches the value

# JES3 //*FORMAT and // OUTPUT

- Non-specific (default): //*FORMAT DDNAME=*null*,
- Specific: //*FORMAT DDNAME=*ddname*

- Default: //*name* OUTPUT with DEFAULT=YES
- Direct: //*name* OUTPUT

- Parameters from //*FORMAT and // OUTPUT statements are never mixed.

# Non-specific //*FORMAT statements

- Non-specific //*FORMAT parameters apply to all DDs in a job:
  - When a default // OUTPUT does not exist in the job.
  - When a direct // OUTPUT does not apply to the DD.
  - Merged when a specific //*FORMAT applies to the DD.
- Non-specific //*FORMAT parameters apply to system-managed data sets.
  - Except when // OUTPUT JESDS= is specified for the data set.
- Multiple non-specific //*FORMAT statements are merged for a single default set of parameters.

# Specific //*FORMAT statements

- Specific //*FORMAT statement identifies the DD statement(s) to which it applies with DDNAME=*name*.
  - Easier to apply the parameters to DDs defined across the job, in a step, or in a procedure.
  - Can also specify system-managed data sets.

- Multiple SYSOUT copies are produced when multiple specific //*FORMAT statements apply to a DD.
  - One for each //*FORMAT.
  - //*FORMAT statements where DDNAME=*name* is more specific apply and less specific will not apply.
    - e.g. DDNAME=SYSUT2 and DDNAME=STEP0001.SYSUT2
    - Only the more specific STEP0001.SYSUT2 will apply.

# How JES3 applies //*FORMAT parameters

- Start with JES3 defaults.
  - JES3 defaults can be altered by OUTSERV initialization statement.
- Apply //*FORMAT DDNAME=*null*, statement parameters.
- Apply SYSOUT class parameters using the first of:
  1) SYSOUT class from // DD SYSOUT = parameter.
  2) SYSOUT class from // JOB MSGCLASS= parameter.
  3) Default SYSOUT class.
- Apply DD statement specific parameters.
- Apply specific //*FORMAT statement parameters.
  - Specific //*FORMAT statement where DDNAME= identifies a DD.
  - Note that this overrides DD statement specific parameters.

# Other JES2 SYSOUT Property Sources

- /*JOBPARM FORMS=, LINECT=, and ROOM= defaults at job level
  - Applies to all SYSOUT as 1st default
  - Even if // OUTPUT or /*OUTPUT cards are present
    - Unless they specify FORMS, LINECT or ROOM
- /*ROUTE PRINT and PUNCH default SYSOUT route codes
  - Applies at the job level to all output
  - Even if //OUTPUT or /*OUTPUT cards are present
    - Unless they specify DEST

# Other JES2 SYSOUT Property Sources

- OUTDISP defaults from OUTCLASS(*x*) parameter setting
  - Applies to both normal and abnormal disposition
  - Overridden by a // OUTPUT JCL card and DD HOLD=YES/NO

- OUTPUT=DUMMY on OUTCLASS(*x*) takes ultimate preference
  - No output is produced

# Default // OUTPUT statements

- Default // OUTPUT parameters apply to all DDs in a job:
  - When a direct // OUTPUT does not apply to the DD.
    - Default and direct // OUTPUT parameters are not merged.
  - When a specific //*FORMAT does not apply to the DD.
- Default // OUTPUT cards cause all /*OUTPUT cards to be ignored
- Multiple SYSOUT copies are produced when multiple default // OUTPUT statements apply.
  - One for each default // OUTPUT.
- // OUTPUT parameters apply to system-managed data sets when JESDS= is specified.

# Direct // OUTPUT statements

- The DD statement directly identifies the // OUTPUT statement(s) to be applied with OUTPUT=.
  - Each DD needs to specify OUTPUT=.
  - Harder to specify for DDs in a procedure.
  - Can also specify the default // OUTPUT
- // OUTPUT JESDS= used to apply parameters to system-managed data sets.
  - Can be used on default and direct // OUTPUT statements.
- Multiple SYSOUT copies are produced when multiple // OUTPUT statements are specified in a DD.
  - One for each // OUTPUT.

# How JES2 applies OUTPUT parameters

- Determine SYSOUT class parameters using the first of:
  1) SYSOUT class from // DD SYSOUT = parameter.
  2) SYSOUT class from // OUTPUT CLASS= parameter (SYSOUT=(,)).
  3) SYSOUT class from // JOB MSGCLASS= parameter.
  4) Default MSGCLASS from device
- Apply general defaults from
  - JOBPARM, ROUTE, and OUTCLASS(*x*)
- Apply values from // OUTPUT or /*OUTPUT (never both)
  - Specific or default // OUTPUT cards (never both)
- Apply values from the DD statement
  - DD statement keywords override all other sources

# How JES3 applies // OUTPUT parameters

- Start with JES3 defaults.
  - JES3 defaults can be altered by OUTSERV initialization statement.
- Apply SYSOUT class parameters using the first of:

  1) SYSOUT class from // DD SYSOUT = parameter.

  2) SYSOUT class from // OUTPUT CLASS= parameter.

  3) SYSOUT class from // JOB MSGCLASS= parameter.

  4) Default SYSOUT class.
- Apply // OUTPUT statement parameters.
  - Either DEFAULT=YES statement or DD OUTPUT= statement.
- Apply DD statement specific parameters.
  - Note that this overrides // OUTPUT statement parameters.

# Using both //*FORMAT and // OUTPUT

- Parameters from //*FORMAT and // OUTPUT statements are never mixed.
- When default // OUTPUT and non-specific //*FORMAT both apply to a data set, only one copy of the data set is created using the default // OUTPUT parameters.
- Multiple SYSOUT copies are produced when specific //*FORMAT and direct // OUTPUT statements apply to a DD.
  - One for each //*FORMAT and one for each // OUTPUT.

# //*FORMAT *v.* // OUTPUT
# Forcing the JES3 initialization default

- Can specify that the JES3 initialization default for a parameter be used.

  //*FORMAT *parameter*=STANDARD, CARRIAGE=6, FCB=6

  // OUTPUT *parameter*=STD

- Only in JES3 for selected parameters
  - Using STD in JES2 may be an error or a valid parameter value.

# //*FORMAT *v.* /*OUTPUT *v.* // OUTPUT Parameters compared

| //*FORMAT | /*OUTPUT | // OUTPUT | Usage notes |
|---|---|---|---|
| CARRIAGE= | None | None | JES3 only 3211 carriage tape name – archaic |
| CHARS= | CHARS= | CHARS= | STANDARD/STD in JES3 only. |
| CHNSIZE= | None | None | JES3 SNA RJP transmission record count |
| None | CKPTLNS= | CKPTLINE= | CKPT interval in lines |
| None | CKPTPGS= | CKPTPAGE= | CKPT interval in pages |
| COMPACT= | COMPACT= | COMPACT= | Compact table name |
| CONTROL= | None | CONTROL= | Different JES2/JES3 defaults. |
| COPIES= | COPIES= | COPIES= | Copies 1-255. JES3 supports 0 (do not print) |
| DEST= | DEST= | DEST= | JES3 values not supported by both.<br>Limited JES2/JES3 common values. |
| EXTWTR= | None | WRITER= | ***These are not the same!*** |
| FCB= | FCB= | FCB= | 6/STD for JES3 initialization default. |
| FLASH= | FLASH= | FLASH= | STANDARD/STD for JES3 initialization default. |
| None | FLASHC= | FLASH= | Second keyword on FLASH= - archaic |

# //*FORMAT *v.* /*OUTPUT *v.* // OUTPUT Parameters compared (continued)

| //*FORMAT | /*OUTPUT | // OUTPUT | Usage notes |
|---|---|---|---|
| FORMS= | FORMS= | FORMS= | STANDARD/STD for JES3 initialization default. (1) |
| None | INDEX= | INDEX= | 3211 left margin |
| None | LINDEX= | LINDEX= | 3211 right margin |
| None | LINECT= | LINECT= | JES2 lines per page |
| MODIFY= | MODIFY= | MODIFY= | |
| None | MODTRC= | MODIFY= | Second parameter on MODIFY=. Archaic. |
| OVFL= | None | OVFL= | JES3 only. |
| PRTY= | None | PRTY= | Different JES2/JES3 defaults. |
| STACKER= | BURST= | BURST= | STACKER=Y / BURST=S (separate sheets)<br>STACKER=N / BURST=C (continuous fanfold)<br>STACKER=STANDARD but no BURST=STD |
| THRESHLD= | None | THRESHLD= | JES3 only. |
| TRAIN= | UCS= | UCS= | TRAIN=STANDARD but no UCS=STD |

1. Form-name is 1 to 8 characters, but only 1 to 4 characters when specified using // DD SYSOUT= parameter.

# // OUTPUT WRITER= *v.* //*FORMAT EXTWTR=

- // OUTPUT WRITER=*name* identifies an external writer to process the SYSOUT data set.
  - JES3 SYSOUT placed on Q=HOLD for WRITER=*name* if not destined for a known node (held for WRITER=*name)*.
  - JES3 SYSOUT placed on appropriate Q for a destination node, then placed on Q=HOLD for WRITER=*name* at the destination node.

- //*FORMAT EXTWTR=*name* identifies an external writer at a destination node that is to process the sysout data set.
  - JES3 SYSOUT placed on Q=WTR if not destined for a known node (not held for EXTWTR=*name)*.
  - JES3 SYSOUT placed on appropriate Q for a known node, then placed on Q=HOLD for EXTWTR=*name* at the destination node.

# JES3 SYSOUT HOLD differences

- JES3 Q=HOLD (hold queue) contains:
  - Data sets for external writers.
  - Data sets for a SYSOUT class with HOLD=TSO.
    - Available for TSO/E OUTPUT command.

- // DD HOLD=YES for JES3 is not the same as for JES2.
  - Sets JES3 hold status of USER with data set on Q=WTR.
  - JES2 data set on Q=HOLD and available for TSO/E OUTPUT.

- //*FORMAT parameters are not applied to JES3 data sets initially put on Q=HOLD.
  - Applied when released (data set moved to Q=WTR).

# Other JCL/JECL

# Other JECL

| JES2 | JES3 | Comment |
|---|---|---|
| /*$*xxxxx* | //***xxxxx* | Issue a JES operator command |
| | //*DATASET<br>//*ENDDATASET | Define data set to process |
| | //*PROCESS<br>//*ENDPROCESS | Process phases for a job |
| | //*NET | JES3 dependent job control |
| /*NETACCT | //*NETACCT | Network accounting info (same in name only) |
| /*NOTIFY | | Archaic. Use JOB card NOTIFY= |
| /*MESSAGE | //*OPERATOR | Issue message to the operator |
| | //**PAUSE | Pause input device waiting for a start |
| /*PRIORITY | | JES2 priority of the job |
| /*SETUP | | JES2 issue setup message to operator |
| /*SIGNON<br>/*SIGNOFF | /*SIGNON<br>/*SIGNOFF | RJE/RJP sign on and off |

# JCL Differences

| Statement | Comment |
|---|---|
| // DD COPIES= | JES2 is 1-255, JES3 0-255 (0 implies does not print)<br>Group value JES2 1-255, JES3 1-254 |
| // DD HOLD= | See hold discussion earlier |
| // DD SEGMENT= | JES2 only |
| // JOB accounting | JES2 has a default parser, JES3 does not |
| // JOB CLASS= | JES2 has only 1 character class, JES3 supports up to 8 using //* MAIN card |
| // JOB NOTIFY= | JES2 supports node.userid, JES3 support only userid |
| // JOB TYPRUN= | JES3 does not support JCLHOLD and COPY.<br>//*MAIN HOLD=YES is same as TYPRUN=HOLD.<br>EXEC PGM=JCLTEST is almost TYPRUN=SCAN except the interpreter is run |

# JCL Differences

| Statement | Comment |
|---|---|
| // (null JCL card) | JES3 stops input processing for the job, JES2 ignores the // card. Can place JES2 JECL after a // and JES2 will honor it. |
| // OUTPUT | Various keywords supported by only one JES |

# Questions?

## Session 11710