#SHAREorg

SHARE
Technology · Connections · Results

Batch Modernization in JES2 z/OS 2.1

Tom Wasik
IBM Rochester, MN
wasik@us.ibm.com

Thursday 8:00AM
Session Number 14919

JES2

Mainframe 50 April 7th 1964 - April 7th 2014
http://www.ibm.com/mainframe50/

SHARE in Anaheim

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.

Overview

- This presentation will cover the following topics
  - JCL changes to JOB, OUTPUT, and JCLLIB cards
  - Eight character job classes
  - Input phase JCL/JECL error processing changes
  - Performing interpretation after conversion
  - JCL symbol processing changes

JOB card JCL changes

- New **SYSAFF=** keyword on JOB card
  - List JES2 member names where job can run
  - Valid names are JES2 members defined in the JESPLEX (MAS)
  - Replaces function of SYSAFF on /*JOBPARM
- New **SYSTEM=** keyword on JOB card
  - Similar to SYSAFF but a list of MVS system names
    - Does not support independent more (IND)
  - Valid names are systems that are/were active in the JESPLEX (MAS)
- SYSTEM and SYSAFF on job card are mutually exclusive
  - Both specified results in JCL error
  - Both support exclusion lists – eg. SYSTEM=(-SY1, SY2)
    - Anywhere except SY1 and SY2
  - ‘*’ indicates submitting system
- JOBPARM SYSAFF ignored if JOB card SYSTEM or SYSAFF specified
  - New warning message generated

JOB card JCL changes

- New **UJOBCORR=** keyword on JOB card
  - Specifies user portion of job correlator
  - 1-32 characters (A-Z, 0-9, Nationals, “_”)
  - “_” requires value be placed in quotes
  - Parsed by JES2 during INPUT phase processing
  - Can also be specified on the start command
    - S DEALLOC,UJOBCORR=TEST

Multi character job class

- Add support for longer job classes (1-8 character)
  - New limit is 512 job classes including 1 character and STC/TSU
  - Valid characters are A-Z and 0-9
- New commands created to ADD and DELETE job classes
  - The 36 traditional classes are predefined and cannot be deleted
- Traditional job class cannot be deleted but they can be set inactive
  - New attribute for BATCH job classes (traditional and 8 character)
  - Set via $T operator command
    - ACTIVE=<u>YES</u>|NO
  - ACTIVE=NO job class cannot be used during input processing
    - Prevents new use of the job class
    - Also applies if class is a default (be careful)
  - Does NOT impact other processing (jobs in class still run)

JCLLIB card changes

- New PROCLIB= keyword on JCLLIB card
  - Selects which JES2 PROCLIB concatenation to use for job
    - Specify entire DDNAME such as PROC01
    - If DDNAME does not exist, PROC00 is used
  - Same as PROCLIB= on JES2 JOBPARM JECL card
    - JCL error if specified on JCLLIB and JOBPARM

OUTPUT card changes

- New MERGE=YES keyword on OUTPUT card
  - Defines base values for OUTPUT level keywords
  - Only one MERGE=YES statement per context is used
    - First at the job or step level
      - *Job level also applies to JESDS data sets*
  - Applies to all SYSOUT in that context
  - Other OUTPUT statements still apply
  - Does not create additional instances of the data sets
  - Similar to JES3 non-specific FORMAT JECL
- New DDNAME= keyword on OUTPUT card
  - Points OUTPUT card to DD statements
    - Specify *ddname* or *stepname.ddname* or *stepname.procstepname.ddname*
  - Applies to all matching SYSOUT DDs in the context
    - Job or step level
  - Creates multiple instances of the data sets

OUTPUT card changes

Sample job (instream data not included):

```
//IBMUSERA JOB  (,2D07),MSGLEVEL=(1,1),CLASS=A,SYSAFF=(*)
//TEST     OUTPUT DEST=R1,MERGE=YES
//*
//STEP1     EXEC  PGM=IEBDG
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  *
//*
//STEP2     EXEC  PGM=IEBDG
//TEST2    OUTPUT DEST=R2,MERGE=YES
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  *
```

Results:

```
DDNAME   CC StepName Forms    Dest
JESMSGLG  1 JES2     STD      R1
JESJCL    1 JES2     STD      R1
JESYSMSG  1 JES2     STD      R1
SYSPRINT  1 STEP1    STD      R1
DATASET1  1 STEP1    STD      R1
SYSPRINT  1 STEP2    STD      R2
DATASET1  1 STEP2    STD      R2
```

OUTPUT card changes

Sample job (instream data not included):

```
//IBMUSERA JOB  (,2D07),MSGLEVEL=(1,1),CLASS=A,SYSAFF=(*)
//TEST1    OUTPUT FORMS=TEST,DDNAME=DATASET1
//*
//STEP1     EXEC  PGM=IEBDG
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  *
//*
//STEP2     EXEC  PGM=IEBDG
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  *
```

Results:

```
DDNAME   CC StepName Forms    Dest
JESMSGLG  1 JES2     STD      LOCAL
JESJCL    1 JES2     STD      LOCAL
JESYSMSG  1 JES2     STD      LOCAL
SYSPRINT  1 STEP1    STD      LOCAL
DATASET1  1 STEP1    TEST     LOCAL
SYSPRINT  1 STEP2    STD      LOCAL
DATASET1  1 STEP2    TEST     LOCAL
```

Improved processing of input phase errors

- Input phase errors are now handled like other JCL errors
  - Most input phase errors do not stop job from converting
  - Messaging similar to existing error messages
  - Support for “warning” messages from input processing
  - JECL statements are now assigned statement numbers
  - JOBDEF JCLERR= is now ignored
    - Jobs with input errors always go to conversion

Sample JCL:

```
//IBMUSERA JOB
   (,2D07),MSGLEVEL=(1,1),CLASS=ABC,SYSAFF=(BAD)
/*JOBPARM PROC=PROC99
//*
//STEP1     EXEC  PGM=IEBDG,REGON=0M
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  DATA,DLM=$$$$
```

Improved processing of input phase errors

Current (pre-2.1) output:

```
//IBMUSERA JOB  (,2D07),MSGLEVEL=(1,1),CLASS=ABC,SYSAFF=(BAD)           JOB00767
************ ILLEGAL JOB CARD - VALUE OF CLASS= EXCEEDS 1 CHARACTER ***********
/*JOBPARM PROC=PROC99
******* NON-VALID JOBPARM  STMT - UNEXPECTED KEYWORD DETECTED - PROC     ******
//*
//STEP1     EXEC  PGM=IEBDG,REGON=0M
//SYSPRINT  DD  SYSOUT=*
//DATASET1  DD  SYSOUT=*
//SYSIN     DD  DATA,DLM=$$$$
******** NON-VALID DD       STMT - VALUE FOR DLM      KEYWORD NOT VALID *******
$HASP106 JOB DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION
------ JES2 JOB STATISTICS ------
           17 CARDS READ
            7 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            0 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
```

Improved processing of input phase errors

Output with 2.1:

```
12.43.45 JOB00042  IEFC452I IBMUSERA - JOB NOT RUN - JCL ERROR  533
------ JES2 JOB STATISTICS ------
           17 CARDS READ
           24 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            1 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //IBMUSERA JOB  (,2D07),MSGLEVEL=(1,1),CLASS=ABC,SYSAFF=(BAD)
        2 /*JOBPARM PROC=PROC99
          //*
        3 //STEP1     EXEC  PGM=IEBDG,REGON=0M
        4 //SYSPRINT  DD  SYSOUT=*
        5 //DATASET1  DD  SYSOUT=*
        6 //SYSIN     DD  DATA,DLM=$$$$
 STMT NO. MESSAGE
        1 HASP110 value of CLASS= parameter is not valid
        1 HASP112 value of SYSAFF= parameter is not valid
        2 HASP107 UNEXPECTED KEYWORD DETECTED - PROC
        3 IEFC630I UNIDENTIFIED KEYWORD REGON
        6 HASP107 value for DLM keyword not valid
```

Performing interpretation after conversion

- First pass of processing JCL is called conversion processing
  - Deals with PROCs and INCLUDEs
  - Does basic parsing of JCL into “text units”
  - Currently run in subtask in JES2 address space in conversion phase
- Interpretation processing creates control blocks for initiator to run job
  - Fully validates JCL (values of parameters)
    - Certain errors only detected by the interpreter
  - Currently runs in the initiator when a job is selected
- OUTPUT card processing cannot be done until the interpreter is run
  - Needs to run against control blocks interpreter creates
  - OUTPUT cards (including JESDS= and MERGE=YES) are not applied if the job does not run
    - JCL error, job canceled, etc.

Performing interpretation after conversion

- New option to run interpreter after converter before job runs
  - New keyword on JOBDEF (scope is a member)
    - INTERPRET=INIT|JES
      - INIT – Run interpreter when job is selected to run
        - *Default traditional way to run*
      - JES – Run interpreter after conversion
        - *Converter and interpreter run in conversion phase*
- When job is selected, interpreter is not run again
  - Control block are read from SPOOL and relocated
- Must be all z/OS 2.1 and $ACTIVATE at LEVEL=Z11 to run this way
  - Otherwise setting has no effect
- Job must have gone through input phase on z/OS 2.1
- **NOTE:** No LOCATE processing is done for data sets used in JCL

EXPORT JCL statement

- Previously, JCL symbols existed only during JCL conversion processing
  - Not saved as part of the output of the converter
- New EXPORT JCL statement places symbols in converter output
  - Select what symbols are to be made available
  - Signals that NEXT setting of symbol should be exported
  - Scope of any symbol is the step
- Exported symbols can be used in a number of ways:
  - Programmatically accessed using new symbol services
    - JCL service IEFSJSYM
    - JES service IAZSYMBL
  - Passed to a job being submitted through an internal reader
  - Substituted in instream data sets

EXPORT JCL statement

- EXPORT statement has the following syntax

```
//MYEXPRT EXPORT SYMLIST=(A,B,C,…)
```

  - SYMLIST is a list of symbols to export, without the “&” character
    - A,B,C in the example are JCL symbol names
      - *Same syntax rules as JCL symbol*
    - Specifying SYMLIST=* exports all JCL symbols
      - *No other use of generics is supported*
- EXPORT statement can appear anywhere after the JOB statement
  - Applies to all steps after the export statement
  - SET statements must set the symbol AFTER the EXPORT
  - Last value SET it a step is the value exported
    - A JCL symbol has only ONE value during the execution of a step
  - No limit as to the number of EXPORT statements in a job

EXPORT JCL statement - Example

```
//EX1     EXPORT SYMLIST=(DSN,MEMB)
//SET1    SET    DSN='SYS1.SAMPLIB‘
//SET2    SET    MEMB=SAMP2
//STEP1   EXEC   PGM=USERPGM1
//INPUT   DD     DSN=&DSN(&MEMB)
//SET3    SET    MEMB=SAMP3
//OUTPUT DD      DSN=&DSN(&MEMB)
```

- In STEP1 the following symbols are exported (available at run time)
  - DSN with value SYS1.SAMPLIB
  - MEMB=SAMP3
- Note that symbol MEMB had 2 values in the JCL for STEP1
  - SAMP2 from SET2 and SAMP3 from SET3
- EXPORT can only set one value per step
  - SAMP3 was the last value set to symbol in the JCL for the step

EXPORT JCL statement - Listing

```
//EX1      EXPORT SYMLIST=(DSN,MEMB)
//SET1     SET    DSN='SYS1.SAMPLIB'
//DSN      EXPORT EXPSET=SYS1.SAMPLIB               GENERATED STATEMENT
//SET2     SET    MEMB=SAMP2
//MEMB     EXPORT EXPSET=SAMP2                       GENERATED STATEMENT
//STEP1    EXEC   PGM=USERPGM1
//INPUT    DD     DSN=&DSN(&MEMB),DISP=SHR
IEFC653I SUBSTITUTION JCL - DSN=SYS1.SAMPLIB(SAMP2),DISP=SHR
//SET3     SET    MEMB=SAMP3
//MEMB     EXPORT EXPSET=SAMP3                       GENERATED STATEMENT
//OUTPUT   DD     DSN=&DSN(&MEMB),DISP=SHR
IEFC653I SUBSTITUTION JCL - DSN=SYS1.SAMPLIB(SAMP3),DISP=SHR
```

EXPORT JCL statement - Example

```
//EX1     EXPORT SYMLIST=(S1,L1)
//SET1    SET    S1=STEWART,J1=JFK,N1=NIAGARA,L1=LAGUARDIA
//SET2    SET    S1=SANDIEGO,F1=FRESNO
//EX2     EXPORT SYMLIST=F1
//STEP1   EXEC   PGM=USERPGM1
//STEP2   EXEC   PGM=USERPGM2
//SET3    SET    S1=MSP
```

- In STEP1 the following symbols are exported (available at run time)
  S1 with value SANDIEGO
  L1 with value LAGUARDIA
  F1 with a null value (not SET after export)
- In STEP2 the following symbols are exported (available at run time)
  S1 with value MSP
  L1 with value LAGUARDIA
  F1 with a null value (not SET after export)

EXPORT JCL statement - Listing

```
//EX1      EXPORT SYMLIST=(S1,L1)
//SET1     SET    S1=STEWART,J1=JFK,N1=NIAGARA,L1=LAGUARDIA
//S1       EXPORT EXPSET=STEWART                  GENERATED STATEMENT
//L1       EXPORT EXPSET=LAGUARDIA                GENERATED STATEMENT
//SET2     SET    S1=SANDIEGO,F1=FRESNO
//S1       EXPORT EXPSET=SANDIEGO                 GENERATED STATEMENT
//EX2      EXPORT SYMLIST=F1
//STEP1    EXEC   PGM=USERPGM1
//STEP2    EXEC   PGM=USERPGM2
//SET3     SET    S1=MSP
//S1       EXPORT EXPSET=MSP                      GENERATED STATEMENT
```

JCL Symbol service (IEFSJSYM)

- New service to access to JCL symbols at runtime
  - Symbols scope is a job step
    - all tasks running in the job step
  - EXPORT JCL statement specifies symbols to make available
  - Symbols are static and cannot be modified
  - Symbol names, length, and value follow JCL rules
  - Must be used after step has started running
    - Not available in exits like IEFUJI
- Also available via LE interface CEEGTJS

JCL Symbol service (IEFSJSYM)

- Usage information
  - All symbols (input and returned) do not include leading '&'
  - Output areas provided by caller
    - Specified as SYMBAREA= and SYMBAREALEN=
    - Mapped by IEFSJSYD
  - Two request types
    - REQUEST=GETALL returns all JCL symbols and values for job step
    - REQUEST=GETBYNAME returns symbol values for the symbol names passed
      - *SYMLISTARRAY= and NUMENTRIES= provides list*
      - *Generics '*' and '?' supported*

JES Symbols

- New type of symbols – JES Symbols
- JES symbols can be used:
  - To communicate between applications running in the same job step
  - To be passed as JCL symbols to jobs submitted through internal reader (SYMLIST feature)
  - To be used for substitution in instream data sets (SYMBOLS feature)
  - To communicate between application and JES internal reader (via a set of special predefined JES symbols)
- JES Symbols are very similar to JCL symbols with some exceptions:
  - Longer name – up to 16 characters
  - Longer value – up to 4096 bytes
  - Can be associated with task or step
  - Can be updated and deleted
- Symbol names starting with “SYS_” should only be set by JES

JES Symbol Service (IAZSYMBL)

- JES Symbol Service (IAZSYMBL) manages JES symbols
- Functions available are
  - CREATE – create one or more JES symbols and assign initial values
  - UPDATE – update values of selected JES Symbols
  - DELETE – delete JES Symbols and their values
  - EXTRACT – return values of selected JES Symbols
- Service supports operations over multiple JES symbols at a time
  - Including use of wildcards for EXTRACT and DELETE requests
- EXTRACT operation searches for requested symbols in this order:
  - Task level JES symbols
  - Step level JES symbols
  - Exported JCL symbols

JES Symbol service (IAZSYMBL)

- Interface to JES Symbol Service includes two macros
  - IAZSYMBL – invocation macro
  - IAZSYMDF – input/output mapping macro
    - Define input parameter list
    - Also defines various structures used by the service
- Similar in style to SSI interface
- Key data structures used by the JES Symbol Service:
  - Parameter structure
    - Filters, options, return and reason codes, feedback data etc
  - JES Symbol table structure which is used
    - for input - to create and update JES Symbols
    - for output – to return symbol information

Using system symbols in batch JCL

- New option on JOBCLASS to allow system symbols substitution
  - SYSSYM=DISALLOW|ALLOW
  - ALLOW performs system symbol substitution in batch jobs
  - Set via $T JOBCLASS command (MAS scope)
    - Remembers setting over restarts
- Most work is in the converter, JES2 owns the external
  - Only applies if job converted on a z/OS 2.1 converter

Controlling where a job converts

- Currently conversion is controlled by factors like system affinity
    - Includes new SYSTEM and SYSAFF from JOB card
    - Also includes minimum MVS level set at input phase
        - Due to use of symbols
- New in z/OS 2.1 is ability to select conversion system based on SCHENV
    - New option on JOBDEF (scope MAS)
        - CNVT_SCHENV=IGNORE|HONOR
        - Scheduling Environment parsed at input phase (or assigned)
        - HONOR says to only convert where SCHENV is available
            - *When not available, job waits to convert (AWAITING CONVERSION)*
    - Only applies if
        - The job that went through input on z/OS 2.1
        - Job is being selected by z/OS 2.1 converters
            - *Down level members can select even if HONOR is set*

Instream symbol substitution

- Symbols can now be used in instream data (similar to in JCL)
  - Instream data created by JES or converter
- Substitution occurs when application reads data
  - Using current value of any symbol
- Symbols can come from
  - JCL symbols made available using the EXPORT JCL statement
  - JES symbols created using JES Symbol Service (IAZSYMBL)
  - MVS system symbols either from conversion or execution system
- Substitution is controlled by a new keyword SYMBOLS
  - On DD data or DD * statement
    - The default is not to perform symbol substitution
- Optional diagnostic logging of substitution can be requested
  - Controlled by the second value of the SYMBOLS keyword

Instream symbol substitution

- Format of SYMBOLS= keyword on DD * or DD DATA
  `SYMBOLS=[(] JCLONLY | EXECSYS | CNVTSYS [, `*`DDname`*` )]`
- SYMBOLS=JCLONLY
  - Substitute EXPORTed JCL symbols and JES symbols (IAZSYMBL)
- SYMBOLS=EXECSYS
  - Substitute EXPORTed JCL, JES symbols and MVS system symbols
    - Execution system MVS symbols used
- SYMBOLS=CNVTSYS
  - Substitute EXPORTed JCL, JES symbols and MVS system symbols
    - Conversion system MVS symbols used (consistent with SYSSYM)
- *DDname*
  - Optional DD name which will receive the symbol substitution log
    - Any output DD name in the step can be used

Instream symbol substitution - Example

```
//        EXPORT SYMLIST=(DSN,VOL)
//        SET DSN='ABC.DATA',VOL='123456'
//STEP1   EXEC PGM=USERPGM1
//DATA    DD  DSN=&DSN,DISP=SHR
//SYSIN   DD  *,SYMBOLS=EXECSYS
   SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL
   FUNCTION='&APPL_NAME'
/*
```

- Application created JES symbol APPL_NAME
  - Value set to 'RECORD SEARCH'.
- Application reading data on SY1 will see these records

```
   SYSTEM=SY1,DSNAME=ABC.DATA,VOLUME=123456
   FUNCTION='RECORD SEARCH'
```

Instream symbol substitution – Log example

```
//MYLOG   DD  SYSOUT=A
//SYSIN   DD  *,SYMBOLS=(EXECSYS,MYLOG)
SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL

. . .
```

- The MYLOG data set will show results of the substitution

```
SYSIN    : RECORD 1 BEFORE SUBSTITUTION
SYSIN    : SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL
SYSIN    : RECORD 1 AFTER SUBSTITUTION
SYSIN    : SYSTEM=SY1,DSNAME=ABC.DATA,VOLUME=123456
```

Instream symbol substitution rules

- Symbols and value can be different lengths
- Substitution attempts to preserve columns
  - Will compress out blanks to right of symbol

```
12345678901234567890123456789012345678901234567890123456789
SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL           Comment

&SYSNAME=SY1 &DSN=ABC.DATA.TEXT  &VOL=WORK12
SYSTEM=SY1,DSNAME=ABC.DATA.TEXT,VOLUME=WORK12     Comment
```

- If not enough blanks, data is shifted right

```
&SYSNAME=SY1 &DSN=ABCDEFG.DATASET.TEXT  &VOL=WORK12
SYSTEM=SY1,DSNAME=ABCDEFG.DATASET.TEXT,VOLUME=WORK12 Comment
```

- Data can be shifted beyond LRECL
  - Will present long record to application (potentially causing I/O error)
  - Application can attempt to recover from error by passing longer buffer
- **JCL coder's responsibility to ensure enough room for substitution**

CATALOG PROC example

- PROC to uncatalog and recatalog a data set

```
//CATALOG   PROC DS=,V=,CAT='PAGE08.CATALOG'
//          EXPORT SYMLIST=(DS,V,CAT)
//          SET  DS=&DS,V=&V,CAT=&CAT
//CATSTEP   EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *,SYMBOLS=JCLONLY
 DELETE &DS                -
   NOSCRATCH CATALOG(&CAT)
 DEFINE NONVSAM (NAME(&DS)  -
   DEVT(3390) VOL(&V))      -
   CAT(&CAT)
```

- Invoking job

```
//SYMTEST JOB    MSGLEVEL=(1,1),CLASS=A
//        SET    DSN='SYS1.CATALOG.TEST',VOLSER=STORAG
//CAT     EXEC   CATALOG,DS=&DSN,V=&VOLSER
```

CATALOG PROC listing

```
//SYMTEST JOB    MSGLEVEL=(1,1),CLASS=A
//        SET    DSN='SYS1.CATALOG.TEST',VOLSER=STORAG
//CAT     EXEC   CATALOG,DS=&DSN,V=&VOLSER
IEFC653I SUBSTITUTION JCL - CATALOG,DS=SYS1.CATALOG.TEST,V=STORAG
XXCATALOG   PROC DS=,V=,CAT='PAGE08.CATALOG'
XX          EXPORT SYMLIST=(DS,V,CAT)
XX          SET  DS=&DS,V=&V,CAT=&CAT
IEFC653I SUBSTITUTION JCL - DS=SYS1.CATALOG.TEST,V=STORAG,CAT=PAGE08.CATALOG
XXDS       EXPORT EXPSET=SYS1.CATALOG.TEST           GENERATED STATEMENT
XXV        EXPORT EXPSET=STORAG                      GENERATED STATEMENT
XXCAT      EXPORT EXPSET=PAGE08.CATALOG              GENERATED STATEMENT
XXCATSTEP   EXEC PGM=IDCAMS
XXSYSPRINT  DD   SYSOUT=A
XXSYSIN     DD   *,SYMBOLS=(JCLONLY,SYMOUT)
XXSYMOUT    DD   SYSOUT=*
XXDS       EXPORT EXPSET=                            GENERATED STATEMENT
XXV        EXPORT EXPSET=                            GENERATED STATEMENT
XXCAT      EXPORT EXPSET=                            GENERATED STATEMENT
```

- Can also be used as a started task

s catalog,ds=sys1.testing.cat2,v=storag

Passing symbols to internal readers

- Symbols (JCL and JES) can now be passed on the internal reader
  - No need to update JCL to set parameters to batch jobs
  - Treated as SET commands after the JOB card
  - Available for substitution during JCL processing
    - Like SET symbols
  - Implicitly exported in the submitted job
    - EXPORT statement is not required for passed symbols
  - Parent job and submitted job can now use consistent set of symbols
- The following symbols can be passed to internal reader:
  - JCL symbols available to a parent job
  - JES symbols that conform to JCL requirements
    - Symbol name and value length
    - Character set for symbol name

Passing symbols to internal readers

- Syntax for statically allocated internal reader:

```
//SYSUT2   DD SYSOUT=(A,INTRDR),SYMLIST=(A,B,C)
```

  - SYMLIST is a list of symbols to pass, without the “&” character
    - A,B,C in the example are symbol names
    - Specifying SYMLIST=* exports all JCL compatible symbols
      - *No other use of generics is supported*
  - SYMLIST is a list of symbol names only
    - Values are extracted during job submission
    - Values of JES symbols CAN change between jobs
- Dynamically allocated internal reader has equivalent function
  - Text unit DALSYML (TU key X'802B')

Symbols on internal reader – example

- Job SYMSAMP submitting job CATALOG with INTRDR symbols

```
//SYMSAMP   JOB   MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=IBMUSER
//          EXPORT SYMLIST=(DSN,VOLSER)
//          SET    DSN=TEST.JES.LINKLIB,VOLSER=STORAG
//STEP1     EXEC   PGM=IEBGENER
//SYSPRINT  DD     SYSOUT=*
//SYSUT2    DD     SYSOUT=(A,INTRDR),SYMLIST=(*)
//SYSIN     DD     DUMMY
//SYSUT1    DD     DATA,DLM='%%'
//CATALOG   JOB    1,CATALOG,MSGLEVEL=(1,1),CLASS=A
//*
//CATUSER   EXEC   PGM=IDCAMS
//SYSPRINT  DD     SYSOUT=A
//SYSIN     DD     *,SYMBOLS=(JCLONLY,SYMLOG)
   DEFINE NONVSAM (NAME(&DSN) DEVT(3390) VOL(&VOLSER))              -
              CAT(PAGE08.CATALOG)
//SYMLOG    DD     SYSOUT=A
%%
```

Symbols on internal reader – example

- Job SYMSAMP as submitted (from SDSF SJ)

```
//CATALOG   JOB    1,CATALOG,MSGLEVEL=(1,1),CLASS=A
//*
//CATUSER   EXEC   PGM=IDCAMS
//SYSPRINT  DD     SYSOUT=A
//SYSIN     DD     *,SYMBOLS=(JCLONLY,SYMLOG)
   DEFINE NONVSAM (NAME(&DSN) DEVT(3390) VOL(&VOLSER))              -
              CAT(PAGE08.CATALOG)
//SYMLOG    DD     SYSOUT=A
```

- Job from JCLIN listing data set

```
//CATALOG   JOB    1,CATALOG,MSGLEVEL=(1,1),CLASS=A
// SET DSN=TEST.JES.LINKLIB                          GENERATED STATEMENT
//DSN      EXPORT EXPSET=TEST.JES.LINKLIB            GENERATED STATEMENT
// SET VOLSER=STORAG                                 GENERATED STATEMENT
//VOLSER   EXPORT EXPSET=STORAG                      GENERATED STATEMENT
//*
//CATUSER   EXEC   PGM=IDCAMS
//SYSPRINT  DD     SYSOUT=A
//SYSIN     DD     *,SYMBOLS=(JCLONLY,SYMLOG)
//SYMLOG    DD     SYSOUT=A
```

Symbols on internal reader – example

- IDCAMS SYSPRINT data set

```
IDCAMS  SYSTEM SERVICES
   DEFINE NONVSAM (NAME(TEST.JES.LINKLIB) DEVT(3390) VOL(STORAG))   -
              CAT(PAGE08.CATALOG)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

- SYMLOG data set

```
SYSIN    : RECORD 1 BEFORE SUBSTITUTION
SYSIN    :    DEFINE NONVSAM (NAME(&DSN) DEVT(3390) VOL(&VOLSER))           -
SYSIN    : RECORD 1 AFTER SUBSTITUTION
SYSIN    :    DEFINE NONVSAM (NAME(TEST.JES.LINKLIB) DEVT(3390) VOL(STORAG))   -
SYSIN    : RECORD 2 BEFORE SUBSTITUTION
SYSIN    :               CAT(PAGE08.CATALOG)
SYSIN    : RECORD 2 AFTER SUBSTITUTION
SYSIN    :               CAT(PAGE08.CATALOG)
```

Symbols returned from internal reader (submit)

- Internal reader now sets JES symbols for jobs processed
  - Processing always performed
  - Symbols set when a job is successfully submitted
- List of JES symbols set:
  - SYS_CORR_LASTJOB
    The 64-character job correlator of the job which was just submitted
  - SYS_LASTJOBID
    The 8-character JES job identifier of the job which was just submitted
- When job submission fails, these symbols are set to a null value
- Use JES Symbol Service (IAZSYMBL) EXTRACT function to access
  - These are task level symbols
    - Extract must be done in same task as job submission

Minimum MVS level

- New minimum z/OS level associated with a job
  - Set if JOB uses certain functions only available in a specific release
  - Can be set by
    - JES2 during input processing.
      - *eg. If symbols passed on the initiator*
    - The MVS converter
      - *eg. If SYMBOL= on DD card is specified*
    - JES2 parameter settings
      - *eg. If DSENQSHR=AUTO on JOBCLASS*
  - Displays using $DJ,LONG keyword REQUIRES_ZOS=
    - Level effectively alters job affinity
  - Future releases or maintenance may add new requirements

Questions?

Questions?

Session 14919

Mainframe 50 April 7th 1964 - April 7th 2014
http://www.ibm.com/mainframe50/