

Batch Modernization in JES2 z/OS 2.1

Tom Wasik
IBM Rochester, MN

Tuesday 1:30PM
Session Number 14233



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
 - Input phase JCL/JECL error processing changes
 - Performing interpretation after conversion
 - JCL symbol processing changes
 - Eight character job classes
 - Job class security checks



JOB card JCL changes

- New SYSTEM= keyword on JOB card
 - List MVS system names where job can run
 - Valid names are systems that are/were active in the JESPLEX (MAS)
- 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
- 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
- JOBPARM SYSAFF ignored if JOB card SYSTEM or SYSAFF specified
 - New warning message generated



JOB card JCL changes

- New UJOB CORR= 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



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



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
- JES2 PROCLIB support has been updated
 - Display now includes concatenation in the JES2 PROC
 - Displayed as “STATIC PROCLIB”s
 - Can be override as in previous releases
 - Static PROCLIBs can be “altered” more easily using \$T PROCLIB
 - Creates override based on static values
 - Quicker close/unallocate of unused concatenations

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



Performing interpretation after conversion

- If INTERPRET=JES is selected
 - Both converter and interpreter run in a new address space
 - Address space name *jesxC1xx*
jesx subsystem name, *xx* is an instance number
 - Define address space to RACF same as JES2
 - *Needs access to all PROCLIB data sets*
 - *Does VERIFY CREATE for jobs being converted*
 - PCEDEF CNVNUM= controls number of converters run
 - CISUB_PER_AS=*nn* – indicates how many subtasks per JES2CI
 - Default is 5 per address space
 - Default number of converters is 10
 - Default number of JES2CI address spaces is 2 (10/5)
JES2CI01 and JES2CI02



Performing interpretation after conversion

- Implications of code change
 - CNVT code to read IOT CBs moved to subtask
 - Calls CBIO exit 8 instead of exit 7
 - Exit 6 compatibly updated with this release
 - Same data passed to exit in registers 0 and 1
 - New is an XPL in register 2
 - *XPL makes it easier to set job CLASS, SCHENV, and HOLD*
 - Existing exits should continue to function without change
 - *Presuming they use data areas passed to the exit*
 - *CNVT DTE (\$DTECNV) structure has changed significantly*
 - *Many field moved to \$CIWORK*
 - New internal data set to store interpreter output
 - \$SWABLKS DD name
 - Always created by z/OS 2.1 JES2 during input phase

Performing interpretation after conversion

- Implications of setting INTERPRET=JES
 - Converter and interpreter run outside the JES2 address space
 - Exit 8 cannot access JES2 private storage
 - New exit 60 to perform function of exit 6
 - Exact same parameters as exit 6 (New XPL in register 2)
 - *XPL includes setting job CLASS, SCHENV, and HOLD*
 - Gets control in USER environment (R11 is HCCT)
 - Run in JES2CI address space (cannot access JES2 private storage)
 - New sample exits HASX60A and HASX60B
- New exit 59
 - Invoked after interpreter call and after processing OUTPUT cards
 - Invoked before writing data to \$SWABLKS DD
 - Same XPL as exit 60 (but passed in register 1)



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=EXEC SYS
            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 , DSNNAME=&DSN , VOLUME=&VOL                               Comment
```

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

- If not enough blanks, data is shifted right

```
&SYSNAME=SY1  &DSN=ABCDEFGH.DATASET.TEXT  &VOL=WORK12
SYSTEM=SY1 , DSNNAME=ABCDEFGH.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**



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

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=YES|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)

Job class groups

- Group batch job classes to simplify specification
 - New attribute of a job class is the GROUP name
 - Each job class can be associated with only one group
 - Applies to traditional and multi-character batch job classes
 - Group names must be unique and cannot match any job class
 - Cannot have 1 character group names
 - Command to display groups and classes in group
 - Group name can be specified on initiator or device selection
 - Round robin selection is done for classes in a group

JOBCLASS commands

- The \$ADD JOBCLASS command defines a new job class
 - Same operands as \$T JOBCLASS or JOBCLASS init statement
 - Class name cannot match an existing job class group
 - All members must be z/OS 2.1
 - Must be in z11 \$ACTIVATE mode

```
$ADD JOBCLASS ( CLASS1 ) , ACTIVE=YES , MODE=WLM
```

```
$HASP837 JOBCLASS ( CLASS1 ) ACTIVE=YES , GROUP= , MODE=WLM ,  
$HASP837 QAFF= ( ANY ) , QHELD=NO , SCHENV= ,  
$HASP837 XEQCOUNT= ( MAXIMUM=* , CURRENT=0 ) ,  
$HASP837 XEQMEMBER ( IBM1 ) = ( MAXIMUM=* ,  
$HASP837 CURRENT=0 )
```

JOBCLASS commands

- The \$DEL JOBCLASS command deletes a job class
 - Cannot delete 1 character job classes or STC/TSU
 - Class must be set to ACTIVE=NO before deleting it
 - Class deleted from
 - Group (if any) it is in
 - Any selection list (removed from list)
 - Default job class (replace with JOBDEF default class)
 - *Cannot delete from all default setting – be careful*
 - Filters are supported

```
$DEL JOBCLASS (CLASS1 )
```

```
$HASP837 JOBCLASS (CLASS1 ) - ELEMENT DELETED
```

JOBCLASS commands

- \$T JOBCLASS command updated with new operands
 - Is job class usable for new jobs entering the systems
 - ACTIVE=YES|NO
 - Specifies the job class group the job is in
 - GROUP=*grpname*

```

$T JOBCLASS(CLASS1,CLASS2),GROUP=XYZ,ACTIVE=YES
$HASP837 JOBCLASS(CLASS1) ACTIVE=YES,GROUP=XYZ,MODE=JES,
$HASP837 QAFF=(ANY),QHELD=NO,SCHENV=,
$HASP837 XEQCOUNT=(MAXIMUM=*,CURRENT=0),
$HASP837 XEQMEMBER(IBM1)=(MAXIMUM=*,
$HASP837 CURRENT=0)
$HASP837 JOBCLASS(CLASS2) ACTIVE=YES,GROUP=XYZ,MODE=JES,
$HASP837 QAFF=(ANY),QHELD=NO,SCHENV=,
$HASP837 XEQCOUNT=(MAXIMUM=*,CURRENT=0),
$HASP837 XEQMEMBER(IBM1)=(MAXIMUM=*,
$HASP837 CURRENT=0)

```

CLASSGRP commands

- New command to display job class groups
 - Displays group name and list of classes in the group
 - `$D CLASSGRP(name)`

```
$D CLASSGRP
```

```
$HASP816 CLASSGRP (TEST)
```

```
B , NONE , A
```

```
$HASP816 CLASSGRP (XYZ)
```

```
CLASS1 , CLASS2
```

Specifying job classes

- Specifying job class selection list (CLASS=)
 - Applies to \$T INITnn, OFFn.JR and OFFn.JT
 - Two syntax now supported
 - List of 1-36 single character classes
 - List of 1-8 job classes or job class groups
- Traditional syntax - 1-36 single character classes
CLASS=**ABCD**
- New syntax – 1-8 classes or class groups
CLASS= (**CLASSX , CLASSY , GROUP1**)
CLASS= (**A , B , C , D**)
- Presence of parenthesis indicates syntax in use
 - Beware of CLASS=TEMP vs CLASS=(TEMP)
 - First is 4 classes T,E,M,P second is 1 class (or group) TEMP

Initialization statements

- New JOBCLASS operands added to initialization statement
 - ACTIVE and GROUP operands
 - As with other operands, only applies on cold start
- New job classes can be added on any start
 - Any JOBCLASS in the init deck that does not exist is added
 - All members must be 2.1 and must be z11 \$ACTIVATE mode
 - Operands set (or defaulted) on new class are honored
 - CAN RESURRECT DELETED JOB CLASS
 - *Always keep init deck current*
- Cannot delete JOBCLASS using init deck
- Changes to CLASS= specification also apply to init statements
 - CLASS=TEMP vs CLASS=(TEMP)

JCL, SMF records, etc.

- CLASS= on JOB card updated to support 1-8 characters
- General philosophy for 1 character class fields in multiple CBs
 - Existing 1 byte fields set to 1st character of multi character class
- MVS JMR updated to support longer class value
 - New extended JMR created to add space to the JMR
 - At end of current JMR area
 - JMRVERSN set to 1 if extension exists
 - New job class field JMRCLAS8 (always set to job class)
- SMF field SMF26WJC set to full job class value
- SMF field SMF30CL8 (formerly JES3 only) used for JES2 class value
- SSIs already had 8 character fields for job class
 - Code now populates/uses entire 8 character field

Coexistence

- MAS with down level releases supported (with appropriate maintenance)
 - However cannot start down level member if
 - A multi character job class is defined
 - A job is in a job class group
 - If a job class is ACTIVE=NO
 - Down levels can start but they ignore ACTIVE setting
- Cannot \$ACTIVATE to z2 level if
 - A multi character job class is defined
 - A job is in a job class group

Exit considerations (exploitation actions)

- Once 8 character job classes defined:
 - 36 class queue heads in the \$HCT (\$JQHEAD1) not updated
 - For new and existing classes
 - JQETYPE does not reflect job class for 8 character classes
 - JQETYPE unchanged for 1 character classes
 - 8 character field set in new field in \$JCT (in an IBM extension)
- To simplify exit writing do not update class in JCT/JQE if possible
 - Use fields in XPLs to update job class (1 or 8 byte value)
 - Exit 6 updated with XPL and field in XPL to set class
 - Other exits (2, 4, 50, 52, 54 etc) already had class fields

SAF/RACF checks for job classes

- JES2 now supports verifying a jobs access to a job class
 - This eliminates a number of customer exits to do similar check
- The check verifies READ access to a entity name in the JESJOBS class
 - `JOBCLASS.nodename.jobclass.jobname`
 - *nodename* is the local NJE node name
 - *jobclass* is the job class to be associated with the job
 - *jobname* is the job name of the job being submitted
- The check is activated by 2 profiles in the FACILITY class
 - JES.JOBCLASS.OWNER controls an owner check
 - JES.JOBCLASS.SUBMITTER controls a submitter check
 - If both profiles exist, 2 AUTH checks are made
 - *Both checks must PASS for job to continue*
 - Only search for discrete (non generic) profiled in the FACILITY class
 - Access list and UACC of FACILITY class profiles are **not** used

SAF/RACF usage example

- User TOM submits a CLASS=C job named TEST with USER=TONI
 - NJE Note name is POK
- If there is a JES.JOBCLASS.OWNER profile in the FACILITY class
 - A check is made if user TONI has READ access to JESJOBS profile
JOBCLASS.POK.C.TEST
- If there is a JES.JOBCLASS.SUBMITTER profile in the FACILITY class
 - A check is made if user TOM has READ access to JESJOBS profile
JOBCLASS.POK.C.TEST
- If both FACILITY class profiles exist, then TOM and TONI must have access to the JESJOBS class profile

JESJOBS class considerations

- Activating JOBCLASS checking considerations:
 - Ensure profiles exist to permit jobs that need to run have access
 - Start with a JOBCLASS.** profile with UACC of read
 - *Blanket permits every job to run*
 - Add profiles to restrict job classes
 - *Consider warn mode at first*
 - *JOBCLASS.*.R.* restricts class R for all job names and nodes*
 - If JESJOBS class never used
 - Ensure before starting that SUBMIT.** has appropriate access
 - Review documentation on CANCEL.**
 - Refrain from a * profile with UACC other than NONE
 - If JESJOBS class is in use, check for * profiles
 - OK if UACC of NONE, not a good idea if not NONE
- Create facility class profile AFTER setting up JESJOBS profiles.

System z Social Media

- System z official Twitter handle:
 - [@ibm_system_z](https://twitter.com/ibm_system_z)
- Top Facebook pages related to System z:
 - [Systemz Mainframe](#)
 - [IBM System z on Campus](#)
 - [IBM Mainframe Professionals](#)
 - [Millennial Mainframer](#)
- Top LinkedIn Groups related to System z:
 - [Mainframe Experts Network](#)
 - [Mainframe](#)
 - [IBM Mainframe](#)
 - [System z Advocates](#)
 - [Cloud Mainframe Computing](#)
- YouTube
 - [IBM System z](#)



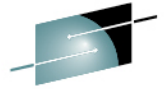
- Leading Blogs related to System z:
 - [Evangelizing Mainframe \(Destination z blog\)](#)
 - [Mainframe Performance Topics](#)
 - [Common Sense](#)
 - [Enterprise Class Innovation: System z perspectives](#)
 - [Mainframe](#)
 - [MainframeZone](#)
 - [Smarter Computing Blog](#)
 - [Millennial Mainframer](#)

Questions?

Questions?

Session 14233





SHARE
Technology • Connections • Results

