

IBM Education Assistance for z/OS V2R1

Item: Eight Character Job Class
SAF Controls on Input Class Use
Element/Component: JES2



Agenda

- Trademarks
- Presentation Objectives
- Overview - Eight Character Job Class
- Usage & Invocation - Eight Character Job Class
- Migration & Coexistence Considerations - Eight Character Job Class
- Overview - SAF Controls on Input Class Use
- Usage & Invocation - SAF Controls on Input Class Use
- Migration & Coexistence Considerations - SAF Controls on Input Class Use
- Presentation Summary



Trademarks

- See url <http://www.ibm.com/legal/copytrade.shtml> for a list of trademarks.



Presentation Objectives

- Introduce enhancements to increase the number of job classes in JES2
 - New 1-8 character job classes
 - Commands to manage and use the job classes
 - JCL and other changes in support of job class change
- Explain new SAF controls to control the use of job classes
 - Profiles used to set up the job class controls



Overview: Eight Character Job Class

▪ Problem Statement

- Job classes in JES2 have multiple uses
 - Scheduling work
 - Associate default attributes with a job
 - Determining service class
- Current 1 character job class creates limitations:
 - Can only have 36 jobs classes using A-Z and 0-9
 - Problem when merging systems
 - » Especially with corporate mergers
 - Problems merging/coexisting with JES3



Overview: Eight Character Job Class

▪ Solution

- 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)



Overview: Eight Character Job Class

▪ Problem Statement

- Specifying multi character job classes on a device/initiator
 - Getting more than 8 or so gets too long to type/manage

▪ Solution

- 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



Usage & Invocation - Commands: Eight Character Job Class

- 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 V2R1
 - 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)
```



Usage & Invocation - Commands: Eight Character Job Class

- The \$DEL JOBCCLASS 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 JOBCCLASS (CLASS1)
```

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



Usage & Invocation - Commands: Eight Character Job Class

- \$T JOBCCLASS 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 JOBCCLASS (CLASS1, CLASS2) , GROUP=XYZ, ACTIVE=YES
$HASP837 JOBCCLASS (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 JOBCCLASS (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)
```



Usage & Invocation - Commands: Eight Character Job Class

- 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
```



Usage & Invocation - Commands: Eight Character Job Class

- 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



Usage & Invocation – Initialization Statements: Eight Character Job Class

- New JOBCCLASS 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 JOBCCLASS in the init deck that does not exist is added
 - All members must be V2R1 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 JOBCCLASS using init deck
- Changes to CLASS= specification also apply to init statements
 - CLASS=TEMP vs CLASS=(TEMP)



Usage & Invocation – JCL, SMF records, etc.: Eight Character Job Class

- 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



Migration & Coexistence Considerations: Eight Character Job Class

- 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



Migration & Coexistence Considerations: Eight Character Job Class

- 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



Overview: SAF Controls on Input Class Use

■ Problem Statement

- Some job class attributes need to be restricted
 - Could be an attribute that needs restriction on usage
 - Could also be the scheduling priority the class provides

■ Solution

- New SAF/RACF authority check
 - Verifies a job's or submitter's access to a job class
 - Applies to traditional 36 and new 8 character job classes
 - Does not apply to the special job classes (STC and TSU)



Usage & Invocation: SAF Controls on Input Class Use

- 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



Usage & Invocation – Usage Example: SAF Controls on Input Class Use

- 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



Migration & Coexistence Considerations: SAF Controls on Input Class Use

- 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.



Presentation Summary

- In this presentation, we discussed new 8 character job classes and job class groups. Also discussed was the new SAF/RACF added in this release to verify a job's access to a job class.

