Users Guide - How to use the features of these Mods

- Control Statements

You take advantage of the Mellon Shared Spool mods via JECL statements. There are currently five supported statements, they are:

"/*CNTL XEQ resource, { EXC | SHR }"

"/*ROUTE XEQ secheduling environment name | HERE "

"/*WITH jobname"

"/*AFTER jobname"

"/*BEFORE jobname"

The /*CNTL XEQ statement is used to add an additional job selection criterion based on other executing jobs current use of the same resource name. The "/*CNTL XEQ" must be placed in columns 1 through 10. The resource name is arbitrary, and is made of up to eight (8) characters with no embedded blanks. The resource name follows the literal "/*CNTL" and must be preceded by at least one, but not more than 30 blank spaces. If the shared (SHR) or exclusive (EXC) keywords are used, they must immediately follow the resource name and be preceded by a comma. If neither the SHR nor EXC keyword is specified SHR is assumed.

You may specify up to eight /*CNTL statements; additional statements are disregarded.

Examples:

Assume jobnames ABC, and CDE are currently in execution with the following /*CNTL XEQ statements -

```
//ABC  JOB (other stuff)
/*CNTL XEQ MYNAME,SHR

//CDE JOB (other stuff)
/*CNTL XEQ MYNAME,SHR
```

Then if job EFG is submitted with the following, /*CNTL statement -

```
//EFG JOB (other stuff)
/*CNTL XEQ MYNAME,EXC
```

The job will not be selected for execution as long as either job ABC or CDE continues to run.

Next if job XYZ is submitted with the following, /*CNTL statement -

```
    //XYZ JOB (acctng info)
    /*CNTL XEQ     MYNAME   Ð SHR is the default
```

It would be immediately available for execution.

When no jobs remain in execution with a /*CNTL XEQ MYNAME job EFG with the EXC requirement would be available for job selection. Assume it has now been selected, and a new job enters the system with the following /*CNTL statement -

```
    //WXY  JOB  (acctng info)
    /*CNTL  XEQ  MYNAME
```

Job WXY will not be available for job execution until job XYZ that holds the resource name exclusively ends. If other jobs are submitted with resource names other than MYNAME, they will be treated separately and only other jobs with /*CNTL statements that reference the same resource name will affect their availability for job selection.

The /*CNTL statement only provides additional job selection criterion, and does not replace other JES2 requirements for job selection such as available initiators, appropriate job class and so on. The resource name is arbitrary, you make it up, there is no need for anyone to add the name to WLM, or any other table before it is used.

The /*ROUTE XEQ is a standard JES2 statement used to route your job to a specific execution node. We have usurped the use of the statement, and when we read an XEQ statement, we test to see if the name following the XEQ literal is a valid WLM scheduling environment name. If the name is a valid scheduling environment name, and if no schenv value is present on the JOB card, we use the XEQ name in the same way the job statement parameter SCHENV is used, otherwise we let JES2 handle it normally.

The literal "/*ROUTE" must begin in column 1. The literal "XEQ" must follow "/*ROUTE", and may be separated by one to twenty blank spaces. The resource name, if not a valid JES2 node name, must follow the "XEQ literal by between 1 and 35 blank spaces. The resource name can be between 1 and 16 characters long, if the name is longer than 16 characters, only the first 16 are used.

There is one specific exception to all of the above, if the literal following XEQ is HERE, then the jobs system affinity is set to the system where you submitted the job, quite a handy feature itself.

You may specify more than one /*ROUTE statement; but only the last will be used.

Examples:

```
//ABC JOB (job acctng),SCHENV=BEFOREALL
/*ROUTE XEQ AFTERALL   <== ignored because a valid schenv is specified

//ABC JOB (job acctng),SCHENV=BEFOREALL
/*ROUTE XEQ  N6  <== a valid JES2 routing node, both jes2 routing and
//*                  schenv are valid for this job.

//ABC JOB (jobacctng)
/*ROUTE XEQ BEFOREALL <== BEFOREALL is setup the same as SCHENV above

//TUV  JOB (jobacctng)
/*ROUTE XEQ  HERE   <== the jobs will only execute where submitted.
```

The /*WITH statement specifies that the job is only available for selection while the jobname named on the /*WITH statement is in execution. The condition is satisfied even if the jobname that must be executing is executing on a different system within the same MAS environment. If more than one /*WITH statement is read; only the last one is kept.

Example:

```
//FGH   JOB  (acctng info)
/*WITH    JKL
```

This job will only be selected for execution if jobname JKL is executing at the time of job selection.

The /*AFTER jobname statement specifies that the job is only available for selection after the jobname in the /*AFTER statement has finished. Because we really do not track all completed jobs, from - from when - the beginning of the day, the week, the last ipl, the last cold start, or maybe for all time, it is more correct to say that if the jobname specified on a /*AFTER statement is in execution at the time our job would otherwise have been selected for execution, then our job will wait until the jobname referenced in the /*AFTER statement has ended. "AFTER" may have been more appropriately names "NOT WHILE", but since I was not around when the original keywords were developed, please do not blame me. Again, the jobname on the /*AFTER need not be executing on the system that our job is potentially selected for execution on, it could be on any system within the MAS complex.

If more than one /*AFTER card is read; only the last one is kept.

Example:

```
//ABC JOB (acctng info)
```

```
/*AFTER    XYZ
```

Job ABC will not be available for execution while job XYZ is in execution anywhere within the MAS complex.

The /*BEFORE jobname statement causes the jobname specified in the statement to not be selected for execution until after this job has completed. Specifically if jobname ABC has a /*BEFORE XYZ statement, then if jobname XYZ is potentially selected for execution by JES2, the Mellon Mods will examine all jobs on the input queue and when jobname ABC is found to have a /*BEFORE for job XYZ, job XYZ will be rejected as a potential candidate for job execution.

If more than one /*BEFORE statement is read; only the last one is kept.

Example:

```
//ABC   JOB  (acctng info)
/*BEFORE   XYZ
```

This before statement will cause JES2 to reject job XYZ as a potential candidate for job execution until after job ABC has completed execution. If job XYZ was already executing at the time job ABC is submitted, the /*BEFORE statement will not affect anything, unless there is another job XYZ waiting to execute. Also please note, it is possible to form a lockout condition where JOBA has a /*BEFORE JOBB, and JOBB has a /*BEFORE JOBA statement. There is no checking done for this type of deadly embrace.

A few final notes concerning the relationship between /*BEFORE, and /*AFTER. Many people try to use these statements, and stack two or more jobs in the same PDS member and submit them all at the same time with one SUBMIT command. This usually works as expected, but sometimes JES2 will NOT PROCESS the jobs in the order they appear in the submitted member. This can result in a job with a /*AFTER statement for a prior job you think JES2 has already seen and processed because of the sequence in the submitted member, being processed and initiated before JES2 ever finishes reading the job that is the object of the /*AFTER statement. This problem can be avoided by making sure that jobs with /*BEFORE and /*AFTER requirements are submitted separately from each other and in an appropriate sequence.

In relationship to any of the statements above, when routing a job to another node, the additional selection criterion defined by the Mellon Mods statements, will follow the job to the new node, where it will likely no longer be appropriate. Of course if the Mellon Mods are not installed at the receiving node, the additional job selection requirements are not honored.

The JES2 $DJ command output has been extended to include information about /*CNTL, up to eight names qualified with an "E" for exclusive, or an "S" for shared, one /*WITH jobname, one /*BEFORE and one /*AFTER jobname. Examples of the extended displays are given below, please note that the information is included in either the standard or long versions of the command.

Altered Display Commands -

```
-$DJ(25926)
 $HASP890 JOB(T0SM0TTY)
 $HASP890 JOB(T0SM0TTY)  STATUS=(AWAITING EXECUTION),CLASS=X,
 $HASP890                PRIORITY=6,SYSAFF=(ANY),HOLD=(JOB),
 $HASP890                AFTER=T0SM0AF,BEFORE=T0SM0BF,WITH=T0SM0WTH,
 $HASP890                CNTL=(HOWDY-E,HOWDO-S,WILDO-S,HELLO-S)
```

```
-$DJ(25926),LONG
 $HASP890 JOB(T0SM0TTY)
 $HASP890 JOB(T0SM0TTY)  STATUS=(AWAITING EXECUTION),CLASS=X,
 $HASP890                PRIORITY=6,SYSAFF=(ANY),HOLD=(JOB),
 $HASP890                CMDAUTH=(LOCAL),OFFS=(),SECLABEL=,
 $HASP890                USERID=T0SM0,SPOOL=(VOLUMES=(JES2T1),TGS=1,
 $HASP890                PERCENT=0.0009),ARM_ELEMENT=NO,CARDS=18,
 $HASP890                REBUILD=NO,SRVCLASS=BATTSTMD,SCHENV=TEST,
 $HASP890                SCHENV_AFF=(),CC=(),AFTER=T0SM0AF,
 $HASP890                BEFORE=T0SM0BF,WITH=T0SM0WTH,
 $HASP890                CNTL=(HOWDY-E,HOWDO-S,WILDO-S,HELLO-S)
```

In addition informational messages, $HASP493 and $HASP494 are written to the log as jobs with /*CNTL, /*WITH, /*BEFORE, or /*AFTER are read. Examples of the messages follow.

These messages were issued for jobname T0SM0TTY

```
$HASP944 T0SM0TTY * -- WITH   JOBNAME = T0SM0WTH   --
$HASP944 T0SM0TTY * -- AFTER  JOBNAME = T0SM0AF    --
$HASP944 T0SM0TTY * -- BEFORE JOBNAME = T0SM0BF    --
$HASP943 T0SM0TTY * -- CONTROL INFO = HOWDY,EXC    --
$HASP943 T0SM0TTY * -- CONTROL INFO = HOWDO,SHR    --
$HASP943 T0SM0TTY * -- CONTROL INFO = WILDO,SHR    --
$HASP943 T0SM0TTY * -- CONTROL INFO = HELLO,SHR    --
```

- end of text -