ESSM (Enhanced Shared Spool Mods compared to Mellon Shared Spool Mods - by Stephen McColley

The Mellon Shared Spool Mods to JES2

The chart that begins on the next page is intended to compare the features and job control statements available in four different environments, JES2, JES3, JES2 with the free Mellon Shared Spool Mods installed, and JES2 with the ESSM product installed.

The Mellon Shared Spool Mods were originally intended to provide job routing or job holds based on resource requirements that might move from one member of a JES2 MAS member to another, or if the required resource was not currently defined the mods would ‘hold’ the job until the resources were defined somewhere within the JES2 MAS. The availability and location of these resources was determined just before a job was allowed to start so that jobs on the input queue that had not yet been started would ‘follow’ a resource that was moved from one JES2 MAS member to another without requiring a JCL change. The SCHENV= JOB parameter was an attempt by IBM to provide support similar to the resource routing provided by the Mellon Shared Spool Mods and when it was made available resource based routing was changed within the Mellon Shared Spool Mods to use the IBM provided job routing.

Later in the continuing development of the product, relative job control or the ability to schedule the release of one job relative to the presence or absence of other jobs or started tasks currently executing, or ending was provided via the //\*AFTER, //\*BEFORE, //\*WITH, and //\*WITHOUT control statements. A means of serializing a group of associated jobs was provided through the use of the /\*CNTL statement that used an arbitrary name to serialize with shared or exclusive control using the specified name. While resource routing required prior setup and definition of the resource to control job routing, no prior setup or definition of the arbitrary name used with the /\*CNTL statement was required other than agreement of the name to be used between the JCL writers of each participating job. The most recent enhancement to the Mellon Shared Mods provided for the /\*HOLDFOR and /\*HOLDTIL statements that required the submitted job to be ‘held’ until either the specified time for /\*HOLDTIL, or the length of time specified with the /\*HOLDTIL statement was satisfied, regardless of other routing requirements that might exist.

ESSM

The ESSM product, Enhanced Shared Spool Mods, is intended to fully support all of the Mellon Shared Spool Mods functions as well as other features that we believe will be helpful to our potential users. Unfortunately ESSM and the Mellon Shared Spool Mods may not both be active in the same JES2 MAS at the same time. ESSM does however support all of the Mellon Shared Spool Mods JCL and features using identical JCL as the Mellon Shared Spool Mods provided and no JCL changes are required to convert from the Mellon Shared Spool Mods to ESSM. ESSM re-introduces user defined and controlled resources independent of WLM managed resources. Both WLM managed job routing via the SCHENV= JOB parameter or the /\*ROUTE XEQ resname statement, and ESSM resource routing via the //\*NET RESOURCE=(##,resname) JECL statement can be used within the same job running on a system with ESSM installed.

The ESSM defined resources are ‘intelligent’ in this implementation, which is to say they come in several different types that include associated values. The ESSM resource values are adjusted up and down as jobs requiring those resources, and specifying them with a numeric value are selected for execution or as they end execution. ESSM already contained relative job routing, run JOB ‘A’ before JOB ‘B’, but only where JOB ‘C’ is already running for instance, because that functionality was carried over from the Mellon Shared Spool mods, but it has been significantly expanded to include Dependent Job Control as defined through JES3 JECL statements providing for very complex JOB NETWORKS. Much of the existing JES3 DJC related JCL structure is recognized by ESSM and supported in a JES2 environment. Even though ESSM will read DJC related JECL statements and produced the same JOB hold/release services that JES3 provides for JOBNETs, ESSM does not attempt to make JES2 act as JES3 does. Specifically JES3 handles device allocation and provides job setup services in a fundamentally different manner than JES2, and no attempt is made to provide JES3 allocation or setup services in a JES2 environment. Of course we are unaware of any product other than ESSM that will support, in the JES2 environment, the type of JOBNETs that can be created under JES3.

JES2 and ESSM running together produce significantly enhanced job display output, that is the output from the $DJ(j###) command is enhanced to include reasons why a job is restricted to specific members of the JES2 MAS or why, and if, it is being held by ESSM processing. This information is either not reflected in job displays for systems running the Mellon Shared Spool Mods, or is poorly represented. Finally, most of the ESSM attributes of a job can be changed through operator commands after a job is on the input queue – for instance, needed resources can be added to a job, or ESSM hold counts can be adjusted.

I hope that the following chart can help explain the services supported in JES2, JES3, JES2 with the Mellon Shared Spool Mods, and JES2 with ESSM.

Thank you,

Stephen McColley

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Native JES2 control statements** | **JES2** | **JES3** | **Mellon Spool Mods & JES2** | **ESSM & JES2** | All JES2 control statements are fully supported in all environments except JES3. |
|  |  |  |  |  |  |
| /\*ROUTE XEQ | YES | --- | YES | YES | See Note – 1 |
|  |  |  |  |  |  |
| **Native JES3 control statements** | **JES2** | **JES3** | **Mellon Spool Mods & JES2** | **ESSM & JES2** | **Ref. - MVS JCL Reference - Chapter 28 JES3 Control Statements** |
| //\*command | --- | YES | No | No | If allowed in JES2 it would cause many JES2 comments to be executed as commands. |
| //\*DATASET | --- | YES | No | No | Only supported in JES3 |
| //\*ENDDATASET | --- | YES | No | No | Only supported in JES3 |
| //\*ENDPROCESS | --- | YES | No | No | Only supported in JES3 |
| //\*FORMAT PR | --- | YES | No | No\* | Only supported in JES3 – Plans for support via ESSM in a future release. |
| //\*FORMAT PU | --- | YES | No | No\* | Only supported in JES3 – Plans for support via ESSM in a future release. |
| //\*MAIN | --- | YES | No | No | Only supported in JES3 |
| //\*NET | --- | YES | No | YES | Jobs belonging to a DJC network cannot be registered with the automatic restart manager (ARM) in JES3 or in JES2 with ESSM. |
| ---- //\*NET KEYWORD details follow ---- |  |  |  |  |  |
| NETID= | ID= | --- | YES | No | YES |  |
| ,ABCMP= | ,AC= {NOKP | KEEP} | --- | YES | No | YES |  |
| ,ABNORMAL | ,AC= {NOKP | KEEP} | --- | YES | No | YES |  |
| ,AFTER=jobname | --- | No | Note 2 | YES |  |
| ,BEFORE=jobname | --- | No | Note 2 | Yes |  |
| ,CNTL = (arbitrary\_resnam,{EXC|SHR|PRG}) | --- | No | Note 2 | Yes |  |
| ,DEVPOOL= {ANY|NET|,dev-name} | --- | YES | No | No | No device allocation features of JES3 are duplicated via ESSM in JES2 |
| ,DEVRELSE= {YES | NO} | --- | YES | No | No | No device allocation features of JES3 are duplicated via ESSM in JES2 |
| ,LASTNET={R|N|C} | --- | No | No | YES | Used on the FIRST job in a job net to indicate disposition of any ‘old’ job net of the same name. |
| ,NETREL=(netid,jobname) | --- | YES | No | YES |  |
| ,NHOLD=n | ,HC=n | --- | YES | No | YES |  |
| ,NORMAL= | NC= {D|F|R} | --- | YES | No | YES |  |
|  |  |  |  |  |  |
| ,NRCMP = {HOLD | NOHO | FLSH}  ,PC = {HOLD | NOHO | FLSH} | - | YES | No | YES |  |
| ,OPHOLD={NO,YES} | OH={NO,YES} | --- | YES | No | YES |  |
| ,RESOURCE=(#,resname) |  ,RS=resname | --- | No | No | Yes |  |
| ,RELEASE = jobname | RL= jobname | --- | YES | No | YES |  |
| ,RELSCHCT =n |RS= n |  | YES | No | NO |  |
| ,WITH=(jobname,jobname,…) |  | NO | Note 2 | YES |  |
| ,WITHOUT=(jobname,jobname,…) |  | NO | Note 2 | YES |  |
| ----- end of /\*NET keywords ------ |  |  |  |  |  |
| //\*NETACCT | --- | YES | No | No |  |
| //\*OPERATOR | --- | YES | No | Yes |  |
| //\*PAUSE | --- | YES | No | No | Note 3 |
| //\*PROCESS | --- | YES | No | No |  |
| //\*ROUTE | --- | YES | No | No |  |
| /\*SIGNOFF | --- | YES | No | No | Note 3 |
| /\*SIGNON | --- | YES | No | No | Note 3 |
| **CONTROL STATEMENTS DEFINED AND USED ONLY BY ESSM and the Shared Spool Mods.** | **JES2** | **JES3** | **Shared Spool mods & JES2** | **ESSM & JES2** | **(notes)** |
| /\*CNTL ArbitraryName,{SHR,EXC|PRG} | No | No | Yes | Yes |  |
| /\*AFTER jobname | No | No | Yes | Yes |  |
| /\*BEFORE jobname | No | No | Yes | Yes |  |
| /\*HOLDFOR hh:mm:ss | No | No | Yes | Yes |  |
| /\*HOLDTIL hh:mm [:ss] [mm/dd] | No | No | Yes\* | Yes | Date specification is only allowed with ESSM |
| /\*RESOURCE resname | /\*RESOURCE=(##,resname) | No | No | No | Yes | RS is accepted as a sub parameter of the //\*NET statement when read by ESSM. |
| //\*SNET | No | No | No | Yes | SNET is a specialized form of DJC, a sequential net. All jobs in the SNET run 1 at a time in job submission order regardless of prior job completion status. |
| /\*WITH jobname | No | No | Yes | Yes |  |
| /\*WITHOUT jobname | No | No | Yes | Yes |  |
| /\*ROUTE XEQ schenvname | No\* | No | Yes | Yes | Note 1. |
| //\*NET NETID=abc,**LASTNET={R|N|C}** | --- | No | No | YES | Used on the FIRST job in a jobnet to indicate disposition of any ‘old’ jobnet of the same name. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

--NOTES--

1. The Shared Spool Mods interrogate /\*ROUTE XEQ node name statements and if the node name is not a valid NODENAME it is assumed to be a resource routing statement and converted to the equivalent of a SCHENV=node name JOB parameter. In other words, the /\*ROUTE XEQ statement has been hijacked by the Mellon Shared Spool Mods for resource routing.
2. This statement is supported via **“/\*NET**” only in ESSM, however Mellon Shared Spool Mods provide the same function via a different control statement.
3. This control statement affects the physical state of a reader or remote node. No direct control of hardware or remote nodes is attempted in JES2 through the use of ESSM or the Mellon Shared Spool Mods.