



SHARE SESSION REPORT			
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE
61	B410	MVSN: Introduction to JES3	60
MVS New Users		David Wichmann	HIO
PROJECT	SESSION CHAIRMAN		INST. CODE
HILTI, Inc., 4115 S. 100th E. Ave., Tulsa, OK 74011, 918-627-9711 X2857			
SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER			

INTRODUCTION TO JES3

Chester Hood
 Maremont Corporation
 Mail Stop 205
 P. O. Box 1488
 Nashville, Tennessee 37202
 (615) 360-1213

LEVEL OF JES3:
 MVS/System Product Version 1 Release 3.1
 MVS/System Product Version 2

Printed: 08-19-83

INTRODUCTION

This presentation is a brief overview of JES3. It is assumed that you have at least a general conceptual understanding of MVS, but no background in JES3.

It is not a comparison of JES2 and JES3.

The main topics to be covered will be concepts, major functions, and configurations. The flow of a standard job through the system will be traced. Some of the important JES3 facilities will be examined. And finally, the user interface will be surveyed.

The presentation is on the JES3 component of OS MVS/System Product, otherwise known as JES3/SP. The JES3 component of OS MVS/System Product is supported in the System/370 environment as part of MVS/System Product Version 1 Release 3.1 and in the System/370 Extended Architecture environment as part of MVS/System Product Version 2.

First, let us examine the general functions that any Job Entry Subsystem should possess. Then, we will see how JES3 relates to MVS as it performs these functions.

The general job entry subsystem functions are:

1. System Input Source. The job entry subsystem gathers jobs and their related data from local and remote input devices.
2. System Output Facility. The job entry subsystem is responsible for disseminating output to its appropriate destination.
3. Job Awareness. The job entry subsystem provides control over the scheduling and execution of jobs. JES3 provides total job awareness in terms of resource allocation for all jobs in the system.

MVS users today can choose between JES2, which was derived from HASP, and JES3, which was derived from ASP.

Under MVS, much of the job entry subsystem executes in its own address space along side of system address spaces such as the Master Scheduler, started tasks, batch jobs, and TSO sessions. The JES address space is normally non-swappable and runs at a high dispatching priority. Also shown is the new optional JES3 Auxilliary Address Space.

The JES3 configuration shown illustrates many key elements. It also serves to point out some major JES3 concepts. A three processor (triplex) installation is being depicted. Each processor shares access to the spool volumes. One of the processors is designated as the JES3 global processor. The other processors are known as JES3 local processors.

The global must be connected to each local by a channel-to-channel adapter (CTC or CTCA). Optionally, there may be a CTC connection between the locals. The use of these connections will be covered later. Any such configuration of two or more processors connected by CTCs and/or shared DASD fits the definition of a loosely coupled multiprocessing system.

The global processor provides a single system image of the entire complex of processors. All of a job's input stream data and all of its system-printed output flow through the JES3 global. The global reads the job stream and writes it out to the spool for later scheduling and execution under MVS on either the global or one of the locals. The data which is sent across the CTC consists of messages relating to the status, ownership, and control of work on the queue, or the control of the processor.

The maximum JES3 configuration represents a significant amount of hardware resources. Performance and physical considerations may limit the actual maximum configuration for an individual installation. JES3 can support the following:

1. Only one global processor (required)
2. Up to seven MVS local processors
3. Up to 31 shared DASD queue volumes

JOB FLOW

A standard job under JES3 is scheduled through five segments of processing and all except a portion of one segment will be performed by a function that executes within the JES3 address space on the global processor. These functions are called Dynamic Support Programs (DSPs).

INPUT SERVICE.

Jobs are read from local and remote input devices attached to the global processor and placed on the spool. Input service supports card readers, tape readers, disk readers, and remote readers in a remote job processing (RJP) or a network job entry (NJE) configuration. The first phase of input processing is the reader, which is skipped by the internal reader. Jobs are read from an input device and placed on the spool in batches. The main purpose of this phase is to separate jobs and their data.

The second phase is control statement processing. This phase builds JES3 control blocks, which will be used throughout the life of the job, and modifies the defaults in the control blocks with the information which was retrieved from the control statements. These control blocks are then written to the spool.

INTERPRETER SERVICE.

The function of interpreter service is to convert JCL statements into scheduler control blocks and additional JES3 control blocks. The converter/interpreter (C/I) is used for this purpose.

The converter reads the JCL from the spool and converts it into internal text. If there are any cataloged procedures in the job, they are resolved by using user defined proclib concatenations.

Next the interpreter is used to convert internal text into scheduler control blocks. Any catalog references by the job are resolved at this time. From the scheduler control blocks and the catalog information, a set of JES3 control blocks which contain a complete profile of the job's resource requirements is created. The scheduler control blocks, job setup table, and job volume table are all written to spool.

If any JCL errors are detected (syntax and logic), only the output and the purge stages of the job will be scheduled.

MAIN SERVICE.

Main service is a segment of job scheduling which collects three areas of JES3 processing which deal with the execution of a job under MVS. The areas are:

1. Main Device Scheduling
- pre-execution setup
2. Generalized Main Scheduling
- controls workload
3. Main Servicing
- interprocessor communications
- job scheduling and termination

MAIN DEVICE SCHEDULING

The first processing a job will have done is Main Device Scheduling (MDS). MDS controls the fetching, allocation, and mounting of the direct access and tape volumes requested in the JCL of each job to be executed on a processor. MDS is broken up into the following areas:

1. Volume Fetch
2. Device Allocation
3. Volume Mount processing
4. Device Deallocation

There are several reasons for JES3 to do allocation:

1. MVS allocation is on a step basis only (delay due to resource unavailability)
2. MVS allocation does not know about other systems (required volume in use on another processor)
3. The required number of devices may never be available
4. Make the maximum use of devices
5. Have a job remain for the minimum amount of wall clock time once it has been passed to the system for execution

Jobs are selected for resource allocation based on installation job priorities in first in/first out, or FIFO, order.

Only those processors with sufficient device resources will be considered for setup. MDS performs allocation on a complex-wide basis for data sets, volumes, and devices.

This provides the ability to have complex-wide data set integrity for data sets on shared devices via an enqueue-type mechanism. This control is effected through a combination of data set name and volume serial number. If two jobs request exclusive control of a data set, they would not be scheduled for concurrent execution on the same (or different) processors.

However, if identically named data sets exist on different volumes, MDS correctly recognizes that they are separate, unique resources and would allow concurrent execution of the two jobs. If the two jobs were scheduled to the same processor, then one would be delayed because the MVS data set enqueue is still performed using only the data set name.

The ability to make a specific volume serial number unavailable for use for long periods of time is provided. MDS keeps track of which volumes are currently available throughout the complex.

It knows which volumes are being used and how they are being used. In addition it will keep a volume mounted, which was released from one job, if it is required by another job.

Fetch messages will be sent to the appropriate library.

Mount messages can be sent to a console near the device to be mounted.

MDS then verifies that the volume which was mounted was indeed the requested one.

If all volume mounts have been satisfied, then the job is ready to be scheduled for execution.

This early allocation of devices guarantees that all of a jobs data set, volume, and device requirements can be met before the job is scheduled for execution.

Usually breakdown for a job occurs as job steps complete and also at the end of the job. For dynamic allocations, breakdown will take place when the corresponding dynamic deallocation occurs.

Resources allocated to a job will be released when they are no longer required by the job.

MDS functions are optional and may be bypassed for a job as well as on a complex-wide basis.

GENERALIZED MAIN SCHEDULING.

After MDS, a job is placed onto the execution queue for the scheduling of the job by Generalized Main Scheduling (GMS). GMS controls the workload and maximizes system throughput, in addition to optimizing job selection for machine loading. GMS provides a flexible framework for establishing priority relations between job classes within groups and between groups.

JES3 has provided the installation a great many controls over job scheduling which will make it easier to manage the work flow through the complex. Effective usage of these facilities can result in a reduction of the manpower which is required for the tasks of scheduling, monitoring, and control over jobs.

These facilities include:

1. Job priority
2. Interaction of jobs within a group
3. Processor dependency (explicit and implicit)
4. Logical storage size
5. Class mix (limit one class by another class)
6. Mix of differing I/O rates (low, medium, high)
7. Initiator availability (by group, the main driving force of GMS)
8. Sequencing of related jobs by Dependent Job Control

MAIN SERVICING.

The main servicing support function controls job execution processing within the JES3 complex. Main servicing on the global processor interfaces with all other processors via a CTC and with the global processor via SRB scheduling. Job execution takes place on the global, or a local main processor.

During execution on a local processor, the job reads SYSIN data sets from the spool and writes SYSOUT data to the spool using JES3 data management routines. JES3 monitors some of the functions which a job uses. JES3 watches out for abnormal

conditions and the overflowing of prespecified output limits. Write-to-operator messages which a job issues are routed back to the global processor to be logged in the jobs output and in addition sent to console service.

Both common and dynamic allocations/deallocations are reviewed by JES3. If major decisions are necessary, then the matter is decided by MDS on the global processor.

OUTPUT SERVICE.

The output service function processes SYSOUT data sets. Output can be destined for local or remote printers, punches, TSO, an external writer, or the internal reader.

Output service consists of two phases: scheduling and writing. Scheduling is done on the basis of output characteristics which are known via JCL, JES3 control cards, or the SYSOUT class table defined at initialization. These characteristics include:

1. Data set priority
2. Data set destination
3. Device type
4. Forms name
5. Carriage tape name
6. Print train name
7. SYSOUT class
8. 3800 characteristics

Spinoff, or immediate printing, of data sets is also supported by JES3. Scheduling for these data sets occurs at the time they are spun off.

A unique output service writer is started for each active output device. At start time, the writer is associated with a given set of SYSOUT characteristics. The writer can be either hot or dynamic. Hot writers are started once by the operator and wait for a specific type of work. Dynamic writers are started and stopped under JES3 control as output to be processed appears.

On tightly coupled multiprocessors, much of the writer processing is dispatchable separately from the mainline JES3

processing, thus effectively using both instruction stream processors.

An external writer is a routine running in its own address space which communicates with JES3 through the subsystem interface and writes output to an output device. The TSO output command uses this same interface.

PURGE.

The final segment of scheduling through which all jobs must go is Purge. This is a very important stage when all remaining resources associated with a job are released for use by other jobs. The main resource held at this time is spool space. This is also the point at which the final job accounting information is written to the SMF data set (record type 26).

FACILITIES

CONSOLE SERVICE.

Console service provides communication between JES3 and the operators of the system. JES3 console support features include:

1. Multiple functional consoles (virtually unlimited vs. 32 MCS consoles)
2. Coexistence with MCS
3. Subsystem interface

MCS commands	(SVC 34)
WTD/WTOR	(SVC 35)
WTL	(SVC 36)
DOM	(SVC 87)

4. Single system image for control of the entire complex
5. JES3 and MCS commands from the same console
6. Extensive MCS and JES3 command language
7. User exits in console service

Shown is a JES3 configuration which has multiple functional consoles. JES3 supports up to 96 unique destination classes for messages as opposed to only 16 for MCS. This facility allows an installation to locate its consoles close to the logical work flow. Consoles may be located close to card readers, printers, card punches, tape drives, disk drives, and in the tape and disk libraries.

Messages pertinent to these operations can be routed to the appropriate consoles. Each RJP work station has a logical console for the control of its work flow. Functional consoles are all attached to the global processor.

JES3 console support co-exists with MVS multiple console support. JES3 consoles can be defined as JES3 consoles, MCS consoles, or both. JES3 consoles permit the entering of JES3 commands to the global or to any of the local processors. MCS support does not apply to remote consoles.

MCS consoles allow the entering of MVS commands to the processor to which they are attached. JES3/MCS consoles allow communication to both the global processor and the local processor using JES3 and MVS commands.

Consoles can also communicate with each other. The need to station an operator at each local processor can be reduced or eliminated. This provides the desired single system image from an operational standpoint.

All console message traffic in the JES3 complex is recorded in the system log (SYSLOG). In addition to the message text is a time stamp and in some cases the id of the associated console or the system which generated the message.

A JES3/MCS console with the proper authority can enter 28 MVS commands and 18 JES3 commands. Four major levels of command authority define what level of control and monitoring is authorized for a given console. Short forms are supported for many commands. In the event of a console failure, the function of the failing console can be switched to an alternate console. Many JES3 parameters which were defined at initialization time can be displayed via the INQUIRY command and changed by the MODIFY command.

On graphics type consoles, commands can be associated with program function keys (PFks). These features illustrate the human factors and RAS capabilities built into the support.

REMOTE JOB PROCESSING.

Another valuable JES3 facility is Remote Job Processing (RJP). Under RJP remote card readers, punches, printers, and console function as logical extensions of similar local devices. RJP supports both system network architecture (SNA) with synchronous data link control (SDLC) line protocols and binary synchronous communications (BSC) line protocols and terminals.

Some key RJP features include:

1. SNA support
2. BSC support for programmable and nonprogrammable terminals
3. Multileaving mode for programmable terminals
4. Password protection for each terminal or work station
5. Remote console support
6. Remote operator control

NETWORKING

JES3 Networking is an important facility which allows communication and interchange of job data between a JES3 complex and other remote JES3 complexes, as well as JES2 and VM systems.

Shown here are each of the types of systems which can participate with JES3 as "nodes" in a network.

Networking provides facilities to transmit:

- Jobs
- SYSOUT Data
- Messages
- Commands

DEPENDENT JOB CONTROL.

Dependent Job Control (DJC) is a general function in JES3 that allows the installation to manage job sequencing. Job sequence dependencies often occur because one job's output may be another job's input. There may also be catalog dependencies. These dependencies may be quite complex. An example is shown:

1. Jobs B and E are dependent on Job A.
2. Job D is dependent on Jobs B and E.
3. Job C is dependent on Job B.

To control this kind of scheduling, users specify DJC "network" control cards in each job. This defines:

1. The identification of the network to which a job belongs
2. The number of predecessor jobs (1-32767)
3. The name of successor jobs to be released (50)

JES3 creates a special queue for the network and schedules the jobs as indicated. The jobs in the network need not execute on the same processor. DJC also permits the reservation of a pool of devices for use by the network and the specification of alternate paths when preceding jobs complete abnormally.

No operator intervention is necessary to activate the DJC network. The operator does have the ability to inquire about the status of jobs in the network and to modify the process (for

example by holding or releasing succeeding jobs). Jobs of one network may release jobs in another network.

DEADLINE SCHEDULING.

Using deadline scheduling, user jobs can be submitted to run to be eligible for scheduling by a certain time on a certain day.

A job submitted for deadling scheduling is not guaranteed to run by the deadline, but JES3 will attempt to schedule and execute the job on time. The job can run as a one-time run or it can be periodically scheduled, such as weekly or monthly. To use this technique, the installation must define deadline types at JES3 initialization. For a given type, the user defines:

1. initial job priority (how critical the jobs of this type are)
2. the aging rate to be applied (how much and how often the job's priority should be increased as the deadline approaches)
3. the lead time (when to activate deadline scheduling for the job)

Up to 36 deadline types can be defined. When the job is first submitted, it must contain a `//*MAIN` JES3 control card specifying:

1. the deadline type
2. the required clock completion time
3. whether it is to be completed once or run periodically on a specified cycle.

As time elapses, JES3 automatically schedules the job, upgrading the priority as required. The operator is notified of the status of a deadlined job:

1. when it is activated under a deadline algorithm
2. if it is past its deadline
3. when it has been completed (purged)

NON-STANDARD JOB PROCESSING.

Non-standard job processing is used when the processing which is required to be done is of such a nature that it does not require all of the job processing which is done by a standard job, it does not require the standard sequence of processing, or it requires special services not included within the standard job. A good description of the latter is an installation written DSP to perform a function which is a unique installation related function such as plotting.

DYNAMIC SYSTEM INTERCHANGE.

In a JES3 environment the global processor assumes an enormous amount of responsibility. With this in mind, let us look at another JES3 facility, Dynamic System Interchange (DSI). Suppose that the global processor suffers a hard red-light failure and will be down for several hours. What is the impact?

On a properly configured local processor, the DSI facility can dynamically assume the role of the global. DSI is invoked by going to the healthy local processor and telling it that it is now the global. The new global will, in effect, do a hot start. It will read the job queue and checkpoint records from the spool and rebuild key control blocks in order to initialize itself as a global. Jobs in execution on the local processors may be unaffected, may be temporarily suspended, or may have to be rerun. Output being printed may be restarted at the last checkpoint. The jobs on the old global may be unaffected by the DSI if the old global is reinitialized as a local processor without an intervening IPL.

"Properly configured" means that a local, to be eligible to become a global, must:

1. have a CTC path to all other local processors
2. be switched to all key input, output, console, and RJP devices that were on the old global.

DSI may be invoked on a scheduled basis if you want to take down the global processor for scheduled hardware changes or maintenance.

USER INTERFACE

How does an installation tailor the JES3 system to its needs? What facilities are available to users submitting jobs? Two main areas will be addressed:

1. what is available to the system, and
2. what is available to the user.

INITIALIZATION

The installation may choose the options and features that are desired through the JES3 initialization parameters in the JES3 initialization deck. The deck itself can be read in through a card reader, or it can be located on a direct access device.

JES3 initialization occurs after MVS is initialized. Once JES3 initialization is complete, you have defined an execution environment in which your user jobs can be run. The scope of the initialization can be divided into three areas:

1. the hardware environment:
 - the processors (global and local mains)
 - the JES3 spool devices
 - the JES3 managed devices
 - global (readers, punches, printers, punches, consoles)
 - locals (tape, DASD, and MSS)
2. job related facilities, features, and options for each stage of job processing
3. other processing facilities. (NJE and RJP)

During system operation, many of these initialization parameters and facilities can be dynamically started, stopped, displayed, and modified.

USER EXITS

A second means of tailoring the JES3 system is through user exits. JES3 supports 30+ user exits in the categories shown below:

Initialization	1
Input Service	7
Interpreter	12
Setup	2
Output Service	5
Miscellaneous	
"Call" command	1
TSO	2
Consoles	3
Networking	7

These are, in effect, hooks into key areas of the system which can go to your routines. These exits give opportunities to:

1. check authority
2. do job accounting
3. monitor activity
4. handle exceptional cases

A macro IATXCUE provides the ability to insert reentrant user exit linkage into JES3 mainline code using only a single source statement. The usage of this macro greatly decreases the likelihood of conflict between user modifications and IBM changes to the code.

USER WRITTEN DYNAMIC SUPPORT PROGRAM (DSP)

An additional way of modifying the JES3 system is to write your own Dynamic Support Programs. Input service, setup, and output writers are a few examples of JES3 standard DSPs. DSPs can be invoked by name, using the CALL command. They can also be related to stages of a job's processing, perhaps part of non-standard job processing.

A large number of macro instructions are available to make the usage of JES3 resources much easier. Programming conventions have been established and documented for the JES3 environment. A majority of these conventions are maintained implicitly by the use of the macros provided with JES3.

JES3 CONTROL STATEMENTS

Users submitting jobs talk to JES3 through MVS JCL (which is fully supported) and through JES3 control cards. JES3 control cards are identified by "//*" followed by the card type in the next position. They are usually inserted after the JOB statement. Users include control cards to specify:

1. specialized control of a job
2. overriding of installation defaults
3. the use of optional facilities

All JES3 control cards are optional. The control card types and their uses are:

1. /*MAIN processing options
2. /*FORMAT output options
3. /*DATASET special input stream data set
4. /*PROCESS non-standard job
5. /*NET dependent job control
6. /*OPERATOR operator message
7. /*PAUSE halt input reader
8. /*command command execution
9. /*NETACCT network accounting data
10. /*ROUTE XEQ route networking job

CONCLUSION

JES3 offers a wealth of function and a great deal of flexibility in how you choose to use that function. You can elect to use features like dependent job control, deadline scheduling, or user exits, or you may choose not to use them.

JES3 was designed and implemented from the ground up as a loosely coupled multiprocessing system with a single system image to the processors in the complex. Through its global functions, JES3 offers a central management of resources to the system programmer and offers central control of the complex to the operator.

If you have any questions about JES3, please do not hesitate to contact a member of the JES3 project or your local IBM representative for more information.

Publications.

- GC23-0039 - Introduction to JES3
- GC23-0040 - JES3 Overview
- GC23-0041 - JES3 SPL: Installation, Planning, and Tuning
- GC23-0042 - JES3 SPL: User Modifications and Macros
- GC23-0043 - JES3 SPL: Diagnosis
- GC23-0044 - JES3 Messages
- GC23-0045 - JES3 Operator's Library
- SX23-0007 - JES3 OL: Reference Summary
- LY24-6005 - JES3 Logic
- GC28-0692 - OS/VS2 MVS JCL