

**Program Product**

**JES3 Overview**

**IBM**

File No. S370-20  
Order No. SC23-0040-0

## **Program Product**

## **JES3 Overview**

Program Number 5740-XYN



### **First Edition (December 1980)**

This edition with Technical Newsletter SN25-0199 applies to the JES3 component of MVS/System Product-JES3, Release 3 and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. For prerequisites and corequisites, see *JES3 Introduction*, GC23-0039.

Changes are continually made to the information herein; before using this publication, consult the IBM System/370 Bibliography, GC20-0001, and the technical newsletters that amend the bibliography, to learn which editions and technical newsletters are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Publications are not stocked at the address given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Publications Development, Department D58, Building 706-2, PO Box 390, Poughkeepsie, New York 12602. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

## Preface

This book will introduce you to the terminology, organization, functions, and features of Job Entry Subsystem 3 (JES3). You need not be familiar with JES3 to read this book, but you should have an introductory level background in the Multiple Virtual Storage (MVS) System. This book contains five chapters, an appendix, and a glossary:

- **Chapter 1. Introduction** consists largely of definitions of concepts and terms. Here, a framework is established for an understanding of the remaining chapters.
- **Chapter 2. JES3 Job Flow** uses a sample job to relate and put into perspective the concepts introduced in Chapter 1.
- **Chapter 3. The Subsystem Interface** shows how MVS communicates with JES3, via the subsystem interface.
- **Chapter 4. Spool Data Management** describes simultaneous peripheral operations online (spool) concepts.
- **Chapter 5. Job Related Control Blocks** describes individually the control blocks introduced in earlier chapters, as well as other control blocks not defined earlier.
- **Appendix A. Additional Facilities** describes key JES3 facilities not defined in earlier chapters. These include deadline scheduling, dependent job control, dynamic allocation, JES3 networking, recovery management, remote job processing, and spool partitions.

The glossary follows Appendix A. Included are definitions of terms used within this publication.

To get the most out of this book, you should first read *JES3 Introduction*, GC23-0039. You will then have a foundation for reading and using the other books that contain information about JES3:

- *JES3 Operator's Library*, SC23-0045
- *JES3 Operator's Library: Reference Summary*, SX23-0007
- *JES3 System Programming Library: Installation Planning and Tuning*, SC23-0041
- *JES3 System Programming Library: User Modifications and Macros*, SC23-0042
- *JES3 System Programming Library: Diagnosis*, SC23-0043
- *JES3 Messages*, GC23-0044
- *JES3 Logic*, LY24-6005
- *OS/VS2 MVS JCL*, GC28-0692



## Contents

<b>Chapter 1. Introduction</b> .....	<b>1-1</b>
The JES3 Environment .....	1-2
Job Processing Concepts .....	1-2
Types of Jobs .....	1-6
<b>Chapter 2. JES3 Job Flow</b> .....	<b>2-1</b>
Construction of a Reader Job .....	2-1
Scheduling of a Reader Job .....	2-1
Dispatching of a Reader Job .....	2-3
Execution of a Reader Job .....	2-3
The Input Service Job .....	2-4
Standard Job Processing .....	2-5
Converter Interpreter Processing .....	2-5
Main Service Processing .....	2-7
Output Service Processing .....	2-10
Purge Processing .....	2-14.1
Chapter 2 Review .....	2-15
<b>Chapter 3. The Subsystem Interface</b> .....	<b>3-1</b>
Types of Communication .....	3-3
Communication Overview .....	3-4
The Request Routine .....	3-5
The Function Routines .....	3-5
The Common Services Router .....	3-5
The Common Services Poster .....	3-6
Chapter 3 Review .....	3-7
<b>Chapter 4. Spool Data Management</b> .....	<b>4-1</b>
JSAM Components .....	4-1
USAM Components .....	4-2
Components Common to JSAM and USAM .....	4-3
Buffer Pool Management .....	4-3
Data Flow for USAM Requests .....	4-4
SYSIN Processing .....	4-5
SYSOUT Processing .....	4-5
Chapter 4 Review .....	4-6
<b>Chapter 5. Job Related Control Blocks</b> .....	<b>5-1</b>
Spool Volume Areas .....	5-1
Single Track Table Space .....	5-1
Job Spool Space .....	5-1
The Job Control Table Data Set .....	5-1
Job Definition and Scheduling Control Blocks .....	5-2
Job Processing Control Blocks .....	5-3
Job Description and Accounting Block .....	5-4
Job Data Sets Block .....	5-4
Job Summary Table .....	5-5
Job Track Allocation Table .....	5-5
Job Management Record .....	5-5
Output Service Element .....	5-5
Format Parameter Buffer .....	5-5
User Parameter Buffer .....	5-5
Job Volume Table .....	5-6
<b>Appendix A. Additional Facilities</b> .....	<b>A-1</b>
Deadline Scheduling .....	A-1
Dependent Job Control .....	A-2
Dynamic Allocation .....	A-4
JES3 Networking .....	A-4
Recovery Management .....	A-5
Recovery for Program Failures .....	A-5
Recovery for Global Processor Failures .....	A-6
Recovery for Spool Failures .....	A-7
Remote Job Processing .....	A-7

Spool Partitions ..... A-8

Glossary ..... GL-1

Index ..... I-1

## Figures

Figure 1. MVS Jobs and JES3 Jobs .....	1-3
Figure 2. Job Control Table Entry .....	1-3
Figure 3. Job Scheduling .....	1-4
Figure 4. Roles of Job Segment Scheduler, Multifunction monitor in Dispatching	1-6
Figure 5. JES3 Job Flow .....	2-2
Figure 6. The MVS Environment .....	3-1
Figure 7. Address Space Communication .....	3-2
Figure 8. SSI Function Codes .....	3-3
Figure 9. SSI-Related Control Blocks .....	3-4
Figure 10. Spool Data Management Functional Components .....	4-2
Figure 11. Storage Map of SDM Components .....	4-4
Figure 12. Job Queue Element Pointers .....	5-3
Figure 13. Job Processing Control Blocks .....	5-4
Figure 14. Dependent Job Control Network .....	A-3





## Summary of Amendments

### Summary of Amendments for SC23-0040-0 as updated by TNL SN25-0199

This TNL includes minor technical corrections, minor editorial corrections, and the following major additions:

- The topic "Converter Interpreter Processing" in Chapter 2 discusses installation imposed DD limits.
- The topic "Output Service Processing" in Chapter 2 discusses the writer output multitasking facility.



## Chapter 1. Introduction

The designers of IBM Operating System/Virtual Storage with Multiple Virtual Storage (MVS) made a clear distinction among the work of readying jobs for execution, that of supervising the execution of jobs, and the work of removing job output from the system. They delegated the readying of jobs and removal of job output to a component of MVS called a **job entry subsystem**, and the supervision of job execution to MVS.

Today, two kinds of job entry subsystems are available for use with MVS: job entry subsystem 2 (JES2), and job entry subsystem 3 (JES3). A fundamental difference between the two subsystems is the manner in which they manage multiple processor complexes. The JES2 concept is one of decentralized control and a common job queue -- each JES2 processor operates independently of the others, but each may add jobs to or select jobs from the common job queue. JES3, in contrast, exercises centralized control over all processors. JES3 also uses a common job queue, but all jobs reach the queue only through one central JES3 component that selects and distributes work to the processors.

There are several areas where benefits are realized as a direct consequence of a separate-subsystem design.

One of these areas is **resource management**. When all jobs, all input required for the jobs, and all output produced by the jobs enters or leaves the system via a single point, the subsystem, the allocation of devices, volumes, and data sets can be coordinated at that central point. With centralized resource control, there is opportunity for fuller resource utilization.

Another area where JES3 builds upon the benefits inherent in a subsystem concept is **workflow management**. The entry of all jobs through a central point, JES3, means that control of the actions needed to prepare jobs for execution can be centralized. The distribution and duplication of job management function becomes unnecessary, and an awareness of the status of all jobs entering or leaving the system can be easily maintained. This awareness is particularly useful in recovery situations, where the scope of recovery is largely a function of the quality of the tracking performed prior to the failure.

Some of the benefits of improved resource utilization and centralized job management can be translated into a benefit of **operator control**. With all system resources known to JES3 and with one job management mechanism, it is a relatively simple step to provide for control over the entire system. And yet, the need for operator control can be minimized because the awareness of job mix and resource availability results in increased coordination by JES3 and in less need for operator intervention and decision-making.

At first glance, the design requirements for a subsystem like JES3 can suggest extreme complexity. A study of the basic mechanisms JES3 uses to perform its work will reveal an organization and distribution of function that is not complex, and the knowledge should become a basis for full use of JES3 features. This is the purpose of the remaining sections of this manual -- to introduce you to the concepts you should understand to use JES3 effectively.

## The JES3 Environment

JES3 installations can be small or large. One installation, for example, may have only one processor while another has eight. Also, the number of I/O devices may vary greatly from installation to installation. One installation could have fewer than ten magnetic tape drives while another installation has one hundred or more.

One processor, the **global processor**, always controls a JES3 complex. JES3 functions assigned to the global processor present to the user a single-system image, regardless of how many operating systems are in use on the various processors. The parts of the JES3 program that are executed in the global processor handle operator communication, manage I/O resources, schedule jobs for execution on the same or other processors, and process output created by the jobs during execution.

Besides the global processor, a JES3 complex can have up to seven other processors called **local processors**. The actual execution of jobs can take place on the global processor as well as on any of the local processors. The term **main**, is used to identify those processors on which jobs can execute. Thus, there can be a global main processor and local main processors. Always associate the execution of jobs with main processors, because global and local are terms used to indicate the distribution of JES3 functions among the processors.

Local processors are attached to the global processor by means of channel-to-channel adapters that serve as communication paths. When the global processor is used for job execution (as a main), communication takes place between address spaces rather than between processors, and the service request facility of MVS is used to accomplish the communication.

## Job Processing Concepts

You are probably acquainted with the term "job." This word has traditionally defined a set of work signified by the presence of a JOB statement in an input stream. Jobs in the MVS context can consist of one or more job steps, signified by EXEC statements in an input stream. JES3 does not change these definitions, but does add to the list of items called jobs.

With JES3, the division of work is made with a **job** being the basic unit of input, but there are special kinds of jobs where no JCL is involved. This is because the term job in the JES3 context means a set of work to be performed to prepare jobs found in the input stream for execution. That set of work can be related to a specific job from the input stream, or (and this is important) it can be related to work JES3 must perform in its role of coordinating the preparation of many jobs for MVS execution. Whatever the reason for the existence of a JES3 job, the JES3 job is defined by means of a control block called a **job control table (JCT) entry**. An analogy can be made between an MVS job being defined by a JOB statement and a JES3 job being defined by a JCT entry (see Figure 1).

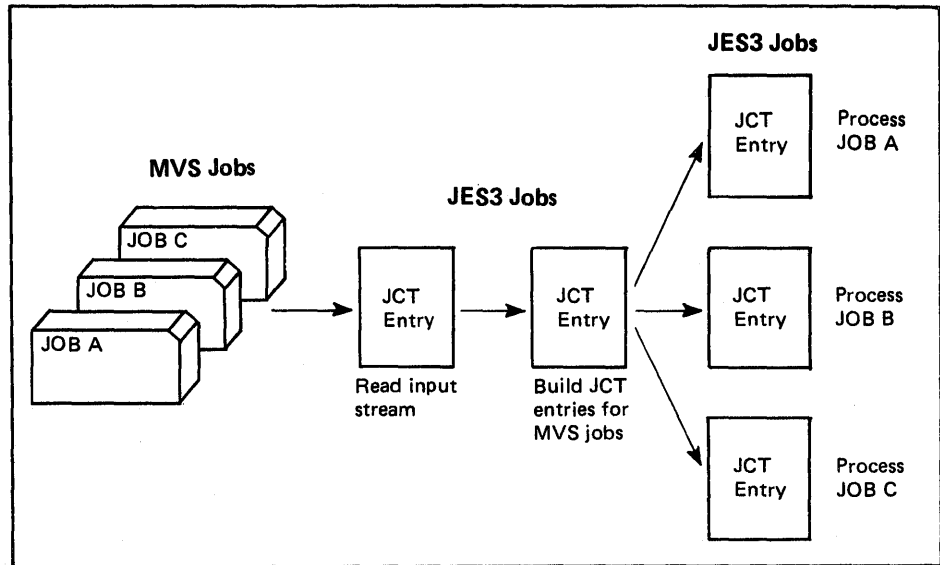


Figure 1. MVS Jobs and JES3 Jobs

JES3 stages its services on behalf of jobs by using **scheduler elements**, which represent units of work JES3 must perform to process the jobs. Each JCT entry contains the definition of a job and one scheduler element for each piece of work that must be done to process the job. For a typical job originating in the input stream (see Figure 2) there would be:

- A scheduler element for converting the job's JCL to a form usable by MVS
- A scheduler element for acquiring I/O resources needed by the job and for passing the job to MVS
- A scheduler element for handling the job's SYSOUT data
- A scheduler element for removing the job from the system

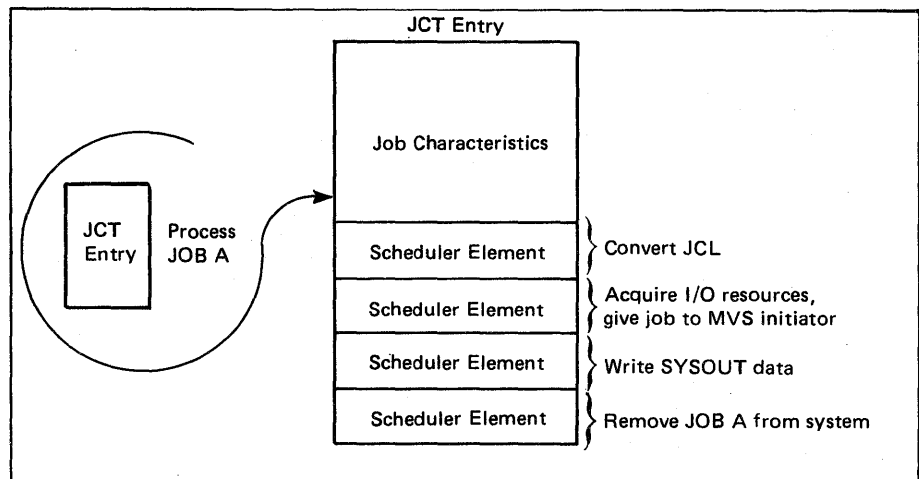


Figure 2. Job Control Table Entry

For the special kind of work mentioned earlier which JES performs on behalf of all jobs, there would be:

- A scheduler element for reading the input stream to obtain JCL statements
- A scheduler element for building JCT entries for the jobs in the input stream

Of course, there is other work JES3 must perform and therefore other types of scheduler elements do exist. You will see more information about specific kinds of scheduler elements in later sections.

Now it is very important to further discussion that you remember the next statement: JES3 does not really schedule jobs (in the MVS sense of scheduling jobs); it schedules the work defined by scheduler elements.

The JES3 programs that perform the work defined by scheduler elements are called **dynamic support programs (DSPs)**. So, when JES3 creates a scheduler element, the eventual result is execution of one or more DSPs.

The scheduling of work to be performed by DSPs is handled by a component of JES3 called the **job segment scheduler**. It is the job segment scheduler which selects scheduler elements that are ready for processing and builds entries in a master JES3 dispatching queue called the **function control table (FCT)** (see Figure 3).

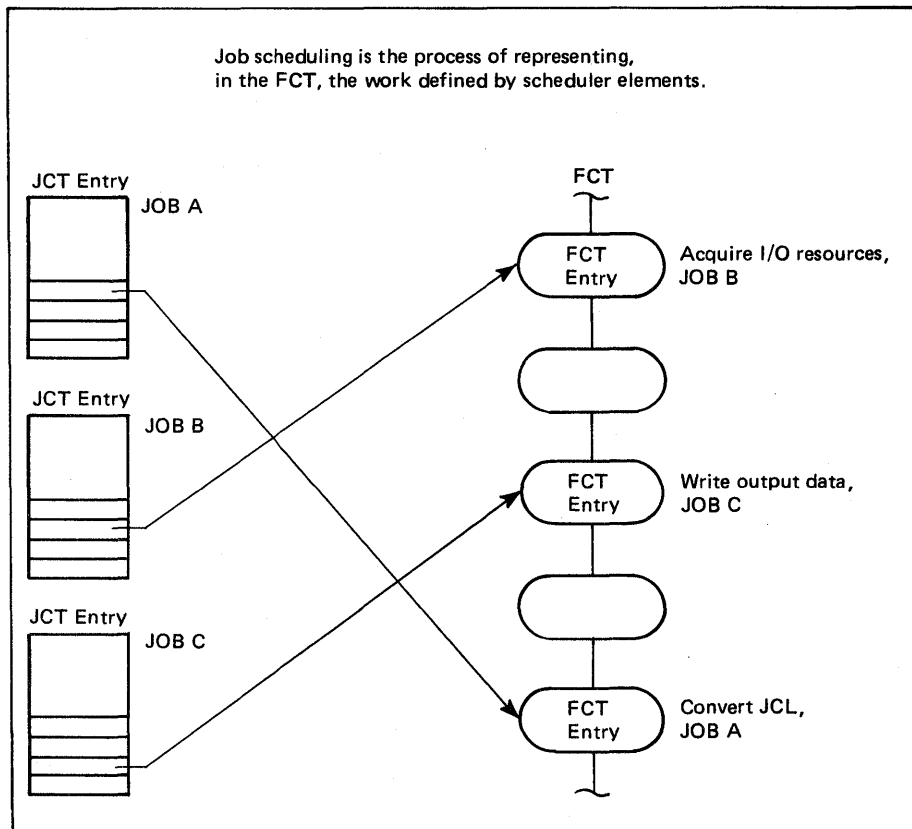


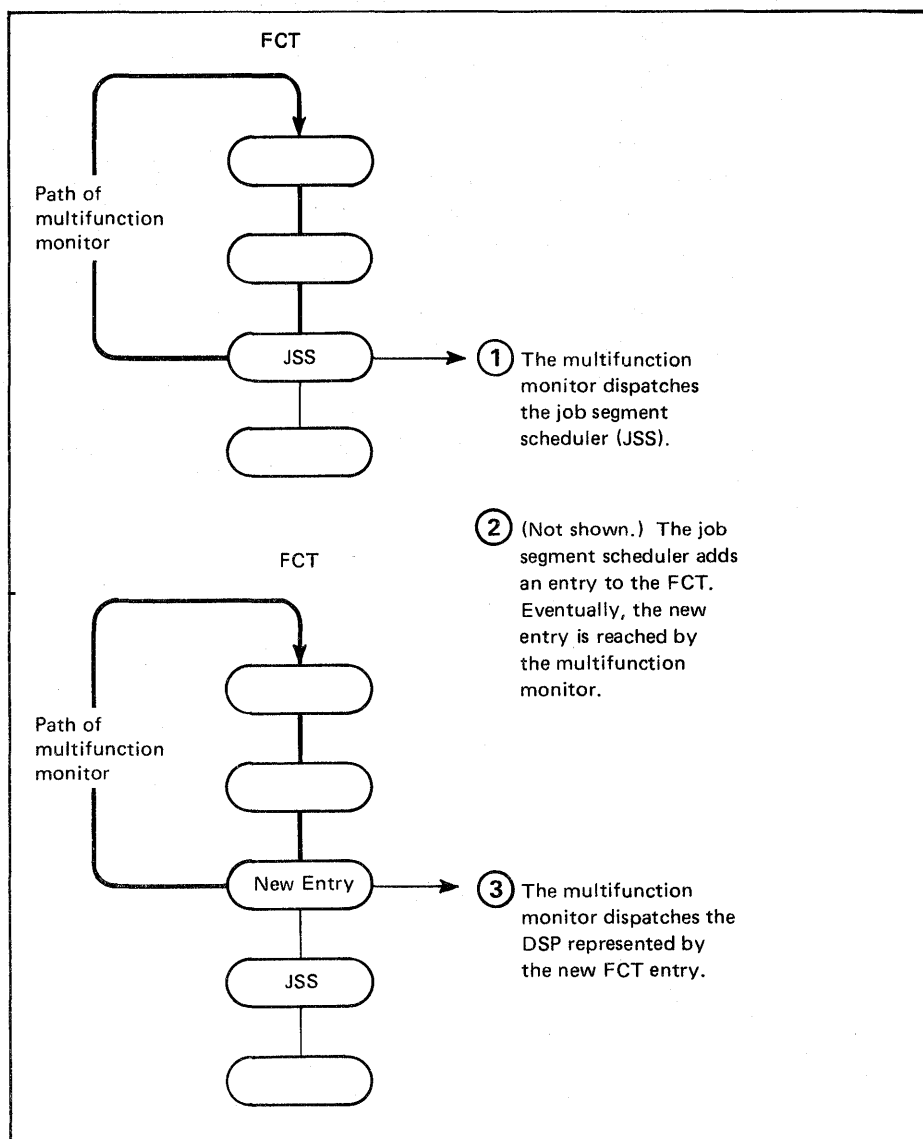
Figure 3. Job Scheduling

Thus, FCT entries represent JES3's dispatchable units of work in an overall sense (for all jobs), while scheduler elements in JCT entries represent units of work for specific jobs. The JCT - FCT structures allow JES3 to track and schedule work on a job-by-job basis (JCT entries), and to dispatch work on a total job mix basis (FCT entries).

The JES3 dispatcher is called the **multifunction monitor**. The job segment scheduler adds entries to the FCT in priority order (the priority of the DSPs to be processed). The multifunction monitor repeatedly scans the FCT searching for a DSP eligible for dispatching. Each scan begins at the top, highest-priority FCT entry, continues until a DSP is dispatched, and resumes again at the highest-priority FCT entry.

A question you might ask now is: "How is the job segment scheduler itself scheduled for execution?" It turns out that not all FCT entries are added to the FCT as the need arises. There are some permanent entries in the FCT, and the job segment scheduler is represented by such a permanent entry, at a relatively low priority. Once the multifunction monitor has dispatched all eligible higher-priority DSPs, it can dispatch the job segment scheduler, which could then add more entries to the FCT (see Figure 4).





**Figure 4. Roles of Job Segment Scheduler, Multifunction monitor in Dispatching**

You will see more about each of the concepts introduced thus far in later discussions. You should now be familiar with the following terms:

- Global and local
- Job and job control table (JCT)
- Scheduler element
- Dynamic support program (DSP)
- Job segment scheduler
- Function control table (FCT)
- Multifunction monitor

## Types of Jobs

There are several types of jobs in the JES3 context. One term you will see often is **standard job**. This is a type of job for which a default set of

scheduler elements is used. That is, JES3 places into the JCT entries of all standard jobs the same scheduler elements, in the same order. Every job containing JCL, but containing no special JES3 control statements, is considered a standard job. In its distributed form, JES3 uses four scheduler elements for standard jobs:

1. Converter interpreter (CI)
2. Main service (MAIN)
3. Output service (OUTSERV)
4. Purge (PURGE)

The names in parentheses are formal names of the DSPs that are executed to perform the work. You can change the number or type of scheduler elements to be used for standard jobs by means of a user exit routine, but such a change would apply to all standard jobs. To specify a different set of scheduler elements on a job-by-job basis, you can code special **JES3 control statements** and include them with JCL statements for the jobs. Jobs for which you do this are called **non-standard jobs**.

There are three more types of jobs, the classification of which is related to the sources of the jobs.

**Normal jobs**, which may be either standard or non-standard, are those entered via:

- Locally attached devices such as card readers, magnetic tape devices, and direct access storage devices
- Remotely attached devices (devices for which the JES3 remote job processing facility is being used)
- The MVS internal reader (from which job streams contained in SYSOUT data sets are received)
- Another global JES3 (in a networking environment)

Most of the work to be done in typical JES3 installations consists of normal jobs.

Another source of jobs is the operator. Those jobs created by operator request are classified as **called jobs**, since the operator uses JES3 CALL commands to make the requests. Called jobs are unique in that they are not defined by JCL; there is, in fact, no JCL involved at all. These jobs are internally generated by JES3 in response to the CALL command, and their JCT entries always contain only two scheduler elements - one to represent the DSP needed for the request, and one to remove the called job from the system.

The final source of jobs is MVS. When an operator issues an MVS START or MOUNT command, or a TSO user enters a LOGON command, MVS builds in-storage JCL and then passes the JCL to JES3 for processing. These MVS-created jobs are called **demand select jobs**.



## Chapter 2. JES3 Job Flow

This chapter describes the sequence of events that takes place during JES3 processing of a job. A sample job is used as a basis for the discussion. Assume that the sample job is a normal job, that it is contained in a card hopper, that the card reader device has not yet been started by the operator, and that JES3 has been started and is processing jobs.

First, the operator must issue a **CALL command** to signal JES3 that an input stream is to be read from the card reader device. This begins a chain of events that includes the creation and scheduling of a **card reader job**, reading of the input stream by a card reader DSP, the building of JCT entries for each job in the input stream, and the execution of DSPs represented by scheduler elements in the JCT entries for each job.

The CALL command that is issued by the operator is read by a DSP named CONSOLES, which is responsible for traffic management on operator consoles. The CONSOLES DSP is represented by a permanent FCT entry, so it will be dispatched automatically (if there is work to do) when the multifunction monitor reaches the entry. The CONSOLES DSP simply routes the CALL command to another DSP, the work-to-do driver (WTDDRVR). Like the CONSOLES DSP, the WTDDRVR DSP is represented by a permanent entry in the FCT. It is the WTDDRVR DSP which actually begins the chain of events mentioned earlier.

### Construction of a Reader Job

The construction of a job, particularly a called job, is a relatively simple procedure. To construct a reader job, the WTDDRVR DSP:

- “Fills in the blanks” of a JCT entry with information about the job.
- Places two scheduler elements into the JCT entry: one to represent the card reader DSP (CR), and one to represent the purge DSP (PURGE).
- Adds the completed JCT entry to the JCT, at priority level 15. (The JCT is managed by job priority levels. That is, the JCT consists of 16 subqueues, one for each of the priority levels 0 - 15. All called jobs are queued at priority 15 to ensure prompt action for operator commands.)

After building a JCT entry for the reader job, the WTDDRVR DSP sets flags to signal the multifunction monitor that the job segment scheduler has work to do and can be dispatched. One of the flags will indicate to the job segment scheduler which of the priority levels in the JCT it is to examine. (Thus, the entire JCT need not be examined every time the job segment scheduler is dispatched.)

### Scheduling of a Reader Job

The term **scheduling**, as it applies to JES3, means the placing of an FCT entry in the FCT. But, as mentioned earlier, the work to be scheduled is defined by scheduler elements which are contained within JCT entries for individual jobs. So the process of scheduling consists of a search of JCT entries to locate scheduler elements for unstarted work, and the building of corresponding FCT entries.

Thus far, the status of the sample job is that the input stream containing the job has not yet been read, but a special reader job has been created by JES3. The JCT entry for that reader job contains a scheduler element

representing the CR DSP, which will read the input stream. Also, flags have been set to make the job segment scheduler dispatchable.

When the job segment scheduler is dispatched by the multifunction monitor, it locates the JCT entry for the reader job at priority level 15 and attempts to schedule the first scheduler element (see Figure 5, Part 1). (There are reasons why the job segment scheduler would not be able to do this, but assume for now that the first scheduler element, CR, can be scheduled.)

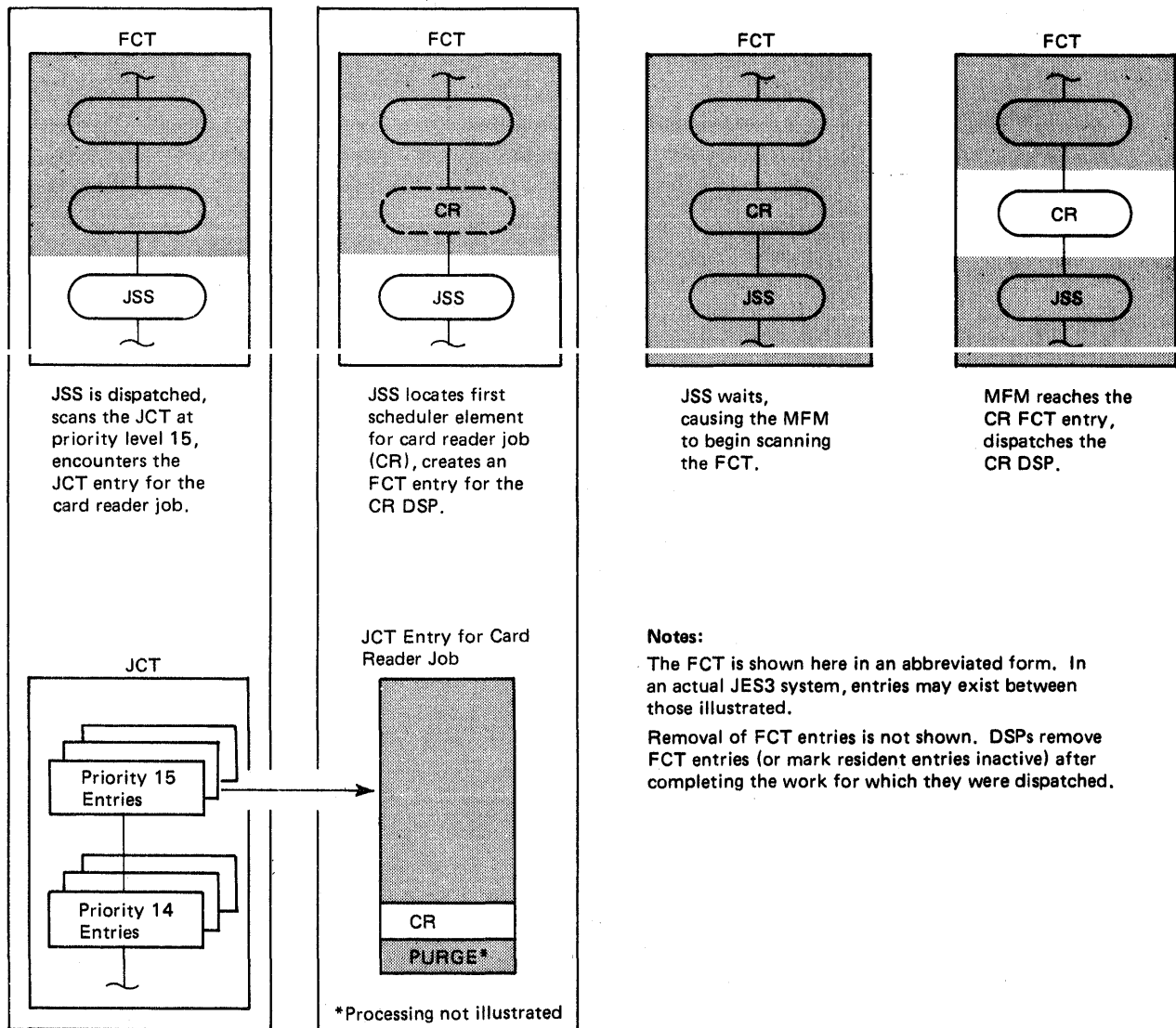


Figure 5 (Part 1 of 5). JES3 Job Flow

The job segment scheduler creates an FCT entry to represent the CR scheduler element (or, more precisely, the CR DSP) and places the entry on the FCT. The job segment scheduler has now completed its scheduling activities, so it causes entry to the multifunction monitor, which begins a scan of the FCT.

## Dispatching of a Reader Job

The process of **dispatching**, as it applies to JES3, consists of a search of the FCT to locate an entry representing a DSP eligible for execution, retrieval

of the load modules for the DSP, and transfer of control to the DSP. Dispatching is performed by the multifunction monitor.

The status of the sample job is that the input stream containing the job still has not been read. However, an FCT entry for the CR DSP, which is responsible for reading the input stream, does exist; and the multifunction monitor has just begun a scan of the FCT.

Because the FCT could contain entries for higher-priority eligible DSPs, some time could elapse before the multifunction monitor reaches the CR DSP. This is because the multifunction monitor always resumes a scan at the top of the FCT rather than at the entry following the last-dispatched entry.

When the multifunction monitor does reach the FCT entry for the CR DSP, it simply retrieves the load modules for the CR DSP and branches to the primary module, in this case (and most others), called a **driver**. Because module loading involves I/O, it is possible that other DSPs will be dispatched by the multifunction monitor during the loading process.

## **Execution of a Reader Job**

The execution of a job (in the JES3 sense of the word job) is the execution of the DSPs represented by scheduler elements contained in the JCT entry for the job. The JCT entry for the reader job has two scheduler elements, CR and PURGE. Therefore, execution of the card reader job means execution of the CR and PURGE DSPs.

The input stream containing the sample job is still in the hopper of a card reader, but modules of the CR DSP have been loaded into storage and execution of the first CR module can begin.

The first thing that the CR DSP does is to retrieve the parameters entered by the operator in the CALL command. Assume that a B=3 parameter was included with the CALL command. This means that the operator wanted the jobs in the input stream to be separated into batches of three jobs each (the last batch might contain fewer than three jobs, of course). With batches, reading of the input stream and transfer of the jobs to a direct access device can take place faster because of the reduction in I/O activity in comparison to that needed for reading on a job-by-job basis.

The CR DSP next reads the input stream. With a batch size of three, it stops after every three jobs and transfers the jobs to a data set on a direct access device. Then, for each batch, the CR DSP constructs a JCT entry and queues the entry at priority level 15 in the JCT. Thus, the CR DSP creates a JES3 job for each of the batches of jobs it obtains from the input stream (see Figure 5, Part 2). Into each JCT entry, the CR DSP places two scheduler elements: one for the input service driver (ISDRVR) DSP, and one for the PURGE DSP.

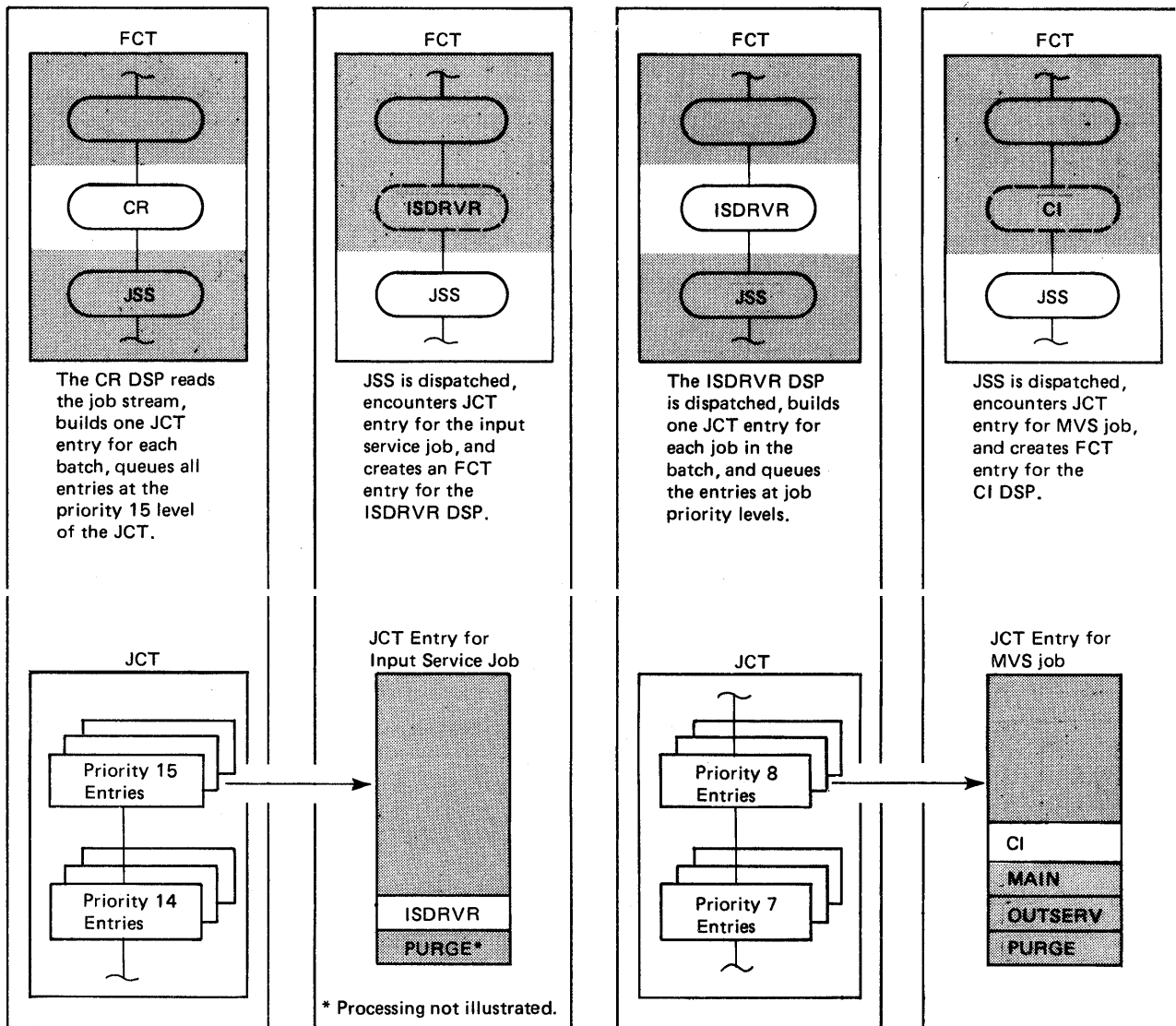


Figure 5 (Part 2 of 5). JES3 Job Flow

The CR DSP continues to construct one JCT entry for each batch until the input stream is exhausted. When that happens, the CR DSP causes entry to the multifunction monitor, which will dispatch the job segment scheduler. The job segment scheduler scans the JCT, discovers that the PURGE scheduler element for the card reader job has not yet been scheduled, and builds an FCT entry for the PURGE DSP. Execution of the PURGE DSP results in removal of the card reader job from the system.

## The Input Service Job

The jobs defined by the JCT entries for batches are called **input service jobs**. Construction, scheduling, and dispatching of an input service job is similar to that of the card reader job. While the purpose of the card reader job was construction of JCT entries for batches of jobs, the purpose of the input service job is construction of JCT entries for individual MVS jobs within each batch.

The scheduler elements contained in the JCT entries for input service jobs represent the input service driver (ISDRVR) and PURGE DSPs. Thus, the

first FCT entry constructed by the job segment scheduler for the input service job will be one to represent the ISDRVR DSP. When the ISDRVR DSP is dispatched by the multifunction monitor, it:

- Retrieves a batch of jobs
- Examines JCL for the individual jobs, looking for JES3 control statements (these could influence the construction of JCT entries for non-standard jobs)
- Constructs a JCT entry for each job in each batch

Information placed into the JCT entry for a job includes:

- A JES3 job number
- The jobname from the JOB statement
- The job priority and class
- A symbolic origin for the job (for returning job output to the origin)
- Information from JES3 control statements
- Scheduler elements representing the DSPs needed to process the job

Assume that the sample job is a standard job, so the scheduler elements would be:

1. Converter interpreter (CI)
2. Main service (MAIN)
3. Output service (OUTSERV)
4. Purge (PURGE)

The ISDRVR DSP adds the completed JCT entries to the JCT at the **job priority levels**. At batch exhaustion, control returns to the job segment scheduler, which will schedule execution of the PURGE DSP for the input service job. The PURGE DSP will remove the input service job from the system.

Remember that there probably would have been several input service jobs active at the same time, because of a number of jobs in the input stream. If the input stream consisted of 10 jobs, and the batch size was 3, then there would be four batches (3 jobs, 3 jobs, 3 jobs, 1 job), and 4 input service jobs would have been created.

## **Standard Job Processing**

As a result of the processing of input service jobs, there is at least one normal job in the job queue. We have assumed the normal job is also a standard job, so the JCT entry contains scheduler elements for converter interpreter (CI), main service (MAIN), output service (OUTSERV), and purge (PURGE) DSPs. The job segment scheduler, dispatched when an entry is added to the JCT, always selects a job by priority and attempts to schedule a DSP, represented either by the first scheduler element or the first one not yet marked complete. For the sample job, the first DSP to be scheduled is the CI DSP.



## ***Converter Interpreter Processing***

A CI DSP exists for one main reason: to gather information concerning the job's device, volume, and data set requirements prior to MVS execution of the job. This information is necessary for preexecution setup, one of the major facilities of JES3. The information is taken from DD statements in the job's JCL.

During JES3 initialization, MVS attaches an installation-defined number of converter/interpreter (C/I) tasks. These C/I tasks are subtasks to the JES3 primary task. When a C/I subtask processes a job, the global processor's **scheduler work area (SWA)** provides temporary storage for the job's JCL statements. These statements remain in the SWA until the C/I subtask finishes processing the job. When several C/I subtasks run concurrently, the SWA contains the JCL statements for all the jobs these subtasks are processing.

The amount of SWA space available for JCL storage is limited, however. To ensure that a C/I subtask does not process a job when there is insufficient space in the SWA for that job's JCL statements, the installation can impose limits on:

- The total number of DD statements each job may contain
- The total number of DD statements all C/I subtasks can process concurrently

The CI DSP serves as an interface between JES3 and the MVS converter/interpreter subtasks. The interface responsibilities of the CI DSP are:

- To fail the job, from a JES3 standpoint, if JCL errors are detected by the MVS converter/interpreter.
- To fail any job that contains more DD statements than the limits allow
- To delay processing of any job that would temporarily cause the C/I subtasks to process more DD statements than the limits allow
- To collect and save on a direct access data set called the spool data set the SWA control blocks that are produced by the MVS converter/interpreter. (The MVS converter/interpreter "thinks" it is placing these SWA control blocks into a user address space.)

The CI DSP stores I/O requirements in job-related control blocks called the **job summary table (JST)** and the **job volume table (JVT)**. It places these control blocks into the spool data set along with others for the job. These control blocks and some that are kept in main storage allow JES3 to maintain a complex-wide awareness of the status of volumes and data sets.

When processing by the CI DSP is complete, the job segment scheduler will be dispatched. The job segment scheduler removes the CI FCT entry from the FCT, marks the CI scheduler element in the JCT entry complete, and attempts to schedule the DSP represented by the next scheduler element (MAIN) for the job (see Figure 5, Part 3).

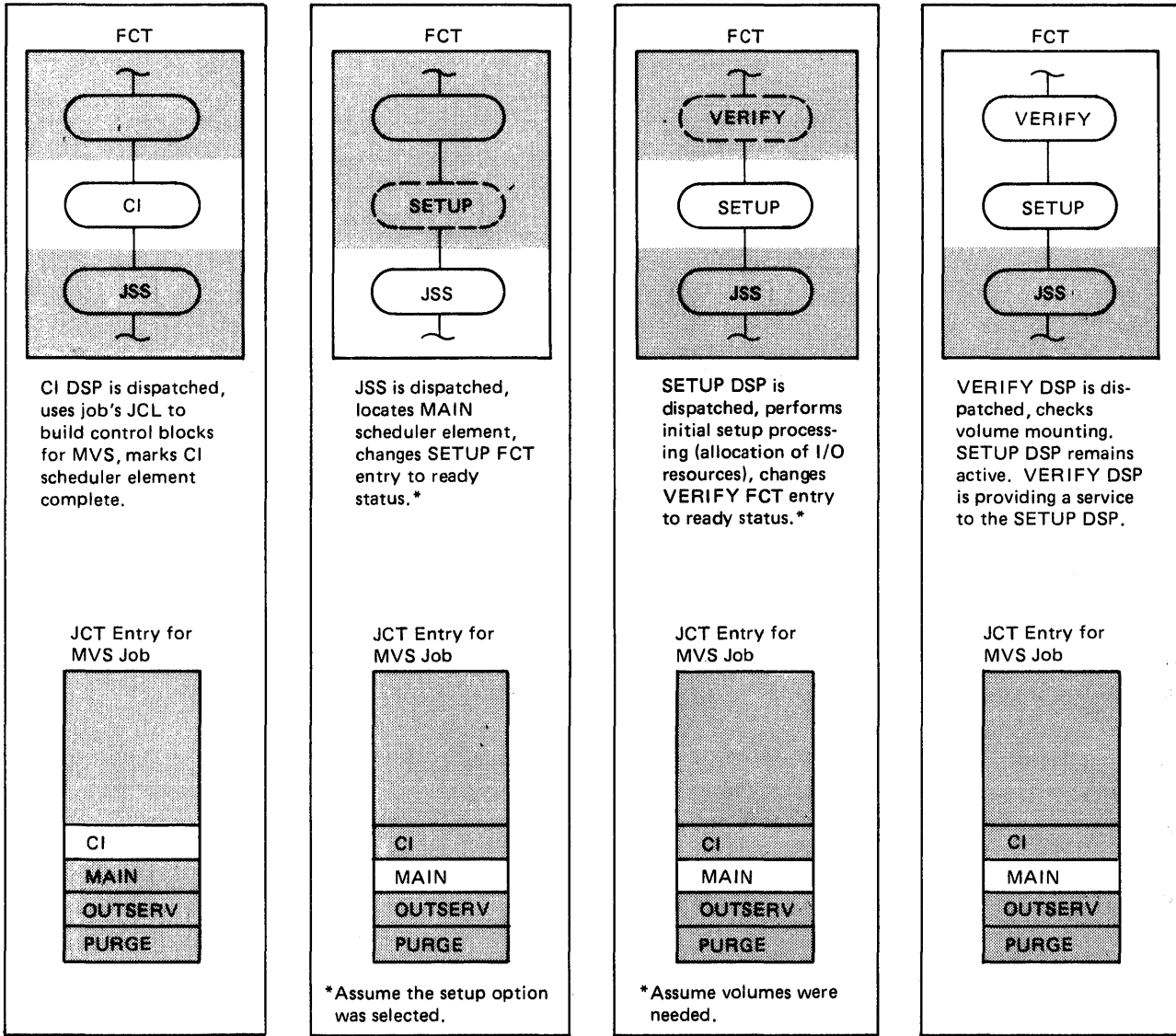


Figure 5 (Part 3 of 5). JES3 Job Flow

### Main Service Processing

Main service processing is the allocation of device, volume, and data set resources to jobs, the selection of jobs to be passed to MVS initiators for execution, and the freeing of allocated resources after the jobs are executed.

For standard jobs such as our sample job, main service processing begins when the job segment scheduler reaches the MAIN scheduler element in the job's JCT entry. The MAIN scheduler element represents two DSPs: setup (SETUP) and **generalized main scheduling (MAIN)**. Both of these DSPs are resident and each has a permanent entry the FCT, so the job segment scheduler need not construct the FCT entries.

The work performed by the SETUP and MAIN DSPs represents the "heart" of JES3 processing -- the reason for the existence of a system like JES3. The goals of the SETUP and MAIN DSPs are effective resource utilization and maximum job throughput. The processing sequence is: initial setup processing to prepare I/O resources, generalized main

scheduling to select and pass a job to an initiator, and then final setup processing to give up I/O resources. Setup processing is an optional JES3 feature.

## Setup Options

An installation may choose whether or not to include the SETUP DSP in its JES3 subsystem. The choice depends upon whether the installation wants one or more of these features:

**Job Setup:** allows the allocation of enough devices for premounting of all the required volumes prior to MVS execution of a job. This saves the time otherwise expended to mount volumes as the job enters its execution. In the JES3 environment, jobs requiring volume mounting are not eligible for MVS processing until the necessary volumes are ready. This option tends to be somewhat extravagant in terms of the number of devices allocated to a single job. For example, with a 3-step job where each step requires 4 different tape data sets, 12 tape devices would be used. The advantage is that little or no operator intervention is necessary between job steps.

**High-Watermark Setup:** an alternative to job setup that is less extravagant in the number of devices allocated to a single job. Where possible, volumes are premounted. High-watermark setup is available for tape, DASD, mass storage system (MSS) virtual units, or all JES3-managed devices. Generally speaking, for each device type JES3 first finds the job step that will use the most volumes. JES3 then allocates to the entire job the number of devices needed to service that step. For example, if steps 1, 2, and 3 of a three-step job will use 6, 4, and 2 tape volumes, respectively, then six tape devices will be allocated to the job.

**Explicit Setup:** combines some advantages of job setup and high-watermark setup. With explicit setup, installations can specify the number of data sets requiring setup and JES3 will determine the number of devices to reserve. Volumes can be kept mounted on devices until they are no longer needed. A certain number of devices are allocated before job execution and the devices are not deallocated until the job completes execution.

## Initial Setup Processing

To review, resources for which the SETUP DSP is responsible are devices, data sets, and volumes. Device types that may be defined are tape, DASD, unit record, graphic, and mass storage system (MSS) virtual units.

Initial setup processing is performed in phases. Assume that one of the options was chosen for our sample job:

During the **volume fetch** phase messages are sent to library consoles to indicate whether or not a volume is to be fetched.

The **allocation** phase may be started automatically after the fetch phase, or manually when the operator specifies that the required volumes have been fetched to the work area. If volumes used are always in the machine room, automatic allocation could be more meaningful because no operator action would be necessary to cause allocation to begin. During the allocation phase, operator messages are issued for premounting of volumes.

During the **verification processing** phase, checking is done to ensure that the proper volumes are being mounted on the proper devices. This work is carried on asynchronously as devices become ready. The SETUP DSP

maintains verify counts relative to individual jobs. When all the necessary mounts for the sample job are accomplished, the job will pass from initial setup processing to generalized main scheduling -- at this point, the job can be passed to MVS for execution.

The SETUP DSP has a built-in early resource release facility. This frequently causes the deallocation of devices, data sets or volumes at the end of the last job step that has a need for the resource.

### Generalized Main Scheduling

To review, the MAIN scheduler element is being processed. The initial phases of setup have been accomplished for the sample job, and after execution the job will reenter setup processing for a final phase. It is now time for processing by the MAIN DSP (see Figure 5, Part 4).

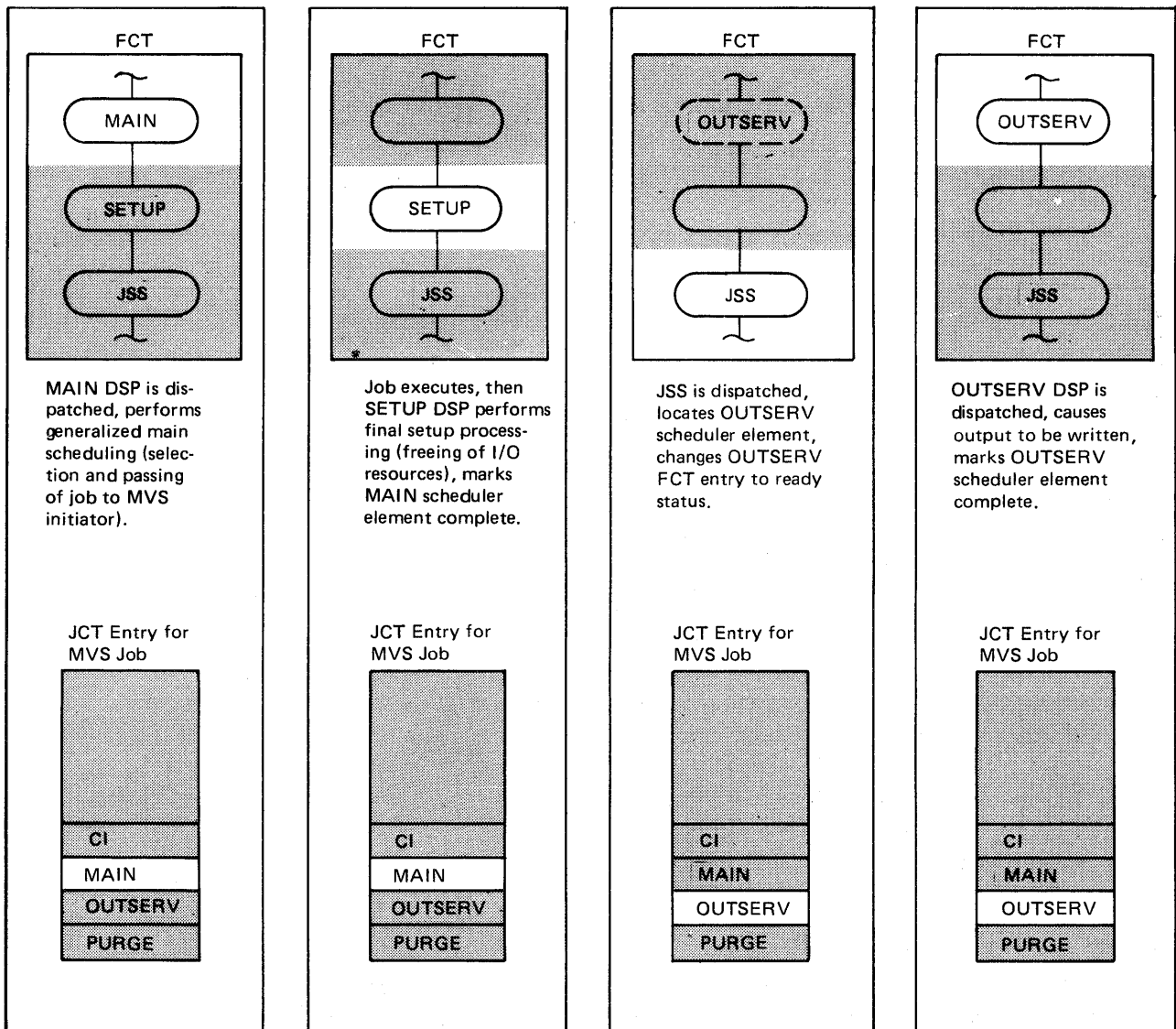


Figure 5 (Part 4 of 5). JES3 Job Flow

The processing of jobs waiting to be selected by MVS is the result of the installation's tailoring of JES3. Triggered by an MVS initiator's request for a job, the MAIN DSP chooses one of several jobs to give to the initiator.

The selection of one job, in preference to another, is influenced by variables such as:

- The type of work being processed during a shift (test vs. production, on-line vs. batch, etc.)
- Eligibility relationships between jobs and processors based upon:
  - Job classes
  - The number of active initiators for the jobs
  - Whether processors are online or offline
  - I/O rates of jobs in execution
  - Virtual storage requirements relative to working set size
- Job priorities, to the extent that the installation wishes to honor priorities

After considering these factors, the MAIN DSP picks our sample job, sends information about the job to the requesting initiator, and indicates that the selected job is now "on main." (It may be valid, due to combinations of eligible job characteristics and scheduling algorithm decisions, for the MAIN DSP to skip a job on a given selection pass.) Having received the job, the initiator schedules it through all its steps, with JES3 being involved only for items such as:

- Notification of step to step transition
- Opening of SYSIN/SYSOUT data sets
- Dynamic allocation and deallocation of data sets on JES3-managed devices
- Requests for spool space

### **Final Setup Processing**

When our sample job completes execution under MVS, it is returned to JES3 (more specifically, to the SETUP DSP) for device breakdown processing. Remember that at the end of each job step which is the last to use a device, volume, or data set, the resources are returned to the SETUP DSP for early resource release. Many, if not all, of the job's resources may already have been returned; but in most cases, the devices required for execution of the last step of the job must be returned to JES3 (along with data sets and volumes). Breakdown processing consists of updating control blocks by removing entries or reducing use counts and issuing appropriate keep or retain operator messages. The purpose, of course, is to make the resources available for use by other jobs which may require them.

The next activity, now that the sample job has executed, is for the job segment scheduler to mark the MAIN scheduler element complete and to schedule the OUTSERV scheduler element.

### ***Output Service Processing***

OUTSERV, the third scheduler element for our sample job, represents a single DSP responsible for the management of SYSOUT data sets generated by jobs during execution. Those we are concerned with were written directly onto the spool data set by our job or by JES3 DSPs running on behalf of our job. To handle these, the OUTSERV DSP first performs preliminary work necessary to schedule the writing of data sets to the output devices. This work consists of generating control blocks for work to be done by a transient writer routine. So, the OUTSERV DSP differs from

many DSPs in that it is a scheduler of work to other DSPs: writers of various types.

### **Matching of Data Set and Device Characteristics**

The actual scheduling process is designed by an installation's system programmers who prepare an algorithm with variables that reflect the characteristics of data sets and output devices. The scheduling process involves a matching of data set characteristics to device characteristics for routing of the data sets. The characteristics are:

- Priority
- Destination
- Device type
- Forms, FCB/carriage tape, and train
- Line limit
- Character set image and forms flash cartridge

**Priority:** Data sets to be managed by the OUTSERV DSP may have an assigned level of priority. The priorities may have been assigned by a JES3 DSP that generated a data set, by a programmer via a control statement, or by default from SYSOUT class definitions. The final result is the same in any case -- the OUTSERV DSP uses the priority to enqueue a data set for subsequent handling (printing, punching, sending to an external writer, etc.).

**Destination:** The JES3 initialization process allows the association of symbolic names with the devices used by DSPs. The devices may be local or remote. The same **device group name** could be assigned to multiple devices. For example, a card reader and a printer could be located in a room near the application programmer work area. Suppose the name "TESTGRP" is assigned to these two devices. The OUTSERV DSP would send back to the TESTGRP printer the output of any jobs that entered the system from the TESTGRP card reader. You can see that the same name can serve as an origin name and a destination name. For remote devices, both input and output devices would use a group name that is the name of the remote work station. Destination names may be overridden by means of specifically assigned destinations in JES3 control statements associated with their jobs.

**Device Type:** This characteristic is probably self-explanatory. Data sets that were defined as "print type" will be sent to printers, "punch type" will be routed to punches, and "sys type" will be routed to TSO or to external writers.

**Forms, FCB/Carriage Tape, and Train:** These three characteristics have been grouped together, since they have a like effect on output. The OUTSERV DSP will route output to a printer having the proper "setup" characteristics or to a printer having characteristics that may be changed to match the requirements of a data set. Notice what was said: "A printer whose characteristics **may** be changed." This statement implies that you **can** define devices with characteristics that are **not** to be changed. It follows, then, that many installations will have some printers with specified, unchangeable characteristics and other printers whose characteristics are changeable as required.

**Line Limit:** Another characteristic assigned to output devices is that of line limit. You can use this limit to cause data sets of less than 10,000 lines, for example, to print on one printer and data sets with more than 12,000 lines to print on another. In this manner, you can route low volume output to a slow printer and high volume output to a fast printer, should you so choose.

**Character Set Image and Forms Flash Cartridge:** These two characteristics are associated with the IBM 3800 printers, giving you the capability of changing the print character set and the forms to be “flushed.”

## Types of Writers

Remember that the OUTSERV DSP is a scheduler of work to writers. Writers may be categorized in several ways, but principally fall into two distinct groups: dynamic writers and hot writers.

A **dynamic writer** is automatically scheduled when:

- There is work (one or more data sets) in the output queue, and
- There is an output device available.

Dynamic writers reduce the amount of control operations personnel have over when and how writing is performed. They do allow changing of setup characteristics of devices. An example of the use of dynamic writers is volume printing on stock paper.

**Hot writers** give operations people total control of output handling. Operators enter commands to call and control hot writers.

Another difference between dynamic and hot writers is that characteristics assigned to dynamic writers are specified in the JES3 initialization statements. These may only be changed with JES3 START or RESTART commands. An operator may allow a hot writer to assume some or all of the installation-default set of characteristics, or the operator may specify an override set, thereby gaining total control over hot writers.

Both dynamic and hot writers could be used to direct classes of output requiring special forms to a particular printer, thereby concentrating operator intervention on a single printer.

Special handling is needed for data sets directed to TSO users, to devices not supported for output service, and to the internal reader. These data sets are transcribed to system routines rather than to devices:

- Data sets to be held for TSO processing are sent, on demand, to the TSO OUTPUT command processor.
- Data sets meant for unsupported output service devices (such as DASD or magnetic tape devices) are held for the **external writer**, which is an MVS routine that handles such devices. A control block is passed to the external writer to indicate there is work to do.
- Data sets meant for the **internal reader** are sent to a JES3 DSP previously discussed: input service (IS). Internal reader data sets contain job streams.

## Scheduling of Output

The processing of our sample job was interrupted at the point where the job segment scheduler scheduled the OUTSERV scheduler element. When the OUTSERV DSP is dispatched, it accesses the control blocks necessary to locate data sets it is to process. These control blocks also contain the

characteristics of each data set. Remember that the OUTSERV DSP constructs other control blocks to represent work to be done by writers, and schedules work for the writers.

### **Writing of Output**

A **writer**, when scheduled by the OUTSERV DSP, is a transient JES3 function. It accesses the control information enqueued by OUTSERV and attempts matching the characteristics of a data set to those of the device obtained by the writer. Data sets are transcribed on a "best-match" basis, allowing for the fact that a given writer function may or may not be able to process all of a job's data sets. Thus, scheduling of several writers may be necessary to complete the handling of the sample job's SYSOUT data. Many dependencies exist which determine where, when, and how data sets are managed. As each data set is written, Type 6 system management facilities (SMF) records are recorded for later use in job accounting. The job's OUTSERV scheduler element remains active until all data sets for the job have been written to the proper destinations.

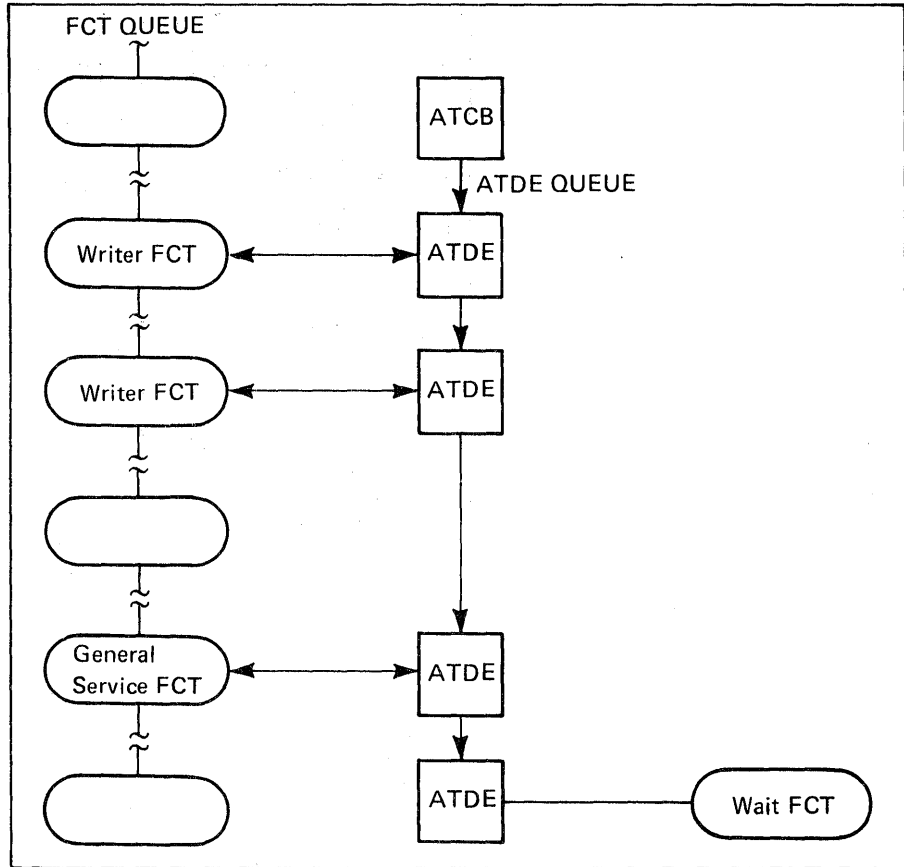
### **Writer Output Multitasking Facility**

The **writer output multitasking facility** allows JES3 output writers to do more work in parallel with other JES3 functions on a tightly-coupled global processor. To enable the writers to do more work in parallel, JES3 provides an additional task under which the writers can execute. This task, called the **JES3 auxiliary task**, is a subtask of the **JES3 primary task**. The JES3 auxiliary task can execute in parallel with the JES3 primary task or with other JES3 subtasks.

After the system programmer or an operator enables the writer output multitasking facility, active output writers execute part of the time under the JES3 primary task and the rest of the time under the JES3 auxiliary task. Writers execute under the auxiliary task while reading the job output files from the spool and while writing these files to an I/O device such as a printer. The rest of the time the writers execute under the JES3 primary task.



**Control Block Structure:** To manage the dispatching of writer FCTs, JES3 uses the control block structure shown in the following diagram. This structure permits JES3 to dispatch an FCT under either the primary task or the auxiliary task. The auxiliary task control block (ATCB) contains information that JES3 needs to manage work done under the auxiliary task. Included in the ATCB is a pointer to the auxiliary task dispatching element (ATDE) queue.



There is an ATDE for each FCT that is dispatchable under the auxiliary task. Thus, there is an ATDE for each active remote or local output writer, one for the general service FCT (discussed later in this topic), and one for the auxiliary task *wait* FCT.

JES3 uses an ATDE to determine whether to dispatch its related FCT under the auxiliary task. When there are no FCTs ready to be dispatched under the auxiliary task, JES3 dispatches the auxiliary task *wait* FCT. This causes the auxiliary task to enter the wait state.

**General Service DSP:** The general services FCT represents the general services DSP. The general services DSP does work for other DSPs when the work involves a resource whose use must be serialized under either the JES3 primary task or the JES3 auxiliary task. For example, if a DSP wants to add an ATDE to the ATDE queue, the DSP must do so through the general services DSP. The general services DSP can execute under either the JES3 primary task or the JES3 auxiliary task.

### Purge Processing

After all data sets are written, there is another return to the job segment scheduler which marks the OUTSERV scheduler element complete and schedules the PURGE scheduler element.

You are about to see the sample job reach completion, since PURGE is its last scheduler element (see Figure 5, Part 5).

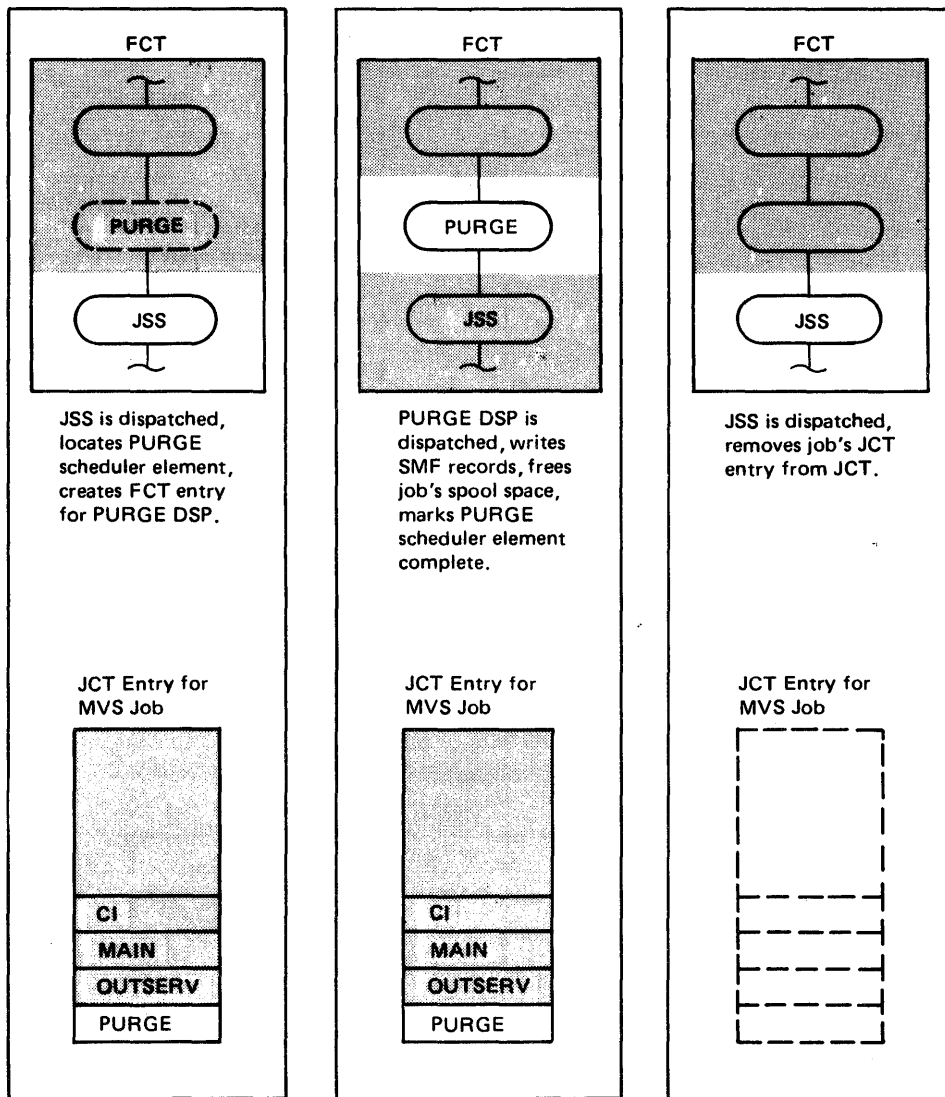


Figure 5 (Part 5 of 5). JES3 Job Flow

There are several items for which the PURGE DSP is responsible:

- Recording of SMF record type 25, generated during MDS processing to indicate the setup requirements of the job
- Recording of SMF record type 26, the job management record

- Most importantly, the freeing of all the spool space that still belongs to the job. Most of the spool space acquired during the life of the job remains assigned until PURGE time. This space contains:
  - The original JCL for the job
  - Message data sets
  - SYSIN/SYSOUT data sets
  - JES3 control blocks which define the job and its processing characteristics
  - The MVS scheduler work area (SWA) control blocks generated during converter interpreter (CI) processing

When the PURGE DSP completes its processing, it returns control to the job segment scheduler. But the job segment scheduler recognizes that the

return is being made by the PURGE DSP and removes the job's JCT entry from the JCT, thus eliminating the job from the system.

## Chapter 2 Review

Scheduler elements represent the pieces into which JES3 divides the work it must perform to ready a job for execution and to clean up afterwards. The JES3 programs that perform the work are called dynamic support programs, or DSPs. DSPs are scheduled for execution by the job segment scheduler, itself a DSP.

For each job in the input stream, and for each special JES3-created job, JES3 builds a job control table (JCT) entry and places into it a description of the job and a set of scheduler elements. The job segment scheduler searches the JCT entries seeking scheduler elements for DSPs to be executed, and it adds corresponding entries to the master JES3 dispatching queue, the function control table (FCT). The entries in the FCT are arranged in priority order. The FCT is searched by the multifunction monitor, which makes the dispatching decisions and causes the DSPs to be executed. The job segment scheduler has a permanent entry in the FCT (at a low priority), so it eventually gets dispatched again to add more entries to the FCT, thereby continuing the process.

Activity typically begins when an operator calls a card, tape, or disk reader. A high priority entry in the FCT results in execution of the CONSOLES DSP, which recognizes the JES3 CALL command and causes another DSP, the work-to-do-driver, to create a JCT entry for a reader job. A reader job is a special JES3 job (in contrast to the user job being processed). The JCT entry for the reader job contains a scheduler element for the reader DSP which causes the job segment scheduler to create an FCT entry for the reader DSP. The job stream will be read after the multifunction monitor searches the FCT and dispatches the reader DSP. The reader DSP reads batches of jobs, places the card images into the spool data set, and constructs one JCT entry for each batch. Each of these JCT entries represents another special JES3 job, called an input service job, and each of these JCT entries contains a scheduler element for the input service DSP. Again, the job segment scheduler comes into play: it creates an FCT entry for each input service scheduler element.

Next, the multifunction monitor dispatches the input service DSP which reads a batch and constructs a JCT entry for each job in the batch. This JCT entry contains four scheduler elements which represent all of the DSPs needed to process the job both before and after its execution. The scheduler elements are converter interpreter, main, output service, and purge. Each of the corresponding DSPs, in turn, is given an FCT entry by the job segment scheduler; each FCT entry is selected by the multifunction monitor; and when all DSPs for the job have been executed, then the job has been processed completely by both JES3 and MVS.



## Chapter 3. The Subsystem Interface

Programs in individual address spaces often need JES3 services, therefore, there is a need for communication among address spaces. Such a communication mechanism has been established and is called the **subsystem interface (SSI)**. A basic understanding of the subsystem interface is essential if you are to grasp the concepts of MVS to JES3 communication. In this chapter you will learn about the general structure of the SSI and the communication paths among:

- MVS system components and JES3
- JES3 and JES3

In preparation, you should review the basic structure of MVS. The typical picture of an MVS environment consists of a large box representing the storage layout of the system (see Figure 6).

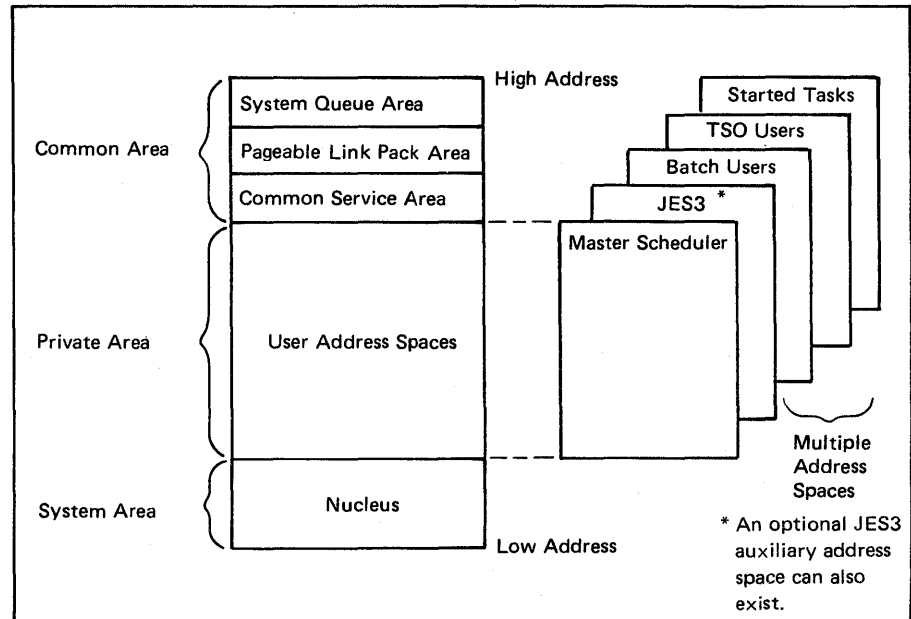


Figure 6. The MVS Environment

Areas of storage usually shown include (from high address to low):

- The system queue area
- The pageable link pack area
- The common service area
- Storage in which virtual address spaces are established for the master scheduler, JES, batch and TSO users, and started tasks (one of these address spaces can be a separate **JES3 auxiliary address space**, containing data that JES3 would otherwise store in the CSA)
- The nucleus

A review of the roles played by global JES3 and local JES3 will help you understand why communication is needed. Remember that global JES3:

- Introduces all jobs into the system, no matter what the source
- Converts and interprets JCL

- Performs preexecution setup of devices
- Schedules MVS jobs to all main processors
- Maintains awareness of all jobs in execution
- Handles all SYSOUT data sets
- Manages the allocation and deallocation of space on the shared-spool devices

When carrying out some of the responsibilities listed above, global JES3 needs the assistance of local JES3. This is true during the scheduling of work onto a local processor.

Local JES3 has these responsibilities:

- Returning needed information to global JES3 to satisfy requests to locate cataloged data sets and to verify device mounts
- Routing of console messages to global JES3
- Generally, keeping the global JES3 aware of what is happening on the local processor

What exactly are the communication paths? From Figure 7 you will see that global users can make requests of global JES3 and responses will be returned.

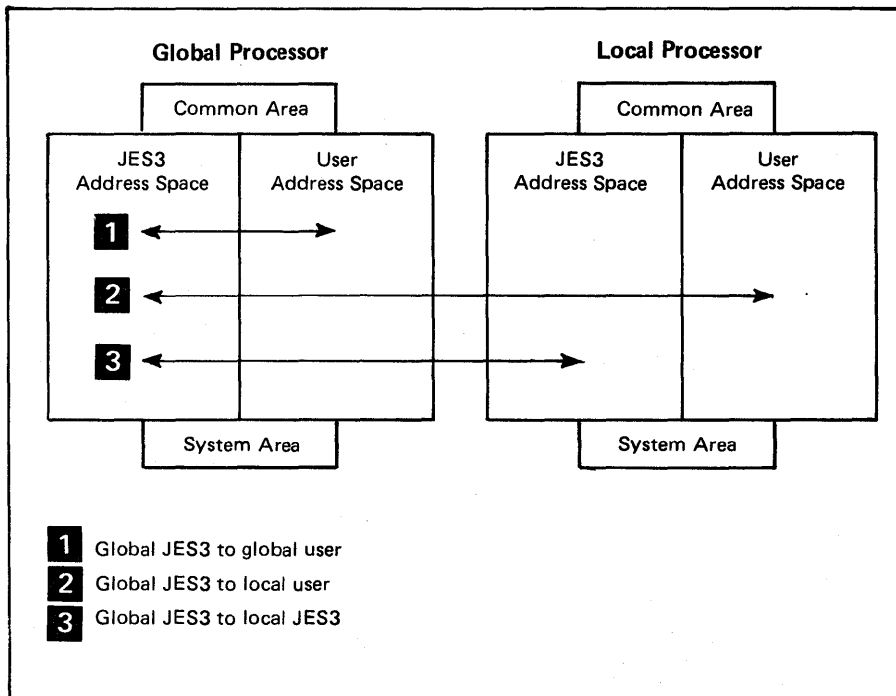


Figure 7. Address Space Communication

For example, an initiator in a user address space would request a job and receive a set of descriptive data defining the job. An extension of this form of request/response would be an initiator in a local processor requesting a job from global JES3. There are also a few cases in which global JES3 and local JES3 must communicate. Either global or local JES3 can initiate the communication, and, in many cases, responses are not required. It would appear from the illustration that a fourth type of communication might be used: local user to local JES3. This last type of

communication, however, is not used in the JES3 environment. Also, no communication paths are supported between a global user and local JES3 or local users. Next, you will see how communication paths are implemented.

## Types of Communication

There are about 40 (the number varies depending upon which features are installed) cases in which MVS communicates with JES3. Each case is assigned a number, called a **function code** (see Figure 8). These SSI function codes represent a request for a service, or control information.

<p style="text-align: center;"><b>Scheduling</b></p> <p>4 Step termination 5 Job selection 8 End of memory 12 Job termination 13 Job requeue 22 Step initiation 32 S INIT command failure</p>	<p style="text-align: center;"><b>SYSIN/SYSOUT</b></p> <p>6 Allocation 7 Unallocation 16 Open 17 Close 18 Checkpoint 19 Restart</p>
<p style="text-align: center;"><b>Resource Management</b></p> <p>23 Dynamic allocation 24 Common allocation 25 Common/dynamic unallocation 26 Change DDNAME 27 Change enqueue use attribute 35 MSS volume inventory control 36 MSS mount equalization 37 MSS OPEN/EOV 40 Early volume release</p>	<p style="text-align: center;"><b>Dynamic Device Reconfiguration</b></p> <p>28 Candidate selection 29 Candidate verification 30 Swap notification 31 Swap completion</p>
<p style="text-align: center;"><b>Consoles</b></p> <p>9 WTO/WTOR 10 SVC 34 14 Delete operator message 33 Switch master console 34 WTL</p>	<p style="text-align: center;"><b>TSO</b></p> <p>1 Output 2 Cancel 3 Status 11 Remote destination validation</p>
	<p style="text-align: center;"><b>Miscellaneous</b></p> <p>15 Verify subsystem ID (unsupported) 20 Request job ID 21 Return job ID 38 Converter subsystem exit (unsupported) 39 Group subsystem request (unsupported)</p>

**Figure 8. SSI Function Codes**

JES3 supports all but code 15, 38, and 39, which are unique to MVS. Several of the request types do not reach global JES3. These are processed directly by JES3 routines in the processor that originated the request.

When JES3 communicates with JES3, another, similar, set of codes called **JES3 destination codes** are used. Both JES3 destination codes and MVS function codes are handled in the same general manner.

## Communication Overview

Prior to following a request through the SSI, you should understand the control blocks involved (see Figure 9).



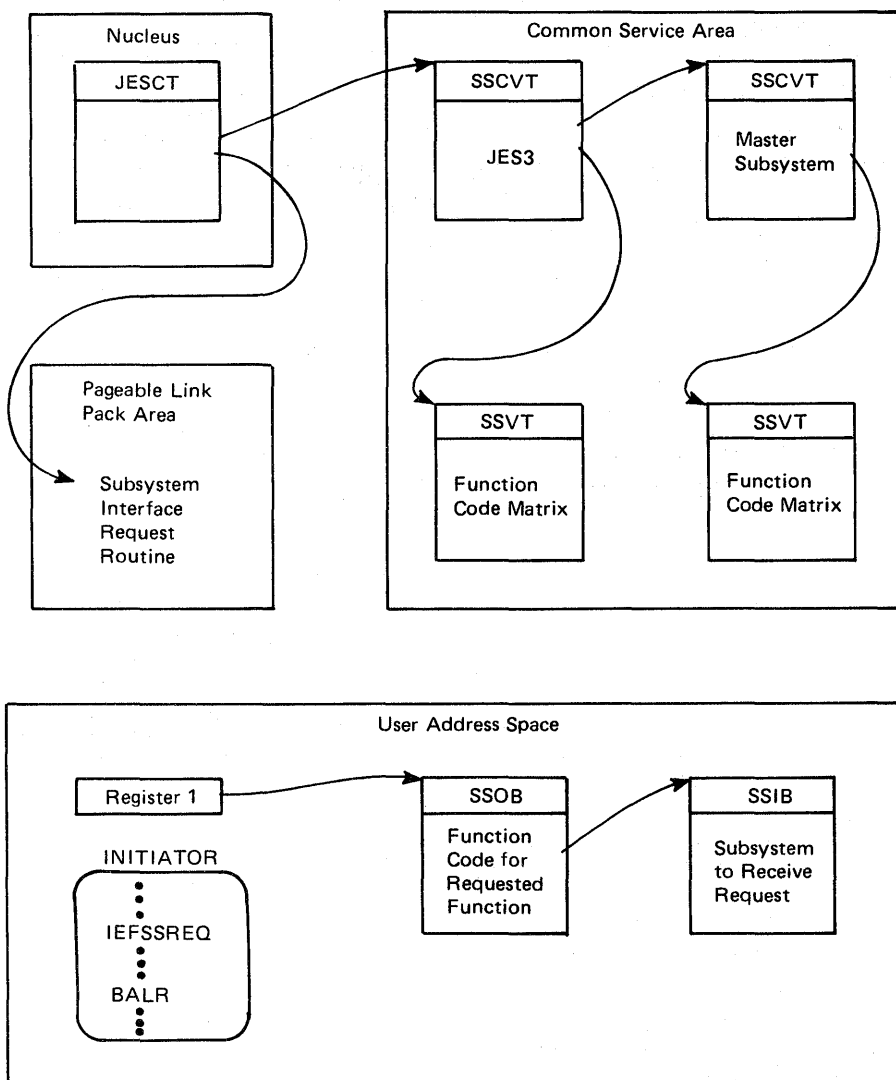


Figure 9. SSI-Related Control Blocks

The chain of control information for SSI begins in the MVS nucleus with a control block named the **JES control table (JESCT)**. This block contains a pointer to the primary **subsystem (JES2 or JES3) communication vector table (SSCVT)**, which is located in CSA. There will always be at least two SSCVTs, including one for the MVS master subsystem. The primary subsystem SSCVT contains three items of interest:

- The ID of the subsystem (specified during initialization).
- A pointer to the master subsystem SSCVT. (SSCVTs are constructed during MVS IPL. They are used for two purposes: checking whether a subsystem is active when needed and locating the address of an SSVT.)
- The address of the **subsystem vector table (SSVT)**. The SSVT is built by JES3 during initialization. A matrix in the SSVT shows which function codes are supported by the subsystem. The SSVT also contains addresses of the routines needed to process the various function codes.

Now, to follow a request through the SSI, assume that an MVS initiator is active in a user address space. At some point in its processing the initiator will ask JES3 for a job by issuing an **IEFSSREQ system macro**. The initiator provides a pointer to a **subsystem options block (SSOB)**, which contains a function code that defines the request being made. (For the job select request, the code is 5.) The SSOB also contains the address of the **subsystem identification block (SSIB)**, which identifies the subsystem to which the request is to go (in this case, JES3). The expansion of the IEFSSREQ macro contains instructions to cause entry to the request routine.

### ***The Request Routine***

This routine, IEFJSREQ, residing in the pageable link pack area (PLPA), performs several activities. After checking the validity of the SSOB and SSIB, the request routine determines that the target subsystem exists and is started. It does this by referring to subsystem IDs in the chain of SSCVTs. It next uses the function code to determine if the subsystem performs the requested function and to derive the address of a routine to which the request is to be passed. For invalid function codes, the IEFJSREQ routine returns an error code to the issuer of the IEFSSREQ macro.

### ***The Function Routines***

There are about 36 SSI functions supported by JES3, but several of the routines perform more than one function. All of the function routines are not described here, but you should understand their general logic.

One of the duties of a function routine is to construct a **service entrance list (SEL)** if global services are needed. (Some requests can be satisfied on the local processor.) The content of an SEL varies with the type of request. An SEL could contain:

- The address of a data area
- The address of an ECB
- The address of a response buffer
- The address of a staging area (which has not yet been described)
- The address of an exit routine

The function routine gathers data, places that data into an SEL, and issues a JES3 macro, SSISERV. This macro causes entry to a routine called the common services routine, which is divided into two parts: the router and the poster.

### ***The Common Services Router***

You may hear the common services router referred to as the top half of common services. The name router is the more descriptive of the routine's services - the routing of a request to the appropriate address space. Seven types of requests can be made with the SSISERV macro:

- **wait type:** communication of data that requires a response. The requestor waits until the response is received.
- **reply type:** similar to a wait type, except the requestor may wait or use an exit routine to process the response.
- **ack type:** communication of data with no required response other than an acknowledgement of receipt.
- **comm type:** sending of data only, no response is necessary.

- **resp type:** a response to an earlier wait, reply, or ack type of request.
- **end of memory type (EOMT):** a request for storage cleanup of SSI communication areas that might still be outstanding for an address space.
- **purge type:** used by JES3 DSPs to free staging areas that the DSPs have processed.

A brief review of events leading to processing by the router routine would be helpful now:

- A system routine requested some service by using the IEFSSREQ macro.
- The subsystem interface request routine verified the request and passed it to an SSI function routine.
- The SSI function routine put the request into an SEL and entered the common services router via the SSISERV macro.

The vehicle used for communication between address spaces is called a **staging area**. Because multiple address spaces are involved, the staging area is located in the common service area (CSA) or in the JES3 auxiliary address space. (A pool of staging areas is created during JES3 initialization.) After acquiring a staging area, the router transfers information to it from the SEL. Then the router determines if the sending address space and the address space of the receiver are in the same processor. This is necessary because if the sender and the receiver are in the same processor, an **MVS service request** can be made to accomplish the communication. (MVS service requests are used by subsystems like JES3 in communicating between address spaces. The requestor must build a service request block and issue a SCHEDULE macro.) If the two address spaces are in different machines, the router must issue a STARTIO macro to write staging areas across the channel-to-channel adapter.

After issuing the SCHEDULE or STARTIO macro the router takes action depending upon the type of communication request being made:

- For wait type requests, the router issues an MVS wait macro instruction.
- For reply, ack, or comm requests, the router returns control to the function routine.

Note at this point that all the code executed in the subsystem interface has been located in PLPA and is run under control of the user address space.

### ***The Common Services Poster***

The request, in a staging area, now enters the second part of the common services routine called the poster. This is referred to as the **read end subroutine**. The poster routine executes on the receiver's processor, under control of the receiver's address space. The poster gets the request to its receiver by:

- Placing the staging area on a queue serviced by the receiver
- Posting the receiver to signify that a request has arrived

Addressability is obtained with an MVS service request, the equivalent of a "cross-storage post."

In CSA, there is a control block to facilitate queuing and posting. The control block is the **destination queue DSQ**), sometimes called the **destination routing table**. It is similar to the subsystem vector table (SSVT);

it has two matrixes (which relate SSI function and JES3 destination codes to receivers) and entries which contain queue origin pointers and data used to post the receiver of a request.

When an initiator requests a job, the poster queues the staging area for the MAIN DSP and posts the MAIN DSP to indicate it has a request to service. The MAIN DSP, when dispatched by the multifunction monitor, will select a job for the initiator and will place scheduling information needed by the initiator into the staging area. Next it will issue the JES3 JSERV macro to send the scheduling information to the initiator. (The JSERV macro causes entry to the router routine of SSI common services — the same process as used before, but now the code for the router is executed under control of the JES3 address space.)

To return scheduling information to an initiator, the router issues either a SCHEDULE or STARTIO macro.

Responses are moved from the staging area, in which they were received from the sender, to the original SSOB, and the user is posted for “receipt of response.” (Notice that the return of a response is **not** the reverse of making a request, it is almost exactly the **same** process.) The common services router, when posted, returns control to the SSI function routine which in turn returns control to the instruction following the IEFSSREQ macro. The initiator now finds the information it requested in the SSOB, and can resume processing.

### Chapter 3 Review

In this chapter, many codes, macros, and routines have been discussed. A brief review is in order.

**Codes:** The MVS subsystem interface function codes are used when a user address space needs to communicate with JES3. Should one JES3 routine need to communicate with another JES3 routine, JES3 destination codes are used instead of MVS SSI function codes.

**Macros:** An SSI request made from a user address space is started on its way by issuance of an IEFSSREQ macro. Subsequently, an SSISERV macro is issued by a function routine to pass the request along to SSI common services (the router). The router routine uses a STARTIO or SCHEDULE macro to send the request through the poster routine of common services to global JES3. Once the request has been satisfied, a JSERV macro is issued by JES3 to route the response (if one is required) back to the requestor.

**Routines:** The first routine entered during the normal course of processing of an SSI request is the request routine, IEFJSREQ. After validation of the request, a function routine that is responsible for supporting the request is given control. In a few cases, the request is satisfied directly by the function routine. When the function routine does not directly satisfy the request, the request is passed to JES3 via the SSI common services routines. The common services routines are also used in returning the response to the original requestor.



## Chapter 4. Spool Data Management

One of the significant features of JES3 is its spooling capability. Though a typical large JES3 installation may have 10 **spool volumes**, JES3 supports as many as 31. These spool volumes must reside on direct access storage devices (DASDs) such as the IBM 3330, 3340, 3350, 3375 or 3380. The environment may consist of a single device type or any combination of the supported device types. Because all JES3 processors share the spool volumes, each processor must have access to the spool space.

JES3 records several types of data on the spool volumes:

- The information (originally taken from the initialization statements) necessary to initialize JES3 in the global and local processors
- The JES3 control blocks that define the scheduling and operational characteristics of the jobs
- The SYSIN (DD\* or DD DATA) data sets and SYSOUT data sets for jobs

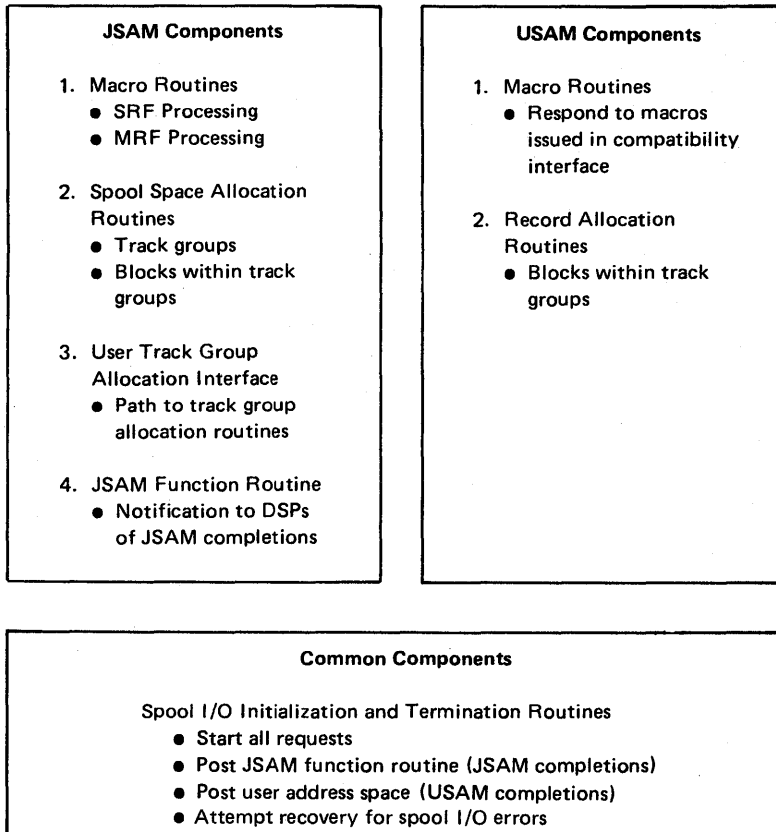
There are two techniques used for recording spooled data. Many control blocks are recorded as **single-record files (SRFs)**. This means that a control block, irrespective of its size, is recorded as a single buffer image (a block of data on a spool volume). While most of the control blocks will fit into a single buffer image, some (due to the variable lengths and multiple parts) may extend across multiple buffers. When this occurs, the buffers are chained together so that there is still a single “record” in the file.

It is more efficient to record files such as a job’s JCL, SYSIN, and SYSOUT with a technique that allows packing of multiple records (card images, print lines) into a single buffer image. These files are called **multirecord files (MRFs)**.

The management of spool space and recording/retrieval of spooled data are the responsibilities of two special access methods. JES3 DSPs use the **JES spool access method (JSAM)** to read and write data. Programs in user address spaces utilize the **user spool access method (USAM)** for handling their spooled data, though USAM is transparent to user programs. If you examine the component parts of JES3 spool data management (SDM), you would quickly see that some of the parts are unique to JSAM, some are USAM-related, and others are common to both access methods.

### JSAM Components

JSAM can be separated into four parts. The first consists of the macros used by DSPs to record and retrieve both SRFs and MRFs (see Figure 10). There are macros to acquire spool space, create spool files, read and write records, and to purge files when they are no longer needed. Many of these macros involve acquisition of space on a spool volume (a DASD block) for the JES3 records. Notice that use of the term “record” is inconsistent with normal DASD terminology. JES3 spool access methods do not use the concept of “blocked” records.



**Figure 10. Spool Data Management Functional Components**

The second JSAM component is a set of routines responsible for the allocation of space. The unit of JES3 spool space is called a **track group**. Track groups are multiple physical tracks on DASD. The number of tracks in a track group is defined by a JES3 initialization parameter. A track group can be either a half cylinder or a full cylinder (the quantity is fixed, once it is defined). There are routines to allocate track groups and other routines to allocate individual blocks within track groups.

The third JSAM component is a routine that acts as an interface between users requesting track groups and the track group allocation routines mentioned earlier. User address spaces are assigned one or more track groups (prior to MVS execution) as a job is being processed by the input service DSP. This initial allocation is used to hold the job's control blocks, SYSIN data, and the MVS scheduler work area (SWA) control blocks generated by the converter/interpreter. While the job is executing, SYSOUT data sets are written into the remaining portion of the space originally allocated, as long as they fit. Should the initial allocation be insufficient to hold all the job's SYSOUT data sets, additional space will be allocated by global JES3 via the interface routine.

The fourth JSAM component is named JSAM. JSAM is responsible for notifying DSPs of I/O completions by setting flags to make the waiting DSPs dispatchable.

## USAM Components

The USAM components can be grouped into two categories. First, there is a set of routines that provide problem programs access to spool data. These routines, called the compatibility interface, intercept requests that the

problem programs make for BSAM or QSAM services, and cause entry to JES3 routines rather than to BSAM or QSAM routines.

When a buffer is filled during creation of a SYSOUT data set, the second USAM component comes into play. Though USAM cannot allocate track groups of spool space, USAM does have responsibility for allocating individual blocks of space into which buffer contents are to be written. These blocks are suballocated from the set of track groups originally assigned to the user job (until, of course, that space is exhausted). If additional track groups are required, requests are forwarded to JSAM's track group allocation routine through the subsystem interface and the allocation interface routine.

## Components Common to JSAM and USAM

JES3 manages I/O to the spool device via the STARTIO macro, and thus JES3 routines use the services of the MVS input/output supervisor (IOS). The routine that schedules spool I/O operations is common to JSAM and USAM. This routine is responsible for initiating and terminating I/O requests. When JSAM requests are completed, the common routine posts JSAM to signal the I/O completion to the appropriate DSP. USAM I/O completions cause the routine to notify the appropriate user address space that I/O activity is complete.

## Buffer Pool Management

Spool I/O is performed between spool volumes and buffers in storage. Both the size and number of buffers involved are predefined at JES3 initialization by means of parameters in JES3 initialization statements. The **buffers used for spool I/O** contain a fixed area which is not transferred to spool volumes. This area contains a channel program used to read or write the data portion of the buffer, flags to indicate the status of the buffer, and any required chaining fields. Next to the fixed area is a data area which is formatted differently for SRFs and MRFs.

The buffer length can be either 1248, 1952, or 4000 bytes. These lengths, when increased by the size of the buffer's fixed area (96 bytes), result in 3 buffers, 2 buffers, or 1 buffer, respectively, per page of virtual storage. The 1952 byte size is most commonly used.

JES3 buffers are always grouped into buffer pools, of which there are three types:

- The JESIO buffer pool, which is in the global JES3 address space and is used by DSPs doing SRF/MRF processing. I/O takes place directly to and from these buffers.
- The user storage buffer pools, which are inside the user address space. These consist of either one page of virtual storage for a SYSIN data set or multiple pages for SYSOUT data sets. The contents of user storage buffers are not transmitted directly to or from spool volumes, but rather are moved to or from a protected buffer pool.
- The protected buffer pool, which can be totally contained in the common service area (CSA) of each processor, split between the CSA and a JES3 auxiliary address space, or entirely in a JES3 auxiliary address space. (Space in the CSA is at a premium. There are parameters in JES3 initialization statements for defining auxiliary address space for protected buffers that JES3 would otherwise put into the CSA.)



JES3 uses the protected buffer pool for I/O to the spool device. The buffer pool is protected so that the integrity of the data can be maintained among address spaces. The buffer pool is in the CSA (and/or the JES3 auxiliary address space) so that it will not be swapped out.

## Data Flow for USAM Requests

Because most readers are assumed to be concerned primarily with the relationships between user address spaces and JES3, the following data flow discussion is directed at USAM requests. Figure 11 is a storage map that highlights the SDM components discussed.

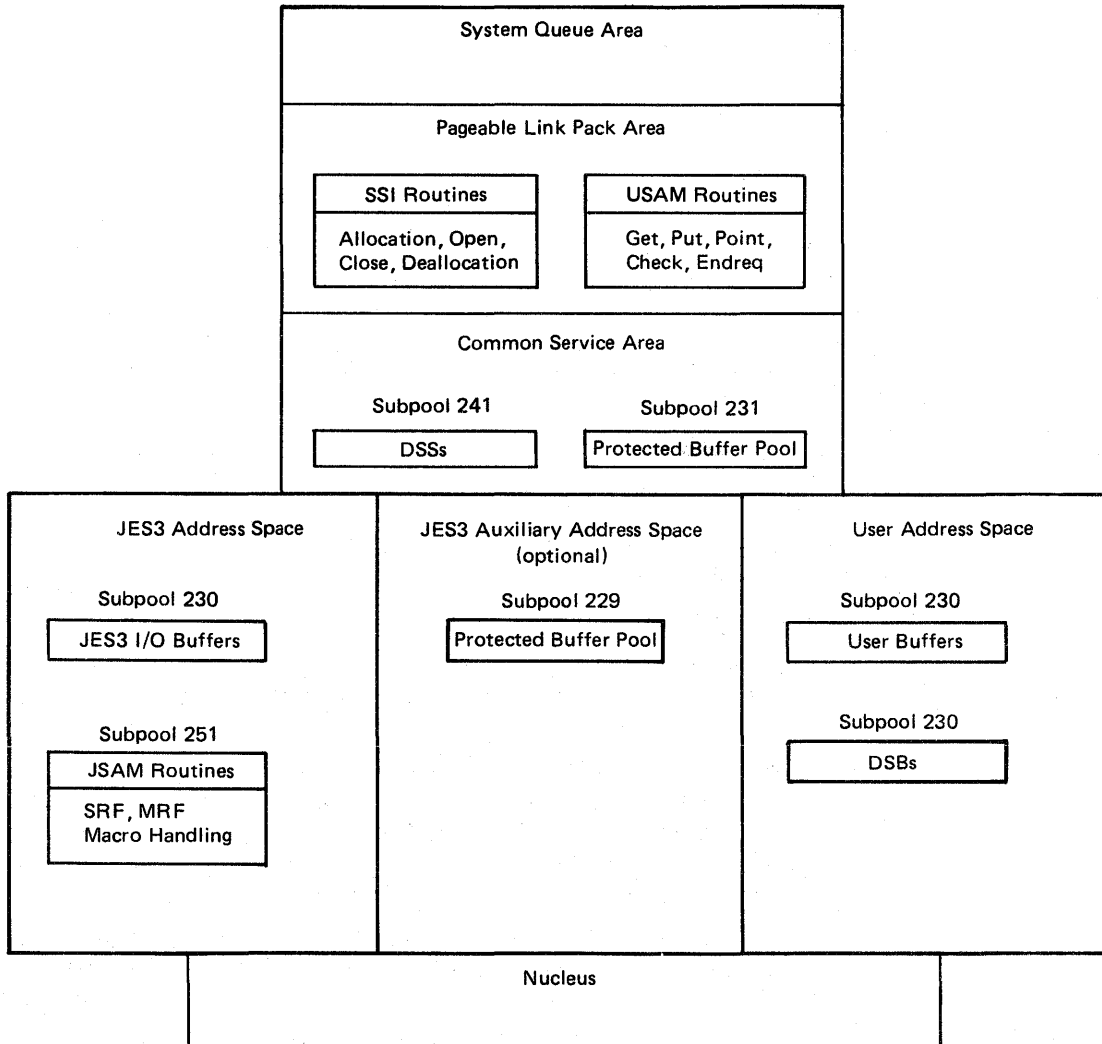


Figure 11. Storage Map of SDM Components

## ***SYSIN Processing***

The MVS allocation routines, when allocating a SYSIN (or SYSOUT) data set, issue the IEFSSREQ macro. The SSI routines satisfy the request without help from global JES3, by constructing a data set status (DSS) block in the CSA and a data set block (DSB) in the requestor's address space. The DSS defines processing for a data set and the DSB defines characteristics of the data set. After building the DSS and the DSB, the SSI function routine returns control to the MVS allocation routines.

When the problem program opens the **SYSIN data set**, another IEFSSREQ macro is issued by the compatibility interface code to aid in opening the spool data set. After the IEFSSREQ macro is issued, the user buffer pool (one page of virtual storage) is constructed in subpool 230 of the user storage. (Only global JES3 knows the location of the SYSIN data set on spool.) A staging area containing information about the data set being opened is passed (via the SSISERV macro) to global JES3. Global JES3 responds to the SSISERV macro (via JSERV) with the disk storage location of the first MRF buffer of the SYSIN data set. This information is kept in a control block built for the job during input service processing. Upon receipt of the response, the OPEN processing can continue. Requests are then queued to prime the user buffer. Remember that this I/O operation consists of reading the data from spool to the protected buffer pool in the CSA. When the reading is finished, the protected buffer's content is moved to a user buffer.

The problem program can now access the data set by issuing GET or READ macros. Each logical record is passed to the program by the USAM routines that were given control by the compatibility interface. Then, as necessary, user buffers are refilled from protected buffers and I/O requests to refill the protected buffers are queued--until all information is read.

User buffers are freed during close processing by the SSI function routine. Thus, the data set and the problem program are disconnected. At the end of the job step, the MVS deallocation routines issue the IEFSSREQ macro so that the SSI function routine can free the DSS and DSB. Processing of the data set is now complete. (The problem program's spool space will be freed by the PURGE DSP.) Notice that only one communication between the SSI routine and JES3 was necessary.

## ***SYSOUT Processing***

SYSOUT processing, though similar to that of SYSIN, is somewhat more complex. This is because JES3 has no need to maintain information about a **SYSOUT data set** until it is opened, and thus such information is not readily obtained.

As before, our involvement begins at data set allocation time. The SSI function routine constructs the DSS and DSB in response to an IEFSSREQ request. For SYSOUT data sets, open processing (by the SSI function routine) consists of constructing an entry to be placed in the control block that JES3 uses to define a job's MRFs. This entry is then sent to global JES3, via SSISERV, where it is placed into the control block. A user buffer pool is then allocated in subpool 230, this time consisting of multiple pages of virtual storage (the default is 2 pages).

Following open processing, the problem program issues PUT or WRITE macros to create the SYSOUT data set. Each PUT macro causes entry to a USAM routine that counts the records, truncates any trailing blanks in the record, and moves the resulting record into a user buffer. When the

user buffer is filled, the USAM put routine causes the contents of the full buffer to be moved to a protected buffer and queued for output to the spool data set.

After creation of the SYSOUT data set, the problem program issues a CLOSE macro. The CLOSE macro causes IEFSSREQ to give control to the close routine in the SSI function code. Close processing by the SSI function routine includes writing the last user buffer to the spool data set and freeing of the user buffer pool. Here, there is another difference between SYSIN and SYSOUT processing. Remember the control block entry was sent to global JES3 at open time. For SYSOUT processing it must be updated during deallocation to reflect items such as the line count of the new data set. The updated entry is passed to global JES3 via the SSISERV macro to complete the entry made at the time of the OPEN. When the updating is complete, the JSERV macro notifies the function routine, which frees the DSS and DSB.

## **Chapter 4 Review**

Spool data sets are clearly the responsibility of JES3. You have seen that routines supplied as part of JES3 are deeply involved in the processing of both SYSIN and SYSOUT data sets. Spool data management is done by two distinct sets of routines, one being entered via a branch from the compatibility interface code, and the other entered as a result of IEFSSREQ macros being issued. A quick review of Figure 11 will help you locate these sets of routines, the buffer pools discussed and the major control blocks involved in JES3 spool data management.

## Chapter 5. Job Related Control Blocks

JES3, like other systems, uses control blocks to define the existence of jobs and to indicate how those jobs are to be processed. Some of a job's control blocks are kept on DASD until required for processing, at which time a copy is brought into storage. Other control blocks reside only in storage, with no copy on DASD. There are two types of storage-resident control blocks: those that exist for the life of the job, and those that are created when DSPs are scheduled. You will examine the general structure of control blocks, but first you should know more about the areas on DASD in which the control blocks are kept.

### Spool Volume Areas

Control blocks are recorded on the spool data set in the form of single-record files (SRFs). Two areas of spool space can contain control blocks:

- A commonly accessible area used by many components of JES3
- The spool space owned by a job

### *Single Track Table Space*

The part of spool space that is used by many components of JES3 is called the **single track table (STT)** space. This name will be more meaningful if you recall the description of tracks and track groups. The unit of spool space was defined as a track group, which consisted of a half or full cylinder of DASD space. You might think a track group consists of physical tracks. But *in JES3 jargon* a **track** is a disk record location or a DASD address at which a block (buffer image) is located. So, to be definitive (in JES3 terminology) a **track group** consists of a consecutive group of tracks or DASD addresses. Because it is often more efficient, in terms of spool space, to allocate a single track rather than a track group, JES3 uses the single track table (STT).

The STT consists of one or more areas inside (or, allocated from within) the DASD space assigned to JES3 as its spool space. Some (or, in special cases, all) of a job's control blocks are written as single-record files (SRFs) into the STT. JES3 uses a bit map of the available tracks in the STT to allocate or deallocate STT space, one track at a time.

### *Job Spool Space*

With the exception of called jobs, all jobs entering JES3 pass through the JES3 input service job and are assigned a minimum of a single track group of spool space. This space is used for many of a job's control blocks. When access is needed to a called job's control blocks, they are retrieved and written into the STT (because the jobs do not own spool space). The STT is used for some of the control blocks belonging to normal jobs as well.

### The Job Control Table Data Set

One of a job's control blocks, the JCT entry (first mentioned in Chapter 1) is recorded into a unique data set. This data set is allocated by MVS during JES3 initialization. Therefore, the JCT data set does not lie within the DASD space allocated for spool.

The JCT data set is in fixed-length, unblocked format. Remember that JCT entries vary in size: called jobs have two scheduler elements, normal jobs have four (or more), and non-standard jobs have a variable number up to an installation-defined maximum.

Even though the JCT data set is not contained within the DASD space allocated for spool, it is read from and written to the JESIO buffer pool by means of JSAM single-record file processing techniques. The JCT data set, depending upon how much space was allocated to it, contains a fixed number of blocks (or "slots"). These blocks are reused after a job is purged. Allocation and deallocation of the blocks is tracked via a storage-resident control block with a bit-map structure identical to that used for the single track table.

## Job Definition and Scheduling Control Blocks

Although many control blocks are associated with a job, two basically define the job. The first, and most important, is the JCT entry. To review, the JCT entry describes the processing to be performed on behalf of the job and the characteristics of the job. It also serves as a continually-updated checkpoint of the status of the job as it progresses through the system. A job exists in the system when its JCT entry is created and ceases to exist when the JCT entry is removed.

In addition to the JCT entry, there is another closely related control block that is kept in storage for each job in the system. This control block is called the **job queue element (JQE)**. The data in the JQE is essentially copied from the job's JCT entry. JQEs exist to reduce the I/O activity that would result from frequent examination of JCT entries for the scheduling of work. The JQE is a highly abbreviated set of information about a job. It contains only that information needed to make scheduling decisions and maintain an in-storage job status. It is the JQE that is used to answer most system or operator inquiries about jobs, further removing a need for JCT access. The JQE is constructed when the job's JCT entry is constructed. In one sense, the JQE is just as important as the JCT. It is the JQE which is first examined by the job segment scheduler when it determines what work is to be done.

When the job segment scheduler is dispatched, it examines JQEs to find an eligible job so it can schedule a DSP. Part of the processing is construction of an FCT entry when the DSP being scheduled is not a resident DSP. In addition to constructing and enqueueing an FCT entry (if necessary), the job segment scheduler always builds a **resident queue element (RQ or RESQUEUE)**. Pointers from the JCT entry to the job's other single-record files (SRFs) are extracted and placed into the RQ. The RQ, a large control block, is a storage area for status flags, job information fields, and queueing pointers. RQs last only for the life of a scheduler element.

Once the job segment scheduler schedules a job, the JCT is read, additional information is extracted, and the status is checkpointed by rewriting of the updated JCT entry. Pointers in the JQE and JCT entry allow JES3 DSPs working on behalf of a job to find all the other job-related information kept by JES3 (see Figure 12).

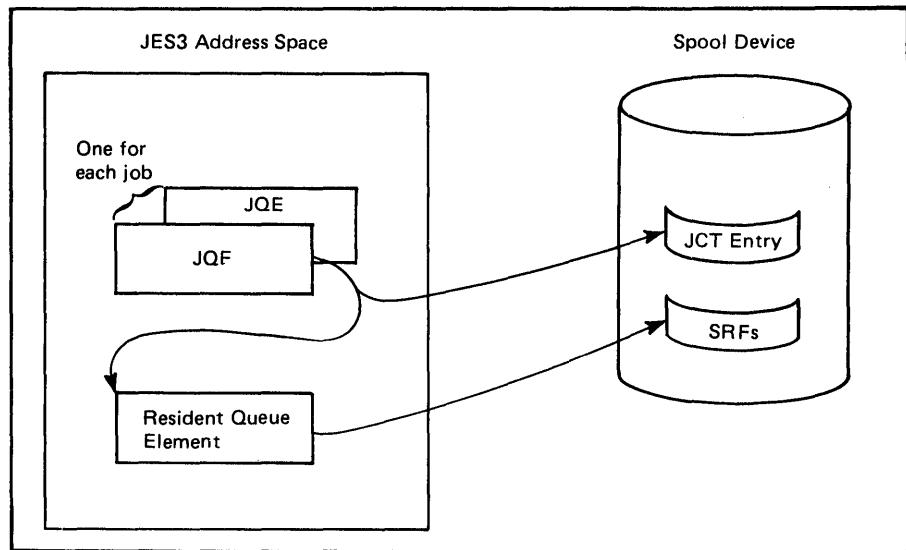


Figure 12. Job Queue Element Pointers

## Job Processing Control Blocks

If you examine the job-related control block structure for a “typical” standard job (see Figure 13), you will find several control blocks involved. The RQ was discussed previously. The remaining control blocks are discussed in the following sections.

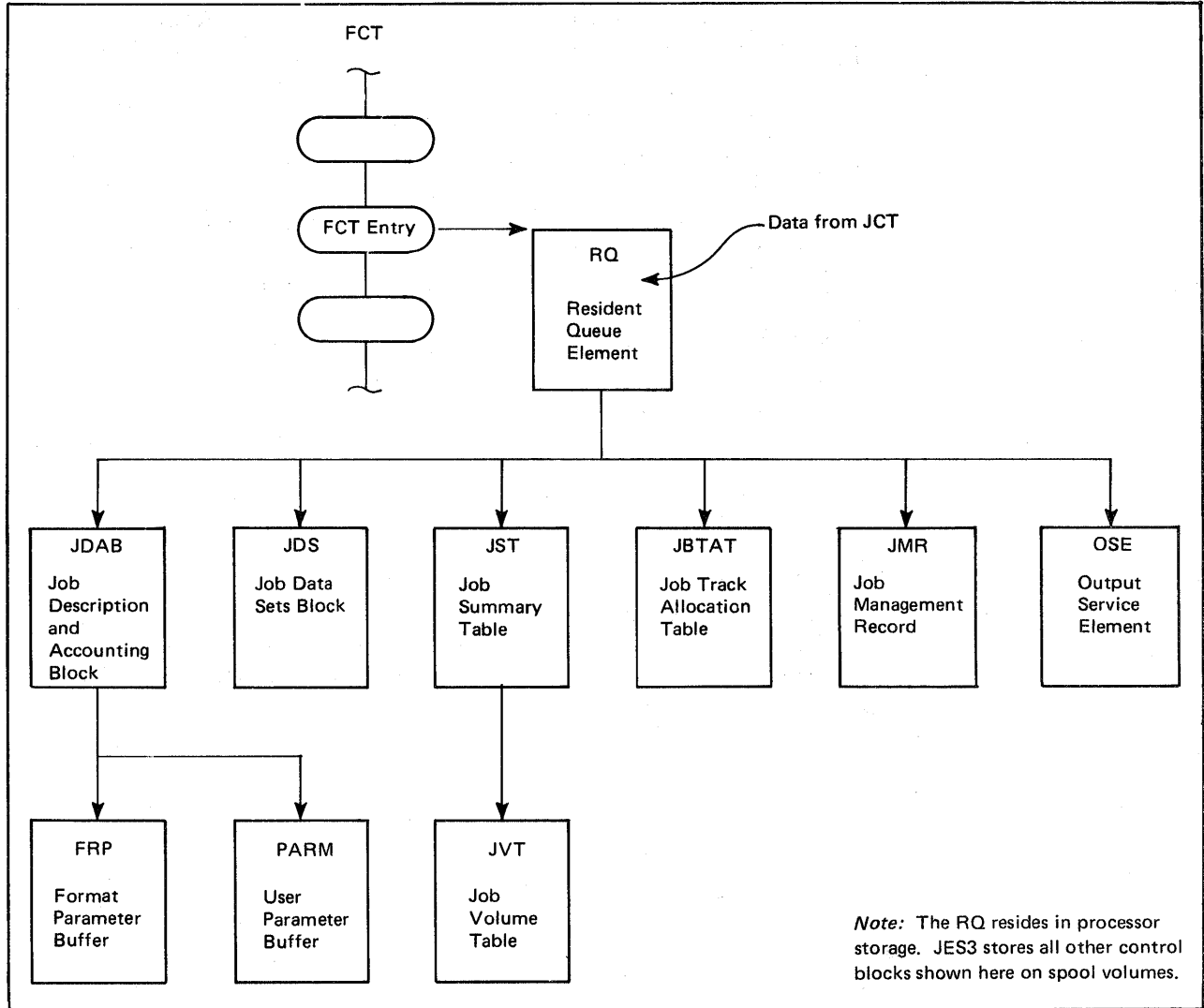


Figure 13. Job Processing Control Blocks

### ***Job Description and Accounting Block***

The **job description and accounting block (JDAB)** is constructed simultaneously with a JCT entry. The JCT and JDAB have much information in common. JDABs give DSPs a source of data about jobs and make JCT access unnecessary. Entries representing each scheduler element in a JCT entry are appended to the JDAB to provide DSP accounting for each DSP needed for the job.

### ***Job Data Sets Block***

All of a job's multirecord files are identified and defined in the **job data sets block (JDS)**. This block, built while the JES3 input service job is being processed, initially contains:

- JCL for the job
- Two message data sets: one for messages generated by DSPs and one for system-generated messages

- The scheduler work area (SWA) control blocks generated by the converter interpreter
- Any SYSIN data for the job

Entries for SYSOUT data sets are added to the JDS when the data sets are opened during MVS execution.

### ***Job Summary Table***

The **job summary table (JST)** contains requirements for preexecution setup for a job. Though the JST is constructed during input service processing, it is not filled in until converter interpreter processing. The JST entries are primarily used during initial setup processing, while devices are being assigned to the job.

### ***Job Track Allocation Table***

Built during input service processing, the **job track allocation table (JBAT)** is a bit map of the track groups of spool space allocated to a job. The table is updated to reflect any subsequent allocations as required. During purge processing, track groups are given back to the system or master track allocation table for reassignment to other jobs.

### ***Job Management Record***

The **job management record (JMR)** is built during input service processing. It contains the job accounting information necessary to construct a system management facilities (SMF) type 26 accounting record for the job.

### ***Output Service Element***

**Output service elements (OSEs)** contain a set of output (SYSOUT) data set characteristics relating to one or more data sets to be managed during output service processing. The first OSE for a job is constructed during input service processing and written to a spool volume. Later, during output service processing, this OSE is completed and others may be generated for the job. These elements become the "output queue."

### ***Format Parameter Buffer***

If the user indicates, via the JES3 FORMAT control statement, that there is to be special handling for any of the SYSOUT data sets, a **format parameter buffer (FRP)** is constructed for the job during input service processing. The FRP is used to store the information, taken from FORMAT statements, relating to individual SYSOUT data sets.

### ***User Parameter Buffer***

The JES3 PROCESS control statement allows the user to define a non-standard job. (That is, it allows design of the set of scheduler elements to be placed in the JCT entry for a job.) Each PROCESS statement may be followed by parameter information to be passed to the DSP named in the PROCESS statement. That parameter information is stored in the **user parameter buffer (PARM)**.

A user parameter buffer is also generated for called jobs if the operator enters optional parameter information with the CALL command.



### ***Job Volume Table***

The **job volume table (JVT)** contains one entry for each of the volumes assigned to jobs during MDS processing. JVT entries identify by serial number the volumes a job will require during MVS execution. The JVT is built during converter interpreter processing.

There are many JES3 facilities that have not yet been addressed. Though it is beyond the scope of this manual to describe all JES3 facilities, a few do deserve attention. This appendix examines several that you should be aware of, as they are key elements of the JES3 design.

### Deadline Scheduling

Production schedules often demand that certain applications be completed at strategic times of the day and on particular days of the week. Deadline scheduling, performed by a JES3 DSP, is designed for scheduling of jobs based upon time intervals, the time of the day, and associated job priority changes.

Your use of deadline scheduling begins with your providing one or more algorithm definitions (up to 36) in the initialization statements for JES3. Each algorithm is given a type name and contains the specifications for time intervals and priority changes. There are two parts of a deadline algorithm you must define:

- **An initial priority value.** The value may be a priority “set” value or a “change” value.
- **A lead time value.** Each job using deadline scheduling will, via a JES3 control statement parameter, identify a time of day by which the job is to be scheduled (the deadline time). The lead time value, expressed in time of day, hours or minutes, represents a time prior to the deadline time at which the set or change value (priority) is to be applied.

Other parts of a deadline algorithm are optional and will be described after an example involving only the required operands.

Suppose that your installation had defined an algorithm, say type C, as

```
DEADLINE,C=(11,1H)
```

At 1 hour before the job’s deadline time, algorithm C will cause the job’s priority to be set to 11. If you wished to set a deadline for a job via JCL, you would include a `//*MAIN` JES3 control statement as in the example below:

```
//MYJOB JOB ...,PRTY=4
//*MAIN ...,DEADLINE=(1500,C)
//STEP1 EXEC ...
```

If the job enters the system at 9:00 am but is not completed by 1400 (1 hour prior to its deadline time of 1500), its priority will be set to 11. If the job is completed prior to 1400, deadline scheduling will have no effect on the job. However, if it is not completed by 1400, the change in priority will enhance the job’s chances of being scheduled. Should the job enter the system after 1500 (its deadline), but before 2400, the deadline would be assumed to be 1500 of the next day.

The other options available for deadline scheduling are:

- The set value in the algorithm could be specified as a change value, such as +3, indicating the job's original priority is to be increased by 3.
- Additional increases and time intervals may be specified.
- A date or a day in a weekly, monthly, or yearly cycle could be included in the DEADLINE operand of the `//*MAIN` statement.

More detailed information will either be supplied by your installation (with regard to its uses of deadline scheduling) or may be found in the publication *OS/VS2 MVS JCL*.

## Dependent Job Control

Applications usually have many processing stages. These stages can be related to each other in simple or complex ways. Some stages may be optional, based upon successful or unsuccessful completion of prior stages. In prior operating system environments, the stages of processing usually had to be related through the mechanisms of job steps, since there were no job-to-job relationships. JES3 provides dependent job control (DJC) as a means of relating the jobs that make up an application.

The term **DJC network** is used to denote a set of related jobs that must run in a predetermined order. The ordering of the jobs is usually determined by data set dependencies, though there may be other reasons as well. To use dependent job control, you include a `//*NET` statement in each job's JCL. This defines the structure of the network. Figure 14 illustrates the relationships among jobs A through F. Job B cannot run until after job A completes. Jobs C and D are to be run, in that order, only after successful completion of job B. Jobs E and F are to be run, in that order, only if job B abnormally terminates.

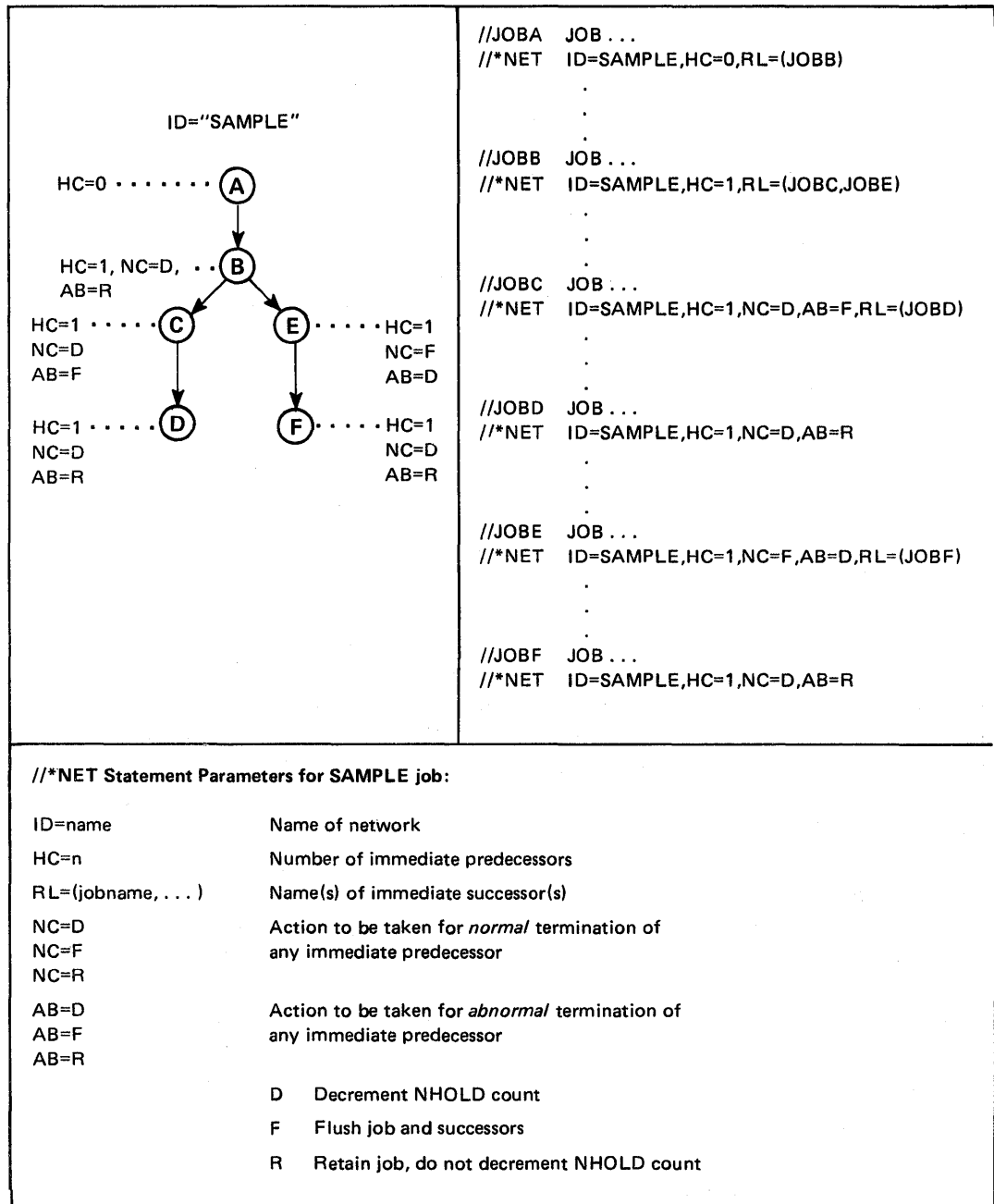


Figure 14. Dependent Job Control Network

Without becoming deeply involved in coding detail, let's examine the dependency structure for the job named SAMPLE. The `//*NET` statements indicate:

- **For JOBA:** There are no predecessors indicated since HC (hold count) is defaulted to zero. Upon completion of JOBA, JOBB (the successor to JOBA) is to be released (RL=JOBB) for normal scheduling by decreasing of its hold count.
- **For JOBB:** There is one predecessor (JOBA), indicated by HC=1. JOBB cannot proceed until its HC is reduced to zero. At completion of JOBB, hold counts for JOBC and JOBE are to be decreased (RL=(JOBC,JOBE)).
- **For JOBC:** This job's hold count is only decreased if JOBB normally completes (NC=D). If JOBB abnormally terminates, JOBC is to be flushed (AB=F) from the system; JOBD will also be flushed.
- **For JOBD:** This job runs upon successful completion of JOBC (NC=D). It is retained if JOBC abnormally terminates (AB=R).
- **For JOBE:** This job is run only if JOBB terminates abnormally (AB=D). Should JOBB run normally, JOBE is to be flushed, along with JOBF.
- **For JOBF:** This job runs upon successful completion of JOBE (NC=D). It is retained if JOBE abnormally terminates (AB=R).

Dependent job control allows jobs to be read into the system in any order and at varying times. The control blocks that are built allow the scheduling of the right job at the proper time, and waiting for missing jobs.

## Dynamic Allocation

Dynamic allocation allows you to change the devices, volumes, and data sets allocated to your jobs by JES3. You can delay specifying device, volume, and data set resources until execution of your jobs; a particularly useful feature if you will not know which resources you need until execution has begun.

To use dynamic allocation, include in your program an MVS DYNALLOC macro or an SVC 99 instruction. MVS will use the subsystem interface to forward the request to JES3. The request can be one of four types:

- Allocate a resource
- Deallocate a resource
- Concatenate a data set
- Deconcatenate a data set

Once of two JES3 DSPs will handle the request: the DYNAL DSP or the SETUP DSP. Upon completion of processing, JES3 sends an appropriate return code to MVS, and execution of the job resumes.

## JES3 Networking

A JES3 complex can be a node in a network of systems. Remaining nodes could be other JES3 complexes or entirely different systems, as long as they have compatible networking facilities. (Examples of compatible systems are specific releases of JES2 and VM/370.) Connections among nodes can be BSC communication lines or CTC adapters. JES3 treats CTC adapters like BSC communication lines.

Job processing at a node is largely unaffected by the fact that the node on which the jobs are being processed is connected to other nodes. The purpose of the network (as well as the programs that support the network) is the transfer of jobs and data among nodes. Thus, job flow in a JES3 system at a node is identical to that for a JES3 system not at a node. All JES3 initialization statements, control statements, and commands for a system not in a network apply equally to a system in a network.

The difference is that additional JES3 programs, statements, and commands are needed for routing jobs and data among nodes. For networks, JES3 has the following DSPs:

- A special consoles DSP (IATCNNJ) handles commands and messages pertaining to network operations.
- A line manager DSP (IATNTDR) performs I/O operations over communication lines.
- A rerouting DSP (IATNTRT) responds to requests for moving a job or data from the node to which it was sent to another node.
- A store-and-forward DSP (IATNTSF) decompresses data, stores it on the spool device, and then compresses the data to ready it for transmission to another node.
- A transmission DSP (IATNTSD) handles transmission details, such as checking and sending job headers, and issuing status messages to the operator.

System programmers define the structure of a network using JES3 initialization statements, including:

- An NJERMT initialization statement to identify passwords, nodes, connections among nodes, and paths JES3 is to use when transmitting and destination nodes are not adjacent.
- An NJECONS initialization statement to define a message class for network-related messages.
- The NETWORK subparameter of a CONSOLE initialization statement to identify the console to which JES3 is to send network-related messages.

Once the network has been defined, JES3 control statements submitted with jobs identify where the jobs and data are to be transmitted. There are special JES3 commands for starting, stopping, and inquiring about network activity.

## **Recovery Management**

JES3 provides recovery for:

- JES3 and MVS software failures
- Hardware failures of the global processor
- Spool volume failures

The objective of any recovery facility is minimum disruption of processing. The JES3 concept of centralized control means centralized tracking of work, and this makes it easier to localize failures.

### ***Recovery for Program Failures***

If a DSP fails, the first step JES3 takes is to automatically reinstate the DSP. Then if the failure persists, JES3 will terminate the DSP and record failure data on the SYS1.LOGREC data set. If the DSP performs a critical

JES3 function, a reinitialization of JES3 will be necessary. There will be no effect upon user jobs executing at the time of the failure. User jobs throughout the complex continue to execute normally until they require a global JES3 service, and then they are placed into an MVS wait state until global JES3 is reinitialized.

The least disruptive form of reinitialization is a JES3 **hotstart**. For a hotstart, JES3 is initialized from information stored in DASD during the last reading of the initialization statements. During a hotstart, JES3 reconstructs an environment based upon checkpoint information, checks the validity of control blocks, and restructures each of its processing queues by reading the job queue (JCT entries). Jobs active at the time of failure are reinstated at the scheduler element upon which the job was active; this is only an internal JES3 function. The active jobs are unaffected.

A JES3 hotstart may also be used if an MVS failure occurs on the global processor. In this case, however, since an MVS IPL must be performed, user jobs in execution on the global processor will be affected. These will be restarted by MVS if they were journaled. Jobs not journaled will be treated according to the JES3 failure option specified in JCL. These options include:

**Restart** To restart the job from the beginning

**Cancel** To print any output generated up to the time of the failure and to cancel the job

**Hold** To place the job in operator hold for later restart

**Print** To print any output, then hold the job for restart

Because the global JES3 address space is lost in an MVS failure, local JES3 processors must resend any outstanding requests for global service once the global processor has been hotstarted.

If a **warmstart** is necessary either because a hotstart was unsuccessful, or because a change was made to the initialization statements, an IPL must be performed for all JES3 processors. This means that all user jobs in execution will be affected. Journaled jobs will be restarted by MVS. Jobs not journaled will be treated according to the JES3 failure option specified.

Failure of a local JES3 address space is transparent to users and causes no loss of user jobs. When an operator starts the local JES3 address space, the local processor is reconnected to the global processor.

A **local start** can also be used if MVS fails on a local processor. But because an MVS IPL must be performed, user jobs in that processor must be restarted via either an MVS checkpoint restart or a JES3 failure option specification.

### ***Recovery for Global Processor Failures***

The global JES3 address space is the heart of a JES3 complex. A hardware failure in the global processor is more serious than the loss of a local processor. The reaction to such a global failure is transfer of the JES3 global capability to another processor in a way that would avoid loss of the JES3 complex. This procedure is called **dynamic system interchange (DSI)**.

DSI must be planned thoroughly during the design of a JES3 installation. This is because:

- The global capability must be moved to another processor.
- No initialization deck is processed. During initialization of the original global processor, local processors that are potential global processors must be identified. This allows CTC paths to be specified to other locals from the new global processor.

All, or major subset, of the devices on the old global processor must be accessible by the new global processor. This accessibility usually involves switching of unit record devices, some magnetic tapes, and communication lines used for remote job processing.

The DSI process is begun and controlled by operations personnel, and it is a multi-step transition of global capability. The local (to become global) JES3 address space is quiesced without loss of jobs, necessary devices are switched to the new global, and a hotstart is simulated. This results in a new global processor. Though there is a loss of computing capacity, the complex can continue to supply JES3 services to its users.

When the original global is repaired, two options are available for using it. It may be local-started and simply run as a local processor or it may be reestablished as a global processor. If the latter option is chosen, the normal course of events is to reestablish the repaired processor as a local, then to use DSI to give it global capability. The interim global could then be local-started to return the complex to its normal operating status.

### ***Recovery for Spool Failures***

If defective tracks cause intermittent spool I/O errors, you can use JES3 commands to identify the affected spool data set and to stop JES3 from using the data set or its tracks.

For permanent I/O errors, you can:

- Have JES3 suspend scheduling of jobs that would use the affected data set or tracks
- Do a warm start and replace the affected spool data set (do this for problems you cannot quickly repair, such as a head crash)

### **Remote Job Processing**

Remote job processing (RJP) allows you to enter a job into the JES3 job stream via an input source connected to the global processor by a data link.

Devices connected directly to processor channels are called **local devices**. Devices connected to channels via data link are called **remote devices**. Jobs introduced at remote devices are transmitted to the host over communication lines or local-channel attachments. The jobs are processed as if they had been submitted locally. Any output from a remotely entered job can, at your option, be retained at the host system, transmitted to the originating location, or sent to another location.

Two types of remote devices can be used for RJP: (1) those attached using **binary synchronous communications (BSC)** protocols and (2) those attached using **synchronous data link control (SDLC)** protocols within the **IBM systems network architecture (SNA)**. The remote devices using BSC protocols are managed by the RJP DSP; those using SDLC protocols are managed by the SNARJP DSP.



With BSC RJP, JES3 uses its own line manager, the Remote, Terminal Access Method (RTAM) to manage line protocols. With SNA RJP, line protocols are managed by either the:

- Virtual Telecommunication Access Method (VTAM), or
- Telecommunication Access Method (TCAM) with application program interface (API) support.

BSC RJP is started when the operator at the host issues an \*CALL,RJP command, SNA RJP is started when the operator at the host specifies an \*CALL,SNARJP command. A JCT entry containing two scheduler elements is always created. For BSC RJP, the two scheduler elements are RJP and PURGE. For SNA RJP, the two scheduler elements are SNARJP and PURGE. Subsequent processing continues in the normal called job manner as described in Chapter 2.

After an operator starts RJP, the operator at the remote device must sign on, via the /\*SIGNON statement for BSC RJP, or via the LOGON command for SNA RJP. Jobs can then be introduced into the JES3 job stream from the remote devices. The system operator must be the one to terminate, via the \*CANCEL command, BSC RJP and SNA RJP when processing is complete.

## Spool Partitions

A spool partition is a named collection of spool data sets. Partitions give system programmers a way of specifying groups they want JES3 to use in organizing spooled data. One advantage to partitions is that if there is a failure of a spool device, most likely the failure will affect a single partition, and JES3 can readily isolate the affected jobs. Likewise, JES3 can identify unaffected partitions and continue processing the jobs having unaffected data. Another advantage to partitions is that JES3 can reduce spool contention. That is, JES3 can arrange partitions on spool devices for shorter access time and efficient use of space.

System programmers identify spool partitions on JES3 initialization statements. In the absence of system-programmer specified partitions, JES3 will create a single partition for all spooled data. There are three ways that a system programmer can group data to assign it to a spool partition. He or she can group:

- All spooled data for all jobs running on a specific processor
- All input and output data for jobs of a specific job class
- All output data for a specific system output class

Operators can use JES3 commands to find out what partitions exist, how much free space remains for a partition, and for changing partition assignments.

## Glossary

This glossary defines JES3 terms and other data processing terms used in this publication. For definitions of terms not included in this glossary, see *IBM Data Processing Glossary*, GC20-1699.

**ATCB:** auxiliary task control block.

**ATDE:** auxiliary task dispatching element.

**auxiliary task:** A subtask under the JES3 primary task. Writer DSPs and the General Services DSP do some of their processing under this task.

**auxiliary task control block:** A control block that JES3 uses to manage work done under the auxiliary task.

**auxiliary task dispatching element:** A control block that JES3 uses to determine whether to dispatch an FCT under the JES3 auxiliary task.

**binary synchronous communication (BSC):** (1)Communication using binary synchronous transmission. (2)A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

**BSC:** Binary synchronous communication.

**called job:** A job created by JES3 in response to a JES3 CALL command

**channel-to-channel (CTC) adapter:** A device for connecting two channels on the same processor or on different processors.

**cold start:** For JES3, the first start after system generation and after some unrecoverable failures. Spool data sets are initialized during a cold start.

**common area:** In MVS, an area of virtual storage that is addressable by all address spaces.

**common service area (CSA):** In MVS, a part of the common area that contains data areas accessible from all address spaces.

**console service:** A DSP that performs traffic management for consoles.

**converter interpreter (CI) DSP:** A DSP that uses MVS converter/interpreter subroutines to process JCL statements. The CI DSP creates internal JCL text for jobs being readied for MVS execution.

**CTC:** Channel-to-channel.

**DASD:** Direct access storage device.

**data link:** The physical connection and the connection protocols between a host and a communication controller nodes via the host data channel.

**DDR:** Dynamic device reconfiguration.

**demand select job:** A job created by MVS and passed to JES3 for processing. MVS creates demand select jobs in response to MVS START or MOUNT commands or the TSO LOGON command. (For processing of these commands, system resources are needed, hence JCL is used to define those resources. It is this JCL that JES3 processes.)

**dependent job control (DJC):** A method of handling multiple jobs that must be executed in a specific order because of job dependencies.

**destination codes:** For JES3, numeric codes used to represent information during communication between JES3 components on different processors via the subsystem interface.

**destination queue (DSQ):** For JES3, a control block used by subsystem interface routines to route requests (represented by destination codes) to the JES3 routines responsible for servicing the requests.

**DJC:** Dependent job control.

**DJC network:** A set of jobs that JES3 must run in user-defined order. Success or failure of one job can cause execution, holding, or cancellation of other jobs.

**DSP:** Dynamic support program.

**dynamic allocation:** For JES3, assignment of system resources to a job while it is executing rather than before it is executed.

**dynamic support program (DSP):** A JES3 routine that accomplishes the work defined by a scheduler element. One or more DSPs may be executed for a single scheduler element.

**dynamic writer:** An output writer that is started automatically by JES3 when there is system output to be processed and a device is available.

**external writer:** An MVS routine that directs system output to unsupported devices such as unit record printers and punches, magnetic tape devices, DASD, and plotters. External writers must be started by the operator as required. Once started, an external writer requests output data sets from the JES3 output service DSP via the subsystem interface.

**FCT:** Function control table.

**function codes:** Numeric codes used by MVS when requesting a service or control information from JES3 via the subsystem interface.

**function control table (FCT):** The master dispatching queue for JES3. Entries in the FCT are arranged in priority order and each represents a DSP to be dispatched.

**global processor:** The processor that controls job scheduling and device allocation for a complex of processors. See also **local processor**.

**high watermark setup:** A type of setup where the minimum number of devices of each type needed is reserved for a job step.

**hot start:** A restart of the global processor using information obtained from the last set of initialization statements processed. Recovery is attempted for all jobs that were in execution at the time of the failure.

**hot start with analysis:** A special form of hot start where the JES3 job queue is examined and the operator is given the opportunity to delete any jobs that would cause another restart.

**hot writer:** An output writer that must be started and stopped by the operator. Hot writers are typically used when operator intervention is anticipated (as for changing forms, etc.).

**initialization:** For JES3, the process of reading JES3 initialization statements and creating tables and control blocks used throughout the JES3 program.

**input service driver (ISDRVR) DSP:** A DSP that reads batches of jobs from the spool data set and constructs a separate JCT entry for each job.

**input service job:** A job created by the card, tape, or reader DSP for each batch written on the spool data set. An input service job is represented by a JCT containing two scheduler elements: one for the ISDRVR DSP and one for the PURGE DSP.

**internal reader:** A JES3 routine that processes input streams contained in SYSOUT data sets obtained from MVS.

**IPL:** Initial program load.

**IS:** Input service DSP.

**JCT:** Job control table.

**JES:** Job entry subsystem.

**JES2:** A subsystem that receives jobs into the MVS system and processes output produced by the jobs. In multiple-processor complexes, the JES2 program supports independently operating processors via a common queue.

**JES3:** A subsystem that receives jobs into the MVS system, optionally schedules resources for the jobs, and processes output data produced by the jobs. In multiple-processor complexes, the JES3 program manages processors so that one processor exercises centralized control over the others and distributes jobs to the others via a common job queue.

**JES3 auxiliary address space:** An address space used exclusively by JES3 for data areas that would otherwise be placed into the CSA. Parameters in JES3 initialization statements specify whether a JES3 auxiliary address space is desired and, if so, the size of each data area.

**JES3 spool access method (JSAM):** An access method used by JES3 DSPs for managing spool space and for reading from or writing to the spool device.

**JES control table (JESCT):** A control block in the MVS nucleus that contains information used by subsystem interface routines.

**job control table (JCT):** A table into which one entry is placed for each job that JES3 is to process. Entries are arranged in the JCT in job priority order to facilitate later job selection by priority.

**job control table (JCT) entry:** A control block into which JES3 places the description of a job to be processed, and scheduler elements representing the DSPs needed to process the job.

**job queue element (JQE):** A control block containing a summary of information from a JCT entry. JQEs remain in storage and are used by JES3 instead of JCT entries for scheduling of work.

**job segment scheduler (JSS) DSP:** A DSP that scans the job control table (JCT) to locate scheduler elements eligible for processing, and then builds function control table (FCT) entries so the corresponding DSPs can be dispatched. JSS itself is represented by an FCT entry.

**job summary table (JST):** A table into which the converter interpreter DSP places job setup requirements.

**job volume table (JVT):** A table into which the converter interpreter DSP places the volume information it obtains from DD statements.

**JQE:** Job queue element.

**JSAM:** JES3 spool access method.

**JSS:** Job segment scheduler.

**JST:** Job summary table.

**local device:** A device attached to a host processor via a channel.

**local processor:** In a complex of processors under control of JES3, a processor connected to the global processor by a CTC adapter, for which JES3 performs centralized job input, job scheduling and job output services via the global processor.

**local start:** A restart of a local processor. Initialization is unnecessary and user jobs are unaffected.

**loosely-coupled multiprocessing:** In System/370, multiprocessing in which two or more processing units share access to direct access storage or are coupled by means of channel-to-channel adapters for passing of control information.

**main DSP:** A DSP that chooses jobs and supplies them to the MVS initiator(s).

**main processor:** Any processor on which jobs can execute. The two types of main processors are (1) global main, and (2) local main.

**MDS:** Main device scheduler.

**MSS:** Mass storage system.

**multifunction monitor (MFM):** The master dispatcher for JES3. The MFM scans the function control table (FCT) for DSPs ready to be executed, and causes execution to begin.

**multiprocessing system:** A computing system employing two or more interconnected processing units to execute program simultaneously.

**MVS:** Multiple virtual storage.

**network:** For JES3, two or more systems and the connections over which jobs and data are distributed to the systems. One or more of the systems can be a JES3 global processor (and its local processors, if any). The other systems can be non-JES3 systems with compatible networking facilities. Connections can be leased BSC communication lines or CTC adapters. (Do not confuse with DJC network.)

**node:** For JES3, one of the systems of a network.

**non-standard job:** A job for which JES3 defines processing from input received on // \*PROCESS control statements.

**normal job:** A job received by JES3 in an input stream. Normal jobs can be standard jobs or nonstandard jobs. Contrast with "called job."

**OSE:** Output service element.

**output service element (OSE):** A control block that describes the characteristics of one or more output data sets of the same job.

**output service (OUTSERV) DSP:** A DSP that schedules output writers for printers or punches, and routes output data to TSO processor, MVS external writers, and the MVS internal reader.

**output writer:** A JES3 routine that transcribes output data sets to printer or punch system output devices. See **dynamic writer**, **hot writer**.

**primary subsystem communication vector table (SSCVT):** A control block in the common storage area that contains information used by the subsystem interface routines.

**primary task:** The task under which most DSPs execute.

**purge DSP:** A DSP that performs post-execution removal of a job from the system, writes system management facilities (SMF) records, and frees spool space used by the job.

**reader DSP:** A DSP that transfers a job's control statements and SYSIN data from an input device to the spool data set. Three types of readers exist: card reader, tape reader, and disk reader.

**reader job:** A called job created by JES3 each time the operator issues a CALL command for a card, tape, or disk reader.

**remote device:** A device attached to a host processor via a data link.

**remote job processing (RJP):** A facility for input and output of jobs via remote JES3 terminals.

**RJP:** Remote job processing.

**scheduler element:** A part of a job control table (JCT) entry. (Each JCT entry may contain multiple scheduler elements.) Each scheduler element represents one or more DSPs needed for JES3 processing of a job.

**SDM:** Spool data management.

**setup DSP:** A DSP that performs volume fetch, job setup, high watermark setup, and explicit setup functions. The setup DSP is optional and additionally the job and high watermark setup options are separately selectable for DASD, tape, and MSS virtual devices.

**SMF:** System management facilities.

**SNA:** Systems network architecture.

**spool:** Simultaneous peripheral operations online.

**spool data management:** For JES3, the recording and retrieval of data on the spool data set and the management of space within the spool data set.

**spool partition:** A named collection of spool data sets.

**staging area:** An area into which subsystem interface routines store data to be transferred between address spaces. Staging areas can be contained in the common service area (CSA), or in an optional JES3 auxiliary address space. The staging areas are accessible from all address spaces.

**Standard job:** A job for which JES3 defines needed processing entirely from the job's JCL statements.

**subsystem identification block (SSIB):** The control block into which MVS places the name of the subsystem to which it is directing a request over the subsystem interface.

**subsystem options block (SSOB):** The control block into which MVS places a function code when communicating with JES3 over the subsystem interface. The function code identifies a requested service.

**subsystem services common services:** A term used to collectively identify JES3 routines that handle communication among JES3 modules running on separate processors. (For example, a subsystem interface service routine and a receiving DSP would be referred to as subsystem interface common services.)

**system management facilities (SMF):** An optional control program feature that provides the means for gathering and recording information that can be used to evaluate system usage.

**systems network architecture (SNA):** The total description of the logical structure, formats, protocols, and operational sequences for transmitting information units through a communication system.

**TCAM:** Telecommunication access method.

**USAM:** User spool access method.

**user spool access method (USAM):** An access method used by programs in user address spaces for managing spool space and for reading from or writing to the spool device.

**VTAM:** Virtual telecommunications access method.

**warm start:** For JES3, a restart where an IPL must be performed on all processors and there is a choice of using the last set of initialization statements processed or a new set of initialization statements.

**warm start with analysis:** For JES3, a special form of warm start where the JES3 job queue is examined and any jobs that would cause another restart are automatically deleted.

**writer:** See output writer.

| **writer output multitasking:** A facility that allows a JES3 output  
| writer to do part of its work in parallel with other JES3 functions  
| on a tightly-coupled global processor.



## Index

\*CALL,RJP command A-8  
\*CALL,SNARJP command A-8  
\*CANCEL command A-8

//\*MAIN statement A-1,A-2  
//\*NET statement A-2,A-4  
/\*SIGNON statement A-8

### A

ack type request 3-5,3-6  
address space communication 1-2,3-1 to 3-7  
allocated resources, freeing 2-7  
allocation  
    devices, volumes, data sets 2-7  
    phase of setup processing 2-8  
    shared-spool devices 3-2  
    use of IEFSSREQ macro for 4-5  
ATCB  
    defined 2-14, GL-1  
    purpose of 2-14  
ATDE  
    defined 2-14, GL-1  
    purpose of 2-14  
auxiliary address space  
    definition of GL-2  
    in storage layout 3-1  
    protected buffer pool in 4-5,4-4  
    staging area in 3-6  
auxiliary task control block (see ATCB)  
auxiliary task dispatching element (see ATDE)  
auxiliary task wait FCT 2-14

### B

basic sequential access method (BSAM) 4-3  
batches of jobs 2-3,2-4,2-5  
binary synchronous communication (BSC)  
    definition of GL-1  
    for networking A-4  
    use of A-7  
blocked records 4-1  
BSAM (basic sequential access method) 4-3  
BSC (see binary synchronous communication)  
buffer pool  
    JESIO 4-3,5-2  
    protected 4-3,4-4,4-5  
    user storage 4-3  
buffers, spool I/O 4-3

### C

call command  
    generating of called jobs 1-7  
    optional parameter information for 5-5  
    reading of input stream 2-1,2-15  
    specifying batches using 2-3

called job  
    definition of GL-1  
    exception for input service 5-1  
    generation of 1-7  
    queueing priority for 2-1  
    scheduler elements for 1-7,2-1,5-2  
cancel failure option A-6  
card reader 1-7,2-1  
card reader (CR) DSP  
    dispatching of 2-2  
    execution of 2-3,2-4,2-15  
    scheduler element for 2-1,2-2,2-15  
card reader job 2-1 to 2-4  
channel-to-channel adapter  
    as communication path 1-2  
    definition of GL-1  
    for networking A-4  
    STARTIO macro for 3-6  
character set image 2-12  
CI (converter interpreter) DSP  
    completion of JST by 5-5  
    creating FCT entry for 2-4  
    definition of GL-1  
    dispatching of 2-6  
    processing performed by 2-6  
    scheduler element for 1-7,2-5  
classes, job 2-10  
codes  
    destination 3-3,3-7  
    function 3-3,3-4,3-7,GL-1  
cold start GL-1  
comm type request 3-5  
commands  
    (see also CALL command)  
    \*CALL,RJP A-8  
    \*CALL,SNARJP A-8  
    \*CANCEL A-8  
    for networking A-5  
    LOGON 1-7,A-8  
    MOUNT 1-7  
    START 1-7  
common area 3-2,GL-1  
common service area (CSA)  
    definition of GL-1  
    DSSs in 4-5  
    location of 3-1  
    protected buffer pool in 4-4,4-5  
    SSI control blocks in 3-4  
    staging area in 3-6  
common services routine  
    poster 3-5,3-6  
    router 3-5  
communication  
    address space 1-2,3-1 to 3-7  
    binary synchronous (BSC) A-7,GL-1  
    BSAM (basic sequential access method) 4-3  
    lines A-7  
    overview of 3-3  
    SDLC (synchronous data link control) A-7  
    types of 3-3  
compatibility interface 4-2,4-5,4-6  
console messages 3-2

- console service
  - consoles DSP for 2-1,2-15
  - definition of GL-1
- consoles, function codes for 3-3
- consoles DSP 2-1,2-15,A-5
- control blocks
  - freeing spool space for 2-14.2
  - job related 5-1 to 5-6
  - SSI related 3-4
- control statements
  - JES3
    - for networking A-5
    - for non-standard jobs 1-7
    - FORMAT 5-5
    - influence upon JCT entries 2-5
    - overriding destination names 2-11
    - PROCESS 5-5
- controller 1-2
- converter interpreter (CI) DSP
  - completion of JST by 5-5
  - creating FCT entry for 2-4
  - definition of GL-1
  - dispatching of 2-6
  - processing by 2-6
  - scheduler element for 1-7,2-5
- converter/interpreter, MVS 2-6
- CR (card reader) DSP
  - dispatching of 2-2
  - execution of 2-3,2-4,2-15
  - scheduler element for 2-1,2-2,2-15
- CSA (see common service area)
- CTC (channel-to-channel) paths A-4,A-7

## D

- DASD (direct access storage device)
  - jobs entered from 1-7
  - setup processing for 2-8
  - spool volumes 4-1
  - track groups on 4-2,4-3,5-1
  - transfer of batches to 2-3
- data link
  - definition of GL-1
  - job entry via A-7
- data set and device characteristics
  - character set image 2-12
  - destination 2-11
  - device type 2-11
  - FCB/carriage tape 2-11
  - forms 2-11
  - forms flash cartridge 2-12
  - line limit 2-12
  - matching of 2-11 to 2-12
  - priority 2-11
  - train 2-11
- data set block (DSB) 4-5
- data set status (DSS) block
  - construction of 4-5
  - freeing of 4-5,4-6
  - location in storage 4-4

- data sets
  - (see also SYSIN, SYSOUT (data sets))
  - allocation of 2-7
  - for internal reader 2-12
  - for TSO users 2-12
  - for unsupported devices 2-12
  - message
    - freeing spool space for 2-14.2
    - in JDS 5-4
- DD statements 2-5
- deadline scheduling A-1
- deallocation
  - early resource release 2-9,2-10
  - of JES3-managed devices 2-10
  - of shared-spool devices 3-2
- demand select job 1-7,GL-1
- dependent job control (DJC) A-2,GL-1
- destination codes 3-3,3-7,GL-1
- destination names, overriding 2-11
- destination queue (DSQ)
  - definition of GL-1
  - use of 3-6
- destination routing table 3-6
- destinations, symbolic names for 2-11
- devices
  - allocation of 2-7
  - breakdown processing for 2-10
  - characteristics of 2-11
  - DASD 2-8
  - deallocation of 2-10,3-2
  - dynamic allocation of 2-10,A-4
  - dynamic device reconfiguration (DDR) 3-3
  - external writer 2-12
  - graphic 2-8
  - group names for 2-11
  - JES3-managed 2-10
  - locally attached 1-7,A-7
  - MSS virtual units 2-8
  - print type 2-11
  - punch type 2-11
  - remote 2-11,A-7
  - setup options for 2-8
  - shared-spool 3-2
  - sys type 2-11
  - tape 1-7,2-8
  - type characteristic for 2-11
  - unit record 2-8
- direct access device (see DASD)
- dispatching
  - description of 1-4,1-5
  - of reader job 2-2
  - queue 1-4,2-15
- DJC (dependent job control) A-2 to A-4,GL-1
- driver module of CR DSP 2-3
- DSB (data set block) 4-5
- DSI (dynamic system interchange) A-6
- DSP (dynamic support program)
  - definition of GL-1
  - dispatching of 1-4,1-5
  - execution of 1-4,2-3
  - failure of A-5
  - job data for 5-4
  - messages generated by 5-4
  - parameter information passed to 5-5

## DSPs

### types of

see entries for:

- CI (converter interpreter)
- CONSOLES
- CR (card reader)
- DYNAL (dynamic allocation)
- general services
- ISDRVR (input service driver)
- JSS (job segment scheduler)
- MAIN (generalized main scheduling)
- networking
- OUTSERV (output service)
- PURGE
- RJP (remote job processing)
- SETUP
- SNARJP (systems network architecture remote job processing)
- VERIFY
- WTDDRVR (work-to-do driver)

### DSQ (destination queue)

definition of GL-1

use of 3-6

### DYNAL (dynamic allocation) DSP A-4

### DYNALLOC macro A-4

### dynamic allocation

definition of GL-1

JES3-managed devices 2-10

purpose A-4

### dynamic device reconfiguration (DDR) function codes 3-3

### dynamic support program (see DSP)

### dynamic system interchange A-6

### dynamic writer

definition of GL-1

scheduling of 2-12

## E

early resource release 2-9,2-10

end of memory type (EMOT) request 3-6

### error recovery

cancel failure option for A-6

for DSPs A-5

for global processor A-5 to A-7

for hardware A-5,A-6

for local processor A-6

for programs

for spool failures

hold failure option for A-6

print failure option for A-6

restart failure option for A-6

software A-5

### EXEC statements 1-2

### execution

DSPs 1-4,2-3

### job

completion of 2-10

supervision of 1-1

on main processor 1-2

### exit routine for

changing scheduler elements 1-7

SSI 3-5

### explicit setup 2-8

### external writer

data sets for 2-11,2-12

definition of GL-1

## F

failures (see error recovery)

FCB/carriage tape 2-11

FCT (function control table)

building of entries 1-4,2-5

definition of GL-1

permanent entries 1-5,2-7

relation to other blocks 5-4

removal of entries 2-2

use for dispatching 1-5,2-3,2-15

use for job scheduling 1-4,2-1

final setup processing 2-10

FORMAT control statement 5-5

format parameter buffer (FRP) 5-4,5-5

forms, printer characteristic 2-11

forms flash cartridge 2-12

FRP (format parameter buffer) 5-4,5-5

function code matrix 3-4

### function codes

definition of GL-1

for consoles 3-3

handling of 3-3

invalid 3-5

purpose of 3-3,3-7

SYSIN 3-3

SYSOUT 3-3

function control table (see FCT)

function routines 3-5,3-7

## G

general services DSP 2-14

generalized main scheduling (MAIN) DSP

definition of GL-2

processing by 2-9

scheduler element for 1-7,2-5,2-7

GET macro 4-5

### global processor

address space communication 3-2

definition of GL-1

failure on A-5 to A-7

role as controller 1-2

graphic devices 2-8

## H

### hardware failure on

global processor A-5 to A-7

local processor A-6

high-watermark setup 2-8,GL-1

hold failure option A-6

### hot writer

definition of GL-1

purpose of 2-12

### hotstart

for program failures A-6,GL-1

with analysis GL-1



## I

IEFJSREQ routine 3-5,3-7  
IEFSSREQ macro  
    invalid function codes 3-5  
    request for service 3-6,3-7  
    response 3-7  
    used by initiator 3-5  
    used for MVS allocation 4-5  
    used for opening spool 4-5,4-6  
initial setup processing  
    JST entries used during 5-5  
    phases of 2-8  
initialization  
    data on spool volumes 4-1  
    definition of GL-1  
    for deadline scheduling A-1  
    for dynamic system interchange A-7  
    for hotstart A-6  
    for networks A-5  
    for partitions A-8  
    for warmstart A-6  
    JCT allocation for 5-1  
    naming devices for 2-11  
    specifying buffers for 4-3  
    specifying number of tracks for 4-2  
    staging area creation for 3-6  
initiator  
    communication, address space 3-2  
    receipt of jobs by 2-10  
    request for jobs by 3-5,3-7  
    selection of jobs for 2-7,2-9  
    use of IEFSSREQ macro by 3-5  
input service  
    FRP (format parameter buffer) construction 5-5  
    JDS (job data sets block) construction 5-4  
    JMR (job management record) construction 5-5  
    job 2-4,2-5,2-15,GL-2  
    job entry via 5-1  
    JST (job summary table) construction 5-5  
input service driver (ISDRVR) DSP 2-3,2-5,GL-1  
input stream  
    JCT entries for jobs in 2-1  
    preparing jobs from 1-2,1-3,2-15  
    reading of 1-4,2-1,2-15  
    separating jobs into batches 2-3,2-4  
input/output (see I/O)  
input/output supervisor (IOS) 4-3  
internal reader  
    definition of GL-2  
    job streams for 1-7,2-12  
I/O (input/output)  
    buffers, location of 4-4  
    completions 4-2  
    dispatching during 2-3  
    errors, spool I/O 1-7  
    rates, jobs in execution 2-10  
    requirements, storing of 2-6  
    resources 1-3  
    spool operations for 4-3  
IOS (input/output supervisor) 4-3  
IPL (initial program load)  
    JES3 A-6  
    MVS A-6  
ISDRVR (input service driver) DSP 2-3,2-5,GL-1

## J

JBTAT (job track allocation table) 5-4,5-5  
JCL  
    converting for MVS 1-3,3-1  
    DD statements 2-5  
    errors 2-6  
    examination for control statements 2-5  
    EXEC statements 1-2  
    failure option, specifying A-6  
    for dependent job control A-2 to A-4  
    freeing spool space for 2-14.2  
    in JDS 5-4  
    JOB statement 1-2  
    jobs without 1-2,1-7  
    packing of multiple records 4-1  
    scheduler element for reading 1-4  
JCT (job control table)  
    (see also JCT entry)  
    definition of GL-2  
    priority level 15 2-1,2-3  
JCT (job control table) entry  
    content 1-3  
    data set for 5-1  
    definition of GL-2  
    for batches 2-3,2-4  
    for defining JES3 job 1-2,2-15,5-2  
    for input service job 2-4,2-15  
    for reader job 2-12,2-15  
    for standard jobs 2-5  
    influence of control statements on 2-5  
    relationship to JDAB 5-4  
JDAB (job description and accounting block)  
JDS (job data sets block) 5-4,5-5  
JES (job entry subsystem) 1-1  
JES control table (JESCT)  
    definition of GL-2  
    pointer to subsystem 3-4  
JESIO buffer pool 4-3,5-2  
JES2 (job entry subsystem 2)  
    component of MVS 1-1  
    definition of GL-2  
JES3 (job entry subsystem 3)  
    address space 2-6  
    (see also auxiliary address space)  
    component of MVS 1-1  
    definition of GL-2  
    destination codes 3-3,3-7,GL-1  
    environment 1-2  
    for non-standard jobs 1-7  
    FORMAT 5-5  
    influence upon JCT entries 2-5  
    initialization 2-6,3-4  
    IPL A-6  
    JES3 control statements  
    job flow 2-1 to 2-15  
    jobs 1-2,1-3,2-15  
    networking 1-7,A-4  
    overriding destination names 2-11  
    PROCESS 5-5  
    spool access method (JSAM) 4-1,4-3,GL-2  
JES3 auxiliary task  
    defined 2-13, GL-1  
    writer execution under the 2-13  
JES3 primary task  
    defined 2-13, GL-2  
    writer execution under the 2-13  
JMR (job management record) 5-4,5-5

## job

- accounting 2-13,5-5
- batches 2-4,2-5
- called (see called job)
- classes 2-10
- control blocks 5-1 to 5-6
- demand select 1-7,GL-1
- entry via data link A-7
- execution 1-1,1-2,2-10
- in network A-5
- management 1-1
- MVS and JES3 1-3
- non-standard 1-7
- normal 1-7,2-5,5-2,GL-2
- priority levels 2-1,2-5
- processing concepts 1-2 to 1-6
- queue 1-1,2-5
- reader 2-1 to 2-4,2-15
- scheduling 1-4
- selection for initiator 2-7,2-9,2-10
- setup 2-8
- spool space 5-1,A-8
- statement 1-2
- steps 1-2
- streams 1-2
- throughput 2-7
- types of 1-6

job control language (see JCL)

job control table (see JCT)

job data sets block (JDS) 5-4

job description and accounting block (JDAB) 5-4

job entry subsystem 2 (see JES2)

job entry subsystem 3 (see JES3)

job management record (MFR) 2-14,5-4,5-5

job queue element (JQE) 5-2,5-3

job segment scheduler DSP (see JSS DSP)

job summary table (JST) 5-4,5-5,GL-2

job track allocation table (JBAT) 5-4,5-5

job volume table (JVT) 5-4,5-5,GL-2

JQE (job queue element) 5-2,5-3

JSAM (JES3 spool access method) 4-1,4-3,GL-2

JSERV macro

- used for
  - response to SSISERV macro 4-6
  - routing response to requestor 3-7
  - sending information to initiator 3-7

JSS (job segment scheduler) DSP

- activities after CI DSP executed 2-6
- activities after OUTSERV DSP executed 2-13
- definition of GL-2
- making dispatchable 2-2
- role in dispatching 1-6,2-15
- scheduling a DSP 2-4,2-5,2-7,2-15
- scheduling of 1-5
- selection of scheduler elements 1-4

JST (job summary table) 5-4,5-5

JVT (job volume table) 5-4,5-5,GL-2

## L

- lead time value for deadline scheduling A-1
- line limit device characteristic 2-12
- line manager DSP A-5
- link pack area, pageable 3-1,3-4,3-5,4-4

## local device

- definition of GL-2
- entry from, for normal jobs 1-7
- for remote job processing A-7

local JES3 3-2

local processor

- address space communication 3-2
- definition of GL-2
- information for initialization 4-1
- job execution on 1-2

local start

- definition of GL-2
- for MVS failure A-6

local-channel attachments A-7

LOGON command 1-7,A-8

loosely-coupled multiprocessing GL-2

## M

macros

- GET 4-5
- IEFSSREQ (see IEFSSREQ macro)
- JSERV (see JSERV macro)
- PUT 4-5
- READ 4-5
- SSISERV (see SSISERV macro)
- STARTIO (see STARTIO macro)
- WRITE 4-5

magnetic tape devices 1-7,2-8

MAIN (generalized main scheduling) DSP

- definition of GL-2
- processing by 2-9
- scheduler element for 1-7,2-5,2-7
- selection of job for initiator 3-7

main processor 1-2,GL-2

main service

- processing 2-7 to 2-10
- scheduler element 1-7,2-5

master subsystem 3-4

messages

- console 3-2
- data sets for 2-14,5-4
- freeing spool space for 2-14.2
- generated by DSPs 5-4
- JDS, in 5-4
- system generated 5-4

MF (multifunction monitor)

- definition of GL-2
- dispatching of
  - CONSOLES DSP 2-1
  - CR DSP 2-3
  - ISDRVR DSP 2-5
  - JSS DSP 2-1,2-2,2-4
  - MAIN DSP 3-7
- role in dispatching 1-6,2-3,2-15
- scanning of FCT 1-5

MOUNT command 1-7

MRFs (multirecord files) 4-1,4-3

MSS virtual units 2-8

multiprocessing system GL-2

multirecord files (MRFs) 4-1,4-3

MVS

- (see also initiator, nucleus)
- allocation routines 4-5
- converter/interpreter 2-6
- internal reader 1-7,2-12,GL-1

IPL A-6  
jobs 1-3,3-2  
service request 1-2,3-6

## N

### network

definition of GL-2  
DJC A-2

### networking

commands for A-5  
control statements for A-5  
description of A-4  
DSPs A-5  
initialization for A-5  
job processing with A-5  
nodes A-4,A-5  
over BSC lines A-4  
over CTC adapters A-4

### networking DPSs

consoles A-5  
line manager A-5  
rerouting A-5  
store-and-forward A-5  
transmission A-5

### node

definition of GL-2  
JES3 complex at A-4  
job processing at A-5

### non-standard job

control statements for 1-7  
definition of GL-2  
number of scheduler elements for 5-2

### normal job

definition of GL-2  
entry sources 1-7  
JCT entries for 2-5  
number of scheduler elements 5-2

### nucleus

in storage layout 3-1,4-4  
JESCT in 3-4

## O

### operator control 1-1

OSE (output service element) 5-4,5-5,GL-2

output, scheduling of 2-12

output queue 5-5

### output service

OSes for 5-5  
processing 2-10  
scheduler element 1-7,2-5  
unsupported devices 2-12

output service element (SE) 5-4,5-5,GL-2

### output writer

scheduling work for 2-11,2-12  
types of 2-12  
writing of output 2-13

### OUTSERV (output service) DSP

definition of GL-2  
dispatching of 2-9  
processing by 2-10 to 2-13  
scheduler element for 1-7,2-5  
scheduling of 2-10

## P

### pageable link pack area (PLPA)

SSI request routine in 3-4,3-5  
storage layout 3-1,4-4

PARM (user parameter buffer) 5-4,5-5

partitions, spool A-8

PLPA (see pageable link pack area)

poster, common services 3-5,3-6

preexecution setup of devices 2-5,3-2

primary subsystem communication vector table (SSCVT)  
3-4,GL-2

print failure option A-6

printer train characteristics 2-11

### priority

data set 2-11  
level 15, in JCT 2-3  
levels, job 2-1,2-5

### PROCESS control statement

for non-standard job 5-5  
parameter information in 5-5

### processor

#### global

address space communication 3-2  
definition of GL-1  
failure on A-5 to A-7  
role as controller 1-2

#### local

address space communication 3-2  
definition of GL-2  
failure on A-6  
information for initialization 4-1  
job execution on 1-2  
main 1-2,GL-2

protected buffer pool 4-3,4-4,4-5

### PURGE DSP

definition of GL-2  
execution of 2-3,2-4  
processing by 2-14.1  
scheduler element for 1-7, 2-1, 2-5, A-8

purge processing 2-14.1 to 2-14.2, 5-5

purge type requests 3-6

PUT macro 4-5

## Q

QSAM (queued sequential access method) 4-3

### QUEUES

#### types of

destination 3-6,GL-1  
job 1-2,2-5  
1-4,2-15

## R

read end subroutine 3-6

READ macro 4-5

### reader

card 1-7,2-1  
internal  
definition of GL-1  
job streams for 1-7,2-12

reader DSP  
  (see also card reader DSP)  
  definition of GL-3  
  function of 2-15

reader job  
  construction of 2-1  
  definition of GL-3  
  dispatching of 2-2  
  execution of 2-3,2-15  
  scheduling of 2-1

records  
  blocked 4-1  
  packing of 4-1  
  SMF 2-14,2-14,5-5

recovery  
  failure options A-6  
  global processor failures A-5,A-6  
  hardware failures A-5,A-6  
  local processor A-6  
  management A-4  
  program failures A-5

reinitialization, JES3 A-6

remote device  
  definition of GL-3  
  jobs from A-7,A-8  
  normal jobs, entry from 1-7  
  using symbolic names for 2-11

remote job processing (RJP)  
  definition of GL-3  
  description of A-7  
  devices for 1-7  
  DSP A-7

remote work station 2-11

reply type requests 3-6

request routine 3-5,3-7

rerouting DSP A-5

resident queue element 5-2,5-3

resource management  
  benefits of 1-1  
  function codes 3-3

resource utilization 2-7

resources  
  early release of 2-9,2-10  
  freeing of allocated 2-7

resp type requests 3-6

RESQUEUE (resident queue element) 5-2,5-4

restart failure option A-6

RJP (remote job processing)  
  definition of GL-2  
  description of A-7  
  devices for 1-7  
  DSP A-7

router, common services 3-5

RQ (resident queue element) 5-2,5-4

## S

SCHEDULE macro  
  issued by router 3-6  
  service request block for 3-6

scheduler elements  
  (see also JCT entry)  
  definition of GL-3  
  for called jobs 1-7,2-1,5-2  
  for processing jobs 1-3  
  for standard jobs 1-7

life of RQ, relationship to 5-2  
  types of 1-3,1-4

scheduler work area (SWA) control blocks  
  freeing of spool space for 2-14.2  
  in JDS 5-5  
  on spool 2-6  
  track groups for 4-2

scheduling  
  deadline A-1  
  function codes for 3-3  
  jobs  
    in controller 1-2  
    use of scheduler elements 1-4  
  of reader job 2-1  
  SYSOUT data sets 2-10 to 2-13

SDLC (synchronous data link control) A-7

SDM (spool data management) 4-1 to 4-6,GL-3

SEL (service entrance list) 3-5,3-6

service request facility, MVS 1-2,3-6

setup, preexecution 2-5,3-2

setup characteristics, writers 2-11

SETUP DSP  
  definition of GL-3  
  final processing 2-10  
  for dynamic allocation A-4  
  initial processing 2-8  
  scheduler element for 2-7  
  work performed by 2-7

setup options  
  explicit 2-8  
  high-watermark 2-8  
  job 2-8

setup processing  
  final 2-10  
  initial 2-8

single-record files (SFRs)  
  data area for 4-3  
  for control blocks 4-1  
  on spool data set 5-1

single-track table (STT) 5-1

SMF (system management facilities)  
  definition of GL-3  
  type 25 records 2-14.1  
  type 26 records 2-14,5-5  
  type 6 records 2-13

SNA (systems network architecture) A-7,GL-3

SNARJP DSP A-7

software failures A-5

space allocation, JSAM 4-2

spool data management (SDM) 4-1 to 4-6,GL-3

spool device  
  access methods for 4-1 to 4-3  
  BSAM and QSAM data sets for 4-3  
  buffers used in I/O for 4-3  
  card images on 2-15  
  data management for 4-1 to 4-6  
  error recovery for A-7,A-8  
  I/O buffers for 4-3  
  partitions for A-8  
  reducing contention on A-8  
  space, freeing of 2-14.2  
  SWA control blocks on 2-6  
  SYSOUT data sets on 2-10  
  techniques for recording on 4-1  
  volumes 4-1

spool partitions A-8

- SRF's (single-record files)
    - data area for 4-3
    - for control blocks 4-1
    - on spool data set 5-1
  - SSCVT (subsystem communication vector table) 3-4
  - SSI (subsystem interface)
    - communication overview for 3-3 to 3-6
    - communication paths of 3-2
    - control blocks for 3-4
    - destination codes for 3-3,3-7,GL-1
    - forwarding of dynamic allocation request by A-4
    - function codes for 3-3,3-5,3-7,GL-1
    - function routine for 3-5,3-7,4-5
    - poster for common services 3-5,3-6
    - read end subroutine for 3-6
    - request routine for 3-5,3-7
    - router for common services 3-5
  - SSIB (subsystem identification blocks) 3-4,3-5,GL-3
  - SSISERV macro for
    - entry to router 3-5,3-6,3-8
    - passing staging area 4-5
    - SYSOUT data sets 4-5,4-6
  - SSOB (subsystem options block) 3-4,3-5,3-7,GL-3
  - SSVT (subsystem vector table) 3-4,3-6
  - staging area
    - address, in SEL 3-5
    - definition of GL-3
    - location of 3-6
    - processing by poster routine for 3-6
    - use in opening of SYSIN data sets 4-5
  - standard job
    - definition of GL-3
    - processing of 2-5 to 2-15
    - scheduler elements for 1-7,2-5
  - START command 1-7
  - STARTIO macro
    - use for
      - I/O to spool device communications 4-3
      - return of scheduling information 3-7
      - sending request to global JES3 3-7
      - staging areas 3-6
  - statements
    - control
      - for non-standard jobs 1-7
      - FORMAT 5-5
      - influence upon JCT entries 2-5
      - overriding destination names 2-11
      - PROCESS 5-5
    - MVS
      - DD 2-5
      - EXEC 1-2
      - JOB 1-2
  - steps, job 1-2
  - storage-resident control blocks 5-1
  - store-and-forward DSP A-5
  - STT (single track table) space 5-1
  - subpools 4-4
  - subsystem, master 3-4
  - subsystem communication vector table (SSCVT) 3-4
  - subsystem ID 3-4
  - subsystem identification block (SSIB) 3-4,3-5,GL-3
  - subsystem interface (see SSI)
  - subsystem options block (SSOB) 3-4,3-5,3-7,GL-3
  - subsystem vector table (SSVT) 3-4,3-6
  - SVC 99 A-4
  - SWA (scheduler work area)
    - freeing of spool space for 2-14.2
    - in JDS 5-5
    - on spool 2-6
    - track groups for 4-2
    - used for temporary storage of JCL 2-6
  - synchronous data link control (SDLC) A-7
  - SYSIN data set
    - freeing spool space for 2-14.2
    - initial space allocation for 4-2
    - MVS allocation for 4-5
    - opened by problem program 4-5
    - opening of 2-10
    - processing of 4-5
    - recorded in spool space 4-1
    - recording technique for 4-1
    - requests for BSAM, QSAM services 4-2 to 4-3
    - virtual storage for 4-3
  - SYSIN function codes 3-3
  - SYSOUT data set
    - characteristics, in OSEs 5-5
    - class definitions 2-11
    - freeing spool space for 2-14.2
    - global JES3 responsibility 3-2
    - initial space allocation for 4-2
    - on spool volumes 4-1
    - opening of 2-10
    - OUTSERV DSP for 2-10
    - processing of 4-5
    - recorded on spool space 4-1
    - recorded technique for 4-1
    - requests for BSAM, QSAM services 4-2 to 4-3
    - scheduler element for 1-3
    - scheduling of writers for 2-13
    - special handling of 5-5
    - virtual storage for 4-3
  - SYSOUT function codes 3-3
  - system area 3-2
  - system management facilities (SMF), definition of GL-3
  - system management facilities (SMF), type 25 records 2-14
  - system management facilities (SMF), type 26 records 2-14,5-5
  - system management facilities (SMF), type 6 records 2-13
  - system queue area 3-1,4-4
  - systems network architecture (SNA) A-7,GL-3
- T
- tape devices 1-7,2-8
  - TCAM (telecommunication access method) A-8
  - track 5-1
  - track groups 4-2,4-3,5-1
  - train printer characteristic 2-11
  - transient writer routine 2-10
  - transmission DSP A-5
  - TSO (time sharing option)
    - data sets for users of 2-12
    - function codes for 3-3
    - OUTPUT command processor for 2-12
    - writers for 2-11

## U

unit record devices 2-8  
USAM (user spool access method)  
  components 4-2,4-3  
  data flow for 4-4 to 4-6  
  definition of GL-3  
  for user programs 4-1  
user address space (see also address space communication)  
user address space 3-4  
user buffers 4-3,4-4  
user exit routine  
  for changing scheduler elements 1-7  
  for SSI 3-5  
user parameter buffer (PARM) 5-4,5-5  
user spool access method (see USAM)  
user storage buffer pools 4-3,4-4

## V

verification processing 2-8  
verify counts 2-9  
VERIFY DSP 2-7  
virtual address spaces 3-1  
virtual telecommunications access method (VTAM) A-8  
virtual units, MSS 2-8  
volume allocation 2-7  
volume fetch 2-8  
VTAM (virtual telecommunications access method)

## W

wait type requests 3-5  
warm start  
  definition of GL-3  
  IPL for A-6  
  with analysis GL-3  
work station, remote 2-11  
workflow management 1-1  
work-to-do-driver (WTDDRVR) DSP 2-1,2-15  
WRITE macro 4-5  
writer  
  dynamic 2-12,GL-1  
  external 2-11,2-12,GL-1  
  hot 2-12  
  output 2-10,2-12,2-13  
  scheduling of 2-11,2-13  
writer output multitasking facility  
  control block structure for dispatching FCTs 2-14  
  defined 2-13, GL-3  
  purpose of 2-13  
  task switching 2-13  
WTDDRVR (work-to-do-driver) DSP 2-1,2-15

## 3

3800 printer 2-12



**This Newsletter No.** SN25-0199  
**Date** December 30, 1981

**Base Publication No.** SC23-0040-0  
**File No.** S370-20

**Prerequisite Newsletters/  
Supplements** None

## JES3 Overview

© Copyright IBM Corp. 1980

OS/VS2 MVS/System Product — JES3, Program No. 5740-XYN

This newsletter contains replacement pages for *JES3 Overview*.

Before inserting any of the attached pages into *JES3 Overview*, read *carefully* the instructions on this cover. They indicate when and how you should insert the pages.

<u>Pages to be Removed</u>	<u>Attached Pages to be Inserted*</u>
Cover - Edition Notice	Cover - Edition Notice
v - vi	v - vi
None	ix - x
1-1 - 1-2	1-1 - 1-2
2-5 - 2-6	2-5 - 2-6
2-13 - 2-14	2-13 - 2-14.2
5-5 - 5-6	5-5 - 5-6
A-7 - A-8	A-7 - A-8
GL-1 - GL-4	GL-1 - GL-4
I-1 - I-8	I-1 - I-10

\*If you are inserting pages from different Newsletters/Supplements and *identical* page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

### Summary of Amendments

This Technical Newsletter includes the following changes:

- Minor editorial corrections to several topics
- The topic "Converter Interpreter Processing" in Chapter 2 discusses installation imposed DD limits.
- The topic "Output Service Processing" in Chapter 2 discusses the writer output multitasking facility.

**Note:** Please file this cover letter at the back of the base publication to provide a record of changes.

**IBM Corporation, Publications Development, Department D58, Building 706-2,  
PO Box 390, Poughkeepsie, New York 12602**









JES3 Overview

Printed in U.S.A.

SC23-0040-0

