

OS/390



JES2 Introduction

OS/390



JES2 Introduction

Note

Before using this information and the product it supports, be sure to read the general information under Appendix B, "Notices" on page 35.

Fourth Edition, September 2000

This is a major revision of GC28-1794-02.

This edition applies to Version 2 Release 10 of OS/390 (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrdfs@us.ibm.com

World Wide Web: <http://www.ibm.com/s390/os390/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1990, 2000. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	vii
Who Should Read This Book	vii
How To Use This Book	vii
Related Information	vii
Additional Information	vii
Summary of Changes	ix
Introduction to JES2	1
What is a JES?	1
How JES2 Fits into the MVS System	2
Relationship of JES2 to JCL and Submitted Jobs	3
JES2 Compared to JES3	3
Overview of JES2	4
JES2 Configurations	4
JES2 Functions	4
Customizing JES2	5
Interacting with JES2	5
Scope of Control and Configurations	7
JES2 Data Set Control	7
Spool Data Sets and Spooling	7
Maintaining Integrity	8
JES2 Configurations	8
Single-System Configuration	9
Multiple-System Configuration	9
Running Multiple Copies of JES2 (Poly-JES)	10
JES2 Remote Job Entry (RJE)	10
JES2 Network Job Entry	13
JES2 Job Processing and Functions	17
Phases of Job Processing	17
Input Phase	18
Conversion Phase	18
Processing Phase	18
Output Phase	19
Hard-Copy Phase	19
Purge Phase	20
JES2 Capabilities and Functions	20
Getting Work Out of MVS	21
Selecting Work to Maximize Efficiency	21
Offloading Work and Backing Up the System	22
Supporting Advanced Function Printers	22
Providing Security	23
Supporting APPC	23
Tailoring Your JES2 System	25
JES2 Initialization Data Set	25
Minimum Required Statements	26
JES2 Table Pairs	26

JES2 IBM-Defined Exits	27
JES2 Installation-Defined Exits	27
Interacting with JES2	29
JES2 Operations	29
Operator Control	29
Automatic Commands	30
Automating JES2 Operations	31
JES2 Communication Mechanisms	31
JES2 Messages	31
JES2 Tracing Facility	32
JES2-IPCS Formatting	32
Appendix A. JES2 Publications	33
Appendix B. Notices	35
Programming Interface Information	36
Trademarks	36
Glossary	39
Index	53

Figures

1. Relationship of JES2 to the Base Control Program	2
2. Remote Job Entry Configuration	11
3. Example of a Remote Job Entry (RJE) Configuration	12
4. Example of a Network Job Entry (NJE) Configuration	14
5. A Network of Various Processing Systems	15
6. Job Processing Phases	17

About This Book

This book provides an introduction to the job entry subsystem 2 (JES2). It is meant to provide an overview of JES2; it is not meant to be an instructional manual. This book presents:

- JES2 as a unified set of related functions whose purpose is to accomplish certain data processing goals for MVS
- JES2 and its relationship to MVS, Time Sharing Option Extensions (TSO/E), and components of the MVS operating system
- An overview of JES2 functions, device control, network job entry (NJE), remote job entry (RJE), initialization, operations, customization techniques, and diagnostic tools.

Who Should Read This Book

This book is intended for the MVS system programmer, operator, data processing manager, or student who is unfamiliar with JES2 and its functional relationship to the base control program of MVS.

This book is specifically designed for installations that are evaluating or running OS/390 JES2. However, much of the function described is available in previous versions of the JES2 component in MVS/System Product Version 1, Version 2, Version 3, Version 4, and Version 5. Therefore, most of the material presented here applies to JES2 at those version/release levels because of the overview-level of detail. The book, however, makes no distinction between functions and in what version/release they were added to JES2.

How To Use This Book

To obtain the most from this book, you should read it sequentially. The information presented in each chapter and topic is built upon previous information. If you are unfamiliar with JES2 concepts and terminology, you will find that reading the book in this manner is most helpful.

Related Information

When this book references information in other books, the shortened version of the book title is used. Refer to *OS/390 Information Roadmap* for their full titles and order numbers.

Additional Information

Additional information about OS/390 elements can be found in the following books.

Title	Order Number	Description
<i>OS/390 Introduction and Release Guide</i>	GC28-1725	Describes the contents and benefits of OS/390 as well as the planned packaging and delivery of this new product.

Title	Order Number	Description
<i>OS/390 Planning for Installation</i>	GC28-1726	Contains information that lets users: <ul style="list-style-type: none"> • Understand the content of OS/390 • Plan to get OS/390 up and running • Install the code • Take the appropriate migration actions • Test the OS/390 system
<i>OS/390 Information Roadmap</i>	GC28-1727	Describes the information associated with OS/390 including OS/390 books and books for the participating elements.
<i>OS/390 Summary of Message Changes</i>	GC28-1499	Describes the changes to messages for individual elements of OS/390. Note: This book is provided in softcopy only on the message bookshelf of the OS/390 collection kit.

Summary of Changes

| **Summary of Changes**
| **for GC28-1794-03**
| **OS/390 Release 10**

| This book contains information previously presented in *OS/390 JES2 Introduction*,
| GC28-1794-02, which supports OS/390 Version 2 Release 4.

| “Spool Data Sets and Spooling” on page 7 is updated to reflect changes to JES2
| that improve spool I/O processing.

| This book includes terminology, maintenance, and editorial changes. Technical
| changes or additions to the text and illustrations are indicated by a vertical line to
| the left of the change.

| **Summary of Changes**
| **for GC28-1794-02**
| **OS/390 Release 4**

| This book contains information previously presented in *OS/390 JES2 Introduction*,
| GC28-1794-01, which supports OS/390 Version 1 Release 3.

| “JES2 Job Processing and Functions” on page 17 is updated to reflect changes to
| JES2 conversion and processing phases when your installation is implementing the
| MVS workload manager (WLM) batch management facility.

| **Summary of Changes**
| **for GC28-1794-01**
| **OS/390 Release 3**

| This book contains information previously presented in *OS/390 JES2 Introduction*,
| GC28-1794-00, which supports OS/390 Version 1 Release 1.

| This book includes terminology, maintenance, and editorial changes. Technical
| changes or additions to the text and illustrations are indicated by a vertical line to
| the left of the change.

| **Summary of Changes**
| **for GC28-1794-00**
| **OS/390 Release 1**

| This book contains information previously presented in *MVS/ESA JES2 Introduction*
| GC28-1420, which supports MVS/ESA System Product Version 5.

| This book includes terminology, maintenance, and editorial changes. Technical
| changes or additions to the text and illustrations are indicated by a vertical line to
| the left of the change.

Introduction to JES2

This chapter answers the following questions:

- What is a JES?
- Why is JES required in an MVS system?
- What is the relationship between JES and the system users?
- Is there really any difference between JES2 and JES3?
- What are some of JES2's key functions?

By the time you pick up this manual, you are probably aware that one of your initial concerns when installing MVS is whether to select an MVS system with the JES2 or the JES3 component. This chapter provides some of the background you will need if you are new to MVS, and the remainder of the book provides a somewhat more detailed discussion of JES2 function. If you require further insight into the functional differences between JES2 and JES3, or are deciding which to install, you should also read *OS/390 JES3 Introduction* which provides a similar overview of JES3.

What is a JES?

MVS uses a job entry subsystem (JES) to receive jobs into the operating system, schedule them for processing by MVS, and to control their output processing. JES2 is descended from HASP (Houston automatic spooling priority). HASP is defined as: *a computer program that provides supplementary job management, data management, and task management functions such as: scheduling, control of job flow, and spooling.* HASP remains within JES2 as the prefix of most module names and the prefix of all messages sent by JES2 to the operator.

JES2 (job entry subsystem 2) is *a functional extension of the HASP II program that receives jobs into the system and processes all output data produced by the job.* So, what does all that mean? Simply stated, JES2 is that component of MVS that provides the necessary functions to get jobs into, and output out of, the MVS system. It is designed to provide efficient spooling, scheduling, and management facilities for the MVS operating system. (Refer to "Spool Data Sets and Spooling" on page 7 for a full definition of Spooling.)

But, none of this explains why MVS needs a JES. Basically, by separating job processing into a number of tasks, MVS operates more efficiently. At any point in time, the computer system resources are busy processing the tasks for individual jobs, while other tasks are waiting for those resources to become available. In its most simple view, MVS divides the management of jobs and resources between the JES and the base control program of MVS. In this manner, JES2 manages jobs before and after running the program; the base control program manages them during processing.¹ Figure 1 on page 2 presents a diagram of the relationship between JES and MVS.

¹ JES3, in contrast, manages jobs throughout the entire process cycle (before, during, and after running the programs).

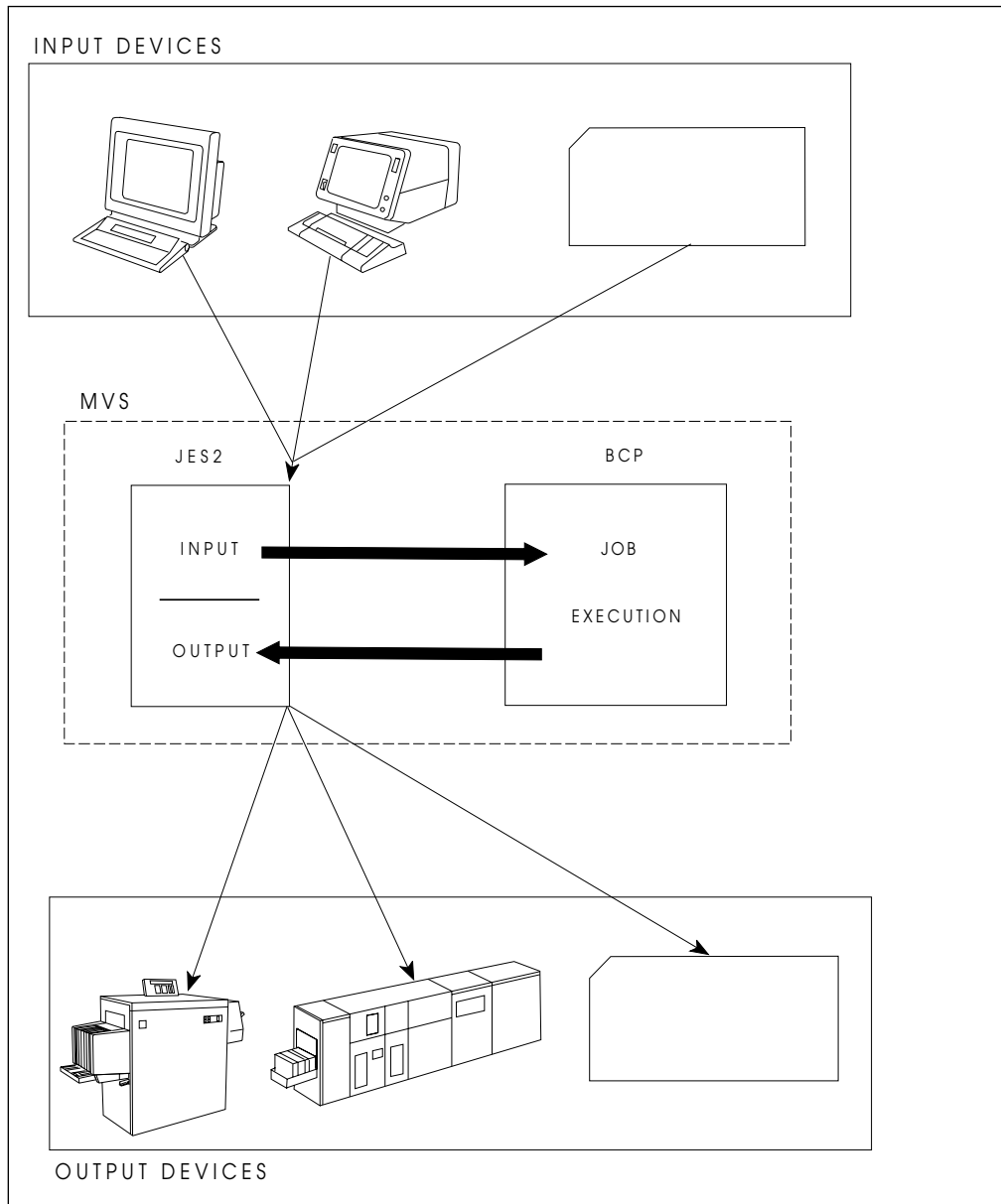


Figure 1. Relationship of JES2 to the Base Control Program

How JES2 Fits into the MVS System

During the life of a job, both JES2 and the base control program of MVS control different phases of the overall processing. Generally speaking, a job goes through the following phases:

- Input
- Conversion
- Processing
- Output
- Print/punch (hard copy)
- Purge

Except for processing, all the job phases are controlled by JES2.

These phases are explained in more detail in Chapter 3, “JES2 Job Processing and Function.”

Relationship of JES2 to JCL and Submitted Jobs

JES2 initialization statements give the system programmer a single point of control to define an installation's policies regarding the degree of control users have over their jobs. For example, consider the confusion an installation would experience if users were allowed to define their own job classes and priorities. Very likely, all users would define all their jobs as top priority, resulting in no effective priority classifications at all. So, although a user can use job control language (JCL) options to define a priority, a JES2 initialization statement (defined by the system programmer) determines whether JES2 acknowledges that priority.

This same type of control (validation of user specification and provision for a default) extends to many JCL definitions. For example, JES2 can specify the maximum time a particular job is allowed to run, the storage a job may consume, the number of copies of output a job can print, and the type of paper (form) on which the output prints.

An installation has the ability to allow its users to override system and JES2 specifications at three distinct levels (user-specified JCL, installation-specified JCL, and JCL defaults). The following list shows the hierarchy of control, from highest priority to lowest priority, of user JCL, JCL defaults, and JES2 defaults:

1. JCL specification on a user job.

The user JCL overrides:

2. JCL default, which MVS uses if there is no user definition or the user specification is disallowed.

The JCL default overrides:

3. JES2 default, which MVS uses if:

- No JCL default exists
- The user specification is disallowed or undefined
- The JCL default definition is not supported.

Note that the user can override the JCL defaults and the JCL defaults can override the JES2 specifications, but the ability to override is either permitted or disallowed by specifications that only the system programmer can control. This structure thereby puts the system control in the hands of the system programmer, not the individual user who is submitting the job. JES2 becomes the base for input and output specifications that can then be overridden, as allowed by your installation, through the JCL and job submitter.

JES2 Compared to JES3

IBM provides two JESs from which to choose: JES2 and JES3. In an installation that has only one processor (computer), JES2 and JES3 perform similar functions. That is, they read jobs into the system, convert them to internal machine-readable form, select them for processing, process their output, and purge them from the system. However, for an installation that has more than one processor in a

configuration, there are noticeable differences in how JES2 exercises **independent control** over its job processing functions. That is, within the configuration, each JES2 processor controls its own job input, job scheduling, and job output processing. In contrast, JES3 exercises **centralized control** over its processing functions through a single **global** JES3 processor. This global processor provides all job selection, scheduling, and device allocation functions for all the other JES3 systems. The centralized control that JES3 exercises provides: increased job scheduling control, deadline scheduling capabilities, and increased control by providing its own device allocation. To gain a more complete understanding of the functional differences between JES2 and JES3, refer to *OS/390 JES2 Initialization and Tuning Guide* and *OS/390 JES3 Initialization and Tuning Guide*.

Overview of JES2

The remainder of this book provides an overview of the processing configurations that JES2 supports, the functional capabilities of JES2, the ability to customize JES2 to meet your specific processing needs, and the means by which JES2 communicates its status through messages and diagnostic information. Each topic is discussed briefly here.

JES2 Configurations

You can run MVS in your installation in many processing configurations that range from a single base control program with a single JES2 that is completely isolated from other processing systems to one that is a part of a worldwide processing network. The choice of configuration complexity is yours and generally is a dynamic one that grows as your business needs grow. Basic configurations are:

- Single-processor
- Multiple-system (multi-access spool)
- Poly-JES
- Remote job entry (RJE)
- Network job entry (NJE)

These are discussed in greater detail, with examples, in Chapter 2, “Scope of Control and Configurations.”

JES2 Functions

To manage job input/output responsibilities for MVS, JES2 controls a number of functional areas, all of which you can customize to your installation's need. Some of the major functional areas and processing capabilities are:

- Getting work into and out of MVS (input/output control)
- Maximizing efficiency through job selection and scheduling
- Offloading work and backing up system workload
- Supporting advanced function printers (AFPs)
- Running multiple copies of JES2
- System security.

Each of these is discussed in Chapter 3, “JES2 Job Processing and Functions.”

Customizing JES2

JES2 has the flexibility to fulfill an installation's unique processing needs. An installation can virtually control every JES2 function. You can perform many customization tasks when JES2 is installed, and you can dynamically customize JES2 whenever your processing needs change. JES2 provides initialization statements, commands, IBM-supplied exit points, the ability to add installation-defined exit points that require minimal source code modification, and the ability to change many commands and system messages, all without the need to modify the IBM-supplied code. Refer to Chapter 4, "Tailoring Your JES2 System," for a more complete description of these customization facilities.

Interacting with JES2

No large data processing system or subsystem can continually operate independently of system programmer or operator intervention. A two-way communication mechanism between the operator and JES2 is required. Based on system workload and device requirements, JES2 must communicate its status to the operator and/or system programmer. JES2 issues messages to communicate job and device status, problem situations, system resource constraint situations, and performance status. Through commands, you can request current status and through the use of various tools you can obtain further information to diagnose and correct problem and system failure situations. Chapter 5, "Interacting with JES2," provides an overview of these topics.

Scope of Control and Configurations

This chapter answers the following questions:

- How does JES2 manage its work?
- Where does JES2 store its data, and how does JES2 provide data integrity?
 - What is spooling?
 - What is checkpointing?
- What processing configurations does JES2 support?
 - What is a multi-access spool complex?
 - What is poly-JES?
 - What is remote job entry?
 - What is network job entry?

The control that JES2 has over certain data sets and devices allows MVS to offload work to JES2. The means by which JES2 maintains these data sets is unique to JES2. Furthermore, the configurations in which JES2 operates range from simple to rather complex. This chapter provides a basic understanding of the JES2 operating configurations; it is important to your complete understanding of the following chapters.

JES2 Data Set Control

JES2 maintains copies of its data sets that contain job and output queues (that is, lists of jobs to be processed by MVS) on direct access storage devices (DASD). Future work is added to these queues and JES2 selects work for processing from them. These data sets and queues must remain current and accurate to maintain system integrity and performance. The following is a discussion of the JES2 spool and checkpoint data sets and the processing JES2 uses to maintain them.

Spool Data Sets and Spooling

Simultaneous peripheral operations online (spool) has several meanings as used in this book and throughout JES2 documentation. **Spooling** is a process by which the system manipulates its work. This includes:

- Using storage on direct access storage devices (DASD) as a buffer storage to reduce processing delays when transferring data between peripheral equipment and a program to be run.
- Reading and writing input and output streams on an intermediate device for later processing or output.
- Performing an operation such as printing while the computer is busy with other work.

Spool also refers to the direct access device that contains the spool data sets. This definition is generally apparent from the context of its use and should not cause any misunderstanding in the following sections of this book or other JES2 documentation.

| Spooling is critical to maintain performance in the MVS-JES2 environment. JES2
| attempts to maximize the performance of spooling operations, which ultimately
| benefits the throughput of work through the JES2 installation.

As noted previously, spooling provides simultaneous processing and a temporary storage area for work that is not yet completed. Once JES2 reads a job into the system, JES2 writes the job, its JCL, its control statements, and its data to a spool data set until further processing can occur.

Maintaining Integrity

Errors can occur while processing jobs and data. Some of these errors can result in the halting of all system activity. Other errors can result in the loss of jobs or the invalidation of jobs and data. If such errors occur, it is preferable to stop JES2 in such a way that allows processing to be restarted with minimal loss of jobs and data. The **checkpoint data set**, **checkpointing**, and the **checkpoint reconfiguration dialog** all help to make this possible.

Checkpoint data set is the general term used to describe the checkpoint data set that JES2 maintains on either DASD or a coupling facility. The checkpoint data set (regardless of whether it resides on a coupling facility structure or a DASD volume) contains a backup copy of the job and output queues, which contain information about what work is yet to be processed and how far along that work has progressed. Similar to the spool data sets, the checkpoint data set is commonly accessible by all members of the multiple-processor (multi-access) spool complex, but only one member will have control (access) of the checkpoint data set at any one time. Furthermore, the checkpoint data set provides communication among all members of the configuration about jobs and the output from those jobs. JES2 periodically updates the checkpoint data set by copying the changed information from the in-storage copy to the checkpoint data set copy that resides on either a coupling facility structure or on DASD. Information in the checkpoint data set is critical to JES2 for normal processing as well as in the event that a JES2 or system failure occurs.

Checkpointing is the concept of keeping a copy of the checkpoint data set that contains essential workload information on either a coupling facility structure or a DASD volume. This copy is updated from the in-storage copy of the checkpoint as each JES2 member of a multi-access spool updates the in-storage checkpoint data set. **Checkpointing** allows JES2 to be stopped and then restarted with little or no loss of job or data integrity.

The **checkpoint reconfiguration dialog** is a dynamic means by which the current checkpoint configuration can be changed (for example, adding a checkpoint device or moving the checkpoint data set to a different device). It is possible to enter the checkpoint reconfiguration dialog to continue processing without necessitating a JES2 restart. This process increases system availability.

JES2 Configurations

The size and complexity of your data processing configuration is based on your business needs. Many factors contribute to the configuration your installation requires. For example, you need to consider the:

- Number of concurrent interactive users
- Size of your data base(s)

- Number of customers
- Geographic location of users and resources
- Number of users that need to run jobs, access data bases, and use programs.

JES2 provides two distinct facilities for the expansion of your JES2 processing configuration: remote job entry and network job entry. **Remote job entry** allows for the extension of your local processing configuration by defining remote locations that can consist of job input terminals and output devices at a different physical site connected to the MVS/JES2 processor through telecommunication links such as telephone lines and satellites. In this manner, an installation can provide input and output support throughout a large office building, to locations across town, or even in another city. But all the remote sites and their attached devices are defined to a single MVS/JES2 configuration. **Network job entry** takes this concept further by allowing individual MVS/JES2 processors, that are geographically dispersed, to be connected in a network of JES2 and non-JES2 systems that can communicate, pass jobs, and route output to any attached output devices.

Any combination of JES2 configurations can coexist in a sysplex, however, IBM recommends that each MVS image in the sysplex have a JES2 that is a member of a single multi-system complex.

JES2 allows you to take advantage of the power of MVS in:

- A single-system complex (one processor and one JES2)
- A multiple-system complex (up to 32 processors, each with its own JES2)
- A poly-JES configuration (more than one JES2 operating concurrently within a single MVS)
- Remote job entry (RJE) workstations (remotely attached to a processor in the configuration)
- A network job entry complex (the linking of two or more single-system or multi-system configurations).

The remainder of this chapter addresses each of these processing configurations.

Single-System Configuration

A JES2 **configuration** can contain as few as one processor (one MVS and JES2 system) or as many as 32 processors linked together. A single processor is referred to as a **single-system configuration**. Such a system is suitable for an installation that has a relatively small work load, or possibly an installation that requires a single processor to be isolated from the remainder of the data processing complex, possibly to maintain a high degree of security.

Multiple-System Configuration

Many installations take advantage of JES2's ability to link processors together to form a multiple-processor complex, which is generally referred to as a multi-access spool (MAS) configuration. A **multi-access spool configuration** consists of two or more JES2 (MAS) processors at the same physical location, all sharing the same spool and checkpoint data sets. There is no direct connection between the processors; the shared direct access data sets provide the communication link. Each JES2 processor can read jobs from local and remote card readers, select jobs for processing, print and punch results on local and remote output devices,

and communicate with the operator. Each JES2 processor in a multiple processor complex operates independently of the other JES2 processors in the configuration.

The JES2 processors share a common job queue and a common output queue, which reside on the checkpoint data sets. These common queues enable each JES2 processor to share in processing the installation's workload; jobs can run on whatever processor is available and print or punch output on whatever processor has an available device with the proper requirements. Users can also specifically request jobs to run on a particular processor and output to print or punch on a specific device. If one processor in the configuration fails, the others can continue processing by selecting work from the shared queues and optionally take over for the failed processor. Only work in process on the failed processor is interrupted; the other JES2 systems continue processing.

Running Multiple Copies of JES2 (Poly-JES)

MVS allows more than one JES2 subsystem to operate concurrently, if one subsystem is designated as the primary subsystem and all others are defined as secondary subsystems. A secondary JES does not have the same capabilities as the primary JES2 and some restrictions apply to its use. Most notably, TSO/E users can only access the primary subsystem. The restrictions are necessary to maintain the isolation from the MVS-JES2 production system, but the convenience for testing is a valuable function. Operation of multiple copies of JES2 is referred to as **poly-JES**. Secondary JES2s can be useful in testing a new release or installation-written exit routines. This isolation prevents potential disruption to the primary JES2 and base control program necessary for normal installation production work.

JES2 Remote Job Entry (RJE)

The **remote job entry** (RJE) facility allows JES2 to define and use RJE workstations. An **RJE workstation** is a workstation that is connected to a member by means of data transmission facilities. The workstation can be a single I/O device or group of I/O devices or can include a processor such as a System/36, or System/390. Generally, RJE workstations either include a programmable workstation (such as a personal computer) or a communication terminal (such as a 3770, 2780, or S/360) connected to the MVS system through a telecommunication link. Such a link utilizes synchronous data link control (SDLC) or binary synchronous communication (BSC) protocols for communicating between JES2 and remote devices. The remote device will be either a system network architecture (SNA) remote, that uses SDLC, or a BSC remote, that uses BSC. Figure 2 on page 11 shows a simple RJE configuration.

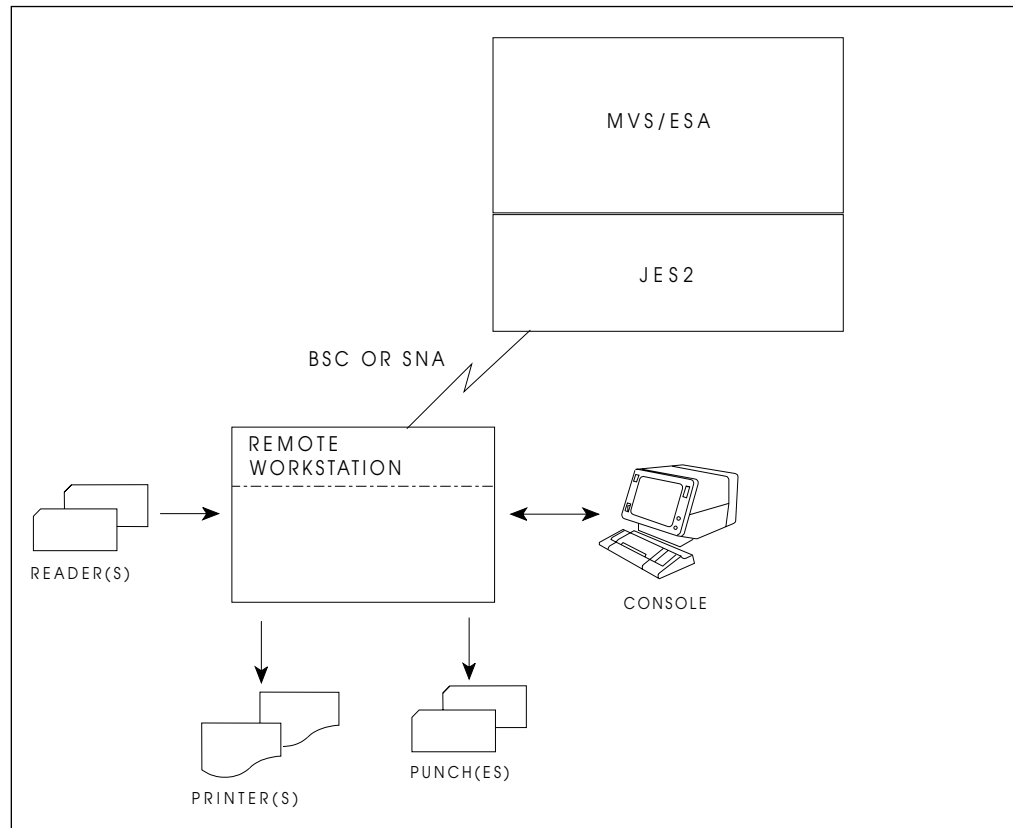


Figure 2. Remote Job Entry Configuration

An RJE workstation is an extension of the local processing facility, except that work is sent across teleprocessing lines. Sending work across teleprocessing lines is convenient for an installation that needs to provide many data entry points at remote locations or route output to many diverse locations. The following illustrates the use of RJE in two familiar examples of daily business:

- In a large department store, many sales clerks need to print output on printers located at the many customer check-out desks throughout the store. Because the location of the building prevents direct connection to the MVS system located in the central office located several hundred miles away, each printer can be defined to JES2 as a remote printer.
- Consider a clothing store chain in which the store managers at seven different stores all need to place orders, access inventory, and provide billing information, all the information for which is maintained on storage devices located in the computer center at the main office. An individual store's workstation is defined to JES2 as an SNA-attached remote to which the individual terminals are defined and as such become an extension of the JES2 configuration located at the main office.

Figure 3 on page 12 presents a diagrammatic view of the RJE configuration as described in the preceding clothing store example.

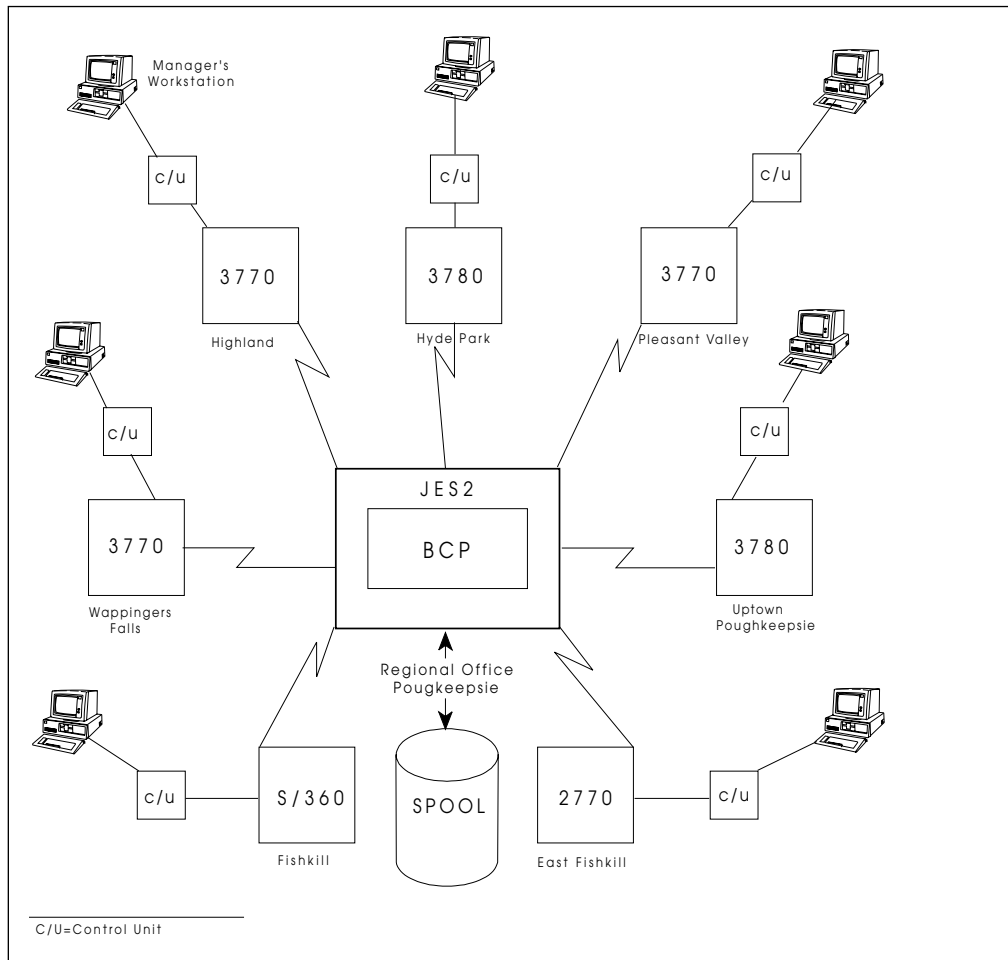


Figure 3. Example of a Remote Job Entry (RJE) Configuration

In Figure 3 each of the clothing stores within a localized region of New York, the Hudson Valley (Highland, Hyde Park, Pleasant Valley, and so forth), is connected to the single processor located at the regional office in Poughkeepsie. One MVS/JES2 system conducts the business of inventory control, shipping, and billing for all of the stores in the region.

JES2 processes remote jobs in the same manner as those received from a local reader. (**Local devices** are printers, punches, card readers, and lines directly attached to the system without the need of transmission facilities.) The terminals and printers located in the Poughkeepsie office are **locally attached**; all other I/O devices (terminals and printers) in all the other branch stores are defined to JES2 as **remote** terminals and printers.

To provide RJE processing, the RJE workstation must be defined to the local processor. There are two protocols available by which JES2 can communicate with the RJE workstations: **synchronous data link control (SDLC)** and **binary synchronous communication (BSC)**.

An RJE workstation can have a processor, like the System/370, that runs a JES2-generated program. The JES2-generated program allows the processor to send jobs to, and receive data from, JES2. Such RJE workstations have generally been replaced by either a programmable workstation, such as a personal computer or a network job entry configuration, and they are rarely used in today's processing

environment. Some RJE workstations do not have a processor. These workstations use a remote terminal, for example, a 2780 or 2770, to enter jobs into, and receive data from, JES2.

Refer to *OS/390 JES2 Initialization and Tuning Guide* for a more complete discussion of RJE concepts.

JES2 Network Job Entry

The JES2 **network job entry** (NJE) facility is similar to remote job entry (RJE) in that they both provide extensions to a computer system. In its simplest terms, NJE is "networking" between systems that interact as peers, whereas RJE is networking between JES2 and workstations. The main difference between them is one of overall compute power and processor location. Remember, RJE is an extension of a *single* computer system (that is, either a single-processor or multi-access spool complex) that allows jobs to be submitted from, and output routed back to, sites that are remote to the location of that system. NJE provides a capability to link many such single-processor systems or multi-access spool complexes into a processing network. Each system can be located on the same physical processor, side-by-side in a single room, or across the world in a network of thousands of nodes. The important difference is that a processor and its local and remote devices make up a **node**. Two or more attached nodes make up an NJE network.

Furthermore, as discussed in the section, "Multi-System Complex," a node can include as few as one processor with MVS or as many as 32, all sharing the same spool and checkpoint data sets. NJE network nodes communicate (send and receive data) using various teleprocessing facilities. Nodes on the same physical processor use the Virtual Telecommunications Access Method (ACF/VTAM) program product to communicate; no hardware is required. Nodes located in the same room or building can utilize channel-to-channel adapters (CTCA) or telecommunication links. Nodes that are geographically dispersed utilize SNA or BSC telecommunication links. The following example further explains this concept.

If we return to the previous example of the clothing store chain, the Poughkeepsie regional office is the location of the processor (MVS and JES2); each of the sites (Highland, Hyde Park, Pleasant Valley, Fishkill, and so forth) are attached to the Poughkeepsie office as remote sites. Together, all the locations in Figure 3 on page 12 comprise the Poughkeepsie node. For a locally owned, relatively small clothing store chain such as the one depicted, this complex may suffice. However, for a national clothier or one dealing in the import/export business, one processor would likely prove to be inadequate. Such a company would likely elect to establish many nodes throughout the world, each connected to all others in an NJE complex. Refer to Figure 4 on page 14 for a diagrammatic view of such a network. Note that the different groups of the ordering/billing department and the business administration/payroll department can be separate members of a MAS making up the New York and London nodes.

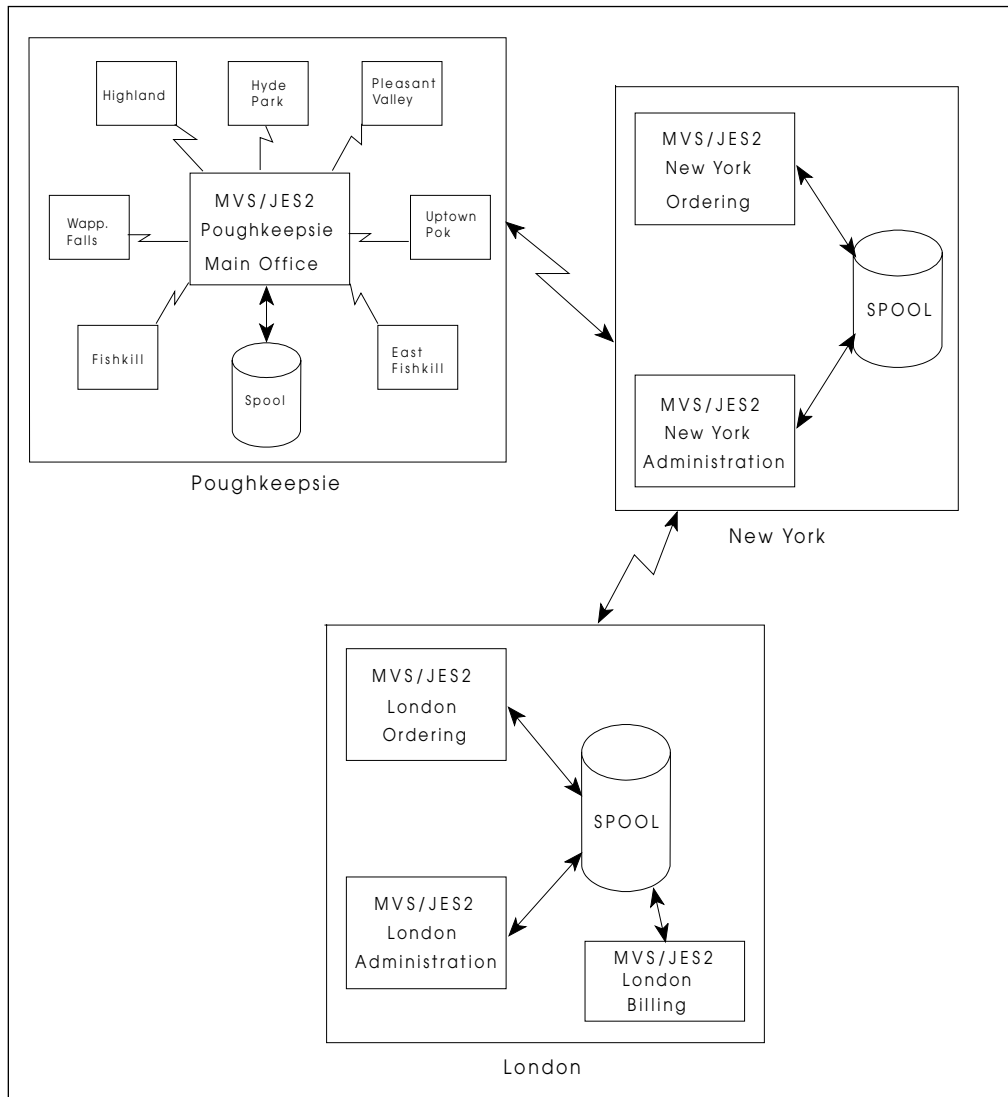


Figure 4. Example of a Network Job Entry (NJE) Configuration

Note that the New York and London nodes are multi-access spool configurations, Poughkeepsie is the configuration as described in Figure 3 on page 12, and all the other nodes located around the world (which are not depicted) are simple single-system configurations. In the network, a store manager in Pleasant Valley can order an item, the Poughkeepsie office (the main office for the Hudson Valley region) tallies the orders from this store and its other six stores and submits its order to the clothier chain headquarters located in New York City. The order is then forwarded to the London export office, where the item is procured and shipped. The request can be routed through Poughkeepsie and New York to London for accounting purposes or it could have been sent directly to London if the business is organized in that manner. The request and subsequent confirmation of the order is not instantaneous due to the distance and the **traffic** in the system but is faster and a more efficient method of doing business than a telephone conversation, particularly if you further consider time zone differences. Traffic refers to the number of users, requests, jobs, and data currently being routed across available teleprocessing lines.

Individual nodes in a network need not all be MVS/JES2 nodes. As Figure 5 on page 15 illustrates, each node can contain different processing systems; nodes

with various levels of MVS/JES2, MVS/JES3, VM/RSCS, and VSE/POWER can all be linked in a network. This is true because the NJE formats and protocols used by each of these systems are, by design, release and product independent.

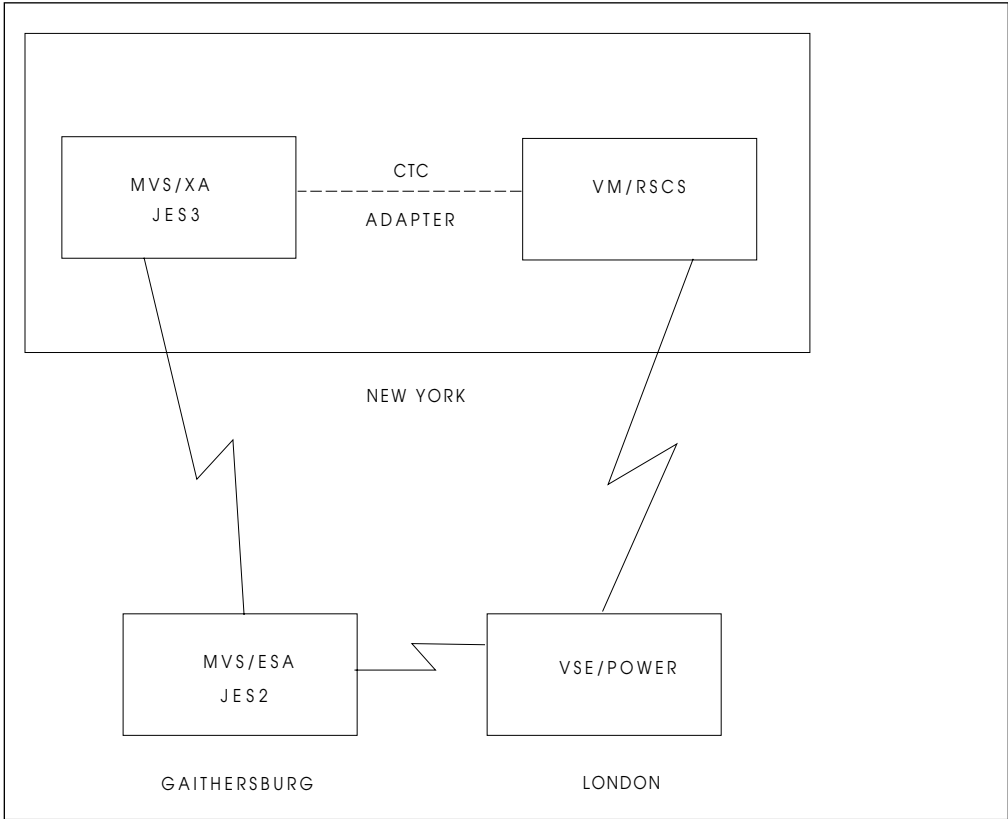


Figure 5. A Network of Various Processing Systems

JES2 Job Processing and Functions

This chapter answers the following questions:

- During the life of a job, what processing does JES2 do for MVS?
- What are some of the major JES2 functions that make it special?

JES2 is responsible for all phases of job processing and monitors the processing phase. This chapter outlines the six phases and presents an overview of some of the major JES2 functions.

Phases of Job Processing

The base control program and JES2 share responsibility in the MVS system. JES2 is responsible for job entry (input), the base control program for device allocation and actual job running, and finally JES2 for job exit (output).

Figure 6 presents a view of the six phases, followed by a description of each.

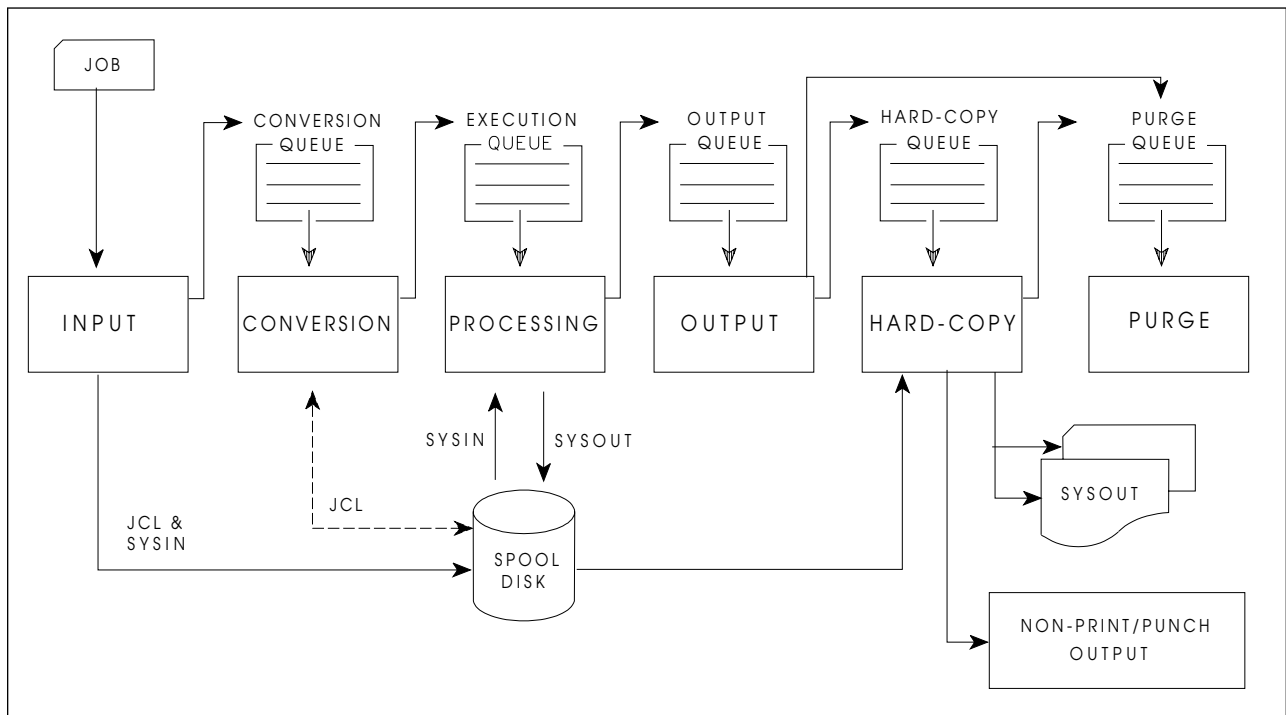


Figure 6. Job Processing Phases

The job queues contain jobs:

- Waiting to run - conversion queue
- Currently running - execution queue
- Waiting for their output to be produced - output queue
- Having their output produced - hard-copy (print/punch) queue
- Waiting to be purged from the system (following completion of all processing) - purge queue.

Input Phase

JES2 accepts jobs (in the form of an input stream) from input devices such as card readers, remote terminals, or other programs. Input streams can also come from other nodes in a job entry network and from internal readers. An **internal reader** is a program that other programs can use to submit jobs, control statements, and commands to JES2. Any job running in MVS can use an internal reader to pass an input stream to JES2, and JES2 can receive multiple jobs simultaneously through multiple internal readers.

MVS uses internal readers, allocated during system initialization, to pass to JES2 the job control language (JCL) for started tasks, START and MOUNT commands, and TSO LOGON requests.

The system programmer defines internal readers used to process all batch jobs other than STCs and TSO requests. JES2 initialization statements define these internal readers which JES2 also allocates during its initialization processing. The internal readers for batch jobs can be used for STCs and TSO requests, if not processing jobs.

As JES2 reads the input stream, it assigns a job identifier to each job and places each job's JCL, optional JES2 control statements, and SYSIN data onto DASD data sets called spool data sets. JES2 then selects jobs from the spool data sets for processing and subsequent running. Refer to "Spooling" in Chapter 2, "Scope of Control and Configurations" for an explanation of spool data sets and the concept of spooling.

Conversion Phase

JES2 uses a converter program to analyze each job's JCL statements. The converter takes the job's JCL, merges it with JCL from a procedure library (such as SYS1.PROCLIB), and converts the composite JCL into converter/interpreter text that both JES2 and the job scheduler functions of MVS can recognize. JES2 then stores the converter/interpreter text on the spool data set. If JES2 detects any JCL errors, JES2 issues messages, and the job is queued for output processing rather than run. If there are no errors, JES2 queues the job for execution. JES2 supports multiple converters; therefore, jobs may not always be processed in a first-in-first-out (FIFO) order. When work load manager (WLM) batch management is in use, JES2 queues the job according to its arrival time.

Processing Phase

In the processing phase, JES2 responds to requests for jobs from the MVS initiators. JES2 selects from a job queue, jobs that are waiting to run and sends them to MVS.

By recognizing the current processing phase of all jobs on the job queue, JES2 can manage the flow of jobs through the system.

JES2 Job Scheduling: To process the jobs on the job queue, JES2 communicates with an initiator. An **initiator** is a system program that starts a job to allow it to compete for system resources with other jobs that are already running.

Initiators are controlled by JES2 or by MVS workload management (WLM).

- JES2 initiators are started by the operator or by JES2 automatically when the system initializes. The initiators select jobs based on the job class(es) that are assigned to the initiator and the priority of the queued jobs. The installation associates each initiator with one or more job classes in a way to encourage the efficient use of available system resources.
- WLM initiators are started by the system automatically based on the performance goals, relative importance of the batch workload, and the capacity of the system to do more work. The initiators select jobs based on their service class and the order they were made available for execution.

After JES2 selects a job and passes it to the initiator, the initiator invokes the interpreter to build control blocks from the converter/interpreter text that the converter created for the job.

The initiator then allocates the resources specified in the JCL for the first step of the job. This allocation ensures that the devices are available before the job step starts running. The initiator then starts the program requested in the JCL EXEC statement.

Priority Aging: When all initiators are busy, throughput of certain jobs might fall below normal expectations. To help in these situations, JES2 uses the additional scheduling function of priority aging. **Priority aging** can help ensure that jobs that have been waiting to run have a chance of being selected to run before those jobs that just entered the system. By using priority aging, an installation can increase the priority of a waiting job. The longer the job waits, the higher its priority becomes, up to a limit, and the greater its chances of being selected to run.

JES2-Base Control Program Interaction: JES2 and the base control program communicate constantly to control system processing. The communication mechanism, known as the subsystem interface, allows MVS to request services of JES2. For example, a requestor can ask JES2 to find a job, do message or command processing, or open (access) a SYSIN or SYSOUT data set. Further, the base control program notifies JES2 of events such as messages, operator commands, the end of a job, or the end of a task.

Output Phase

JES2 controls all SYSOUT processing. **SYSOUT** is system-produced output; that is, all output produced by, or for, a job. This output includes system messages that must be printed, as well as data sets requested by the user that must be printed or punched. After a job finishes, JES2 analyzes the characteristics of the job's output in terms of its output class and device setup requirements; then JES2 groups data sets with similar characteristics. JES2 queues the output for print or punch processing.

Hard-Copy Phase

JES2 selects output for processing from the output queues by output class, route code, priority, and other criteria. The output queue can have output that is to be processed locally or output to be processed at a remote location (either an RJE workstation or another node). JES2 handles each of these situations in different ways.

- Local Output:

When output is to be processed at a local or remotely-attached output device, JES2 uses these local and remotely-attached output devices to produce a job's output. JES2 queues a job's print and punch data sets on the output queue for the local and remote output devices. The active devices, that are attached locally or through RJE connections, select the output data sets with characteristics that best match their selection criteria.

- Network Job Entry Output:

Job output passing through to another JES2 node resides on the network output queue. JES2 selects a job's output from the network output queue for transmission to another node based upon the priority and the desirability of reaching the output-processing node over the available transmission line. After the receiving node signals that it has accepted total responsibility for the output, the transmitting JES2 node releases the resources used to represent the output.

After processing all the output for a particular job, JES2 puts the job on the purge queue.

Purge Phase

When all processing for a job completes, JES2 releases the spool space assigned to the job, making the space available for allocation to subsequent jobs. JES2 then issues a message to the operator indicating that the job has been purged from the system.

JES2 Capabilities and Functions

JES2 (in conjunction with VTAM) is the link between the Time Sharing Options/Extensions (TSO/E) user and MVS. As such, it is very externally oriented. That is, it is visible to the data processing personnel and provides the ability to specify and tailor many installation-specific functions through JES2 initialization statements and JES2 commands. These statements and commands are analogous to the knobs, buttons, and handles with which you control a machine. You set them, knowing what response the machine will provide, but you need not be concerned with how the button sets the gears and belts in motion.

This section provides an overview of the following major functions JES2 provides to manage its job input/output responsibilities for MVS; all of which are under system programmer control.

- Getting work out of MVS
- Selecting work to maximize efficiency
- Offloading work and backing up the system
- Supporting advanced function printers
- Providing security.

All are discussed in greater depth in *OS/390 JES2 Initialization and Tuning Guide*.

Getting Work Out of MVS

As the central point of control over the job output (or exit) function, JES2 controls output devices: local and remote printers, punches, and card readers. You can use JES2 initialization statements to define each device. All are directly under JES2's control with the exception of those printers that operate under the Print Services Facility (PSF). These printers provide all-points-addressable capabilities, and although defined by JES2, are driven by PSF. PSF thereby assumes the processing overhead that JES2 typically performs to support printer operation. (Refer to "Supporting Advanced Function Printers," for an introduction to PSF.)

Printed and punched output can be routed to a variety of devices in multiple locations. The control JES2 exercises over its printers ranges from the job output classes and job names from which the printer can select work to such specifications such as the forms on which the output is printed. This control allows the system programmer to establish the job output environment most efficiently without causing unnecessary printer backlog or operator intervention.

Through JES2, the installation defines the job input classes, reader specifications, and output device specifications. As a result, JES2 is the central point of control over both the *job entry* and *job exit* phases of data processing.

Selecting Work to Maximize Efficiency

To minimize contention for output devices, JES2 allows the installation to define work selection criteria that can be specific for each output device (local and remote printers and punches and offload devices). The work selection criteria on the respective device initialization statements define the:

- Specification of job and output characteristics that JES2 considers when selecting work for an output device
- Order of importance (priority) of the selection characteristics
- Characteristics of the printer and job that must match exactly.

JES2 initializes the setup characteristics of a device based on the specifications supplied on a device (printer/punch) initialization statement.

A job's output is grouped based on the data sets' output requirements. These requirements are defined by the job submitter with the job's job control language (JCL) or by JES2-supplied defaults.

When selecting work to be processed on a device, JES2 compares the device's characteristics to the output requirements associated with the pieces of work awaiting processing. In addition, JES2 compares each piece of work against all others to determine the best match for the device. If work output requirements are found that match a device's characteristics, JES2 sends the output associated with that piece of work to that output device. If a piece of work and an output device cannot be paired, the work will not be selected for output until the operator changes the output device specifications or the output requirements for the piece of work.

Work selection control provides efficient use of output devices by allowing them to print or punch specified output classes without requiring continual operator intervention of setup characteristics such as forms or print trains (the piece of hardware that carries the print type). Also, when too much work is adversely affecting system performance, the system programmer or operator can specify work

to be **offloaded** (that is, moved off the work queues and temporarily out of contention for system resources. Refer to the topic, “Offloading Work and Backing Up the System” for an explanation of offloading.)

Offloading Work and Backing Up the System

All input jobs and system output are stored on spool. JES2 gives your installation the capability to offload data from and later reload data to the spool. This is useful if you need to:

- Preserve jobs and SYSOUT across a cold start, which entails the total rebuilding of work and output queues.
- Migrate your installation to another release of JES2. (You can use spool offload to reload the spool to the new or previous release.)
- Converting to another DASD type for spool.
- Archive system jobs and SYSOUT.
- Relieve a full-spool condition during high-use periods. (You can reload at a later time.)
- Provide a back-up for spool data sets.
- Back-up network connections.

Many selection criteria can be used to limit the scope of the spool offload operation. For example, work can be selected based on: a system on which the job is to run, job number, job disposition (for example, only held data sets), destination, class, or many other criteria. These selection criteria can be changed by operator command following your initial specification with JES2 initialization statements.

Supporting Advanced Function Printers

Advanced function printers (AFPs) such as the 3800-3 and 3820 provide all-points addressable capability. Because every point on the page is available, these printers support graphics and a range of font types and sizes unavailable to impact printers. AFPs are controlled by Print Service Facility (PSF) and used by JES2. JES2 and PSF work together by communicating through the functional subsystem interface (an extension of JES2 device control that removes the device-specific support) to obtain full-function page-mode capability.

Full-function mode is defined as using those functions of the printer that produce page-mode output. The concept of **page-mode** permits printed pages to contain both text data and graphic presentations. The user can define and request attributes such as **segments** (predefined portions of a page), **overlays** (predefined page templates), **images** (pictures and graphics), and **type fonts** (collections of unique or stylized characters). For example, the graphical material can include a wide range of print font types and sizes for use in text headings, logos, and imbedded artwork; shading of textural and user-produced graphics; and graph plotting, some of which you can see in this book. ²

² All the illustrations and headings in this book are imbedded within the text files, and all print as a single image on IBM's AFPs.

Providing Security

Security in a data processing environment involves controlling and auditing access to resources that are important to your installation. In the JES2 environment, these resources include:

- JES2-owned data sets
- Input (from nodes, remote workstations, readers, offload devices, and commands)
- Job names
- System input/output residing on spool (SYSIN/SYSOUT)
- Output (to nodes, printers, punches, remote workstations, and offload devices).

JES2 provides a basic level of security for some of these resources through initialization statements. For example, each node in a network can be defined as having a certain level of control over work at each of the other nodes in the system, which can give one operator limited control over each of the other nodes. This level of control is based on mutual agreement between the nodes and the degree of trust one node has for the other's security procedures.

The control available through initialization statements can be broadened by implementing several JES2 exits available for this purpose. You can implement a more complete security policy by using the system authorization facility (SAF) component of the base control program and a security product such as Resource Access Control Facility (RACF). SAF provides a link to the security product to define any additional security controls your installation may require.

JES2 passes information to SAF to perform password validation, to request authority to access a resource, and to determine security information in various environments. When SAF and the security product indicate a decision on a security request, JES2 bypasses its own security processing.

Supporting APPC

JES2 processes SYSOUT data sets for **APPC transaction programs**, which are application programs that communicate with other APPC transaction programs through the Advanced Program-to-Program Communication/MVS (APPC/MVS). For details about APPC/MVS, see *OS/390 MVS Planning: APPC/MVS Management*. SYSOUT processing is the only function JES2 provides for APPC transaction programs.

Tailoring Your JES2 System

This chapter answers the following questions:

- How can JES2 be tailored at initialization to meet my installation's needs?
- Do I need to get so involved in customizing JES2 that I need to write 'operating system' code?
- What other means can I use to enhance JES2 if it does not meet all my specific processing needs?
- What are JES2 'table pairs'?
- What's the difference between an IBM-defined exit and an installation-defined exit?

JES2 is designed to be tailored to meet an installation's particular processing needs. You can address your basic customization needs when you create the JES2 initialization data set. If you find that this control over JES2 processing to be insufficient, JES2 also provides exits and table pairs to change many JES2 functions or processes.

If you need to modify JES2 processing above the capability provided by initialization statements, and elect to do so through installation-written code, such code should be isolated from the IBM-shipped source code. Changes to JES2 processing implemented through direct source code modification are error prone, counter-productive during migration to future releases, and can prove to be very time consuming when debugging, diagnosing, and applying IBM-written fixes (program temporary fixes (PTFs) and authorized program analysis report (APARs) fixes) to code. Further, alteration of JES2 processing in this manner complicates IBM service assistance. Therefore, JES2 provides several means of allowing you to tailor its processing without direct source code modification.

Recommended methods for tailoring JES2 processing include JES2 table pairs, IBM-defined exits, and installation-defined exits. A general discussion of each is provided later in this chapter. Refer to *OS/390 JES2 Installation Exits* for a complete description of each IBM-defined exit.

JES2 Initialization Data Set

Because every installation that uses JES2 to manage its work input and output is unique, so too are the requirements each installation has on JES2. To meet these needs, JES2 is highly tailorable, and with minimal effort, a system programmer can customize most JES2 functions by changing and using the **JES2 initialization data set** that is provided with the product in the HASIPARM member of the SYS1.VnRnMn.SHASSAMP data set. Although this data set will not run as shipped without some installation additions, it is a valuable model that can save hours of system programmer input time.

With a set of approximately 70 initialization statements, you can control all JES2 functions. The JES2 initialization data set provides all the specifications that are required to define output devices (printers and punches), job classes, the JES2

spool environment, the checkpoint data sets, the trace facility, and virtually every other JES2 facility and function.

Each initialization statement groups **initialization parameters** that define a function. The use of most JES2 initialization statements is optional. That is, you need not define them unless you need to implement or tailor a particular function. Further, many of the parameters provide default specifications that allow your installation to perform basic JES2 processing with no explicit definition on your part. JES2 requires only a minimal set of initialization statements (and/or parameters) that you define when first installing JES2.

Minimum Required Statements

When you are first getting started, you need not define or even be concerned about some of the more sophisticated processing environments such as a multi-access spool complex, nodes, or remote workstations. You are simply building a base on which your installation can grow. There is no need to be overwhelmed with the potential complexity of your JES2 system. As your installation grows, you will likely use more and more of JES2's available function. To assist the task of JES2 initialization, there is a sample initialization data set shipped with the product.

The sample data set shipped in SYS1.PARMLIB requires only the addition of installation-defined devices and installation-specific values. It contains all the JES2 initialization statements and the defaults for all parameters for which they are provided. A new installation would initially want to delete many of these definitions. A more realistic initialization "starter set" is shipped by IBM if you request Custom Built Product Delivery Offering (CBPDO)³ to build your system.

JES2 Table Pairs

Table pairs provide a facility to change, delete, or add to JES2 processing and/or function. Changes made to JES2 processing using table pairs are generally less prone to error than are changes made through installation exits. This is true because JES2 macros generate the tables and generally require you to write less code to be run.

A number of JES2 functions (such as initialization statement processing, command processing, and message building) use tables. You can customize these JES2 functions, and others, by extending their associated tables. JES2 examines two tables, known as a **table pair**. The first table (the **JES2 table**) provides the default processing specifications; the second table (the **user table**) is used to extend, change, or delete the default processing specifications. For example, you can add your own JES2 commands and messages, add a new initialization statement or parameter, abbreviate the length of a JES2 command, or delete an unnecessary command to prevent its accidental misuse.

To simplify the use of this facility you can use the JES2 default tables as templates for construction of installation-written tables. Depending on the table(s) from which

³ CBPDO is an IBM offering which builds, tailors, and delivers a full MVS operating system, such that all you need do is install and use the system. CBPDO is useful for all new MVS installations to ease the entire migration effort from the previous operating system. This option is particularly useful to new MVS installations that may lack the expertise to perform this task.

you choose to add, change, or delete, using table pairs generally takes less detailed knowledge of JES2 internal structure than does writing an exit.

Table pairs do not replace the need for exits. Table pairs and exit points can provide added capability either independently or in conjunction with one another.

JES2 IBM-Defined Exits

JES2 exits provide a clean, convenient, relatively stable interface between JES2 and your installation-written code. Installation-written exit routines are invoked from standard JES2 processing at various strategic locations in JES2 source code. These strategic locations in JES2 source code are called **exit points**. A JES2 exit is established by one or more exit points.

JES2 can have up to 256 exits; IBM has defined only a number of these to allow customization of the most commonly modified functions. New exits are often added when new function is added with each new release of the product. These new exits provide a facility for you to alter the new processing as appropriate to meet your installation's needs.

The exits allow a wide range of JES2 customization. For example, you can add code to:

- Design your own print job separator page
- Verify or change jobs submitted by TSO/E users
- Change or disallow selected commands
- Define alternate processing for a job that uses too many resources
- Provide increased security and password checking for remote terminals and system data sets.

For the **IBM-defined exits** you need only write your own exit routines and incorporate them through use of two initialization statements. IBM has already placed these JES2 exit points in the code. To ensure a proper implementation you should thoroughly understand the IBM-defined exit and its JES2 operating environment. A comprehensive description of each exit is presented in *OS/390 JES2 Installation Exits*.

If you find that none of these exits meet your installation's needs you can establish your own exit point and provide your own exit routine. This, of course, requires a more thorough knowledge of JES2 processing than the use of an IBM-defined exit point. But remember, the use of exits is still far superior to adding in-line source code modification that may necessitate the addition of many lines of code; code that cannot be dynamically disabled as can an exit.

JES2 Installation-Defined Exits

The JES2 exit facility allows you to establish your own exits, should the IBM-defined exits not suffice. Exits established by you are modifications to JES2 and are called **installation-defined** exits. You define them by placing an exit point at appropriate points in the JES2 code (or in your own exit routine code). (IBM-defined exit points are established in the same manner, but the exit point is placed appropriately for the specific function for which it is intended.) Note,

however, that implementing your own exit can be considerably more difficult than writing an exit routine for an IBM-defined exit. Refer to *OS/390 JES2 Installation Exits* for explanations of JES2 coding conventions, restrictions, how JES2 exits are packaged, and to *OS/390 JES2 Macros* for the JES2 macros that are available and intended for your use.

Interacting with JES2

This chapter answers the following questions:

- How does the data processing staff communicate with JES2?
- How much control do operators have over JES2 once it is running?
- Are there ways to automate some of the routine processing needs?
- If JES2 experiences problems, how will it inform the operator?
- Are there any tools available to assist diagnosis and recovery?

JES2 Operations

To help you maintain your overall work environment, JES2 provides an interactive means to control much of its function and the devices under its control. Although the JES2 environment is initially determined through initialization statements, you can alter many of those definitions as the system's workload changes or your installation adds new devices or needs to redefine its overall configuration. JES2 provides commands to request current status of devices and functions and JES2 responds with informational messages. Based on those messages, the operator, system programmer, or automated operations program, such as NetView or Automated Operations Control/MVS (AOC/MVS), can issue further commands to change processing (such as, implement a newly written exit routine), start or stop a printer, or start diagnostic functions (such as the JES2 trace facility). Much of the processing can be changed without disrupting the remainder of the system. The following sections provide a brief overview of the control you have over JES2. (Refer to *OS/390 JES2 Commands* for an explanation of JES2 commands, and *OS/390 JES2 Messages* for the text and explanation of JES2 messages.)

Operator Control

Almost all JES2 initialization statement definitions can be changed by operator commands. These commands are available to both operators and system programmers to allow them to change current definitions. The system programmer can implement certain security features or customization techniques to limit the degree of control an individual or group can have over the operating system.

As your JES2 complex becomes more sophisticated, you might connect your system to others to form a network of systems. You can use operator commands to control the lines that connect the separate systems as well as defining the separate systems to yours. This is typically a very dynamic environment, as different systems are added or deleted from the network due to maintenance, hardware reconfiguration requirements, workload balancing, or the need to access a data base at a different location. JES2 permits you to use commands to alter most of your original network definition, as required.

If this dynamic control were not available, the operator or system programmer would need to change the initialization data set definition, stop JES2 processing, and then restart the system so those changes can take effect. This is, of course, required when redefining some areas of processing, but it denies system users valuable time.

Stopping and Restarting JES2

However, there are instances when JES2 must be stopped and restarted either by a warm or cold start. For example, redefining the number of systems in a network job environment requires a warm start. A **warm start** is a restart of JES2 that does not cause the current work and output queues to be rebuilt, but does provide the required changed information to be propagated and used by all necessary components. A warm start is far superior to a cold start.

The definition (or redefinition) of some JES2 facilities and resources require that the JES2 system be totally shut down. JES2 must be restarted with a cold start to allow all component systems to be aware of the changed facilities and resources. A **cold start** is a JES2 restart that causes all current work and output queues to be lost and rebuilt with new data. The time to restart JES2 in this manner is based on the work in the system and, if not scheduled, causes a disruption in data processing services.

JES2 Commands

JES2 processes its initialization statements and commands in a way that allows most initialization statements to be changed by operator command. The following is a partial list of the control that JES2 commands have over JES2 processing.

Operator commands can be used to:

- Add function and functional subsystems
- Modify previously defined processing, such as: output definition, the dynamic alteration of the checkpoint definition, enabling installation-defined exits, offload devices, printer and punch characteristics, and job characteristics
- Delete function, and network systems, exits, and diagnostic traces
- Start, stop, and halt devices under JES2's control
- Assign units to local printers, punches, card readers, and lines or reassign units to these devices
- Display current facility and device definition.

All JES2 commands are entered using standard MVS command interfaces (such as through an MVS console or within the JES2 initialization data set). The command prefix character, which defaults to a dollar sign (\$), distinguishes JES2 commands and messages from other components of the operating system. For commands, the prefix character defines the scope of the command as being JES2 only; for messages, the prefix character is informational in that it designates that the message was issued by the JES2 component.

Automatic Commands

The operator can specify that certain commands or strings of commands take effect automatically at specific times or at regular intervals. Automatic command processing can be used to provide status displays and to lessen the operator's work for common, preset routines or schedules. For example, if an installation normally does one specific kind of work at 8 AM and another at 9 AM every day, it is possible to preset automatic command processing to issue the operator commands that would ordinarily be necessary at those times. The set of commands can be entered directly into the initialization data set, and the command processor will then issue them daily. Such commands can be a single command, a rather

complicated set of commands, or just commands that are tedious (and error prone) when entered manually.

The use of the automatic command facility is, of course, optional, and provided to ease the operator workload. A second method of automating the operating system is to prompt system response based on JES2 messages through the use of automation products such as Automated Operations Control/MVS (AOC/MVS).

Automating JES2 Operations

JES2 messages (similar to all MVS messages) contain various elements. All messages contain a unique identifying number, some messages contain non-variable text, others contain variable text (the text is based on the specific error condition or status), others contain specific reason codes (a numeric value - the explanation of which is documented in *OS/390 JES2 Messages* and some contain combinations of the preceding elements. Based on the message number, reason code value or variable text, an installation can interpret the system status and have a programmable console automatically issue the desired command(s) to respond to the situation. Also, there are other IBM products, NetView and the application product Automated Operations Control/MVS (AOC/MVS), that can be installed to help you automate your operation based on your installation's policies regarding the handling of particular message or error conditions.

For example, AOC/MVS can be used to issue the appropriate JES2 commands to relieve a spool shortage condition or issue the appropriate MVS command(s) to complete the shutdown of JES2 after a serious error condition.

JES2 Communication Mechanisms

The following diagnostic tools are available for detecting and diagnosing problems occurring in the JES2 environment.

- Messages - JES2 provides a set of messages to alert initially the JES2 operator and system programmer of processing errors.
- Traces - Optionally, your installation can use the JES2 tracing facility, a function that records events associated with specific functions, such as each time JES2 is initialized or terminated or each time an exit routine is taken.
- IPCS - JES2 exploits the interactive problem control system (IPCS) facility to allow you to view, on-line, the formatted contents of JES2 control blocks and dumps of system data necessary when diagnosing and recovering from processing errors.

JES2 Messages

Similar to all components of the MVS operating system, JES2 maintains its own set of informational and error messages. Each is typically prefixed by \$HASP. That is, the HASP prefix is constant; however, you can change the dollar sign (\$) to another character. You must alter the prefix character for a secondary JES2 (which you might have if, for example, the primary JES2 supports normal production work while the secondary supports a test system). To distinguish between messages issued from the two systems, JES2 provides the ability to set the symbol to another unique character. The prefix symbol can also be changed to provide for national language support. For example, the pound sterling symbol (£) is often the symbol of choice in England.

There are approximately 500 JES2 messages in the range \$HASP000 to \$HASP949. (Each is documented in *OS/390 JES2 Messages*.) Message text ranges from the \$HASP000 message that simply reads OK (indicating that your request was received) to messages, such as \$HASP095 JES2 CATASTROPHIC ERROR, CODE=\$K04, that indicate JES2 has experienced a serious error that needs to be corrected immediately if processing is to continue. To assist the operator or system programmer, documentation for each message contains information such as: an explanation of the message text, the action that the operating system takes, and the recommended operator and/or system programmer responses.

Message numbers in the range \$HASP950 through \$HASP999 are reserved for your use. You can add installation-written messages by using the JES2 message build facility or the table pair facility. (Refer to *OS/390 JES2 Macros* for details on implementing the table pairs and the message build facility through use of JES2 macros.)

JES2 Tracing Facility

To record register contents and data at specific points in JES2 processing, you can use the JES2 tracing facility. JES2 has defined a number of the 255 trace types that can be defined; new ones are occasionally added to new product releases as their need arises. Here, again, is another JES2 facility that your installation can augment. (Your installation can add installation-specific trace points if the available trace functions do not meet your requirements.)

The use of the facility is optional. Tracing is initially activated under system programmer control (by specifying initialization statements) and subsequently controlled through several operator commands. Generally, trace points are only activated as a diagnostic tool for short periods of time.

Traced data can be accessed either through a dump of unformatted trace tables or in formatted system output. (Refer to *OS/390 JES2 Initialization and Tuning Guide* and *OS/390 JES2 Diagnosis* for further information regarding the initialization and use of the JES2 trace facility.)

JES2-IPCS Formatting

To facilitate diagnosis of errors that require the viewing of storage dumps or control blocks, both the base control program and JES2 components exploit the interactive problem control system (IPCS). When diagnosing a problem, IPCS panels provide you the opportunity to continue the diagnostic process through the base control program and into selected JES2 data areas.

IPCS is a menu-driven facility that lets you interactively select control blocks that you need to examine. Selected control blocks are formatted, displayed, and available for printing. (Refer to *OS/390 JES2 Diagnosis* for a discussion of IPCS-JES2 support.)

Appendix A. JES2 Publications

The following list of JES2 publications includes the basic set of those publications provided with the product.

OS/390 JES2 Migration — **GC28-1797**

Describes the system programming and operational changes that are required to migrate from selected releases of JES2 to the most current JES2 release.

OS/390 JES2 Initialization and Tuning Guide — **SC28-1791**

Contains descriptions of JES2 initialization process and the JES2 facilities and task descriptions for the system programmer to use the JES2 initialization statements and their parameters to use and tune those facilities.

OS/390 JES2 Initialization and Tuning Reference — **SC28-1792**

Contains complete descriptions of the JES2 initialization statements and their parameters to include syntax and defaults. Reference tables provide lists of hardware supported by JES2, JES2 values for optimizing hardware usage, and a copy of the sample initialization data set shipped in SYS1.VnRnMn.SHASSAMP.

OS/390 JES2 Commands — **GC28-1790**

Describes how to operate JES2. The book explains the use of each JES2 operator command in detail.

OS/390 JES2 Commands Summary — **GX22-0041**

Contains a summary of all JES2 commands presented in *OS/390 JES2 Commands*.

OS/390 JES2 Messages — **GC28-1796**

Describes all JES2 messages and suggests appropriate operator and system programmer responses.

OS/390 JES2 Installation Exits — **SC28-1793**

Provides information to customize JES2 through JES2 installation exits and describes how to use JES2 programmer macro instructions.

OS/390 JES2 Macros — **SC28-1795**

Describes how to use JES2 programmer macro instructions.

OS/390 JES2 Diagnosis — **SY28-1086**

Describes procedures to use and interpret the output of diagnostic aids/tools and their output. The book assists the system programmer to: identify a problem, collect information about the problem, report the problem to IBM, and fix the problem.

Appendix B. Notices

This information was developed for products and services offered in the USA. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This book documents information that is NOT intended to be used as Programming Interfaces of OS/390.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- AFP/VTAM
- Advanced Function Printing (AFP)
- AnyNet
- BookManager
- CBPDO
- CICS
- DFSMS/MVS
- DFSMSdfp
- DFSMSdss
- DFSMShsm
- DFSMSrmm
- ESCON
- ES/3090
- ES/9000
- GDDM
- IBM
- IBMLink
- IMS
- MVS/DFP
- MVS/ESA
- NetView
- OS/2

- OS/390
- PR/SM
- Print Services Facility (PSF)
- RMF
- RACF
- RMF
- SOMobjects
- SP
- SP1
- SP2
- SystemView
- System/36
- System/360
- System/370
- System/390
- VTAM

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines technical terms and abbreviations used in JES2 documentation. If you do not find the term you are looking for, refer to the index of the appropriate JES2 manual or view *IBM Dictionary of Computing*, located at:

<http://www.ibm.com/networking/nsg/nsgmain.htm>

This glossary includes terms and definitions from:

American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by an asterisk (*) that appears between the term and the beginning of the definition; a single definition taken from ANSI is identified by an asterisk after the item number for that definition.

A

ACB. Access control block

ACF. Advanced communication function

address space. The complete range of addresses available to a program. See also *virtual address space*.

affinity. The condition under which one or more members of a JES2 multi-access spool configuration may be designated as qualified to execute certain jobs.

AFP. Advanced function printing

all-member warm start. A JES2 member restart of the first member in a multi-access spool (MAS) configuration. Either the JES2 member previously ended without error or there must be an IPL of the MVS system.

allocate. To assign a resource for use in performing a specific task.

APAR. Authorized program analysis report

APPC. Advanced Program-to-Program Communication.

APT. Application table

artificial JQE. An artificial JQE consists of the base JQE, the JQX, and additional fields defined in the JQA.

automatic restart. A restart that takes place during the current run, that is, without resubmitting the job. An

automatic restart can occur within a job step or at the beginning of a job step. Contrast with *deferred restart*. See also *checkpoint restart*.

automatic volume recognition (AVR). A feature that allows the operator to mount labeled volumes on available I/O devices before the volumes are needed by a job step.

AVR. Automatic volume recognition

B

background. (1) In multiprogramming, the environment in which low-priority programs are executed. (2) Under TSO/E the environment in which jobs submitted through the SUBMIT command or SYSIN are executed. One job step at a time is assigned to a region of central storage, and it remains in central storage to completion. Contrast with *foreground*.

background job. (1) A low-priority job, usually a batched or non-interactive job. (2) Under TSO, a job entered through the SUBMIT command or through SYSIN. Contrast with *foreground job*.

BAL. Basic assembler language

batch processing. (1) *Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started. (2) *Pertaining to the sequential input of computer programs or data. (3) *Loosely, the serial execution of computer programs. (4) Under TSO, the processing of one job step in a region, so called because jobs are submitted in a group or batch.

baud. (1) A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one-half dot cycle per second in Morse code, one bit per second in a train of binary signals, and one 3-bit value per second in a train of signals each of which can assume one of eight different states. (2) In asynchronous transmission, the unit of modulation rate corresponding to one unit interval per second; that is, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud.

binary synchronous communication (BSC). Communication using binary synchronous transmission.

binary synchronous transmission. Data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

bind. In SNA products, a request to activate a session between two logical units.

broadcast data set. Under TSO, a system data set containing messages and notices from the system operator, administrators, and other users. Its contents are displayed to each terminal user when he logs on the system, unless suppressed by the user.

BSAM. Basic sequential access method

BSC. Binary synchronous communication

BSCA. Binary synchronous communication adapter

burst. *To separate continuous-form paper into discrete sheets.

C

cataloged data set. A data set that is represented in an index or hierarchy of indexes that provide the means for locating it.

cataloged procedure. A set of job control statements that has been placed in a library and that can be retrieved by name.

CCW. Channel command word

central storage. (1) In System/390 virtual storage systems, the storage of a System/390 computing system from which the central processing unit can directly obtain instructions and data, and to which it can directly return results. (Formerly referred to as "real storage".) (2) Synonymous with *processor storage*.

centralized control. Control in which all the primary station functions of the data link are centralized in one data station. Contrast with *independent control*.

CES. Connection event sequence

chain printer. An impact printer that has a revolving chain with links that carry the type slugs.

change log. Area of the checkpoint data set that contains the specific control blocks changed by the last member of the multi-access spool configuration to own the checkpoint data set.

channel-to-channel (CTC). A method of connecting two computing devices.

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

checkpoint. (1) *A place in a routine where a check, or a recording of data for restart purposes, is

performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be later started. (3) To record information about the status of a job and the system for restart purposes.

checkpoint data set. A data set in which information about the status of a job and the system can be recorded so that the job step can be restarted later.

checkpoint reconfiguration. A process used by JES2 to dynamically redefine checkpoint data set specifications for a JES2 MAS.

checkpoint reconfiguration dialog. An interactive form of a JES2 checkpoint reconfiguration where the operator directs the reconfiguration process with replies to a series of WTOR messages.

checkpoint restart. The process of resuming a job at a checkpoint within the job step that caused abnormal termination. The restart may be automatic or deferred, where deferred restart involves resubmitting the job. See also *automatic restart*; *deferred restart*. Contrast with *step restart*.

checkpoint write. Any write to the checkpoint data set. A general term for the primary, intermediate, and final writes that update any checkpoint data set.

checkpoint/restart facility. (1) A facility for restarting execution of a program at some point other than at the beginning, after the program was terminated due to a program or system failure. A restart can begin at a checkpoint or from the beginning of a job step, and uses checkpoint records to reinitialize the system. (2) Under TCAM, a facility that records the status of the teleprocessing network at designated intervals or following certain events. Following system failure, the system can be restarted and continue without loss of messages.

checkpointing. Preserving processing information during a program's operation that allows such processing to be restarted and duplicated.

CKPT1. The checkpoint data set designed as the one on which the reserve is acquired. In a DUAL mode configuration, CKPT1 is one of the alternately used primary data sets from which JES2 reads and writes the checkpoint. In a DUPLEX mode configuration, CKPT1 is the primary checkpoint data set.

CKPT2. In a DUAL mode configuration, CKPT2 is one of the alternately-used checkpoint data sets from which JES2 reads and writes the checkpoint. In a DUPLEX mode configuration, CKPT2 is the back-up copy (generally down-level) of the primary checkpoint data set (CKPT1) which can be used to replace CKPT1 if necessary. CKPT2 is formatted the same as CKPT1.

(Previously CKPT2 was the DUPLEX checkpoint data set).

CLPA. Common link pack area

CMB. Console message buffer

CMS. Cross memory services

cold start. A JES2 member start that initializes data areas and accounting information in central storage and the job and output queues.

communication line. Any physical link, such as a wire or telephone circuit, for connecting geographically dispersed computer systems.

complex. The maximum set of hardware and software resources that support one or more images of a single operating system.

configuration. The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units.

connection event sequence. A clock value that indicates the time a connection took place or was broken. This is copied to NCC records and used by the path manager to determine the "most current" record when keeping track of NJE connections.

console. Any device from which operators can enter commands or receive messages. For JES2, the same device from which an operator also enters MVS base control program commands.

control statements. Statements placed into an input stream to identify special JES2 processing options for jobs.

CSA. Common service area

CSECT. Control section

CTC. Channel-to-channel adapter

D

DASD. Direct access storage device

data integrity point. The generic name given to the point in the 3800 model 3 printing process at which the data is known to be secure. (Also called the stacker.)

data set forwarding. The dynamic replacement of the checkpoint data set specifications (data set name and volume) with new specifications.

data set separator pages. Those pages of printed output that delimit data sets.

DCT. Device control table

deallocate. To release a resource that is assigned to a specific task.

dedicated. Pertaining to the assignment of a system resource - a device, a program, or a whole system - to an application or purpose.

deferred restart. A restart performed by the system when a user resubmits a job. The operator submits the restart deck to the system through a system input reader. See also *checkpoint restart*. Contrast with *automatic restart*.

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

despooling. The process of reading records off the spool into central storage. During the despooling process, the physical track addresses of the spool records are determined.

destination. A combination of a node name and one of the following: a userid, a remote printer or punch, a special local printer, or LOCAL (the default if only a node name is specified).

destination identifier (destid). The 8-character subscript on the DESTID(jxxxxxx) initialization statement or command that corresponds to a combination of a first-level destination and a second-level destination that determines where data should be sent in a JES2 installation. A destid can be either a symbolic destination or an explicit destination.

destination node. Node to which data is sent.

device partitioning. A pool of devices (called a fence) to be used exclusively by a set of jobs in a specific job class allowing an installation to tailor its device usage to its anticipated workload.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data.

DJC. Dependent job control.

DUAL mode. A checkpointing mode that provides the alternate use of two primary checkpoint data sets (CKPT1 and CKPT2). The data sets are referred to as the to-be-read-from and to-be-written-to data sets.

dump. A report showing the contents of storage. Dumps are typically produced following program failures, for use as diagnostic aids.

DUPLEX mode. A checkpointing mode that provides the continuous use of only one checkpoint data set. A

second (backup) data set is defined, but it is written to less frequently than the primary.

dynamic allocation. Assignment of system resources to a program at the time the program is executed rather than at the time it is loaded into central storage.

dynamic connection. A connection created via sign-on or NCC record sent from another node
Synonymous with *non-static connection*.

dynamic table. An installation-defined table that is used to extend, modify, or delete the JES2 default processing specifications. See also *table pair*.

E

EBCDIC. Extended binary coded decimal interchange code

ECSA. Extended common service area

EM. End of media

end of block (EOB). A code that marks the end of a block of data.

end-of-file mark (EOF). A code that signals that the last record of a file has been read.

EOB. End of block

EOF. End of file

EPVT. Extended private storage area

execution node. The JES2 network job entry node upon which a job is to be executed.

exit points. The place in the code where a routine (exit) receives control from the system.

explicit destination. A destination identifier of the form Nnnnn, Rmmmm, RMmmmm, RMTmmmm, NnnnnRmmmm or Unnnn. See also *destination identifier* and *symbolic destination*.

extended binary coded decimal interchange code (EBCDIC). A set of 256 characters, each represented by 8 bits.

external writer. A program that supports the ability to write SYSOUT data in ways and to devices not supported by the job entry subsystem.

F

facility. (1) A feature of an operating system, designed to service a particular purpose, for example, the checkpoint/restart facility. (2) A measure of how easy it is to use a data processing system. Together with system performance, a major factor on which the total productivity of an installation depends. (3) Anything used or available for use in furnishing communication service. Commonly, a general term for communications paths.

FCB. Forms control buffer

final write. A write of the same information as the intermediate write done at the end of the checkpoint cycle. See also *intermediate write*.

first-level destination. The nodal portion of a destination (the node to which the data goes).

foreground. (1) in multiprogramming, the environment in which high-priority programs are executed. (2) Under TSO, the environment in which programs are swapped in and out of central storage to allow CPU time to be shared among terminal users. All command processor programs execute in the foreground. Contrast with *background*.

foreground job. (1) A high-priority job, usually a real-time job. (2) A teleprocessing or graphic display job that has an indefinite running time during which communication is established with one or more users at local or remote terminals. (3) Under TSO, any job executing in a swapped region of central storage, such as a command processor or a terminal user's program. Contrast with *background job*.

forms control buffer (FCB). A buffer that is used to store vertical formatting information for printing; each position corresponding to a line on the form.

forwarding. The dynamic replacement of the checkpoint data set specifications (data set name and volume) with new specifications.

FSA. Functional subsystem application

FSA startup. That part of system initialization when the FSA is loaded into the functional subsystem address space and begins initializing itself.

FSI. Functional subsystem interface

FSI connect. The FSI communication service which establishes communication between JES2 and the FSA or functional subsystem.

FSI disconnect. The FSI communication service which severs the communication between JES2 and the FSA or functional subsystem.

FSI services. A collection of services available to users (JES2) of the FSI. These services comprise communication services, data set services, and control services.

FSS. Functional subsystem

full-function mode. The state that permits a printer to produce page-mode output.

functional subsystem (FSS). An address space uniquely identified as performing a specific function related to the JES. For JES2, an example of an FSS is the Print Services Facility program that operates the 3800 Model 3 and 3820 printers.

functional subsystem application (FSA). The functional application program managed by the functional subsystem.

functional subsystem interface (FSI). The interface through which JES2 or JES3 communicate with the functional subsystem.

functional subsystem startup. That process part of system initialization when the functional subsystem address space is created.

G

global command. A command that is recognized and honored by any node in a JES2 network.

global processor. In JES3, the processor that controls job scheduling and device allocation for a complex of processors.

GMT. Greenwich mean time.

Greenwich mean time (GMT). The mean solar time of the meridian of Greenwich used as the prime basis of standard time throughout the world. See also *TOD clock*.

H

handshaking. Exchange of predetermined signals when a connection is established between two data set devices.

HASP. Houston automatic spooling priority. A computer program that provides supplementary job management, data management, and task management

functions, such as: control of job flow, ordering of tasks, and spooling. See also *JES2*.

HASP table. See *JES2 table*.

HCT. HASP communication table

host processor. (1) *In a network, the processing unit in which resides the access method for that network. (2) In an SNA network, the processing unit that contains a system services control point (SSCP).

host system. *The data processing system to which a network is connected and with which the system can communicate.

host-id. The unique 10-digit CPU identification made up of the 6-digit CPU serial number followed by a 4-digit model number.

hot start. A JES2 member restart performed when a member ends abnormally and the MVS system is not re-IPLed.

I

I/O. input/output

IBM-defined exit. The point in source code where IBM has added an exit point where an installation routine can receive control from the operating system. Contrast with *installation-defined exit*.

impact printer. *A printer in which printing results from mechanical impacts.

independent control. In JES2, the process by which each processor in a complex controls its own job input, scheduling, and job output. Contrast with *centralized control*.

independent mode. A means of isolating a processor for testing purposes. A processor so designated will only process jobs that are both routed to it and are themselves designated to execute on a processor in independent mode.

initial program load (IPL). The initialization procedure that causes an operating system to commence operation.

initialization data set. The data set that contains the initialization statements and their parameters that controls the initialization and ultimate processing of JES2.

initialization parameter. An installation-specified parameter that controls the initialization and ultimate operation of JES2.

initialization statement. An installation-specified statement that controls the initialization and ultimate operation of JES2.

initiating task. The job management task that controls the selection of a job and the preparation of the steps of that job for execution.

initiator. That part of an operating system that reads and processes operation control language statements from the system input device.

initiator/terminator. The job scheduler function that selects jobs and job steps to be executed, allocates input/output devices for them, places them under task control, and at completion of the job, supplies control information for writing job output on a system output unit.

input service processing. In JES2, the process of performing the following for each job: reading the input data, building the system input data set, and building control table entries.

input stream control. Synonymous with *JES2 reader*.

installation-defined exit. The point in source code where an installation adds an exit point where an installation routine can receive control from the operating system. Contrast with *IBM-defined exit*.

interface. Hardware, software, or both, that links systems, programs, or devices.

intermediate write. In DUAL mode, the write of the change log records containing the control blocks that have been updated since the last checkpoint write. In DUPLEX mode (or DUAL mode where the change log overflows the first track) the checkpoint write of the 4K records.

internal reader. A facility that transfers jobs to JES.

interrupt. (1) *To stop a process in such a way that it can be resumed. (2) In data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.

IOT. input/output table

IPL. initial program load

IPS. Installation performance specification

J

JCL. Job control language

JCT. Job control table

JES2. Job entry subsystem 2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each processor's JES2 subsystem independently controls job input, scheduling, and output processing.

JES2 reader. In MVS, the part of the job entry subsystem that controls the input stream and its associated job control statements. Synonymous with *input stream control*.

JES2 table. A JES2-defined table that is used to specify the default characteristics of many of its initialization parameters, commands, and other externals. See also *table pair*.

JES2 writer. In MVS, the part of the job entry subsystem that controls the output of specified data sets. Synonymous with *output stream control*.

JES3. Job entry subsystem 3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with multiple processors (a JES3 complex), one processor's JES3 subsystem exercises centralized control over the other processors and distributes jobs to them through use of a common job queue.

JIX. Job queue index

JMR. Job management record

job. A unit of work for an operating system. Jobs are defined by JCL statements.

job class. Any one of a number of job categories that can be defined. With the classification of jobs and direction of initiator/terminators to initiate specific classes of jobs, it is possible to control the mixture of jobs that are performed concurrently.

job control language (JCL). A programming language used to code job control statements.

job control language (JCL) statements. Statements placed into an input stream to define work to be done, methods to be used, and the resources needed.

job control statement. *A statement in a job that is used in identifying the job or describing its requirements to the operating system.

job entry subsystem (JES). An MVS facility that receives jobs into the system and processes output data produced by the jobs. See also *JES2* and *JES3*.

job entry subsystem 2. See *JES2*.

job entry subsystem 3. See *JES3*.

job output element (JOE). Information that describes a unit of work for the output processor and represents that unit of work for queuing purposes.

job priority. A value assigned to a job that is used as a measure of the job's relative importance while the job contends with other jobs for system resources.

job queue element (JQE). A control block that represents an element of work for the system (job) and is moved from queue to queue as that work moves through each successive stage of JES2 processing.

job separator page data area (JSPA). A data area that contains job-level information for a data set. This information is used to generate job header, job trailer or data set header pages. The JSPA can be used by an installation-defined JES2 exit routine to duplicate the information currently in the JES2 separator page exit routine.

job separator pages. Those pages of printed output that delimit jobs.

JOE. Job output element

JOT. Job output table

K

keyword. A part of a command operand that consists of a specific character string (such as *DSNAME=*).

keyword parameter. A parameter that consists of a keyword, followed by one or more values. Contrast with *positional parameter*. See also *parameter*.

L

label. (1) *One or more characters used to identify a statement or an item of data in a computer program. (2) An identification record for a tape or disk file.

line mode. A type of data with format controls that only allow a printer to format data as a line.

line-mode data. A type of data that is formatted on a physical page by a printer only as a single line.

LMT. Load module table

local devices. Those devices that are directly attached to the operating system without the need for transmission facilities.

local processing environment. The collection of devices all of which are locally attached. That is, they are connected without the need for transmission facilities.

local system queue area (LSQA). In MVS, one or more segments associated with each virtual storage region that contain job-related system control blocks.

locally attached. A manner of device connection without the need for transmission facilities.

logical unit (LU). The combination of programming and hardware of a teleprocessing subsystem that functions like a terminal to VTAM.

logoff. (1) The procedure by which a user ends a terminal session. (2) In VTAM, a request that a terminal be disconnected from a VTAM application program.

logon. (1) The procedure by which a user begins a terminal session. (2) In VTAM, a request that a terminal be connected to a VTAM application program.

loop. A situation in which an instruction or a group of instructions execute repeatedly.

LPA. Link pack area

LRECL. Logical record length

LSQA. Local system queue area

LU. Logical unit

M

machine check interruption. An interruption that occurs as a result of an equipment malfunction or error.

MAS. See *multi-access spool configuration*.

MCS. Multiple console support

member. A JES2 instance of a MVS system

message. For communication lines, a combination of characters and symbols transmitted from one point to another. See also *operator message*.

MIT. Module information table

MLU. Multiple logical unit

multi-access spool complex. See *multi-access spool configuration*.

multi-access spool configuration. Multiple systems sharing the JES2 input, job and output queues (via a checkpoint data set or coupling facility) .

multi-access spool multiprocessing. Two or more computing systems interconnected by an I/O channel-to-channel adapter. The CPs can be different types and have their own unique configurations.

multiple console support (MCS). A feature of MVS that permits selective message routing to up to 32 operator's consoles.

Multiple Virtual Storage (MVS). An operating system that manages resources and work flow while jobs are running.

multiprocessing. (1) *Pertaining to the simultaneous execution of two or more computer programs or sequences of instructions by a computer network. (2) *Loosely, parallel processing. (3) Simultaneous execution of two or more sequences of instructions by a multiprocessor.

multiprocessing system. A computing system employing two or more interconnected processing units to execute programs simultaneously.

multiprocessor. (1) A computer employing two or more processing units under integrated control. (2) A system consisting of two or more CPs (or ALUs, or processors) that can communicate without manual intervention.

MVS. Multiple virtual storage.

N

NACT. Network account table

NAT. The nodes attached table, which is an internal JES2 control block containing information about each pair of nodes connected, or recently disconnected.

NCC record. The network connection and control records.

NCP. Network control program

NCP/VS. Network control program/VS

NDH. Network data set header

network. For JES2, two or more systems and the connections over which jobs and data are distributed to the systems. The other systems can be non-JES2 systems with compatible networking facilities. Connections can be established through communications paths using SNA or BSC protocols.

network job entry (NJE). A JES2 facility that provides for the passing of selected jobs, system output data, operator commands, and messages between communicating job entry subsystems connected by binary-synchronous communication lines, channel-to-channel adapters, and shared queues.

Network Job Entry (NJE) facility. In JES2, a facility which provides for the transmission of selected jobs, operator commands, messages, SYSOUT data, and accounting information between communicating job entry nodes that are connected in a network either by binary synchronous communication (BSC) lines channel-to-channel (CTC) adapters, or by System Network Architecture (SNA).

Network Job Entry facility. In JES2, a facility which provides for the transmission of selected jobs, operator commands, messages, SYSOUT data, and accounting information between communicating job entry nodes that are connected in a network either by binary synchronous communication (BSC) lines or by channel-to-channel (CTC) adapters.

network operator. (1) The person responsible for controlling the operation of a telecommunication network. (2) A VTAM application program authorized to issue network operator commands.

NIP. Nucleus initialization program.

NIT. The node information table, which is an internal JES2 control block containing information about each NJE node.

NJE. Network job entry

NJH. Network job header

node. (1) One of the systems in a network of systems connected by communication lines or CTC adapters. (2) In VTAM, an addressable point in a telecommunication system defined by a symbolic name. (3) In JES2 NJE, one or more job entry subsystems sharing a common job queue.

node name. An 8-character alphameric name which represents a node to other parts of the NJE network.

non-impact printer. *A printer in which printing is not the result of mechanical impacts; for example, thermal printers, electrostatic printers, photographic printers.

non-static connection. A connection created via sign-on or NCC record sent from another node
Synonymous with *dynamic connection*.

nonpageable dynamic area. *In MVS, an area of virtual storage whose virtual addresses are identical to real addresses; it is used for programs or parts of

programs that are not to be paged during execution. Synonymous with *V=R dynamic area*.

nonpageable region. In MVS, a subdivision of the nonpageable dynamic area that is allocated to a job step or system task that is not to be paged during execution. In a nonpageable region, each virtual address is identical to its real address. Synonymous with *V=R region*.

nucleus. That portion of a control program that always remains in central storage.

nucleus initialization program (NIP). The MVS component that initializes the resident control program.

O

offload. Moving jobs and work off the work queues to remove them from contention for system resources, or off spool to free up system work space.

operand. (1) *That which is operated upon. An operand is usually identified by an address part of an instruction. (2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

operator commands. Statements that system operators may use to get information, alter operations, initiate new operations, or end operations.

operator message. A message from an operating system directing the operator to perform a specific function, such as mounting a tape reel; or informing the operator of specific conditions within the system, such as an error condition.

operator orientation point. The generic name given to the point in the 3800 model 3 printing process at which the data becomes visible to the operator, and is therefore the point at which all operator commands are directed. Synonymous with *transfer station*.

output group. A set of a job's output data sets that share output characteristics, such as class, destination, and external writer.

output stream control. Synonymous with *JES2 writer*.

output writer. A part of the job scheduler that transcribes specified output data sets onto a system output device independently of the program that produced the data sets.

overlays. A collection of predefined data such as lines, shading, text, boxes, or logos, that can be merged with the variable data on a page while printing.

P

page. (1) In virtual storage systems, a fixed-length block of instructions, data, or both, that can be transferred between central storage and external page storage. (2) To transfer instructions, data, or both, between central storage and external page storage. (3) The unit of output from a 3800-3 running with full function capability.

page. The unit of output from a 3800 model 3 running with full function capability or 3820 printer.

page data set. In System/390 virtual storage systems, a data set in external page storage in which pages are stored.

page fault. In System/390 virtual storage systems, a program interruption that occurs when a page that is marked "not in central storage" is referred to by an active page.

page mode. The mode of operation in which the 3800 Printing Subsystem can accept a page of data from a host processor to be printed on an all-points-addressable output medium.

page-mode data. A type of data that can be formatted anywhere on a physical page. This data requires specialized processing such as provided by the Print Services Facility for the 3800-3 and 3820 printers.

page-mode environment checkpointing. That process which preserves the information necessary to resume page-mode printing.

page-mode printer. A printer (such as the 3800 model 3 and 3820) that can print page-mode data.

pageable region. In MVS, a subdivision of the pageable dynamic area that is allocated to a job step or a system task that can be paged during execution. Synonymous with *V=V region*.

paging. In System/390 virtual storage systems, the process of transferring pages between central storage and external page storage.

paging device. In System/390 virtual storage systems, a direct access storage device on which pages (and possibly other data) are stored.

parameter. (1) *A variable that is given a constant value for a specific purpose or process. (2) See *keyword parameter*, *positional parameter*.

password. A unique string of characters that a program, computer operator, or user must supply to meet security requirements for gaining access to data.

patch. *To modify a routine in a rough or expedient way.

path. In VTAM, the intervening nodes and lines connected a terminal and an application program in the host CPU.

path manager. The part of JES2 that controls NJE sign-on, sign-off, keeps track of all other nodes and connections in the network, and determines the best path to reach those nodes. (JES2 is unique among other NJE subsystems in keeping track of the network topology through NCC records.)

PCE. Processor control element

Pddb. Peripheral data definition block

PEP. Partitioned emulator program

physical unit (PU). (1) The control unit or cluster controller of an SNA terminal. (2) The part of the control unit or cluster controller that fulfills the role of a physical unit as defined by systems network architecture (SNA).

PLPA. Pageable link pack area

poly-JES. Concurrent operation of multiple copies of JES2 on a single MVS system to allow an installation to separate its primary production system(s) and test system(s).

positional parameter. A parameter that must appear in a specified location, relative to other parameters. Contrast with *keyword parameter*. See also *parameter*.

PPL. Purge parameter list

PRE. Processor recovery element

primary write. The write of the 4K records to the down-level checkpoint data set to make it current.

Print Services Facility. The program (code) that operates the 3800 model 3 and 3820 printers. The Print Services Facility operates as a functional subsystem.

priority aging. A function of JES2 by which the longer a job waits to be selected for processing, the greater become its chances of being selected to run.

private connection. A connection known only to the two nodes making the connection.

process mode. The mode in which SYSOUT data exists and is to be processed by a JES output device. There are two IBM-defined process modes: line mode and page mode.

processor storage. See *central storage*.

program temporary fix (PTF). A temporary solution or bypass for a problem diagnosed by IBM as the result of a defect in a current unaltered release of the program.

protocols. Rules for using communication lines. Protocols can identify the direction of data flow, where data begins and ends, how much data is being transmitted, and whether data or control information is being sent.

PSF. Print Services Facility

PTF. Program temporary fix

PU. Physical unit.

Q

QSE. Shared queue element

queue. A line or list formed by items in a system waiting for processing.

quick start. A JES2 member restart in an existing multi-access spool (MAS) configuration. The JES2 member previously ended without error.

quiescing. *The process of bringing a device or a system to a halt by rejection of new requests for work.

R

RACF. Resource Access Control Facility

read 1. A read of the first track of a checkpoint data set. Usually performed as the initial I/O operation to a checkpoint data set.

read 2. A read of the 4K page data records and any change log records not contained on the first track from a checkpoint data set. Usually performed after a READ 1 as the second checkpoint I/O operation in a checkpoint cycle.

reader. A program that reads jobs from an input device or data base file and places them on the job queue.

real address. In virtual storage systems, the address of a location in central storage.

real storage. See *central storage*.

remote. RMT

remote job entry (RJE). Submission of job control statements and data from a remote terminal, causing the jobs described to be scheduled and executed as though encountered in the input stream.

remote station. *Data terminal equipment for communicating with a data processing system from a location that is time, space, or electrically distant.

remote terminal. An input/output control unit and one or more input/output devices attached to a system through a data link.

remote terminal access method (RTAM). A facility that controls operations between the job entry subsystem (JES2) and remote terminals.

remote workstation. (1) *Data terminal equipment for communicating with a data processing system from a location that is time, space, or electrically distant. Synonymous with *remote station*. (2) A workstation that is connected to a system by means of data transmission facilities.

RJE. Remote job entry

RMS. Recovery management support

RMT. Remote

RMT generation. Generation of remote workstations for remote job entry.

routing. (1) The assignment of the communications path by which a message or telephone call will reach its destination. (2) In NJE, the path, as determined by NJE or explicitly by the operator, that a job or SYSOUT data set will take to reach its destination.

routing code. A code assigned to an operator message and used, in systems with multiple console support (MCS), to route the message to the proper console.

RPL. Request parameter list

RPS. Rotational position sensing

RTAM. Remote terminal access method

RTP. Remote terminal program

S

SAF. Security authorization facility

SAM. Sequential access method

SDLC. Synchronous data link control

| **SDSB.** Spool data set browse

second-level destination. Specifies a remote workstation, special local route code, userid, or LOCAL or ANYLOCAL (for data not associated with a specific routing).

secondary console. In a system with multiple consoles, any console except the master console. The secondary console handles one or more assigned functions on the multiple console system.

security classification. (1) An installation-defined level of security printed on the separator pages of printed output. (2) In RACF, the use of security categories, a security level, or both, to impose additional access controls on sensitive resources. An alternative way to provide security classifications is to use security labels.

segments. A collection of composed text and images, prepared before formatting and included in a document when it is printed.

session. (1) The period of time during which a user of a terminal can communicate with an interactive system; usually, the elapsed time from when a terminal is logged on to the system until it is logged off the system. (2) The period of time during which programs or devices can communicate with each other. (3) In VTAM, the period of time during which a node is connected to an application program.

setup. The preparation of a computing system to perform a job or job step. Setup is usually performed by an operator and often involves performing routine functions, such as mounting tape reels and loading card decks.

shared broadcasting. The two TSO data sets SYS1.UADS (TSO user definition) and SYS1.BROADCAST (TSO message transmission definition) are shared by all systems in the multi-access spool (MAS) complex.

simultaneous peripheral operations online (spool). The reading and writing of input and output streams on auxiliary storage devices, concurrently while a job is running, in a format convenient for later processing or output operations.

single-member warm start. A JES2 member restart of a new member in an existing multi-access spool (MAS) configuration. The JES2 member previously ended abnormally. Before the restart can occur, there must be an IPL of the MVS system.

single-processor complex. A processing environment in which only one processor (computer) accesses the spool and comprises the entire node.

SMF. System management facilities

SNA. Systems Network Architecture

special local. A routing in the form Unnnn, where 'nnnn' signifies a numeric value in the range of

1–32767. Usually, installations use this routing to specify local printers and punches.

spin data set. A data set that is deallocated (available for printing) when it is closed. Spin off data set support is provided for output data sets just prior to the termination of the job that created the data set.

spool. Simultaneous peripheral operations online.

spooled data set. A data set written on an auxiliary storage device and managed by JES.

| **spooled data set browse (SDSB).** An application that
| allows a program to read spool data sets.

spooling. The reading and writing of input and output streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

SQA. System queue area

SRM. System resources manager

static connection. A connection (also called “predefined connection” in earlier releases) between two nodes created by either a JES2 initialization or an operator command.

STC. Started task control

step restart. A restart that begins at the beginning of a job step. The restart may be automatic or deferred, where deferral involves resubmitting the job. Contrast with *checkpoint restart*.

subnet. Subset of a NJE network identified by an eight-character ‘SUBNET’ name on the JES2 NODE initialization statement. The grouping of nodes into “SubNets” is based on the assumption that if you have access to any node in the subnet, you have access to them all.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

SVC. Supervisor call instruction

SVC interruption. An interruption caused by the execution of a supervisor call instruction, causing control to be passed to the supervisor.

SWA. Scheduler work area

swap data set. A data set dedicated to the swapping operation.

swapping. An MVS paging operation that writes the active pages of a job to auxiliary storage and reads

pages of another job from auxiliary storage into central storage.

symbol. (1) *A representation of something by reason of relationship, association, or convention. (2) In MVS, a group of 1 to 8 characters, including alphanumeric characters and the three characters: #, @, \$. The symbol begins with either an alphabetic character or one of the three characters (#,@,\$).

symbolic address. *An address expressed in symbols convenient to the computer programmer.

symbolic destination. A destination identifier specifying a symbolic name that represents a JES2 destination. See also *destination identifier* and *explicit destination*.

synchronous data link control (SDLC). A discipline for managing synchronous, transparent, serial-by-bit information transfer over a communication channel. Transmission exchanges may be duplex or half-duplex over switched or nonswitched data links. The communication channel configuration may be point-to-point, multipoint, or loop.

syntax. (1) *The structure of expressions in a language. (2) The rules governing the structure of a language.

SYSIN. A system input stream; also, the name used as the data definition name of a data set in the input stream.

SYSLOG. System log

SYSOUT. A system output stream; also, an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system affinity. See *affinity*.

system control programming. IBM-supplied programming that is fundamental to the operation and maintenance of the system. It serves as an interface with program products and user programs and is available without additional charge.

system management facilities (SMF). An MVS component that provides the means for gathering and recording information that can be used to evaluate system usage.

system output writer. A job scheduler function that transcribes specified output data sets onto a system

output unit, independently of the program that produced the data sets.

system queue area (SQA). In MVS, an area of virtual storage reserved for system-related control blocks.

system services control point. *In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

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

T

table pair. A set of JES2-defined, USER-defined, and dynamic tables that an installation can use to modify JES2 processing.

TCAM. Telecommunications access method.

telecommunications access method (TCAM). A method used to transfer data between central storage and remote or local terminals. Application programs use either GET and PUT or READ and WRITE macro instructions to request the transfer of data, which is performed by a message control program. The message control program synchronizes the transfer, thus eliminating delays for terminal/output operations.

teleprocessing. The processing of data that is received from or sent to remote locations by way of telecommunication lines.

terminal. A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

text transparency. A provision that allows BSC to send and receive messages containing any or all of the 256 character combinations in EBCDIC, including transmission control characters. Transmission control characters sent in a message are treated as data unless they are preceded by the data link escape (DLE) control character.

TGB. Track group block

TGBE. Track group block entry

tightly-coupled multiprocessing. Two computing systems operating simultaneously under one control program while sharing resources.

Time Sharing Option Extensions (TSO/E). A licensed program that is based on the Time Sharing Option (TSO). It allows MVS users to interactively share computer time and resources.

time tolerance. The difference between the TOD clocks on two adjacent nodes, beyond which the path manager will not allow a session to be established.

time-of-day clock. See *TOD clock*.

TOD. Time-of-day

TOD clock. A timing device that counts units of time based on the starting point of 00 hours, 00 minutes, and 00 seconds on January 1, 1900. Time-of-day (TOD) information is used, for example, to monitor computer operations and events.

token. Specifically defined for JES2 checkpoint processing as a checkpoint identifier that is used to determine checkpoint I/O status.

trace. (1) The record of a series of events. (2) To record a series of a events as they occur.

trace. A report showing data relevant to a particular point in the processing of a program. Traces are typically produced for analysis of program performance, but they can also be valuable diagnostic aids.

tracing routine. *A routine that provides a historical record of specified events in the execution of a program.

traffic. In data communication, the quantity of data transmitted past a particular point in a path.

train printer. A printer in which the type slugs are assembled in a train that moves along a track. Contrast with *chain printer*.

transfer station. The point in the 3800 model 3 printing process at which the data set becomes visible to the operator, and is therefore the point at which all operator commands are directed. Synonymous with *operator orientation point*.

TSO. Time-sharing option. See *Time Sharing Option Extensions (TSO/E)*.

TSO/E. Time Sharing Option Extensions

TSU. Time-sharing user

TTE. Trace table entry

type font. In printing, a set of type that is of a particular size and style (for example, 10-point century school book).

U

UCB. Unit control block

UCS. Universal character set.

unallocate. See *deallocate*.

unit. (1) *A device having a special function. (2) A basic element.

unit address. The address of a particular device, specified at the time a system is installed; for example, 191 or 293.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays.

user identification (USERID). A 1-8 character symbol identifying a system user.

user table. An installation-defined table that is used to extend, modify, or delete the JES2 default processing specifications. See also *table pair*.

USERID. User identification.

V

V=R dynamic area. Synonymous with *nonpageable dynamic area*.

V=R region. Synonymous with *nonpageable region*.

V=V region. Synonymous with *pageable region*.

VIO. virtual input/output

virtual address space. In virtual storage systems, the virtual storage assigned to a job, terminal user, or system task. See also *address space*.

Virtual Telecommunications Access Method (VTAM). A set of programs that control communication between terminals and application programs running under MVS.

VTAM. Virtual Telecommunications Access Method.

W

warm start. A general term for a JES2 member restart. See also *hot start*; *quick start*; *single-member warm start*; *all-member warm start*.

writer. See *output writer*.

WTO. Write-to-operator

WTOR. Write-to-operator with reply

X

XFER. Transfer

XIT. Exit information table

XRT. Exit routine table

Numerics

3800 compatibility mode. Operating the 3800 model 3 printer as a 3800 Model 1 printer.

3800 model 3 startup. That process part of system initialization when the 3800 model 3 printer is initializing.

Index

Special Characters

\$HASP message

JES2 31

A

ACF/VTAM ix

See also VTAM

Advanced Program-to-Program Communication ix

See also APPC

aging

priority 19

AOC/MVS (Automated Operations Control/MVS)

JES2 message processing 31

APPC (Advanced Program-to-Program Communication)

JES2 SYSOUT support 23

attribute

print

for full-function printers 22

Automated Operations Control/MVS ix

See also AOC/MVS

automatic

JES2 command 31

automatic operation

of JES2 31

automatic operation program

AOC/MVS 31

NetView 31

B

binary synchronous communication ix

See also BSC

BSC (binary synchronous communication)

JES2 protocol 12

C

capability

of JES2 20

CBPDO (Custom Built Product Delivery Offering)

to build system 26

checkpoint 8

data set 8

reconfiguration dialog 8

cold start

of JES2 30

command

automatic 31

JES2 prefix character 30

operator 30

communication

with JES2 31

comparison

JES2 to JES3 3

component of MVS 1

configuration

JES2 7, 8

multiple system 10

of processors 8

single processor 9

control

centralized 4

data set 7

independent 4

operator 29

output 21

conversion

job phase 18

Custom Built Installation Process Offering ix

See also CBIPO

customization

JES2 25

IBM-defined exit usage 27

installation-defined exit usage 28

table pair usage 26

D

data set

checkpoint 8

JES2 initialization 25

sample 26

device

local 12

diagnostic tool

for JES2 31

IPCS 32

messages 31

trace 32

dialog

checkpoint reconfiguration 8

E

exit

IBM-defined 27

installation-defined 28

exit point

definition 27

extension

of JES2 complex 9

F

FSS (functional subsystem)

support 22

function

of JES2 20

functional subsystem ix

See also FSS

H

hard-copy

job phase 19

HASP (Houston Automatic Spooling Priority)

definition 1

Houston Automatic Spooling Priority ix

See also HASP

I

initialization

JES2 statements 25

input

job phase 18

interactive problem control system ix

See also IPCS

internal reader

definition 18

IPCS (interactive problem control system)

JES2 output 32

J

JES2

capability and function 20
communication mechanisms 31
configurations 7, 8
customization 25
data set control 7
device control 7
diagnostic tools 31
job scheduling 18
running multiple copies 10

JES3

in NJE network 15

job

entry
remote 10
process phase 2
scheduling
functions 19
JES2 18
priority aging 19

job control language ix

See also JCL

job phase

conversion 18

job phase (continued)

hard-copy 19
input 18
output 19
processing 18
purge 20

L

local

device 12

local complex

JES2 9

local output

processing 19

M

message

JES2 31
\$HASP 31
prefix character 31

modification

installation-written 25

multiple system

configuration 10

N

NetView

JES2 message process 31

network job entry ix

See also NJE

NJE (network job entry)

compared to RJE 13
example 14
network 13
node 13
possible systems 15
processing 20

node

in network 13

O

operation

automatic 31
of JES2 29

operator

control of JES2 29

output

control 21
job phase 19

P

phase

job process 2

poly-JES 10

prefix character 31

command 30

print

attributes

for full-function printers 22

print mode

full-function 22

page-mode 22

print/punch

job phase 19

printer

locally attached 12

remotely attached 12

processing

job phase 18

purge

job phase 20

R

RACF (Resource Access Control Facility)

for JES2 security 23

remote

terminal 13

workstation 13

remote job entry ix, 10

See also NJE

See also RJE

Resource Authorization Control Facility ix

See also RACF

RJE (remote job entry)

compared to NJE 13

example of use 11

workstation 10

S

SAF (Security Authorization Facility)

for JES2 security 23

SDLC (synchronous data link control)

JES2 protocol 12

security

in JES2 system 23

Security Authorization Facility ix

See also SAF

selection

of work 21

simultaneous peripheral operations online ix

See also spool

single-system

configuration 9

spool 7

multi-access 10

offload 22

start

cold

of JES2 30

warm

of JES2 30

statement

JES2 initialization

minimum set 26

synchronous data link control ix

See also SDLC

SYSOUT data set 19

T

table pair

for JES2 customization 26

JES2 table 26

user table 26

terminal

remote 13

Time Sharing Option Extended ix

See also TSO/E

tool

diagnostic

for JES2 31

trace

JES2 32

traffic

definition 14

transaction program

APPC

JES2 SYSOUT support 23

V

VM/RSCS

in NJE network 15

VSE/POWER

in NJE network 15

VTAM (Virtual Telecommunication Access Method)

as NJE access method 13

W

warm start

of JES2 30

work

selection 21

workstation

remote 13

RJE

definition 10

Communicating Your Comments to IBM

OS/390
JES2 Introduction
Publication No. GC28-1794-03

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
 - FAX: (International Access Code)+1+845+432-9405
- If you prefer to send comments electronically, use one of these network IDs:
 - Internet e-mail: mhvrcfs@us.ibm.com
 - World Wide Web: <http://www.ibm.com/s390/os390/webqs.html>

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

OS/390

JES2 Introduction

Publication No. GC28-1794-03

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: _____

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

- | | | | |
|--------------------------|-------------------------------|--------------------------|------------------------|
| <input type="checkbox"/> | As an introduction | <input type="checkbox"/> | As a text (student) |
| <input type="checkbox"/> | As a reference manual | <input type="checkbox"/> | As a text (instructor) |
| <input type="checkbox"/> | For another purpose (explain) | | |

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

Name

Address

Company or Organization

Phone No.

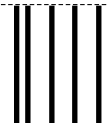


Cut or Fold
Along Line

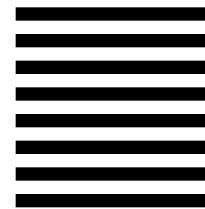
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1794-03

