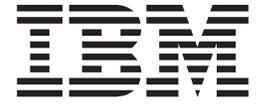
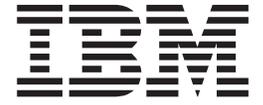


z/OS



JES3 Introduction

z/OS



JES3 Introduction

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 37.

Third Edition, September 2002

This is a major revision of SA22-7551-01.

This edition applies to Version 1 Release 4 of z/OS (5694-A01) and z/OS.e(5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM® welcomes your comments. A form for readers' comments may be provided at the back of this document, 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: mhvrcfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/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 document
- 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 1988, 2002. All rights reserved.**

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

Contents

Figures	v
About This Document	vii
Who Should Use This Document	vii
Where to Find More Information	vii
Additional Information	vii
Summary of Changes	1
z/OS Version 1 Release 4	1
z/OS Version 1 Release 2	1
Chapter 1. Introduction	3
Single-System Image	3
Workload balancing	4
Availability	4
Control Flexibility	4
Physical Planning Flexibility	5
Chapter 2. JES3 Environments	7
Single-Processor JES3 Environment	7
Multiprocessor JES3 Environment	8
Devices	10
Remote Job Processing JES3 Environment	13
JES3 Networking	16
Support for Advanced Program-to-Program Communication (APPC)	19
Chapter 3. JES3 from an Operator's Point of View	21
JES3 Operator Commands	21
Resource Management for Operators	21
Workflow Management for Operators	22
Utility Programs for Operators	22
Chapter 4. JES3 from an Application Programmer's Point of View	23
JES3 Control Statements	23
Deadline Scheduling	24
Dependent Job Control	24
Priority Aging	24
Chapter 5. JES3 from a System Programmer's Point of View	25
Resource Management for System Programmers	25
Workflow Management for System Programmers	26
Service Aids for System Programmers	29
Chapter 6. The JES3 Library	33
Appendix. Accessibility	35
Using assistive technologies	35
Keyboard navigation of the user interface	35
Notices	37
Programming Interface Information	38
Trademarks	39

Glossary	41
Index	51

Figures

1.	JES3 in a Single-Processor (Single-System Sysplex) Environment	7
2.	Spooling as a Method of Distributing Work to Processors	8
3.	Spooling as a Method of Collecting Job Output	9
4.	JES3 Multiprocessing in a Sysplex	10
5.	Remote Job Processing with JES3	14
6.	A Network Environment with JES3 and Other Systems	18
7.	Overview of JES3 Networking with SNA and BSC Protocols	19
8.	Placement of JES3 Control Statements	23

About This Document

This document introduces you to the Job Entry Subsystem 3 (JES3) and supports z/OS (5694-A01) and z/OS.e(5655-G52).

Who Should Use This Document

Whether you are an installation manager, system programmer, application programmer, operator, student, are currently using JES3, or are just considering or planning a JES3 installation this document will benefit you. Read this document to get a perspective on JES3 and to get the maximum benefit from the other documents in the JES3 library.

Where to Find More Information

The following table lists documents that contain information related to the information contained in this document. The table shows the complete titles, the short titles, and order numbers that are not listed in *z/OS Information Roadmap*. Refer to that document for all z/OS publications.

Most licensed documents were declassified in OS/390 V2R4 and are now included on the z/OS Online Library Collection, SKT2T-6700. The remaining licensed documents appear in unencrypted BookManager softcopy and PDF form on the z/OS Licensed Product Library, LK2T-2499.

Additional Information

Additional information about z/OS elements can be found in the following documents.

Title	Order Number	Description
<i>z/OS Introduction and Release Guide</i>	GA22-7502	Describes the contents and benefits of z/OS as well as the planned packaging and delivery of this new product.
<i>z/OS and z/OS.e Planning for Installation</i>	GA22-7504	Contains information that lets users: <ul style="list-style-type: none">• Understand the content of OS/390• Plan to get z/OS up and running• Install the code• Take the appropriate migration actions• Test the z/OS system
<i>z/OS Information Roadmap</i>	SA22-7500	Describes the information associated with z/OS including z/OS documents and documents for the participating elements.
<i>z/OS Summary of Message Changes</i>	SA22-7505	Describes the changes to messages for individual elements of z/OS. Note: This document is provided in softcopy only on the message bookshelf of the z/OS collection kit.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using

LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.¹

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Determining If a Publication Is Current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. Here's how to determine if you are looking at the most current copy of a publication:

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

1. At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.
2. If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
3. To compare softcopy publications, you can check the last two characters of the publication's filename (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk that indicates whether a publication is new or changed.

Summary of Changes

z/OS Version 1 Release 4

Summary of Changes for SA22-7551-02 z/OS Version 1 Release 4

This book contains information previously presented in *z/OS JES3 Introduction*, SA22-7551-01 which supported z/OS Version 1 Release 2. The following summarizes changes to that information:

New Information

- Information has been added to indicate this document supports z/OS.e.

Changed Information

- None

This book contains 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.

You may notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

z/OS Version 1 Release 2

Summary of Changes for SA22-7551-01 z/OS Version 1 Release 2

This book contains information previously presented in *z/OS JES3 Introduction*, SA22-7551-00 which supported z/OS Version 1 Release 1. The following summarizes changes to that information:

Changed Information

Updates to describe JES3 support of advanced function presentation (AFP) printers. See updates at “Supporting Advanced Function Presentation (AFP) Printers” on page 11 for details.

Chapter 1. Introduction

A major goal of operating systems is to process jobs while making the best use of system resources. Thus, one way of viewing operating systems is as resource managers. Before job processing, operating systems reserve input and output resources for jobs. During job processing, operating systems manage resources such as processors and storage. After job processing, operating systems free all resources used by the completed jobs, making the resources available to other jobs. This process is called *resource management*.

There is more to the processing of jobs than the managing of resources needed by the jobs. At any instant, a number of jobs can be in various stages of preparation, processing, and post-processing activity. To use resources efficiently, operating systems divide jobs into parts. They distribute the parts of jobs to queues to wait for needed resources. Keeping track of where things are and routing work from queue to queue is called *workflow management*, and is a major function of any operating system.

With the z/OS MVS JES3 system, resource management and workflow management are shared between MVS and its Job Entry Subsystem 3 (JES3) component. Generally speaking, JES3 does resource management and workflow management *before* and *after* job execution, while MVS does resource and workflow management *during* job execution.

JES3 considers job priorities, device and processor alternatives, and installation-specified preferences in preparing jobs for processing job output.

Features of the JES3 design include:

- Single-system image
- Workload balancing
- Availability
- Control flexibility
- Physical planning flexibility

Single-System Image

JES3 runs on either one processor or up to thirty-two processors in a sysplex. A *sysplex* is a set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

In a sysplex, your installation must designate one processor as the focal point for the entry and distribution of jobs and for the control of resources needed by the jobs. That processor, called the global processor, distributes work to the processors, called local processors.

It is from the global processor that JES3 manages jobs and resources for the entire complex, matching jobs with available resources. JES3 manages processors, I/O devices, volumes, and data. To avoid delays that result when these resources are not available, JES3 ensures that they are available before selecting the job for processing.

JES3 keeps track of I/O resources, and manages workflow in conjunction with the workload management component of MVS by scheduling jobs for processing on the processors where the jobs can run most efficiently. At the same time, JES3

maintains data integrity. JES3 will not schedule two jobs to run simultaneously anywhere in the complex if they are going to update the same data.

JES3 may be operated from any console that is attached to any system in the sysplex. From any console, an operator can direct a command to any system and receive the response to that command. In addition, any console can be set up to receive messages from all systems, or a subset of the systems in the sysplex. Thus, there is no need to station operators at consoles attached to each processor in the sysplex.

If you want to share input/output (I/O) devices among processors, JES3 manages the sharing. Operators do not have to manually switch devices to keep up with changing processor needs for the devices.

The JES3 architecture of a global processor, centralized resource and workflow management, and centralized operator control is meant to convey a single-system image, rather than one of separate and independently-operated computers.

Workload balancing

JES3 balances workload among processors in a way consistent with and in conjunction with the workload management (WLM) and system resources management (SRM) functions of MVS by considering the resource requirements of jobs. The method JES3 uses is the same whether one or several processors make up the configuration. Thus, addition of another processor does not mean a new operational and scheduling environment.

Availability

If a problem develops with the global processor, you can have one of the other processors assume the global functions. (Operators have JES3 commands to cause the switch.) Jobs running on other processors, including the one to become the new global processor, will usually be unaffected (except for a waiting period until global activities are resumed).

If part of JES3 fails, JES3 collects failure symptoms, records error data, and attempts recovery. All major JES3 components are protected by at least one special JES3 recovery routine. If recovery is unsuccessful, the failing component gets insulated from the rest of JES3, resources are freed, and the failing component will not be used again. If the component is not critical to overall JES3 processing, complete JES3 failure may be avoided.

Control Flexibility

Operating systems must be easy to control. Internal complexity must be offset by features that make the systems easy to operate, to monitor, and to change. JES3 has designed-in features for operators, application programmers, and system programmers:

- Operators have special JES3 commands. Some commands activate programs to handle I/O devices, while others obtain or change the status of jobs being processed. With multiple processing-unit systems, JES3 operators have less to do than for an equal number of individual systems, because they can control the entire complex from a central point, and because JES3 decides where and when jobs will be processed.

JES3 applies installation policies, as defined in the JES3 initialization stream, to perform job scheduling, thus freeing the operator from this task.

Even though JES3 handles job flow control, there are operational controls in JES3 for use at the discretion of the operator to override JES3 decisions, and to take control in unusual situations.

- Application programmers have special JES3 control statements (similar to JCL statements). There are control statements to make some jobs run only after successful (or unsuccessful) processing of other jobs, and for specifying the time-of-day, week, month, or even the year when jobs should run.
- System programmers have special JES3 initialization statements to define the way JES3 is to manage devices and jobs. Operators or application programmers can override many of these initialization options on a job-by-job basis. JES3 gives system programmers a unique way of setting installation policy for device and job management. They define separate groups of processing rules. Application programmers select and apply the groups of rules in various ways to individual jobs.

Where JES3 does not provide the exact function that your complex requires, such as special printing requirements, the system programmers can write their own routines and make them part of the JES3 program. This is done through installation exits provided with JES3 and through dynamic support programs that can be added to JES3. For diagnostic purposes, JES3 provides traces and dumps. Traces are copies of data made during normal processing to help monitor activity in the system, and dumps are copies of data made at times of failure.

Physical Planning Flexibility

You can give I/O devices specific assignments, and then put the devices in convenient places. For example, you can put card readers, card punches, and printers where programs get submitted and where output is needed. You can put I/O devices with removable volumes near tape and disk libraries, and you can have tape and disk librarians receive requests for the volumes on nearby consoles. Thus you can put processors into areas free of the congestion that often surrounds peripheral units.

You can define your MVS/JES3 complex so that only appropriate data, such as system messages, is sent to certain specified devices. This device tailoring capability lets you set up your system so that much of the activity at the main operator's console can be reduced.

Chapter 2. JES3 Environments

The work environment for JES3 consists of processors and I/O devices. JES3 covers a range of data processing needs, partly because it can accommodate various combinations of processors and devices. JES3 is a manager of its environment, so if you understand that environment, it will be easier for you to understand JES3.

This topic describes the I/O devices used by JES3 and MVS and the following environments in which JES3 runs:

- Single-processor environment
- Multiprocessor environment
- Remote job processing environment
- JES3 networking environment
- APPC environment

A JES3 complex can involve any combination of these environments.

Single-Processor JES3 Environment

Figure 1 shows JES3 in a single-processor environment, also known as a *single-system sysplex*. Besides the processor, two categories of I/O devices are shown: JES3 devices (those used by JES3); and JES3- and MVS-managed devices (those used by jobs). The spool device is a direct-access storage device (DASD) that is treated in a special way by JES3, so it is shown and explained separately.

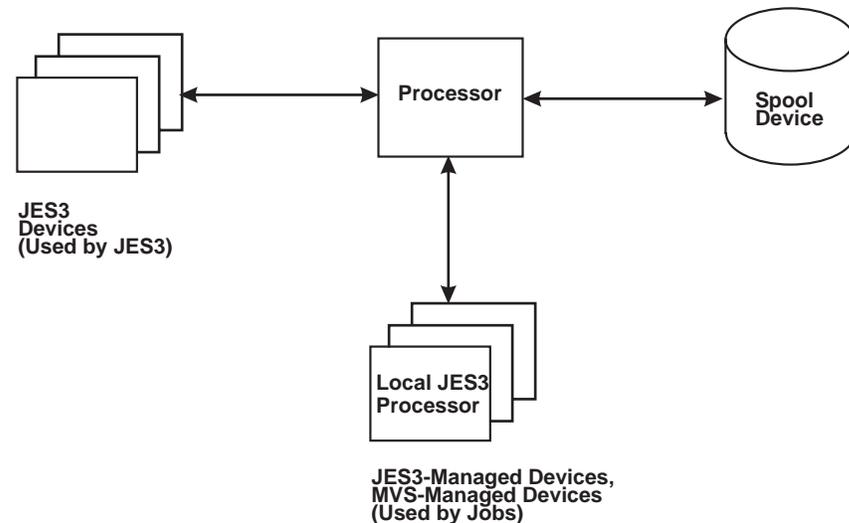


Figure 1. JES3 in a Single-Processor (Single-System Sysplex) Environment

The JES3 Processor

In installations with one processor, JES3 coexists in that processor with MVS and with running jobs. In many respects, JES3 is treated like a job by MVS. (MVS handles JES3 in much the same way it handles the jobs it receives from JES3 for processing). JES3 becomes a processable "job" during initialization, while jobs submitted to the system become processable by the actions of JES3.

Multiprocessor JES3 Environment

Multiprocessing is a way of doing work with two or more connected processors. In a multiprocessing environment, also known as a multisystem sysplex, JES3 allows up to 32 processors, also known as mains, to be configured into the complex. JES3 uses one processor (called the global) to do work and also to distribute work to up to 31 other processors (called locals). Some of the advantages of running JES3 in this environment are:

- Elimination of much of the overhead of scheduling work for and operating separate processors
- Sharing devices by processors, which means that the devices can be used more efficiently
- Movement of work to other processors, should one processor become overworked, need maintenance, or need to be removed from the complex for any reason.

Use of the Spool Device For Shared Data Storage

Most multiprocessing systems use shared data storage. In JES3 the spool device becomes the shared data storage. Thus, besides being a buffer (as for single-processor systems), in multiprocessing systems the spool device becomes a collection point for data to be distributed to individual processors (see Figure 2). Also, the spool device becomes a collection point for data coming from local processors routed to JES3 output devices connected to the global processor (see Figure 3).

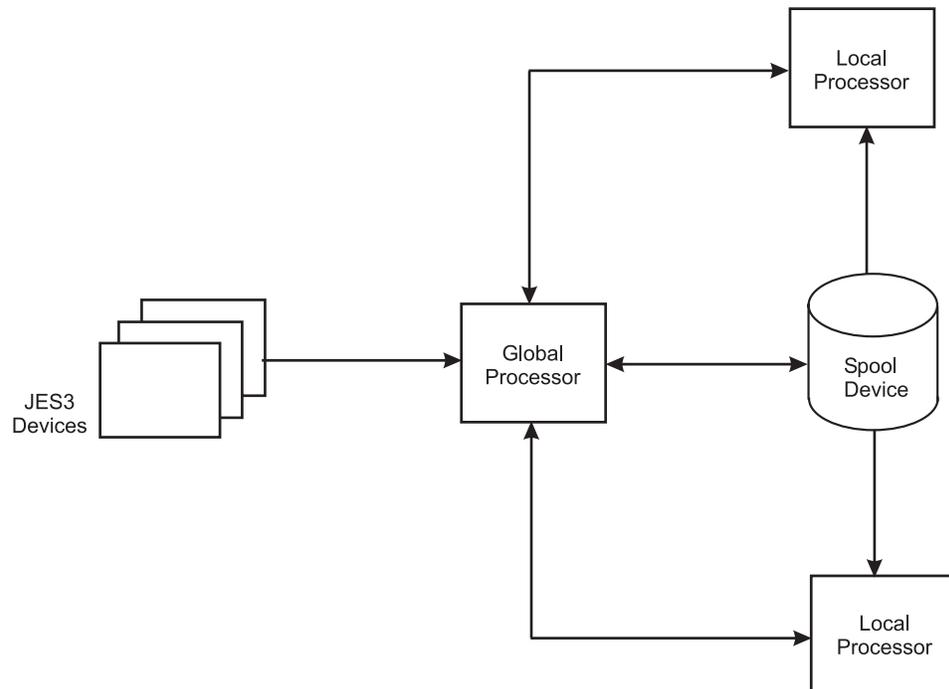


Figure 2. Spooling as a Method of Distributing Work to Processors

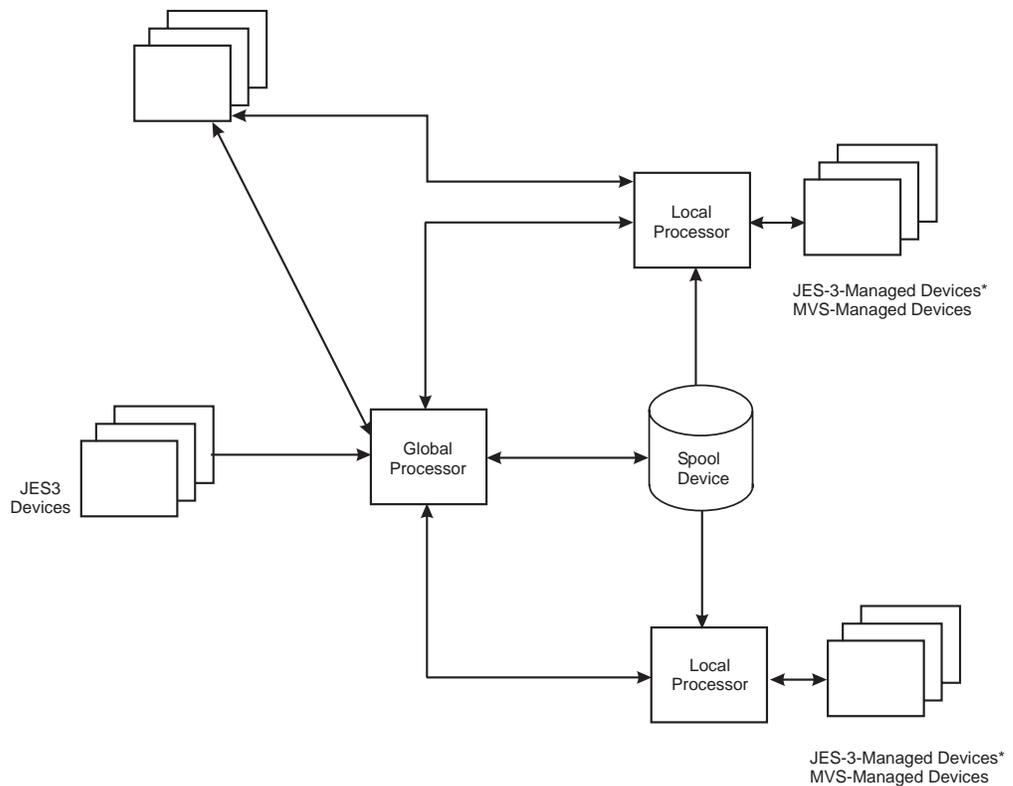


Figure 3. Spooling as a Method of Collecting Job Output

Processor Communications

In a JES3 multisystem sysplex environment, one processor is called the global processor while the other processors are called the local processors. In a JES3 complex there can be a maximum of 32 processors, one global and up to 31 locals. The global processor and the local processors communicate using the MVS cross-system coupling facility (XCF). JES3 devices (the devices from which JES3 reads jobs, and on which JES3 stores jobs awaiting processing, stores job output, and writes job output) are connected to the global processor. Some JES3 devices that write job output can also be connected to the local processors (not shown in Figure 3).

Figure 4 shows a JES3 multiprocessing system that has three processors (that communicate via XCF signalling). The global processor runs most of JES3, for it is from the global processor that JES3 allocates resources to jobs and sends jobs to local processors for processing. In addition, JES3 can also pass jobs to MVS in the global processor for processing.

Each local processor has a complete MVS system and JES3 routines for spooling and for communication with the global processor. When MVS in a local processor needs a job, JES3 in that local processor sends the job request to JES3 in the global processor via XCF signalling. JES3 in the global processor returns identifying information about a job, and JES3 in the local processor uses that information to retrieve the job from the spool device.

If a problem arises on the global processor while JES3 is running, you can transfer global functions to a local processor. This transfer is called dynamic system

interchange (DSI). The processor you choose must be capable of assuming the same workload as the previous global.

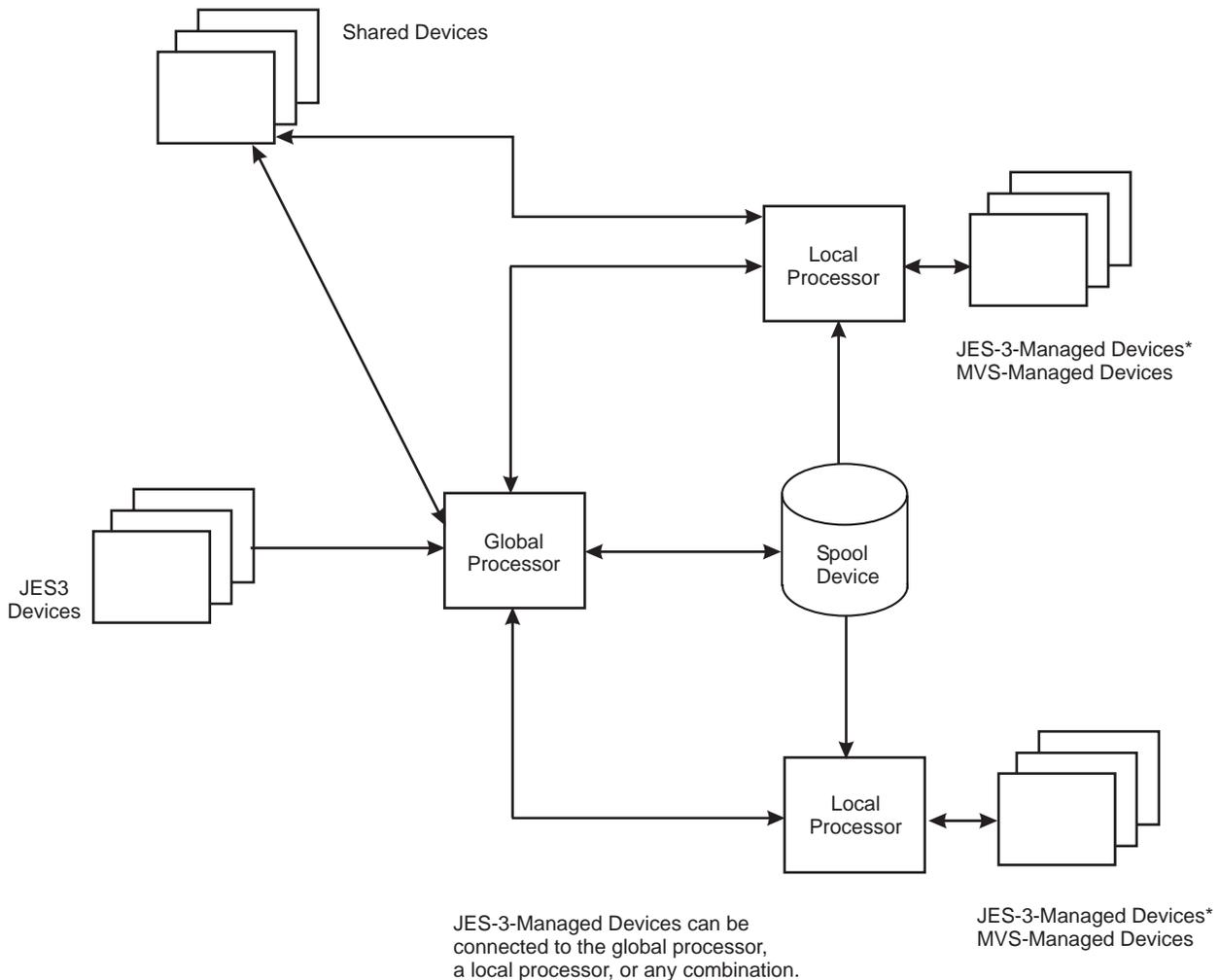


Figure 4. JES3 Multiprocessing in a Sysplex

Devices

In a JES3 complex, the devices in the system configuration are identified to JES3 when the system programmer creates a JES3 initialization stream defining the devices. There are five categories of devices:

- JES3 devices
 - JES3 Devices for Reading Jobs
 - JES3 Devices for Writing Job Output
 - The Spool Device
- JES3-managed devices
- MVS-managed devices
- Jointly-managed devices
- Shared devices

JES3 Devices

JES3 I/O devices are 'owned' by JES3. With these devices, JES3 reads jobs, stores jobs awaiting processing, and writes job output. JES3 devices are typically:

- card readers

- punches
- printers
- magnetic tape devices

System programmers identify JES3 devices during JES3 initialization. While JES3 is running, operators control JES3 devices with JES3 commands.

In JES3 SP5.2.1, the MVS Multiple Console Support (MCS) sysplex operations environment is enabled. MCS function replaces that previously provided by JES3, including:

- JES3 support for operator console devices
- JES3 command routing
- JES3 message formatting and queuing

JES3 continues to provide console support for RJP workstation consoles and networking.

JES3 Devices for Reading Jobs

The individual jobs that are the input to JES3 are combinations of job control language (JCL) statements, data accompanying the JCL statements, and JES3 control statements. The set of jobs to be processed is called an input stream. JES3 can read input streams from card readers, magnetic tape units, or magnetic disk units. (There are JES3 programs called *utility programs* that operators may use to transfer input streams or any other kinds of data from one storage medium to another.)

Jobs themselves can create input streams. That is, the output of one job could very well be an input stream containing other jobs. This eliminates the need to reenter job-created input streams in the conventional manner using an I/O device. A special *JES3 internal reader program* can transfer job-produced input streams directly to the JES3 devices for reading jobs.

JES3 Devices for Writing Job Output

Data produced by jobs during processing is called *system output data*. JES3 stores system output data temporarily on the spool device. After job processing JES3 prints or punches the system output data to the appropriate output device, which was already identified during system initialization. Operators have JES3 commands for controlling the writing of system output data.

Supporting Advanced Function Presentation (AFP) Printers

Advanced function presentation printers (AFPs), such as the 3900 and 3827, provide all points addressability. All points addressability allows every point on the page to be available for reference. These points on the page include graphics and a range of font types and sizes unavailable to impact printers.

AFP printers are controlled by Print Service Facility (PSF), which interacts with JES3. JES3 and PSF work together by communicating through the functional subsystem interface (an extension of JES3 device control that removes the need for JES3 to be dependent on specific characteristics of the printing device). This independence allows JES3 to utilize AFP printers in **full function mode** and **page mode**.

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

The Spool Device

In a system that has one processor, the spool device serves:

1. as a buffer between input devices and routines that read input data, and between routines that write output data and output devices
2. as a storage place for the control blocks and data that JES3 builds to process jobs.

A spool device must be a DASD. Transfer of job data to and from the spool device is called *spooling*. (One spool device is shown in Figure 1, but more may be used.) *Spooling* refers only to use of a DASD(s) by JES3 for storage of jobs and job-related data. Use of the same device(s) for other purposes is *not* spooling.

Spooling allows reading or writing to take place continuously at or near the full speed of I/O devices. Without spooling, there would be frequent delays (and processing overhead) during reading or writing of data.

For example, JES3 would have to read a job, process the job, and then wait while the job reads its input data (if there is any) before reading the next job. With spooling, JES3 reads an input stream containing many jobs and their input data, and sends the input stream to the spool device. JES3 then retrieves jobs individually and processes them, returning the jobs to the spool device to await processing. During processing, jobs read their input data from the spool device and write their output data to the spool device, thereby running faster.

JES3-Managed Devices

JES3-managed devices are the devices that JES3 allocates to jobs. Jobs use these devices during processing. As they do for JES3 devices, system programmers identify JES3-managed devices during JES3 initialization.

MVS-Managed Devices

MVS-managed devices are the devices that MVS allocates to jobs. Jobs use these devices during processing just like they use JES3-managed devices. System programmers do not identify MVS-managed devices during JES3 initialization. Rather they must follow MVS rules and identify the devices to MVS during MVS initialization. Because JES3 is unaware of MVS-managed devices, JES3 ignores them when allocating devices to jobs. Thus the devices will be left for allocation by MVS.

As you can see, you can choose whether or not you want JES3 to allocate I/O devices. At one extreme, you could have JES3 allocate all devices to jobs before it forwards the jobs to MVS for processing. At the other extreme, you could have JES3 forward jobs to MVS without allocated devices, and MVS would allocate devices to the jobs. Or, you could have JES3 allocate some devices to jobs, and have MVS allocate others. The key is whether system programmers define the

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 AFP printers.

devices as JES3-managed devices or MVS-managed devices during the JES3 initialization process. (JES3 allocates devices to jobs, while MVS allocates devices to job steps.)

Jointly-Managed Devices

DASDs with volumes that you cannot physically remove (called permanently-resident devices) can be managed jointly by JES3 and MVS. You could have JES3 allocate these to some jobs and have MVS allocate them to other jobs. System programmers should define these as *both* JES3-managed and MVS-managed devices.

Shared Devices

If an I/O device is not a DASD, it can be both a JES3 device or a JES3-managed device. Then you could have JES3 use the device for special purposes (such as for labeling tapes), and jobs could use the device the rest of the time. System programmers would define shared I/O devices by making them *both* JES3 devices and JES3-managed devices.

Remote Job Processing JES3 Environment

With remote job processing (RJP), you can submit work to JES3 from locations significantly distant from the JES3 global processor. The points of origin for RJP jobs are called workstations (see Figure 5). A workstation can be a single I/O device, a number of separate devices, or one of the allowable processors with its devices.

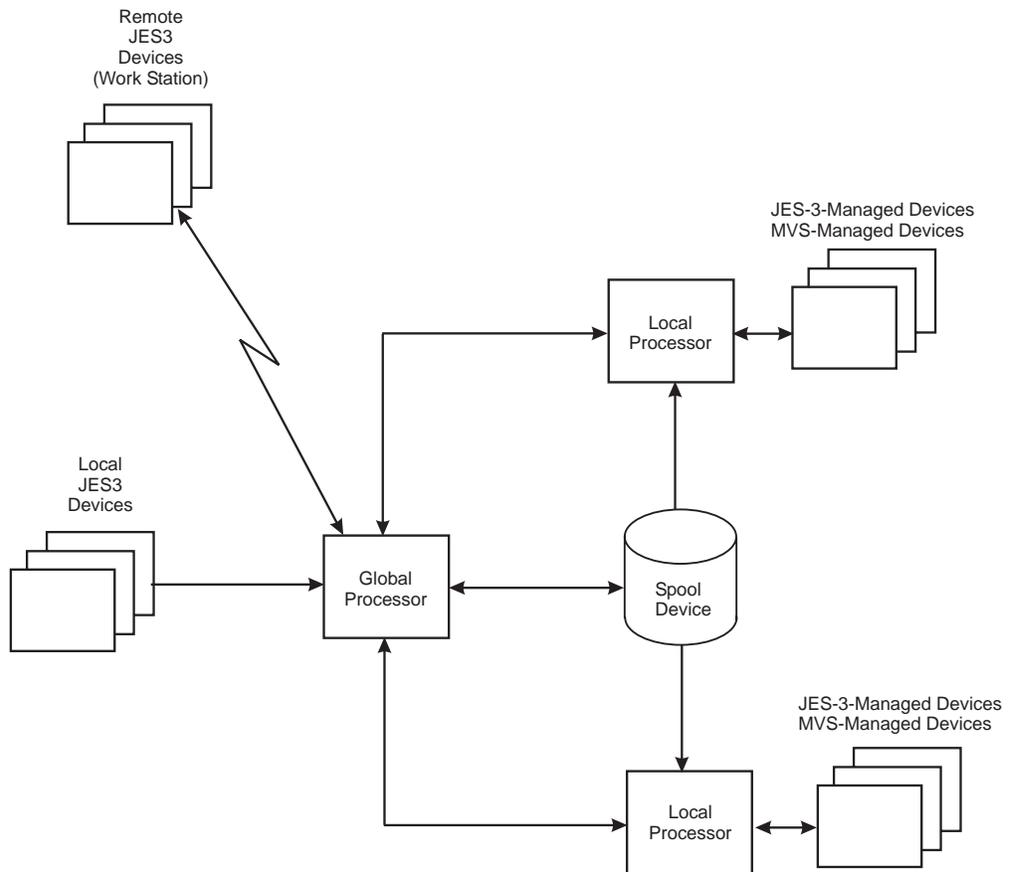


Figure 5. Remote Job Processing with JES3

Data travels between workstations and the JES3 global processor over communication lines or adapters that substitute for communication lines. JES3 processes the jobs it gets from workstations as if it had received the jobs locally. JES3 can write output of remotely-entered jobs on local devices or it can transmit the output to the originating workstation. JES3 can also transmit output to any other workstation connected to the global processor.

With communication lines, special rules exist to identify:

- the direction of data flow
- where data begins and ends
- how much data is being transmitted
- whether data or control information is being sent

Additionally, precautions have to be taken to ensure that data is not altered in transit. There must also be a means of identifying work stations and for routing data to the workstations. The rules for doing this are called protocols. JES3 offers installations two kinds of protocols:

- Binary synchronous communication (BSC) multi-leaving protocols, where a separate communication line is needed for each device at a workstation.
- IBM systems network architecture (SNA) protocols, where many devices can share a communication line.

Remote job processing with BSC multi-leaving protocols is called BSC remote job processing (RJP), and that with SNA protocols is called SNA remote job processing (RJP).

BSC Remote Job Processing

Features of BSC remote job processing include:

- Password protection. You can restrict access to an RJP line and/or a device by using an 8-character password.
- Message routing. JES3 sends job-started, job-ended, and abnormal-ending messages to workstations where the corresponding jobs originated.
- Support for programmable and non-programmable devices. JES3 handles programmable devices in an interleaved manner. (There can be concurrent activity for up to 15 devices at a workstation: seven input devices, seven output devices, and one console.) JES3 handles non-programmable devices in a non-interleaved manner. (Only one device at a time can be active at a workstation.)
- Output suspension for non-programmable devices. An operator can interrupt printing to use a card reader.
- Compressed data support. JES3 accepts compressed inbound data (compression is removal of repeated characters from data).
- Connection via leased or dial-up communication lines.
- Remote console support. You can make a device at a workstation a full-function console, a console only for work originating at that workstation, or an inquiry-only console. If one of the devices typically used for consoles does not exist at a workstation, an operator can enter commands on a card reader and receive messages on a printer.
- Inquiry by data set origin or destination. An operator at a work station can find out if there is output for a workstation by specifying the data set origin or destination in a command. An operator at a local console can inquire about any workstation.
- Error recovery and error statistics. JES3 automatically attempts error recovery and creates reports of line error statistics.
- Message queueing for signed-off consoles. If a console at a workstation cannot accept a transmission, JES3 will optionally hold the messages for later transmission.
- System management facilities (SMF) recording. JES3 creates SMF records for every workstation signon, signoff, and line start. (SMF is an MVS feature that supplies processing data useful for purposes such as job accounting.)
- Interface with remote terminal processor (RMT) programs. These are programs for managing devices connected to processors at workstations.

SNA Remote Job Processing

Features of SNA remote job processing include:

- Password protection. You can restrict access to an RJP workstation by using an 8-character password.
- Message routing. JES3 sends job started, job ended, and abnormal termination messages to workstations where the corresponding jobs originated.
- SNA data transmission protocols.
- Connection via leased or dial-up communication lines.
- Multiple logical unit support. You can have concurrent data transmission between JES3 and the console, readers, and printers at a workstation.

- Compression and compaction. JES3 can compress inbound data, and it can compress or compact outbound data. Compression is removal of repeated characters from data; compaction is substitution of characters according to installation-specified rules.
- ASCII support. JES3 can handle ASCII or EBCDIC character sets, but does not compress or compact ASCII data.
- Block size selection. Remote workstations operators can choose compression, compaction, the ASCII character set, and block sizes during logon.
- Remote console support. You can make a console at a workstation a full-function operator console, a console only for work originating at that workstation, or an inquiry-only console. If one of the devices typically used for consoles does not exist at a workstation, an operator can enter commands on a keyboard and receive messages on a printer.
- Inquiry by data set origin or destination. An operator at a workstation can find out if output for the workstation by specifying a data set origin or destination in a command. An operator at a local console can inquire about any workstation.
- Error recovery and error statistics. JES3 automatically attempts error recovery and creates reports of line error statistics.
- Message queueing for signed-off consoles. If a remote console cannot accept a transmission, JES3 will optionally hold its messages for later transmission.
- System management facilities (SMF) recording. JES3 creates SMF records for every workstation signon, signoff, and line start. (SMF is an MVS feature that supplies processing data useful for purposes such as job accounting.)

JES3 Networking

JES3 network job entry (NJE) allows JES3 users at one location to send jobs to another JES3 location for execution, to send output (SYSOUT) data to another JES3 location for processing, and to send jobs, commands, and messages to another JES3 location or to a non-JES3 location.

Therefore, JES3 can become a part of a network comprised of a Job Entry Subsystem 2 Network Job Entry (JES2 NJE) configuration, a Virtual Machine/Remote Spooling Communications Subsystem (VM/RSCS) configuration, a VSE/POWER system, as well as other JES3 complexes.

Note: Networking is not to be confused with remote job processing (RJP), since the concept of nodes does not really apply in an RJP environment. Users are connected to the computer from a remote distance, but are connected directly and not through another computer serving as a node. For information about RJP, see section “Remote Job Processing JES3 Environment” on page 13.

Each complex in the network is called a **node** and is identified by a unique node name. System programmers define nodes by name during JES3 initialization. The name of a JES3 node will always represent a global processor *and* all of its local processors within the JES3 complex. (The only time a node name can represent a single JES3 processor is when JES3 at that node has no local processors.)

When defining a JES3 networking environment, your JES3 complex becomes the *home node* and part of a job entry network that can include one or more additional systems or complexes - the *remote nodes*. Nodes communicate with each other on an equal basis.

Some features of JES3 networking are:

- Store-and-forward. The spool device at each node holds jobs in transit. JES3 automatically routes jobs to their destinations, passing them from node to node when necessary. (JES3 delays forwarding until an entire job reaches a node.) At each node, JES3 checks whether or not a job is to be passed on, so all a node has to 'know' is where the job goes next.
- Output and job routing. JES3 can send output and jobs to one or more nodes.
- Operator commands for network control. Operators can alter the status of nodes or change routing tables. Also, operators can enter commands at any node, and JES3 will pass the commands to other nodes for processing. JES3 will return responses to commands to consoles at originating nodes. Also, operators can use commands to add or delete a node from the network.
- Data compression. JES3 removes repeated characters from data to improve transmission efficiency.
- Collecting accounting information. JES3 collects job-forwarding accounting data at each transmitting node.
- Tracing facilities. Problem diagnosis is made easier because JES3 traces (creates records of) transmission activity.

A way you can link systems to form a network is shown in Figure 6. Data can travel between nodes over binary synchronous communications (BSC) lines or over Systems Network Architecture (SNA) communications lines.

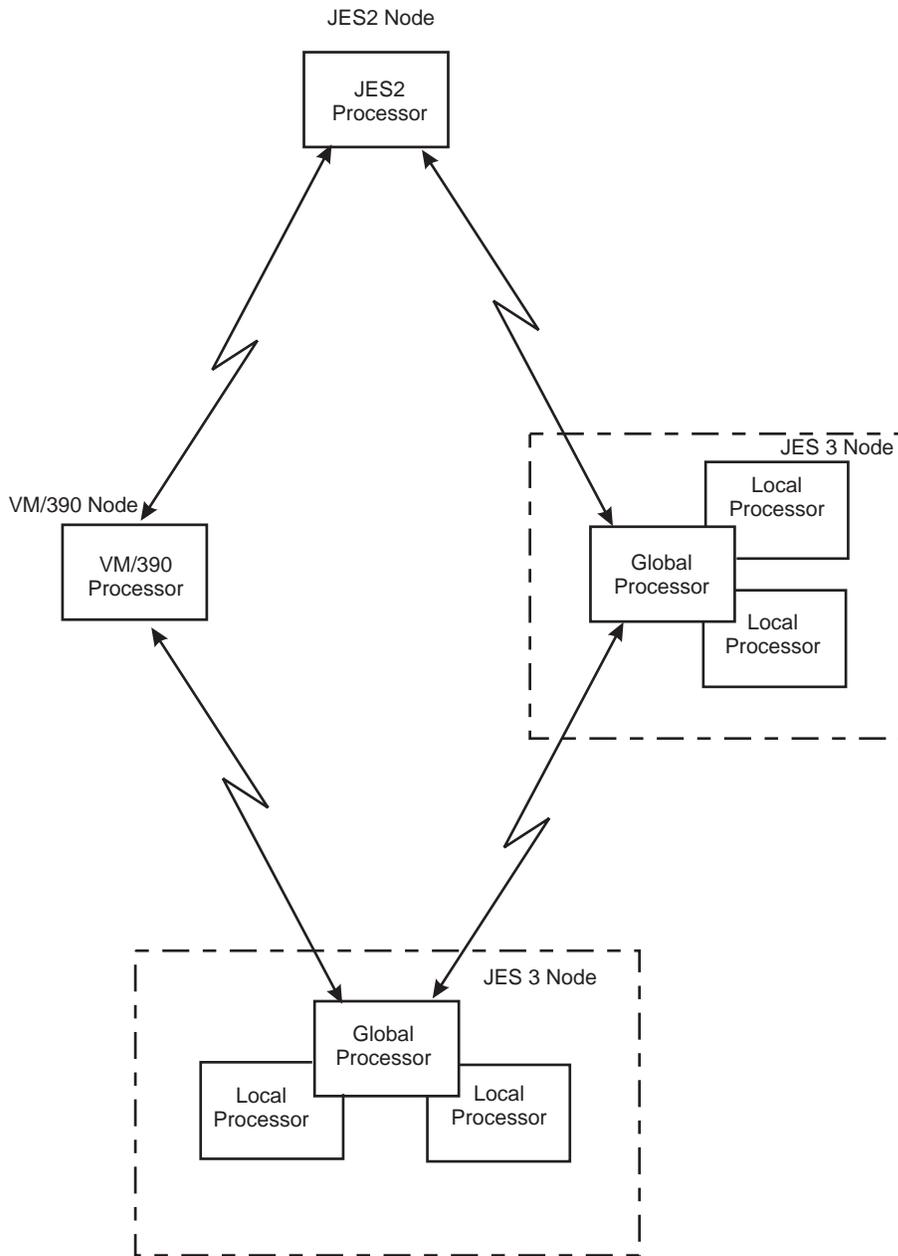


Figure 6. A Network Environment with JES3 and Other Systems

JES3 Networking Protocols

JES3 provides two types of networking protocols, binary synchronous communication (BSC) protocols and systems network architecture (SNA) protocols. Either type enables a complex to participate in a data communications network, and to pass jobs, commands, messages, and system output (SYSOUT) data between nodes in that network.

The JES3 SNA/NJE protocol in combination with MVS/BDT Version 2 provides a JES3 complex with systems network architecture/network job entry (SNA/NJE) capability.

Figure 7 shows an overview of JES3 networking where four JES3 complexes (or nodes) are connected by teleprocessing lines using both SNA and BSC protocols.

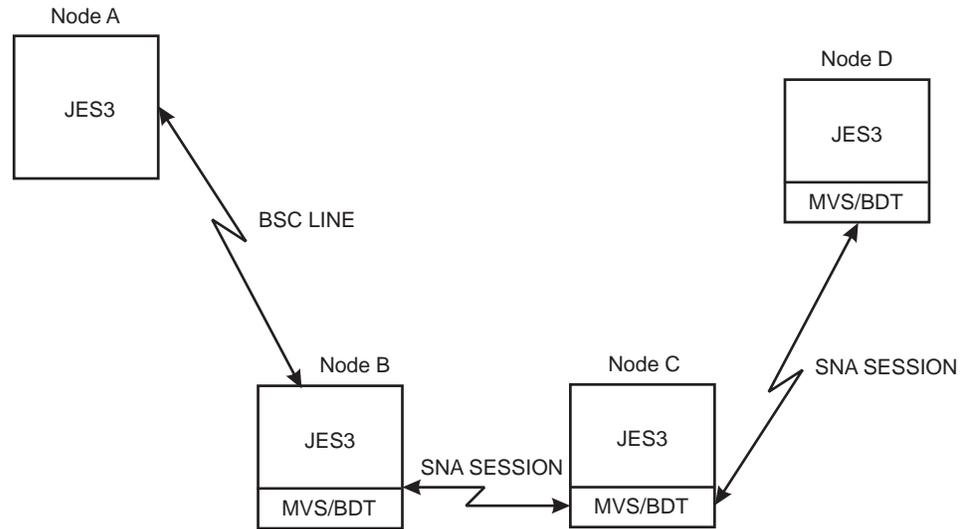


Figure 7. Overview of JES3 Networking with SNA and BSC Protocols

The JES3 networking capability allows a JES3 complex to use SNA or BSC communications protocols to:

- Send a job to another node in the network to be processed.
- Receive a job from another node in the network and either process the job or send it to another node.
- Send the SYSOUT data produced by a job to another node in the network.
- Receive SYSOUT data sent from another node and either store or print the data or send it to another node.
- Reroute jobs or SYSOUT data to another node in the network.

A JES3 complex that uses the SNA/NJE protocol may also communicate with nodes that use binary synchronous communications protocol (BSC/NJE nodes). This means that the NJE network may consist of SNA/NJE nodes only, BSC/NJE nodes only, or a mixture of the two. Installations that plan to convert nodes from BSC protocol to SNA protocol need not convert all nodes at the same time.

When an end user submits an NJE job, the user does not need to know the type of communications protocol JES3 will use to send the job to another node. JES3 determines the proper routing and proper protocol.

With the MVS XMIT JCL statement, a user can send non-MVS JCL (such as VSE/Power JECL), as well as MVS JCL statements, to another node for processing.

Support for Advanced Program-to-Program Communication (APPC)

z/OS MVS supports Advanced Program-to-Program Communication (APPC). APPC allows MVS systems to participate in a cooperative processing environment. APPC is an implementation of the Systems Network Architecture (SNA) LU 6.2 protocol that permits interconnected systems to communicate and share the processing of programs.

APPC/MVS is a Virtual Telecommunications Access Method (VTAM) application that extends APPC support to the z/OS MVS operating system. Although APPC/VTAM previously provided some LU 6.2 capability, APPC/MVS, in cooperation with APPC/VTAM, provides full LU 6.2 capability to programs running in MVS. With APPC/MVS, installations can create cross-system applications that exploit the unique strengths of different computer architectures.

Application programs that use APPC communication calls to perform cross-system requests are called **transaction programs**. Transaction programs can communicate with other transaction programs that reside on local or remote systems.

JES3 supports APPC/MVS. As a result, APPC transaction programs can exploit certain JES3 functions, such as JES3 output service processing and complex-wide data set awareness.

Transaction programs do not run as JES3 jobs; instead, transaction programs are associated with APPC initiators, which are long-running jobs that can sponsor many transaction programs. APPC initiators utilize JES3 functions on behalf of transaction programs.

Chapter 3. JES3 from an Operator's Point of View

Most installation standards for how JES3 is to manage devices and jobs are defined in the JES3 initialization stream created by system programmers. Some initialization statements define system input devices and system output devices. Others tell JES3 where to read jobs from, how to schedule the jobs for processing, and where to write data produced by the jobs.

There could be times when operators may want to change what system programmers specified on JES3 initialization statements. For example, there may be times when printers or other devices should be made unavailable to JES3 or when jobs should be canceled. Or, an operator may want to get the status of a job or device, and then modify that status.

Operators can alter JES3 processing by using JES3 operator commands. Console authority levels govern which commands or sets of commands operators can enter at specific consoles. Installations can control commands that are allowed from specific consoles by using facilities such as SAF authorization checks and RJP console authority levels.

JES3 Operator Commands

All JES3 operator commands consist of (1) a command verb (commonly called the command name), (2) the name of the affected device, processor, JES3 component or job, and (3) parameters for indicating what is to be done. There are two general types of JES3 commands:

- Commands for getting status information
- Commands for changing the status

JES3 operators need not enter commands repetitively. A single command will get the status of a specific device, devices with certain volume serial numbers, all JES3-managed devices, or all devices. Likewise, a single command can put a specific device online or offline, or it can affect all devices within a range of device numbers, all devices for a specific device control unit, or all devices.

Resource Management for Operators

JES3 operators can change I/O device assignments. Requests for these kinds of changes typically come from system programmers. However, operators may decide changes are needed after seeing a sustained increase or decline in I/O activity, or after receiving JES3 messages. Their actions may be to start or stop I/O devices, or to make I/O devices available or unavailable to JES3 (or to MVS) for allocation to jobs.

In addition to changing I/O assignments, operators can react to a slow-running system by using commands to alter JES3 processing. For instance, operators can change the processor on which converter/interpreter (C/I) processing is performed. (C/I processing is the phase when JES3 converts information on JCL statements into internal control blocks for use by the system.) This capability gives operators an active role in monitoring and controlling system activity.

Starting and Stopping JES3 Devices

As discussed previously, JES3 devices are the ones JES3 uses for reading job input and for writing job output. JES3 disk reader, tape reader, and output writer

programs do the I/O. As the need arises, operators can start or stop I/O activity by using JES3 commands, which cause JES3 reader and writer programs to be activated or deactivated. I/O activity will stop when a reader or writer program runs out of data, or when an operator enters a command to stop the reading or writing. If I/O stops for lack of data, JES3 can resume I/O when more data is introduced.

An operator who stops reading or writing with a command can resume the reading or writing with a command.

Placing Devices Online or Offline

With a single JES3 command, an operator can change the status of one I/O device, all I/O devices connected to a processor, or all I/O devices on all processors. This is possible because the devices can be identified in JES3 commands by name, by number, or by a range of numbers.

Workflow Management for Operators

Operators can alter how JES3 processes jobs, but they may not get explicit JES3 messages or instructions from programmers for some situations. Messages generally indicate that a problem or potential problem was recognized by JES3. System programmers may know overall job mix, but may not see day-to-day job mixes that affect intermittently system efficiency.

With JES3, operators have ways of searching for potential workflow problems. They can use commands to learn:

- What jobs JES3 has read, but has not begun processing
- What resources are required for a job
- What jobs JES3 has processed and has scheduled for processing
- What jobs are processing on each processor
- Job names, job numbers, job priorities, region sizes, whether or not jobs are being held up, and what JES3 routines have yet to process jobs
- How much output data is ready for printing or punching
- Determine why a job is not being scheduled for processing

Operators may then use JES3 commands to:

- Suspend further processing of jobs.
- Change the attributes of a job.
- Cancel jobs.
- Resume processing of suspended jobs.
- Limit the number of jobs in execution.
- Change (dynamically) how JES3 is to process and manage the job workload.

Utility Programs for Operators

JES3 offers utility programs, or utilities, for handling the routine work related to running JES3. Utilities do not affect how JES3 processes jobs. Rather, they affect the input to jobs or the output from jobs. To learn more about the various utilities that are available, see *z/OS JES3 Commands*.

Chapter 4. JES3 from an Application Programmer's Point of View

Application programmers identify resources their programs need by using job control language (JCL) statements. Generally speaking, JES3 examines JCL statements on behalf of MVS. That is, MVS requires JES3 to prepare jobs for processing, and the information JES3 needs to do that is contained in JCL statements.

JES3 has its own statements called control statements, for defining special processing that application programmers want JES3 to do for their jobs.

The combination of JCL, JES3 control statements and the MVS workload management mechanisms ensure that jobs are scheduled to those systems that have the resources the jobs need.

JES3 Control Statements

JCL statements and JES3 control statements serve the same purpose: they identify resources and services to be used for individual jobs. JES3 control statements are optional, being needed only for the special processing options they represent. When JES3 control statements are used, they are to be used *in addition to* JCL statements (see Figure 8).

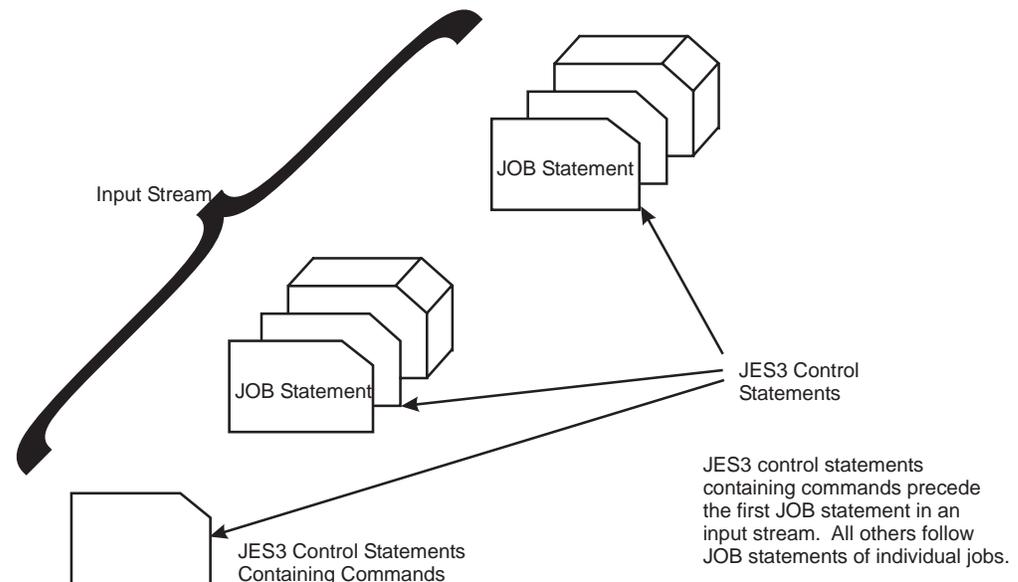


Figure 8. Placement of JES3 Control Statements

With JES3 control statements, application programmers can:

- Use operator commands
- Send messages to operators (such as special instructions for handling printed or punched data)
- Have secondary input sources containing JCL statements and/or data
- Route jobs to specific processors
- Route job output to specific devices

- Define limits for output data, and tell JES3 what to do when the limits are exceeded (examples of limits are number of lines to be printed or number of cards to be punched)

Two other things that programmers may use JES3 control statements for are:

- Deadline scheduling, for getting jobs processed before a specified deadline
- Dependent job control, for getting jobs processed only upon successful or unsuccessful processing of other jobs

Deadline Scheduling

Deadline scheduling is a way of scheduling jobs by time of day or by week, month, or year. Job priorities remain in force, but as deadlines approach, JES3 increases the priorities, thereby increasing the likelihood the jobs will be processed before a certain deadline.

System programmers must prepare for deadline scheduling with JES3 initialization statements that tell when a priority is to be increased, and by how much. Then application programmers can put deadlines on JES3 control statements they include with the jobs.

Dependent Job Control

Dependent job control (DJC) is a means of coordinating the processing of jobs. With DJC, one job will not run unless some other job has run first. (DJC has an MVS counterpart for conditional execution of job steps.)

DJC requires no advance preparation with JES3 initialization statements. Application programmers can specify all relationships between jobs on JES3 control statements they submit with the affected jobs.

Priority Aging

When all initiators are busy, throughput of certain jobs might fall below normal expectations. To help in these situations, JES3 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.

Chapter 5. JES3 from a System Programmer's Point of View

One way of visualizing JES3 is as a control program that performs job entry and job exit services. Another way of visualizing JES3 is as a language. Saying that JES3 is a control program is analogous to saying FORTRAN is a compiler. In reality, there is a FORTRAN language and a FORTRAN compiler. Similarly, there is a JES3 language and there is a JES3 program. Just as a FORTRAN programmer uses the FORTRAN language to define work the compiler is to do, JES3 system programmers use the JES3 language to define the work that JES3 is to do.

The JES3 language is made up of initialization statements, control statements, and commands. Operators and application programmers use parts of the language, but it is system programmers who define the total JES3 environment. Often, what is specified on JES3 initialization statements can be overridden with JES3 commands or control statements. While system programmers may not themselves use commands or control statements, they may have to instruct operators or application programmers on when and how they should be used.

A further level of defining how JES3 should do its work comes in the form of installation exits supplied with JES3. For example, in the area of controlling the print output from jobs, a system programmer can:

- Define initial values in the initialization stream
- Allow the operator to make certain modifications to the initial values
- Write an installation exit routine to further modify or control what should happen to the output.

As in other parts of this manual, specific commands, control statements, and initialization statements are not mentioned in this section. Commands are described in *z/OS JES3 Commands*. JES3 control statements are described in *z/OS MVS JCL Reference*. Initialization statements are described in *z/OS JES3 Initialization and Tuning Reference*. What is described here are some of the ways that system programmers can control processing by JES3 and, where applicable, where a designed-in flexibility exists for defining and redefining how JES3 is to process jobs.

Resource Management for System Programmers

System programmers use JES3 initialization statements to define resources to JES3. As part of resource definition, the system programmer establishes the policies that govern the use of installation resources. The process of resource definition is one of:

- Identifying the resources that are available to JES3
- Defining how JES3 is to use the resources

The resources that a system programmer defines include processors, I/O devices, network nodes, and remote workstations. In addition to defining resources the system programmer can use initialization statements to enable or disable specific JES3 facilities.

System programmers must define the global processor and each local processor in the JES3 complex. Additionally, they define each processor's I/O configuration.

Besides defining resources, the system programmer can turn on or turn off specific JES3 facilities or features. For example, the writer output multitasking facility

enables JES3 output writers to do more work in parallel with other JES3 functions. The system programmer (or the operator) must turn on this facility before JES3 can use it.

When defining JES3, system programmers can distribute the workload between the global processor and the local processors. Balancing the workload can relieve storage constraints on the global processor and result in a better running system.

An example of distributing work from the global processor to a local processor is to offload part of C/I phase of JES3 processing onto a local processor. The C/I portion of JES3 analyzes JCL statements on incoming jobs and converts them into internal control blocks usable by the operating system. The process of analyzing JCL frequently uses large amounts of storage, especially in installations that process large numbers of jobs.

If all of this processing is done on the global processor, your system may not run efficiently. By initializing JES3 so that part of the C/I processing occurs on local processors, you may reduce the amount of work to be done on the global processor, which can improve overall system performance.

JES3 system programmers can initialize JES3 in ways that efficiently make use of system-wide workload management functions in MVS. This can maximize JES3's efficiency at processing jobs. The system programmer can dynamically initialize JES3 with minimal disruption to normal operations.

Dynamic system interchange (DSI) is a JES3 feature that enables an installation to make a local processor become the global processor without the loss of work. During a planned or unplanned outage of the global processor, DSI provides a way for the installation to continue doing its work. To enable the installation to use DSI, the system programmer must define one or more local processors as alternate global processors.

Workflow Management for System Programmers

One important feature of JES3 is the amount of control users have over the way JES3 processes jobs. Initialization statements do not 'freeze' JES3 into a framework within which all jobs are similarly handled. Job processing flexibility results not only from:

- The variety of options available.
- The design of the statements and commands that identify desired options.
- The flexible and non-disruptive ways you can change how JES3 processes jobs.

Application programmers and operators can override many of the options specified on JES3 initialization statements by using JES3 control statements or commands. Also, named subsets of options identified on initialization statements can be used in control statements to apply the subsets to individual jobs. There is even an initialization statement to define default options to be used by JES3 when specific options are omitted from control statements.

To provide more flexibility in the way your system runs jobs, JES3 allows you to insert your own programs at certain pre-defined locations in JES3 processing. This feature is called *installation exits*. The programs you write are called installation exit routines.

Each installation exit has a name, and at each place where an installation exit is defined, there is a sample JES3 routine. You may, if you wish, write another routine, give it the name of the exit, and place it in the JES3 library that contains exit routines. The new routine will replace the JES3 routine with the same name and will be processed in its place.

JES3 has many ways to identify and define job workload so that the maximum use of system-wide workload management mechanisms are used. Through initialization statements, operator commands, and installation procedures, your installation has the flexibility to adjust its processing of jobs to meet its own unique criteria.

If more flexibility is desired, system programmers can alter the way JES3 does its work. JES3 is composed of routines called dynamic support programs, or DSPs. With the same mechanisms it uses for processing jobs, JES3 schedules and processes DSPs. JES3 users can write their own dynamic support programs to customize JES3 processing in their system.

Job Classes and Job Class Groups

Job classes and job class groups give installations a disciplined approach to specifying rules for JES3 to follow in processing jobs, scheduling jobs, and assigning resources to jobs. Instead of each application programmer specifying rules for each job, system programmers define sets of rules, and application programmers choose which sets apply to their jobs. Generally speaking, job classes contain rules for processing and scheduling jobs, while job class groups contain rules for assigning resources to jobs.

Job Classes

A job class is a named set of job processing and scheduling rules. System programmers define job classes using JES3 initialization statements. On these, they give classes unique names and specify the rules they want JES3 to apply to jobs. The link between a job and a class is established when an application programmer uses a class name on a JES3 control statement accompanying a job. JES3 will apply the rules represented by the class name to the job.

The job processing and scheduling rules that can be grouped by job class name on initialization statements are the following:

- The priority JES3 should use while processing the job (this is a special JES3 priority that can be overridden by a priority specified in a JCL statement)
- The estimated I/O rate (high, medium, or low) for the job (this is a default I/O rate, to be used unless overridden by an I/O rate on a JES3 control statement)
- The maximum number of jobs of the class to be run on a particular processor
- The maximum number of jobs of the class that can be running in the JES3 complex at one time
- The maximum number of jobs of other classes that can run on a particular processor and still allow jobs of this class to be scheduled
- The maximum number of jobs of other classes that can run in the JES3 complex and still allow other jobs of this class to be scheduled
- The maximum number of jobs of the class requiring volume mounting that can be set up at one time in the complex
- Whether jobs of the class must run only on specific processor(s), or must not run on specific processor(s)
- The actions JES3 will take if a job fails (JES3 can cancel the job, hold the job for later restart, or restart the job)

- Whether JES3 should keep a journal for a job

Job Class Groups

A job class group is a named set of resource assignment rules to be applied to a group of job classes. System programmers define job class groups on JES3 initialization statements. They establish a link between a job class group and a job class by specifying a job class group name when they define the job classes.

The resource assignment rules that apply to job classes are:

- The maximum number of jobs of the job class group that can run concurrently on one processor
- Whether JES3 alone or JES3 in conjunction with the workload management (WLM) component of MVS assigns initiators to run jobs.
- The number of MVS initiators to be dedicated to jobs of the job class group, and when the MVS initiators are to be assigned. (Initiators can be assigned as needed, all can be assigned when the first job is ready)
- When MVS initiators are to be released (initiators can be released when there is no more work, when there are fewer executable jobs than free initiators, or when there are no more executable jobs for the job class group). This applies when JES3 is responsible for initiator management.
- Which devices requiring volume mounting are to be dedicated for use by jobs in the job class group
- Whether non-dedicated devices can be used in addition to dedicated devices
- How many of the named devices are necessary
- How many jobs JES3 should examine when selecting a job for processing
- A job priority limit (JES3 will schedule all jobs in the group with a priority at or above the limit before attempting to schedule jobs below the limit)

Job Output

Using JES3, a system programmer can set limits involving the output produced by the jobs in your installation. For example, a programmer could define a limit for the number of pages of printed output produced by any job. If a job's output exceeds the limit, JES3 may follow a predetermined action, such as notifying the operator, who may cancel the job or let it complete.

Also, a system programmer can set a limit on the amount of spool space that can be used by a job. If a particular job's output uses enough spool space to reach the limit, JES3 notifies the operator, who can take a predefined action. This limit on spool space usage helps control and make better use of system resources.

Device Allocation Options

System programmers may choose from three types of device allocation:

- *Job setup*, where jobs will not be sent to MVS until all devices needed by the jobs are available and all mountable volumes (in the case of multivolume files, the first volume) have been mounted.
- *High-watermark setup*, where JES3 finds the job steps requiring the most devices of each type, allocates that number of devices, and mounts the first volumes to be used on those devices.
- *Explicit setup*, which has the processing advantages of job setup and the device allocation advantages of high-watermark setup. JES3 allocates devices for all data sets; however, the user can request that devices be scheduled based on a

job's processing phase. For example, a particular data set might request a tape volume and that volume is not to be mounted until JES3 has scheduled the job for execution.

Job setup and high-watermark setup are specified on JES3 initialization statements. If installation practices allow, JES3 control statements may be used to override, on a job-by-job basis, what was specified on JES3 initialization statements.

Installation Exits

Installation exits are places in JES3 code where exit routines can receive control. Optionally, system programmers can write their own exit routines and replace the IBM-supplied exit routines. Installation exit routines are one way for an installation to customize JES3. For example, installation exit routines can:

- Change JCL statements
- Change JES3 initialization statements
- Take corrective action when needed data cannot be found
- Enforce installation standards
- Cancel jobs
- Put text into header or trailer pages of printed output
- Align paper for printed output
- Verify nonstandard tape labels
- Monitor, accept, or reject information sent to or from a network node

Installation-Written Dynamic Support Programs

Most of the JES3 program in the global processor is divided into parts called dynamic support programs, or DSPs. There are DSPs for reading job input, for processing jobs, and for writing job output. What distinguishes DSPs from ordinary routines or subroutines is that DSPs are schedulable units. Before a DSP is executed, it must be scheduled by JES3. (DSPs have priorities that govern their position in a JES3 dispatching queue.)

System programmers can alter what DSPs do (with installation exit routines), or they can write new DSPs to supplement or replace the DSPs shipped with JES3.

Service Aids for System Programmers

JES3 has many service aids to help system programmers test for, diagnose, and correct problems.

JES3 Monitoring Facility (JMF)

The JES3 monitoring facility collects data from your system to see how your installation uses its resources. This information can help detect any performance problems which help you tune your installation. A JES3 command invokes the facility. You have the choice of producing either a hard-copy report or SMF records.

Control Block Print Programs

Control block print programs print selected JES3 and MVS control blocks. Special JES3 control statements containing control block names will cause the printing to take place.

Display Dependent Job Control Network Program

The display dependent job control (DJC) network (this is not related to a teleprocessing network) program prints a report describing a DJC network. Included

are the number of jobs in the network, the number of completed jobs, the names and status of each job in the network, and dependencies of jobs on other jobs. A JES3 command causes printing of the report.

Display JES3 Job Queue Program

The display JES3 job queue program prints information about jobs in the JES3 job queue. (JES3 uses the job queue to track jobs being readied for processing). A JES3 command will cause printing of the data. Users can print information about a specific job or about all jobs of a specific priority.

Display Selected Areas of Storage Program

This utility displays storage, alters data in storage, produces dumps, causes a wait (a suspension of execution) at a specific address, displays register contents at a wait, and displays JES3 control blocks. A JES3 command causes the desired activity to take place. The readable text is displayed as well as the hexadecimal values.

Dump Job Queue to Tape Program

The dump job queue to tape program transfers the contents of the JES3 job queue to tape. This program also returns the JES3 job queue to storage, so that JES3 can resume processing jobs where processing stopped when the job queue was dumped. A JES3 command causes dumping or restoration of the JES3 job queue.

Event Tracing Facilities

Event tracing is the collection and recording of system data at various times during the processing of jobs. The reports produced are records of key system events and show data collected during the events.

One advantage to event tracing is that a sequence of events is recorded, so the data will reflect a continuous picture of JES3 processing. Also, tracing can be started whether or not a failure has occurred, and reports can be a valuable source of tuning information.

Generally, a trace report identifies the JES3 module in which the data was collected, the contents of registers, addresses of key tables, and other data related to the events. Some of the events traced are:

- Activity on communication lines (used for remote job processing and network configurations)
- Job processing activity, including both normal and error job processing conditions.

Initialization Debugging Aid Program

The initialization debugging aid program produces storage dumps during JES3 initialization. Error messages trigger the dumps. The text of any initialization message can be put into a special JES3 initialization statement. Then, whenever JES3 writes the message, the initialization debugging aid will produce a storage dump.

JCL Test Program

The JCL test program checks JCL statements for errors. Basically, a job is handled normally by JES3 until after the job's JCL is interpreted and internal tables are constructed. JES3 does not process the job any further. JES3 does not set up any

devices and will not schedule the job for processing. A special parameter on the job's EXEC statement causes testing of JCL in place of normal job processing.

Initialization Stream Checker Utility

To help avoid errors during JES3 initialization, JES3 provides a utility called the initialization stream checker. This utility simulates the JES3 initialization process and enables you to verify your initialization stream before actually initializing JES3. This utility scans the entire initialization stream for syntax errors and certain inconsistencies in the statements which would cause JES3 to either fail to initialize or to initialize with errors.

A sample JES3 initialization stream is shipped with JES3. The sample stream can be used to help a new JES3 user get started on creating an initialization stream.

Interactive Problem Control System (IPCS)

IPCS allows installations to format and display JES3 control blocks. System programmers can use IPCS to:

- Locate and view JES3 control blocks
- View JES3 formatted dumps

IPCS can display trace data for all active trace entries and functions for any JES3 trace table. See *z/OS JES3 Diagnosis* for more information about using IPCS.

Providing Security

Security in a data processing environment involves controlling and auditing access to resources that are important to your installation. In the JES3 environment, these resources include:

- JES3-owned data sets
- Input (from nodes, remote workstations, readers, offload devices, and commands)
- Job names
- System input/output residing on spool (SYSIN/SYSOUT)
- Output devices (nodes, printers, punches, remote workstations, and offload devices).

JES3 provides 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.

The control available through initialization statements can be broadened by implementing several JES3 exits available for this purpose. You can implement a more complete security policy by using the security authorization facility (SAF) component of the BCP and a security product such as Resource Access Control Facility (RACF). SAF provides an interface to the security product to define additional control and whatever additional security controls your installation may require.

JES3 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, JES3 bypasses its own security processing.

Chapter 6. The JES3 Library

The books in the JES3 library are your primary information sources about JES3. Frequently, the text of a JES3 book will direct readers to other, non-JES3, books for related information. The preface of each JES3 book contains a list of all such books, with their order numbers.

All JES3 books have a Reader's Comment Form, which you are encouraged to use. This form is sent to the writer who is responsible for the book.

The books in the JES3 library are:

- *z/OS JES3 Introduction*

This is an introductory book; it describes JES3 from an external point of view, emphasizing what JES3 is and how it can be used. Read it first.

This book contains no specifics on getting work done, but it makes the information in the rest of the JES3 library more meaningful.

- *z/OS JES3 Commands*

This book, for JES3 operators, includes directions for using consoles; for starting, stopping, and restarting JES3; for controlling jobs, devices, volumes, and data; and for using utility programs. Each JES3 operator command is explained in detail. A table is included which lists all JES3 commands, and shows where to look for commands not described in this manual.

- *z/OS JES3 Commands Summary*

This handy summary of all JES3 commands contains operator commands as well as special commands for diagnosing and correcting problems. Detailed descriptions are absent, but the complete syntax of every command is shown.

- *z/OS JES3 Initialization and Tuning Guide* and *z/OS JES3 Initialization and Tuning Reference*

These books describe how to get a JES3 system running, and how to get the most out of it once it is running. Included are a description of the JES3 initialization process, and descriptions of individual initialization statements. Where appropriate, information is given about command or control statement overrides for what is specified on initialization statements.

- *z/OS JES3 Migration*

This book is intended for JES3 system programmers responsible for converting their installations from versions prior to OS/390 Release 1 JES3 to OS/390 Release 1 JES3.

- *z/OS JES3 Customization*

This book is for users of JES3 who want to tailor JES3 by using JES3 installation exits, dynamic support programs, and JES3 macros.

- *z/OS JES3 Messages*

This book describes all JES3 messages. The messages are listed by message number and each message includes:

- The causes of the messages
- Accompanying actions
- Appropriate responses
- Modules that contain the messages, detect the messages, and issue the messages

Where applicable, problem determination information is included.

- *z/OS JES3 Diagnosis* and *z/OS JES3 Diagnosis Reference*

These books describe tools JES3 users may use for debugging. Included are descriptions of commands for collecting information about JES3 processing, details on how to obtain dumps and traces, and explanations of the contents of dumps and traces.

- *z/OS MVS JCL Reference* and *z/OS MVS JCL User's Guide*

Actually a part of the MVS library, these publications contain descriptions of *JES3 control statements* you will not find in JES3 manuals. (If and when JES3 control statements are needed, they must accompany JCL statements in an input stream, so the JCL manual was chosen as a convenient place for the JES3 control statement descriptions.)

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

This information was developed for products and services offered in the U.S.A. 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
522 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

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

Programming Interface Information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of z/OS.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/OS. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

- Programming Interface information
- End of Programming Interface information

Trademarks

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

- AnyNet
- BookManager
- CICS
- DFSMS
- DFSMSdfp
- DFSMSdss
- DFSMSHsm
- DFSMSrmm
- DFSMS/MVS
- ESCON
- GDDM
- Hardware Configuration Definition
- IBM
- IBMLink
- IMS
- Language Environment
- MVS/ESA
- OS/2
- OS/390
- RACF
- RMF
- SOMobjects
- SystemView
- VisualLift
- VTAM
- z/OS
- z/OS.e

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

Glossary

This glossary defines technical terms and abbreviations used in JES3 documentation. If you do not find the term you are looking for, refer to the index of the appropriate JES3 manual or view *IBM Glossary of Computing Terms*, located at:

www.ibm.com/ibm/terminology

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

action message. A request for operator intervention from the operating system. In JES3, action messages are typically displayed on the operator's console.

address space. The virtual storage assigned to a job, TSO user, or a task initiated by the START command. Each address space consists of the same range of addresses.

Advanced Function Presentation (AFP). A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information. See *presentation device*.

Advanced Program-to-Program Communication (APPC). A set of inter-program communication services that support distributed transaction processing in a SNA network. See also *logical unit type 6.2*.

AFP. See *Advanced Function Presentation*.

all points addressability. The ability to address, reference, and position text, overlays, and images at any defined position or pel on the printable area of the paper. This capability depends on the ability of the hardware to address and to display each picture element.

APA. See *all points addressability*.

APPC. See *Advanced Program-to-Program Communication*.

APPC/VTAM. The implementation of APPC on VTAM.

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

auxiliary task control block (ATCB). A control block that JES3 uses to manage work done under the auxiliary task.

auxiliary task dispatching element (ATDE). A control block that JES3 uses to determine whether to dispatch a function control table (FCT) under the JES3 auxiliary task.

B

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

binary synchronous communications remote job processing (BSC RJP). A facility that permits the input and output of jobs to and from BSC workstations.

Bulk Data Transfer (MVS/BDT). (Multiple Virtual Storage/Bulk Data Transfer) An IBM program product that uses SNA protocols to copy sequential or partitioned data sets within an SNA network.

C

call. See *communication call*.

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

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

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

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

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

Common Programming Interface. Provides languages, commands and calls that allow the

Glossary

development of applications that are more easily integrated and moved across environments supported by Systems Application Architecture.

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

communication call. A conversation statement that transaction programs can issue to communicate through the LU 6.2 protocol boundary. The specific calls that a transaction program can issue are determined by the program's current conversation state. See also *verb*.

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

console authority level. A numeric value from 0-15 assigned to RJP consoles which governs the set of commands that can be issued from the console.

console destination classes. A set of named classes used by JES3 to direct messages to certain consoles. Also used in specifying the messages to be received at an RJP console.

control section (CSECT). The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

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

control statements. Statements placed into an input stream to identify special JES3 processing options for jobs.

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

CPI. See *Common Programming Interface*.

CPU. Central processing unit (equivalent to the term **processor**).

CTC. Channel-to-channel.

D

data link. The physical connection and the connection protocols between a host and a communication controller nodes by using the host data channel.

DC. Dump core.

DDR. Dynamic device reconfiguration.

deadline scheduling. A method of scheduling jobs by time of day, or by week, month, or year.

deferred-printing mode. A printing mode that spools output through JES to a data set instead of printing it immediately. Output is controlled by JCL statements.

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

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

dependent job control (DJC). The organizing of a collection of jobs that must execute in a specific order. DJC manages jobs that are dependent upon one another.

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

device fencing. Reserving devices for use only by jobs within a specified job group, or jobs with a specified job network.

DJ. Dump job.

DJC. Dependent job control.

DJC network. A set of jobs that JES3 must run in a predetermined order. Success or failure of one job can cause execution, holding, or cancelation of other jobs.

DR. Disk reader.

DSI. Dynamic system interchange.

dump job (DJ). A JES3 dynamic support program, invoked by operator command to write JES3 jobs to tape and later to restore them back to JES3 by reading them from tape back into the system.

dyadic. A multiprocessor that contains two CPUs (hardware term that is not normally used in software documentation).

dynamic destination queuing. The facility that allows the separate queueing of staging areas received by the JES3 global address space from the FSS address space.

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

dynamic device reconfiguration (DDR). A facility that allows a demountable volume to be moved, and

repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

dynamic support program (DSP). Multiprogrammed JES3 system components that are scheduled by JSS and cause the implementation of some function of JES3. DSPs can be directly related to job execution (e.g., main service, output service) or can be a background utility such as card-to-tape.

dynamic system interchange (DSI). A JES3 recovery facility that allows the operator to switch the JES3 global functions to a local processor in case of global processor failure.

dynamic writer. An output service function that controls printing or punching of data sets with characteristics that are not assigned to a specific device but are assigned by JES3 to appropriate devices as they become available.

E

early resource release. The releasing of resources (devices, volumes, and data sets) after they are no longer needed.

explicit setup. The programmer's specification, on a JES3 control statement, of precisely which devices are to be set up.

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

F

FCB. Forms control buffer.

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

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

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

functional subsystem (FSS). A functional subsystem performs JES3 functions on behalf of the JES3 global address space while residing in its own address space, which may be on any processor in the complex. The functional subsystem off-loads some of the work from the JES3 address space.

functional subsystem application (FSA). Contained within the functional subsystem address space, these routines handle a specific piece of JES3 work normally done by the JES3 global processor.

functional subsystem intercommunication (FSI). Provides formal communication between JES3 and the functional subsystem application or FSS.

G

generalized main scheduling (GMS). A set of algorithms that allow the JES3 system programmer to tailor job scheduling and selection to the specific needs of the installation.

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

global main (and local mains). The **global main** controls job scheduling and device allocation for a complex of JES3 processors. Each **local main** in the complex exists under control of the JES3 global main and is connected to the global main by CTC adapters. The JES3 on the global main can perform centralized job input, job scheduling, and job output services. Only the global main performs scheduling functions, although scheduled work executes on the local mains. See also **local main**.

GMS. Generalized main scheduling.

H

high watermark setup (HWS). An attempt to allocate a minimum number of unique device types that fulfill the requirements for each job step. Devices used in one step can be released and used again in later steps.

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

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

hot start with refresh. A special form of hot start where the JES3 initialization stream is read.

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

Glossary

I

initialization. In JES3, the process that reads the JES3 initialization statements and creates the tables and control blocks used throughout the JES3 program.

input service. The function that accepts and queues all jobs, entering the JES system, except those invoked via the *CALL command.

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

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

installation exit. A part of JES3 specifically designed for replacement by user-written routines.

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

IPL. Initial program load.

J

JCL. See *Job Control Language*.

JECL. See *Job Entry Control Language*.

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

JES managed. The system mode of operation where JES3 batch initiators are controlled by JES3.

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

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

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

JES3 devices. The devices that JES3 uses to communicate with the operator, read jobs, store jobs awaiting execution, and write job output. See also *shared devices*.

JES3-managed devices. The devices that JES3 allocates to jobs. See also *MVS-managed devices*, *jointly-managed devices*, *shared devices*.

JES3 spool access method (JSAM). Data management routines that serve JES3 address space requests such as allocation and deallocation of JES3 buffers.

job class. A named collection of JES3 job processing and scheduling rules. Use of job class names on JES3 control statements is a way of specifying what job processing and scheduling rules JES3 should use for jobs.

job class group. A named collection of resources to be associated with a job class. Use of job class names on JES3 control statements is a way of specifying what resources will be needed for jobs.

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

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

Job Control Language (JCL). A problem-oriented language designed to express statements in a job that identify the job or describe its requirements to an operating system.

Job Entry Control Language (JECL). A problem-oriented language designed to express statements in a job that describe its requirements to an operating system's job entry subsystem.

job ID. An 8-character identifier used by JES3 to uniquely identify any job in a JES3 complex at any moment in time. The job identifier is of the form "JOBnnnnn" where nnnnn is the job *number* with the appropriate number of leading zeroes, if the job number is 99,999 or less. Otherwise, the job identifier is of the form "Jnnnnnnn", where nnnnnnn is the job number with the appropriate number of leading zeroes.

job number. A unique number assigned to a job by JES3. To create a job ID, JES3 adds the letters JOB in front of the job number if the job number is 99,999 or less; otherwise the job number is left padded with zeroes up to seven digits and the letter J is added in front of this number.

job queue element (JQE). A control block containing a summary of information from a JCT entry. JQEs

remain in storage and are used by JES3 instead of JCT entries for scheduling of work.

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

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

job validation. The process during JES3 initialization where JES3 examines the job-related spool control blocks to verify their validity. If JES3 finds incorrect control blocks, JES3 gives the system operator an opportunity to take corrective action to insure that JES3 initialization completes.

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

jointly-managed devices. A special case where the same device is both a JES3-managed device and an MVS-managed device. Only direct-access devices with volumes that cannot be physically removed can be jointly-managed devices.

JSAM. See *JES3 Spool Access Method*.

L

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.

local console. Any console that is dedicated to a single main within a JES3 installation. A remote job processing console cannot be a local console.

local device. A device attached to a host processor by using a channel.

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

local start. A restart of a local processor. Initialization is unnecessary and user jobs are not affected.

logical storage. The amount of central storage required by a job or a job step to execute efficiently on a processor when running under JES3.

loosely-coupled multiprocessing. Two or more computing systems interconnected by an I/O

channel-to-channel adapter. The processors can be of different types and have their own unique configurations.

logical unit. 1) a type of network addressable unit that enables end users to communicate with each other and gain access to network resources. 2) A port providing formatting, state synchronization, and other high-level services through which an end user communicates with another end user over an SNA network.

logical unit type 6.2. The SNA logical unit type that supports general communication between programs in a distributed printing environment; the SNA logical unit type on which CPI communications is built.

LU. See *logical unit*.

M

main. A processor named by a JES3 MAINPROC initialization statement, on which jobs can execute; represents a single instance of MVS. The two types of mains are (1) global main, and (2) local main.

MAINPROC. A JES3 initialization statement that defines a processor to JES3.

main device scheduler (MDS). Controls the setup of I/O devices associated with job execution.

main device scheduler (MDS). A phase of JES3 that controls the setup of I/O devices associated with job execution.

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

main service. A dynamic support program that schedules problem programs for execution and manages the flow of data (system input, print, and punch) across the channel-to-channel adapter to and from the global processor.

MDS. Main device scheduler.

migration. The changing over from an installation's production operating system to an upgraded or entirely new operating system.

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

multiple console support (MCS). A feature of MVS that permits selective message routing of up to 99 operator's consoles.

multiple virtual storage (MVS). A virtual storage facility that allows each user a private address space.

Glossary

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

multiprocessor. A processor complex that consists of more than one CPU.

MVS. See *Multiple Virtual Storage*.

MVS/APPC. The implementation of APPC on an MVS system.

MVS-managed devices. The devices that MVS allocates to jobs. See also *JES3-managed devices*, *jointly-managed devices*.

N

network. For JES3, two or more systems and the connections over which jobs and data are distributed to the systems. One or more of the systems can be a JES3 global (and its local mains, if any). The other systems can be non-JES3 systems with compatible networking facilities. Connections can be established through communications paths using SNA or BSC protocol.

network job entry (NJE). The process in which a user at one installation can submit a job/output to be executed at or sent to a different installation (node to node). NJE is networking between installations using SNA or BSC protocol.

network job stream. A network job stream includes:

- a job header
- an MVS job comprised of JCL an/or SYSIN data
- a job trailer.

See also the definition of network SYSOUT stream.

network stream. A network stream contains either a network job stream or a network SYSOUT stream. See the respective definitions for each.

network job. Same as network stream.

network SYSOUT stream. A network SYSOUT stream includes:

- a job header
- a data set header (where there may be more than one data set header per SYSOUT data set transmitted)
- a SYSOUT data set
- a job trailer.

Note: There may be more than one SYSOUT data set-data set header pair. See also the definition of network job stream.

networking protocol. 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. The two protocols that JES3 uses to establish a networking environment are binary synchronous communication (BSC) and systems network architecture (SNA).

NJE. An installation to installation data communication network.

node. 1) An end point of a link, or a junction common to two or more links in a network. Nodes can be processors, controllers, or workstations. Nodes can vary in routing and other functional capabilities. 2) In JES3, one of the systems in a network of systems connected by communication lines. Each node defined to itself is the home node. All others are defined as remote nodes, directly or indirectly connected. The home node and the remote nodes are identified as such in the installation's initialization stream (NJERMT statement).

non-partitionable processor complex. A processor complex that cannot be partitioned.

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

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

O

operating system. The software that controls the operation of a processor complex.

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

operator messages. 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.

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

output service. The function that processes SYSOUT data sets. Processing includes printing, punching, or directing output to an external writer.

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

output writer. A JES3 routine that transcribes output data sets to the printer or punch system output devices.

P

page mode. The mode of operation in which the AFP print (such as 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 AFP printers, such as the 3800-3 and 3820.

page mode printer. An AFP printer, such as the 3800 model 3 and 3820, that can print page-mode data.

partition. Equivalent to the term **physical partition.**

partitionable processor complex. A processor complex that can be partitioned.

partitioned mode. Equivalent to the term **physically partitioned mode.**

partitioning. The process of forming multiple physical partitions from one processor complex.

physical partition. A set of hardware resources, formed by partitioning, that can support a single operating system.

pel. Picture element

physically partitioned mode. The state of a processor complex when its hardware resources are divided into multiple configurations.

pre-execution setup. That portion of setup performed by MDS prior to a job entering execution.

presentation device. A device that produces character shapes, graphics pictures, images, or bar code symbols on a physical medium. Examples of physical media are display screens, paper, foils, microfilm, and labels.

primary job entry subsystem. The active job entry subsystem. The primary job entry subsystem is determined during the system generation process.

primary task. The task under which most DSPs execute.

Print Services Facility (PSF). An IBM licensed program that produces printer commands from the data set to it. PSF programs run on the z/OS, OS/390, MVS, VM, VSE, OS/2, AIX, and OS/400 operating platforms. For JES, PSF programs operate the 3800 model 3 and 3820 printers. PSF operates as a functional subsystem.

process SYSOUT (PSO). An interface to JES3 to allow access and control of SYSOUT data sets from

other address spaces. It is used primarily by TSO OUTPUT and RECEIVE commands and external writers.

processor. A hardware unit that contains software to interpret and process instructions.

processor complex. The maximum set of hardware resources that support a single operating system.

protected buffer pool (PBUF). An area in the common storage area and JES3 auxiliary address space that has been divided into buffers.

protocol. The meaning of, and the sequencing rules for, requests and responses used for managing a network, transferring data, and synchronizing the states of network components.

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

R

RACF. Resource Access Control Facility

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

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

reconfiguration. The process of adding hardware units to, or removing hardware units from, a configuration.

remote device. A device attached to a host processor by using a data link.

remote job entry (RJE). A process in which a user at a remote site is connected to the host system by a data link (telecommunication lines). RJE and RJP is networking between the user and the host system.

remote job processing (RJP). A facility that permits the input, processing, and output of jobs to and from terminals remote from the JES3 installation.

RJP. Remote job processing.

remote terminal processor (RTP). A programmable remote workstation.

resident queue (RESQUEUE). A control block built in storage by the job segment scheduler to represent a scheduler element during the life of the scheduler element. It contains status information and queuing pointers.

Glossary

Resource Access Control Facility (RACF). An IBM program product that provides for access control by identifying and verifying users to the system, authorizing and logging access to protected resources, and logging detected unauthorized attempts to enter the system.

RMT. Remote terminal processor program.

remote terminal processor (RMT). A self-loading object deck created as a result of an RMT generation. RTP programs allow JES3 to communicate with programmable remote workstations.

routing code. An MVS identifier that you use to route MVS messages to a specific console(s).

RTAM. Remote terminal access method.

RTP. Remote terminal processor.

S

SAA. See *Systems Application Architecture*.

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

scheduling environment. A list of resource names along with their required states. If an MVS image satisfies all of the requirements in the scheduling environment associated with a given unit of work, then that unit of work can be assigned to that MVS image. If any of the requirements are not satisfied, then that unit of work cannot be assigned to that MVS image.

service class. A group of work which has the same performance goals, resource requirements, or business importance. For workload management, you assign a service goal and optionally a resource group to a service class.

server mode. A processing mode of the JES3 dump job function that runs in its own address space and can utilize any tape devices in the system.

session. A logical connection between two logical units that can be activated, tailored to provide various protocols, and deactivated as requested.

setup. The phase of JES3 processing that performs volume fetch, device, volume, and dataset allocation.

setup DSP. A DSP that performs volume fetch, job setup, high watermark setup, and explicit setup functions.

shared devices. (1) Devices that are connected to more than one processor. (2) Devices that are both JES3 devices and JES3-managed devices.

side. Equivalent to the term **physical partition**.

single-image mode. The state of a processor complex when all of its hardware resources are in a single configuration.

SNA. See *Systems Network Architecture*.

solicited message. A message that is a response to a command (also see unsolicited message).

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

spool device. A direct-access device that JES3 uses for intermediate storage of control blocks and data needed for processing jobs. When JES3 is used for multiprocessing, the spool device becomes a collection point for job input data to be distributed to local processors, and for job output data coming from local processors enroute to I/O devices attached to the global processor.

spool device. A direct-access device that JES3 uses for intermediate storage of control blocks and data needed for processing jobs. When JES3 is used for multiprocessing, the spool device becomes a collection point for job input data to be distributed to local mains, and for job output data coming from local mains enroute to I/O devices attached to the global.

spool partition. A named collection of spool data sets.

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

staging drive group. A collection of staging drives for space management and recovery. It is created by the user with the Mass Storage Control Table Create program.

standard job. A job for which JES3 defines needed processing entirely from Input Service, Converter Interpreter, MAIN service, Output Service, and PURGE service.

statistics data area (SDA). A data area used to collect JES3 processing statistics by using the IATXSTAT macro.

Storage management subsystem (SMS). An MVS subsystem responsible for managing data sets and volumes. This subsystem supports JCL constructs such as storage class and storage group.

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

SSI. Subsystem interface.

subsystem interface (SSI). A set of program routines that allows two-way communication between a JES3 address space and other address spaces.

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

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

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

systems application architecture (SAA). A set of software interfaces, conventions, and protocols that provide a framework for designing and developing applications with cross-system consistency.

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

systems network architecture remote job processing (SNA RJP). A facility that permits the input and output of jobs to and from SNA workstations.

systems network architecture/network job entry (SNA/NJE). A networking capability that works in combination with MVS/Bulk Data Transfer (MVS/BDT). Networking is established between nodes through MVS/BDT "sessions." Sessions can be established over telephone lines, microwave links, by satellite, or by channel-to-channel adapters.

T

TP. See *transaction program*.

transaction program. An application program that allows users to access resources in a SNA network.

U

uniprocessor. A processor complex that consists of only one CPU.

unsolicited message. A message that is not a response to a command (also see solicited message).

USAM. User spool access method.

user buffer pool (UBUF). An area in each user's address space that has been divided into buffers.

user spool access method (USAM). Data management routines that do not execute in the JES3 address space but provide the subsystem interface for allocation, deallocation, SYSIN/SYSOUT, OPEN, and CLOSE functions of user data sets.

V

volume. That portion of a single unit of storage that is accessible to a single read/write mechanism; for example, a drum, a disk pack, or part of a disk storage module.

VTAM. Virtual telecommunications access method.

W

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

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

warm start to replace a spool data set (WR). For JES3, a special form of warm start where a spool data set can be replaced by another data set with the same dname; all jobs with data on the replaced spool data set are lost.

warm start with analysis to replace a spool data set (WAR). For JES3, a special form of warm start (W) combining warm start with analysis (WA) and warm start to replace a spool data set (WR) processing.

WLM managed. The system mode of operation where JES3 batch initiators are controlled by the workload management component of MVS.

Workload Management (WLM). WLM is a component of MVS that manages system resources.

workstation. A station at which an individual can send data to or receive data from a computer for the purpose of performing a job.

writer. See *output writer*.

writer output multitasking. For JES3, a facility by which writer output processing can be performed concurrently with other JES3 functions on a multiprocessor global processor.

WTO. Write to operator.

WTO/R. Write to operator with a reply request.

Glossary

Index

A

- accessibility 35
- accounting information
 - for network 17
- adjacent node 18
 - See remote node
- allocation
 - explicit setup 28
 - high-watermark setup 28
 - job setup 28
- APPC (advanced program-to-program communication) 19
- application programmer
 - features 5
- ASCII support
 - for SNA remote job process 16
- attribute
 - print
 - for full function printers 12
- availability
 - of global processor 4
 - of JES3 4

B

- balance of workload 26
- BDT (bulk data transfer) 18
 - See MVS/BDT
- block size
 - for SNA remote job process 16
- BSC (binary synchronous communications) protocol 17
 - remote job process 15
- BSC RJP (binary synchronous communication remote job processing) 15
- BSC/NJE 16

C

- card punch
 - location 5
- command
 - JES3 control statements 23
 - operator 21
- communication line
 - for SNA remote job process 15
 - to workstation 14
- compaction
 - for SNA remote job process 15
- compressed data
 - for BSC remote job process 15
 - for network 17
 - for SNA remote job process 15
- console
 - remote support 16
- control flexibility 4

- control statement
 - for deadline scheduling 24
 - for dependent job control 24
 - in input stream 11
 - placement 23
 - purpose 23
- control statements
 - for JES3 usage 5
- CR (card reader)
 - location 5
- customizing JES3
 - dynamic support program 27
 - installation exit 26

D

- data
 - integrity 3
- deadline scheduling
 - description 24
- debug
 - initialization stream service aid 30
 - traces and dumps 5, 30
 - tracing facilities for network 17
- dependent job control
 - description 24
- device
 - as workstation 14
 - categories 10
 - control by JES3 10
 - for BSC remote job process 15
 - JES3-managed, purpose 12
 - jointly-managed 13
 - location 5
 - spool 12
- device sharing
 - JES3 as manager 4
- disability 35
- display of storage 30
- documents, licensed viii
- DSP (dynamic support program) 27, 29
- dump
 - job queue, utility 30
- dynamic system interchange
 - defined 10
 - when a processor is down 26

E

- environment
 - for JES3 7
- error recovery
 - description 4
 - for BSC remote job process 15

F

FSS (functional subsystem)
support 11

G

global processor
as focal point 3
availability 4
glossary 41
growth
non-disruptive 4

I

I/O activity
controlling I/O device 21
I/O resources 3
rate 27
initialization statement
initializing JES3 25
override 5
usage 5
verification of accuracy 31
input stream
reading job input 11
installation exit
types of function 29
usage 26

J

JCL statement
purpose 23
JCL test program 30
JES3
and the application programmer 23
and the system operator 21
and the system programmer 25
approach to a single-system image 4
availability 4
command
for switching global functions 4
usage 4
control of devices 10
control statement
for deadline scheduling 24
for dependent job control 24
placement 23
purpose 23
control statements 5
customizing
using installation exit 26
device
connection 9
for writing job output 11
environment 7
multiprocessor 8
single-processor 7
failure 4

JES3 (continued)

features 4
home node 16
initialization statement
override 5
network 17
operator commands for network control 17
publications library 33
remote node 16
resource and workflow management by 3
service aids 29
switching global functions with 4
terminology
glossary 41
usage 4
JES3 design feature
availability 3
control flexibility 3
non-disruptive growth 3
physical planning flexibility 3
single-system approach 3
JMF (JES3 monitoring facility)
performance problem 29
job
class
definition 27
definition of job group 28
scheduling
functions 24
priority aging 24
setup 28
job execution
job execution 3
job priority
for job execution 3
for job output 3
jointly-managed device 13

K

keyboard 35

L

licensed documents viii
limit
output data 24
local processor
connected to global 9
for writing job output 9
LookAt message retrieval tool vii

M

message
BSC message routing 15
SNA message routing 15
message retrieval tool, LookAt vii
multiprocessing environment
attached processors 8

- multiprocessing environment (*continued*)
 - environment
 - multisystem sysplex 8
 - multiprocessing
 - JES3 8
 - uniprocessors 8
- multisystem sysplex
 - multiprocessor JES3 environments 8
- MVS
 - resource and workflow management 3
- MVS device
 - identification 12
 - purpose 12
- MVS-managed device 12

N

- network control with operator commands 17
- networking 18
 - adjacent node 18
 - definition 16
 - NJE (network job entry) 16
 - node
 - definition 18
 - SNA protocol 17
- node
 - home 16, 18
 - remote 16, 18
- Notices 37

O

- operating system
 - major goal 3
- operator
 - control over device 21
 - control over system processing 21
 - features 4
 - utility program 22
- operator command 21
- output data
 - limit 24
- output processing
 - send output to other nodes 17
 - write job output 11
- override
 - JES3 initialization statements 5
- overview
 - application development 19

P

- password protection
 - for BSC remote job process 15
 - SNA remote job process 15
- physical planning, flexibility 5
- policy
 - for device and job management 5
- print
 - attributes
 - for full function printers 12

- print mode
 - full function 11
 - page mode 11
- printer
 - location 5
- priority
 - for job execution 3
 - for job output 3
- priority aging
 - for job 24
- processor
 - communication 9
 - global
 - focal point 3
 - location 5
 - sharing I/O devices 4
- protocol
 - BSC 14
 - for BSC 17
 - for SNA 17
 - SNA 14

R

- RACF (Resource Access Control Facility) 31
- recovery
 - for JES3 failure 4
- recovery routine 4
- refid=mess.entering system commands 4
- resource
 - definition 25
 - processor space 3
 - processor time 3
 - reserving 3
 - system 3
- Resource Access Control Facility
 - See RACF
- resource management
 - by JES3 3
 - by MVS 3
 - for operator 21
 - for system programmer 25
- RJP (remote job processing)
 - defined 13
 - workstation 13
- RSCS (Resource Access Control Facility) 16

S

- SAF (security authorization facility) 31
- security 31
- security authorization facility
 - See SAF
- see=multiple virtual storage 3
- service aids 29
- shared devices
 - definition 13
 - purpose 13
- shortcut keys 35
- single-system sysplex
 - JES3 example/figure 7

- SMF recording
 - for BSC process 15
 - for SNA processing 16
- SNA (system s network architecture)
 - protocol 17
- SNA (systems network architecture)
 - remote job process 15
 - SNA/RJP 15
- SNA/NJE (systems network architecture network job entry) 16
- spool device 8, 12
- Sysplex
 - definition 3
- system resource
 - usage 3

T

- traces and dumps 30
 - for debugging 5
- tracing events (utility) 30

U

- utility program
 - supplied with JES3 22

V

- VM/RSCS 16
- VSE/POWER 16

W

- workflow management
 - by JES3 3
 - by MVS 3
 - for operator 22
 - for system programmer 26
 - workload balance 26
- workload balance 4
- workload managment 4
- workstation
 - for remote processing 13



Program Number: 5694-A01

Printed in U.S.A.

SA22-7551-02

