GC23-0001-0
File No. S370-36

Systems

OS/VS2 MVS
System Programming Library: JES2

VS2 Release 3.7

IBM

First Edition (February, 1976)

This edition applies to JES2 as a component of OS/VS2 MVS Release 3.7 and to all subsequent releases until otherwise indicated in new editions or technical newsletters. Changes are continually made to the information contained herein. Before using this publication in connection with the operation of IBM systems, refer to the *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the end of this publication. If the form has been removed, comments may be addressed to: IBM Corporation, Department 4F1, 18100 Frederick Pike, Gaithersburg, Maryland 20760. Comments become the property of the IBM Corporation.

PREFACE

This publication provides information for system programmers who are to install the job entry subsystem JES2. It consists of seven chapters that include information about the installation and initialization of JES2, JES2 processing, remote job entry supported by JES2, and factors that affect JES2 performance.

"Chapter 1. Introduction to JES2" briefly describes the job entry subsystem. It also provides short descriptions of JES2 generation and initialization, RMT generation, and JES2 processing.

"Chapter 2. Installing JES2" provides you with information necessary to install JES2. This chapter discusses the programming requirements for JES2 and RMT generations, and includes information about the generating system, the distribution libraries, and the required spool data sets. Coding the JES2 generation parameters and the processing of a JES2 generation are also described. (RMT parameters and the execution of an RMT generation are described in the chapter "Remote Job Entry.")

"Chapter 3. JES2 Initialization" describes the JES2 initialization procedure. The chapter includes the meaning and use of each of the initialization parameters and the rules for coding the parameters. It also contains descriptions of starting and restarting JES2.

"Chapter 4. JES2 Processing" discusses how you can affect JES2 processing by means of the JES2 generation and initialization parameters and the JES2 operator commands. The chapter describes configuration, job submission and queuing, conversion and execution, and output.

"Chapter 5. Remote Job Entry" contains information about remote devices supported by JES2, the RMT generation procedure (including the RMT parameters and the execution of an RMT generation), and the operation of a remote station.

"Chapter 6. Miscellaneous JES2 Facilities" describes automatic command processing, the JES2 patching facility, the flow for time sharing and started tasks, and the multi-access spool configuration.

"Chapter 7. JES2 Performance" discusses factors affecting JES2 performance, operator control of the batch job workload, a comparison of HASP II Version 4.0 and JES2 performance factors, and changing JES2 from nonswappable to privileged.

Related Publications

The following manuals should be available for reference when you are using this publication.

Operator's Library: OS/VS2 Reference (JES2), GC38-0210

Operator's Library: OS/VS2 Remote Terminals (JES2), GC38-0225

OS/VS2 JES2 Logic, SY28-0622

OS/VS2 JCL, GC28-0692

OS/VS2 IBM 3540 Programmer's Reference, GC24-5111

OS/VS2 Conversion Notebook, GC28-0689

OS/VS2 System Programming Library: Storage Estimates, GC28-0604

OS/VS2 System Programming Library: System Generation Reference, GC26-3792

OS/VS2 System Programming Library: Data Management, GC26-3830

OS/VS2 System Programming Library: Job Management, GC28-0627

OS/VS Message Library: VS2 System Messages, GC38-1002
OS/VS Message Library: VS2 System Codes, GC38-1008
OS/VS Utilities, GC35-0005
OS/VS System Management Facilities (SMF), GC35-0004
OS/VS-DOS/VS-VM/370 Assembler Language, GC33-4010

Contents

Figures

CHAPTER 1. INTRODUCTION TO JES2

JES2 is a job entry subsystem, provided with MVS, that is generally compatible with HASP II. JES2 serves as the point of entry for all jobs, and the function which produces all hardcopy job output. To accomplish these functions, JES2 controls local and remote job entry devices and output devices. A special job entry source, the internal reader facility, allows MVS to submit system jobs: started tasks and time-sharing LOGONs. Tape and disk input are also supported through the internal reader facility. See Figure 1-1 for input/output relationships to the job entry subsystem and MVS.

An output interface allows MVS to retrieve output for TSO terminals, and allows a special output facility – the External Writer – to process output to tape, disk, and installation-written writer routines. The output interface also supports an output facility to the 3540 diskette writer (see *OS/VS2 IBM 3540 Programmer's Reference*).

While the job is in MVS, the JES2 job queue residing in pageable storage maintains a record for the job. Job-related system records plus records related to job input and output are maintained on external spool volumes.

JES2 Processing: By means of JES2 and RMT generation and JES2 initialization, you can define and control the configuration of job entry sources and job output destinations. JES2 provides centralized control of job input, queuing, and output, such that all jobs are controlled in the same manner whether submitted from local or RJE (remote job entry) devices, or through the internal reader facility.

Before you attempt to install JES2 programs, you should be familiar with the information about JES2 programs contained in this chapter and in the chapters "Remote Job Entry" and "JES2 Initialization."

To estimate storage requirements, refer to *OS/VS2 System Programming Library: Storage Estimates*. For operator's procedures, refer to *Operator's Library: OS/VS2 Reference (JES2)*.

JES2 Generation: The JES2 generation procedure, which is the process of installing JES2, is in two parts. The first part, JES2GEN, may be performed while Stage II of the system generation process is in progress. It consists of assembling the job entry subsystem from source modules that have been tailored to reflect the characteristics that you have specified through JES2 generation parameters. The second part, JES2BLD, can occur only after Stage II of the system generation has been completed. During JES2BLD, the assembled object modules are link-edited into the MVS system control program.

RMT Generation: In addition to installing JES2, you may also generate one or more JES2 MULTI-LEAVING remote terminal processor (RTP) programs. The RTP programs allow jobs to be submitted from a remote terminal to the job entry subsystem in the central computer for processing.

JES2 Initialization: Initialization is JES2's means of readying itself for processing. JES2 performs an initialization the first time it is started and every time it is restarted after a normal shutdown or after a system failure. By specifying a set of initialization parameters, you indicate which of the JES2 functions and devices defined at JES2 generation are to be initialized; the initialization parameters define the functions and device characteristics JES2 will use during its current execution.

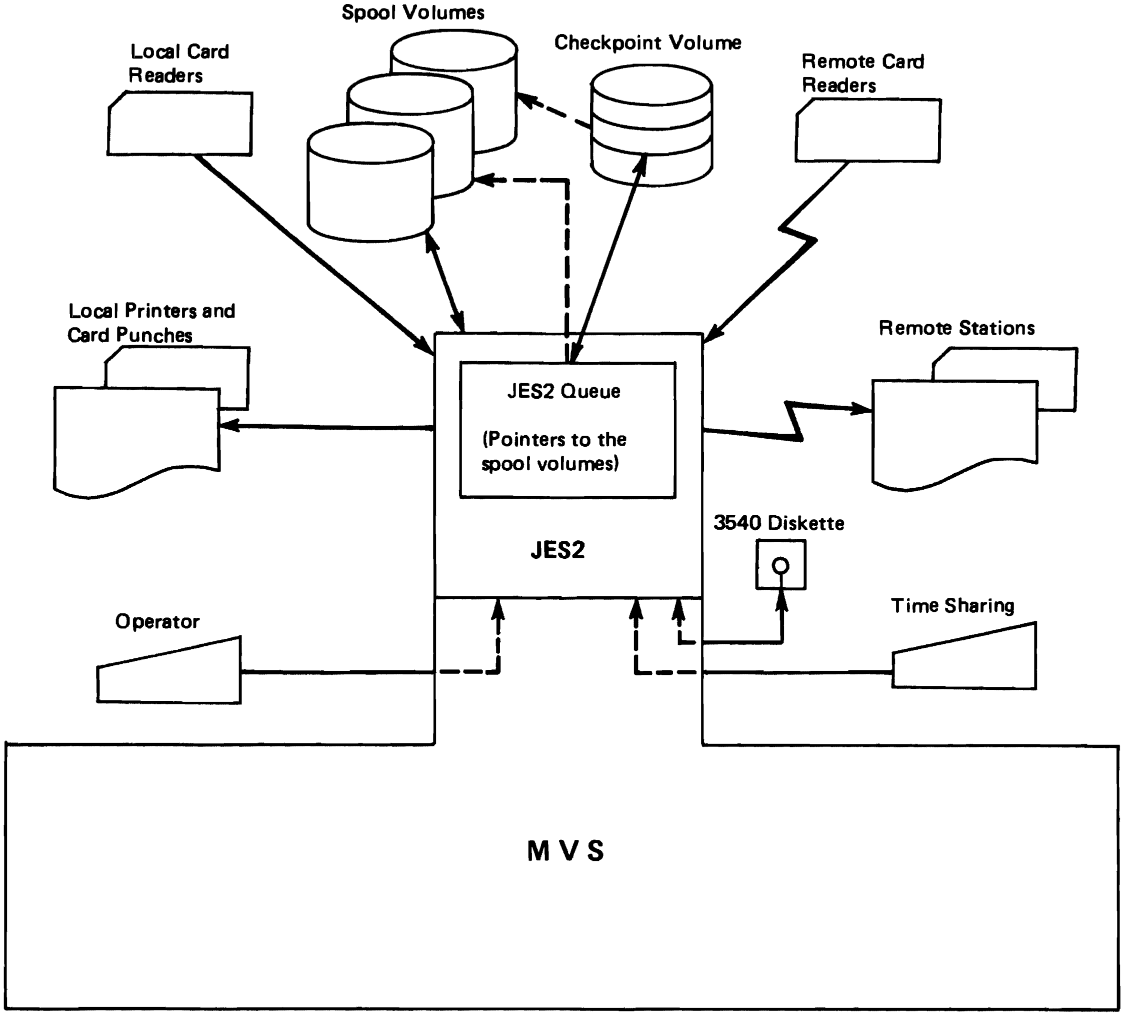

Figure 1-1. JES2 I/O Relationships

CHAPTER 2. INSTALLING JES2

The process of installing JES2 is referred to as JES2 generation. JES2 generation consists of assembling the JES2 job entry subsystem from source modules that have been tailored to reflect the features, options, and limits that you have selected through the JES2 parameters, and then link-editing the assembled object modules into the MVS system control program.

Once JES2 has been installed, partial JES2 generations may be performed if only minor changes need to be made. In this case, instead of reassembling all of the JES2 modules, only those modules that need updating need be reassembled and link-edited.

In addition to the JES2 generation, you can perform an RMT generation to generate JES2 MULTI-LEAVING remote terminal processor (RTP) programs for job entry from remote terminals. The RTP programs, generated separately or concurrently, are punched as self-loading object decks. The RTP programs that can be generated are:

- System/360 and System/370 binary synchronous communication (BSC) RTP program
- System/360 Model 20 BSC RTP program (including the 2922 RTP program)
- 1130 RTP program
- 1130 loader program
- System/3 RTP program

This chapter describes the procedures required for JES2 and RMT generation (for example, those for allocating space and cataloging), and the spool data sets required for job processing after generation. It also contains detailed information about coding the JES2 generation parameters and executing programs to install JES2. For a description of the RMT parameters and the processing involved in an RMT generation, refer to the chapter "Remote Job Entry."

Overview

To install JES2 and generate RTP programs, you need a generating system to "drive" the JES2 generation and RMT generation processes and perform the other jobs that must be done before processing can begin. In addition, you must do the following:

1. Copy the distribution libraries from the PID distribution library tapes to a direct-access volume(s).
2. Define the data sets that are required to contain the source and object modules that are used during processing.
3. Code the JES2 and RMT parameters that specify the features, options, and limits to be included in the JES2 subsystem to be installed and in the RTP programs to be generated. In addition, if you are applying changes to the source code, you would specify these changes in update control statements.
4. Execute the programs to install JES2 and, if specified, generate the RTP programs.

The following sections give detailed information that applies to both JES2 and RMT generation for the first two steps. Information about the last two steps for JES2 generation is also contained in the following sections. This information for RMT generation is included in the chapter "Remote Job Entry."

Requirements for JES2 and RMT Generation

To perform JES2 and RMT generations, you are required to have an MVS system control program for use as the generating system and four distribution libraries.

The Generating System

For your first JES2 and RMT generations, you must use a starter system that is provided by IBM. The starter system is a minimum MVS system control program that contains all of the programs and procedures required for JES2 and RMT generations. After your first JES2 and RMT generations using the starter system, an existing VS2 system control program of the current release can be used for subsequent JES2 and RMT generations.

The Distribution Libraries

The distribution libraries are distributed by IBM as unloaded partitioned data sets on magnetic tape. They contain all of the macro definitions and components necessary to install a system control program as well as to install the JES2 job entry subsystem and generate RTP programs.

The contents of the distribution libraries must be copied to direct-access volumes prior to the JES2 and RMT generations. These volumes must be mounted during the JES2 and RMT generations. Refer to *OS/VS2 System Programming Library: System Generation Reference* for information about copying the distribution libraries to direct-access volumes.

Four of the distribution libraries are required for JES2 and RMT generations: SYS1.AMODGEN, SYS1.AMACLIB, SYS1.AOSH1, and SYS1.AOSH2. SYS1.AMODGEN and SYS1.AMACLIB contain macro definitions and components that are used during the JES2 and RMT generation processes. SYS1.AOSH1 and SYS1.AOSH2 contain the following:

SYS1.AOSH1: This distribution library contains the following utility programs:

JESIIGEN A utility program that reads JES2 and RTP program source modules and the JES2 parameters and applies changes, based on parameter specifications, to the JES2 source modules in SYS1.HASPSRC.

EXRMTGEN A utility program that acts as an executor to the RMT generation. This utility checks the JES2 parameters to determine if workstation programs are required. If any RMT CPU parameters have been specified, it issues a message to place the RMT parameters in the card reader. Otherwise, it takes no action and returns.

REMOTGEN A utility program that acts as a monitor linking to other remote terminal utility programs and to the assembler during the RMT generation.

GENRMT A utility program that reads the card input stream during the RMT generation for the RTP program identification, selects the appropriate standard option list for the RTP program to be generated, prints the parameter default values, and updates the source modules with the changes read from the RMT parameters.

LETRRIP This utility program is an 1130 post-processor that creates an 1130 object-deck image on the SYSPUNCH data set.

SYS3CNVT This utility program is a System/3 post-processor that produces an object-deck image on the SYSPUNCH data set.

SYS1.AOSH2: This distribution library contains the JES2 and RTP program source modules.

Defining the Data Sets for Generation

Before JES2 can be installed and RTP programs can be generated, space must be allocated to the two data sets that are used during JES2 and RMT generations and they must be cataloged in the master catalog of the generating system.

Before JES2 can be initialized, at least one spool data set must have space allocated for it. Although spool data sets are not required for a JES2 generation, space can be allocated at the same time the required data sets are defined.

This section gives information about the required data sets, SYS1.HASPSRC and SYS1.HASPOBJ, the data set(s) used for spooling, SYS1.HASPACE, and the data set used for the two checkpoint records of JES2, SYS1.HASPCKPT. The space allocations that are given for the data sets are recommended when full volume allocation is not used.

SYS1.HASPSRC

SYS1.HASPSRC is a partitioned data set that contains JES2 and RTP program source modules.

Figure 2-1 lists the contents of SYS1.HASPSRC after the contents of the SYS1.AOSH2 distribution library have been placed into it.

Requirements for Specification

Location: This data set must be on a direct-access volume. The volume that contains this data set can be on a:

2305 model 1 fixed-head storage facility
2305 model 2 fixed-head storage facility
2314/2319 direct-access storage facility
3330/3333 model 1 disk storage drive
3330/3333 model 11 disk storage drive
3340/3344 disk storage drive
3350 direct-access storage

Spac? Allocation: Space allocation for this data set is:

SPACE=(1680,(3600,200,10))

The following DCB subparameters must be specified:

RECFM=FB,LRECL=80,BLKSIZE=1680

Cataloging: This data set must be cataloged in the master catalog of the generating system.

Refer to Figure 2-3 for an example of defining this data set.

SYS1.HASPOBJ

SYS1.HASPOBJ is a partitioned data set that contains object modules that have been assembled from the source modules in SYS1.HASPSRC.

Figure 2-2 lists the contents of SYS1.HASPOBJ after the JES2 object modules have been assembled into it.

Requirements for Specification

Location: This data set must be on a direct-access volume. The volume that contains this data set can be on a:

2305 model 1 fixed-head storage facility
2305 model 2 fixed-head storage facility
2314/2319 direct-access storage facility
3330/3333 model 1 disk storage drive
3330/3333 model 11 disk storage drive
3340/3344 disk storage drive
3350 direct-access storage

	Member Names	**Description**
SYS1.HASPSRC	$$POST thru $XXC	JES2 Macros
	HASPACCT	Accounting Routine
	HASPCOMM	Command Processor
	HASPCON	Console Support
	HASPDOC	Control Block Documentation
	HASPINIT	Initialization Routine
	HASPMISC	Miscellaneous Routines
	HASPNUC	JES2 Nucleus
	HASPPRPU	Print/Punch Processor
	HASPRDR	Input Processor
	HASPRTAM	Remote Support
	HASPSSSM	Subsystem Support Module
	HASPXEQ	Execution Processors
	HRTPB360	System/360 and M20 BSC Remote Program
	HRTPLOAD	1130 Loader Program
	HRTPOPTS	RMTGEN Standard Option Lists
	HRTPSYS3	System/3 Remote Program
	HRTP1130	1130 Remote Program
	NULL	JES2 Macro

Figure 2-1. Contents of SYS1.HASPSRC

	Member Names	**Description**
SYS1.HASPOBJ	HASPNUC	JES2 Nucleus
	HASPRDR	Input Processor
	HASPXEQ	Execution Processors
	HASPPRPU	Print/Punch Processor
	HASPACCT	Accounting Routine
	HASPMISC	Miscellaneous Routines
	HASPCON	Console Support
	HASPRTAM	Remote Support
	HASPCOMM	Command Processor
	HASPINIT	Initialization Routine
	HASPSSSM	Subsystem Support Module

Figure 2-2. Contents of SYS1.HASPOBJ

Space Allocation: Space allocation for this data set is:

SPACE=(400,(400,100,10))

The following DCB subparameters must be specified:

RECFM=FB,LRECL=80,BLKSIZE=400

Cataloging: This data set must be cataloged in the master catalog of the generating system.

Refer to Figure 2-3 for an example of defining this data set.

SYS1.HASPCKPT

SYS1.HASPCKPT is used for the two checkpoint records of JES2.

Requirements for Specification

Name Convention: The data set must be named SYS1.HASPCKPT and must exist on the checkpoint volume whose volume serial number is specified by the JES2 initialization parameter, &CHKPT. If &CHKPT is not specified, the JES2 generation parameter (&USPOOL) or the JES2 initialization parameter (&SPOOL) will be the default value for the volume serial number. The JES2 initialization parameters (&SPOOL and &CHKPT) are described in the chapter "JES2 Initialization."

Location: The checkpoint volume may reside on any direct-access device type. All shared JES2 systems must have at lease one channel path to the device.

The checkpoint data set is frequently referenced and is important to throughput in shared JES2 systems. This should be considered when choosing other data sets for allocation on the same volume.

It is strongly recommended that the RESERVE macro not be used for any other data set on the checkpoint volume. If necessary, the RESERVE macro should be used infrequently and for short periods of time. Serious degradation of shared JES2 throughput may result if this recommendation is ignored.

```
//ALLOCATE   JOB    (...),'PREPARE FOR JES2GEN',MSGLEVEL=1
//ALLOCAT    EXEC   PGM=IEFBR14
//HASPSRC    DD     DSN=SYS1.HASPSRC,UNIT=SYSDA,VOL=SER=JES2,
//                  DISP=(NEW,CATLG),SPACE=(1680,(3600,200,10)),
//                  DCB=(RECFM=FB,LRECL=80,BLKSIZE=1680)
//HASPOBJ    DD     DSN=SYS1.HASPOBJ,UNIT=SYSDA,VOL=SER=JES2,
//                  DISP=(NEW,CATLG),SPACE=(400,(400,100,10)),
//                  DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)
//SPOOL1     DD     DSN=SYS1.HASPACE,UNIT=3330,VOLUME=SER=SPOOL1,
//                  DISP=(NEW,KEEP),SPACE=(ABSTR,(7642,2)),
//                  LABEL=EXPDT=99366
//SPOOL2     DD     DSN=SYS1.HASPACE,UNIT=2314,VOLUME=SER=SPOOL2,
//                  DISP=(NEW,KEEP),SPACE=(ABSTR,(3998,2)),
//                  LABEL=EXPDT=99366
//CHKPT      DD     DSN=SYS1.HASPCKPT,UNIT=3330,VOLUME=SER=CHKPT,
//                  DISP=(NEW,KEEP),SPACE=(ABSTR,(42)),LABEL=EXPDT=99366
/*
```

Figure 2-3. Defining JES2 Data Sets

Space Allocation: The checkpoint data set should be allocated as a single extent within one cylinder. JES2 will use only the first extent. Space allocation should be:

SPACE=(ABSTR,(tracks,address))

tracks specifies the number of full tracks to be allocated and *address* specifies the first track.

The first checkpoint record requires one or more tracks sufficient to hold the following number of bytes:

400+3(&NUMRJE+1)+(3(&NUMRJE)+7)/8+&NUMDA((&NUMTGV+7)/8)
+28(&MAXJOBS)

The second checkpoint record requires one or more tracks sufficient to hold the following number of bytes:

100+26*&NUMJOES

The number of tracks for each record must be determined independently for each record (from the formulas and track capacity of the device type) and summed up to give the total requirement.

Refer to Figure 2-3 for an example of defining this data set.

SYS1.HASPACE

SYS1.HASPACE is the name of one or more data sets used for queuing JCL internal text, for queuing input data, and for saving job output and system messages for later output to printers and punches.

For shared JES2 systems, all spool volumes must reside on devices which have at least one channel path to each JES2 system in the multi-access spool environment.

Requirements for Specification

Naming Conventions: One or more volumes may be designated as spool volumes. The first five characters of the volume serial number of each volume defined must be identical to the first five characters specified in the JES2 parameter &SPOOL. The sixth character can be any character that is valid in volume serial numbers. One volume must be designated as the primary spool volume. All six characters of its volume serial number must agree with the six characters specified in the &SPOOL parameter. Refer to the information on the &SPOOL and &NUMDA parameter in the chapter "Installing JES2."

Each spool volume must contain a data set named SYS1.HASPACE. JES2 uses only the first extent of this data set for spooling space. If more than one extent is allocated, only the first extent is used.

Location: Spool volumes may reside on any combination of direct-access device types. JES2 utilizes space from each spool volume, ensuring full use of all allocated space.

It is strongly recommended that each spool volume be entirely devoted to JES2 usage. To allocate other frequently referenced data sets on a spool volume would degrade the efficiency of JES2's direct-access allocation algorithm.

Space Allocation: Spool data sets must be allocated as single extent data sets. Use the following specification for allocating space for the spool data sets:

SPACE=(ABSTR,(primary quantity,address))

ABSTR
specifies that the data set is to be placed at a specific location on the volume.

primary quantity
specifies the number of tracks to be allocated to the data set.

address
specifies the track number of the first track to be allocated.

To allocate a spool volume, specify both *primary quantity* and *address* as integral multiples of the number of tracks per track group. For example, specify SPACE=(ABSTR,(1000,16)) if the number of tracks per group is 8.

If other data sets must be allocated space on a spool volume, SYS1.HASPACE should be allocated such that it contains no dead space. The following text illustrates how to do this.

The unit of direct-access space allocation for JES2 is the track group. The number of tracks in a track group is obtained by dividing the total number of tracks on a volume by the value specified in the &NUMTGV parameter (the number of track groups per volume). The number of tracks for a 2314 volume is 4000 (regardless of the size the SYS1.HASPACE data set). If the value of &NUMTGV is set to 500, the number of tracks per track group is 4000/500 or 8 tracks per track group. JES2 uses only those track groups that fall completely within the SYS1.HASPACE space allocation.

Refer to Figure 2-3 for an example of defining this data set.

Specifying JES2 Generation Parameters and Control Statements

Before JES2 can be installed, you must specify parameters that define the features, options, and limits of JES2. Additionally, if changes are to be made to the JES2 program source modules, you must specify these changes in control statements.

This section discusses the rules for coding the JES2 generation parameters and control statements, and presents each parameter's function, format, and default value. (Additional information is given for some of the parameters to expand the explanations and refer to related parameters.)

Note: For a description of how to code the RMT generation parameters, refer to the chapter "Remote Job Entry."

JES2 Parameters

Each parameter is coded, beginning in column 1, in the format:

```
parameter=value
```

parameter represents a JES2 parameter, and *value* represents a permissible value you have assigned to that parameter.

The format cannot have embedded blanks. Comments can be included in a parameter statement but they must be separated from the value by one or more blanks.

JES2 parameter cards may be arranged in any order. If the same parameter occurs more than once, the last occurrence determines the parameter value. An input deck of one or more parameter cards is terminated by a card with END coded in columns 1 through 3 unless update cards are included in the input deck (see "JES2 Update Control Cards" below).

JES2 Update Control Cards

Source modules in SYS1.HASPSRC may be updated by cards punched according to formats acceptable to the IEBUPDTE utility program. This method is used to apply IBM-supplied JES2 maintenance changes and your own modifications to JES2. Updates are placed after the JES2 parameter cards, immediately following a card with UPDATE punched in columns 1 through 6.

All IEBUPDTE control cards, except the ./ALIAS... detail statement can be used to update JES2 source modules. The ./NUMBER... detail statement is accepted but it is ignored. Only the NAME, SEQ1, and SEQ2 keywords are interpreted. Other keyword parameters are ignored and may be omitted.

The update cards immediately following the control cards replace existing source card images in SYS1.HASPSRC (if columns 73 through 80 match an existing card image in the module) or are inserted between existing source card images, according to ascending collating sequence based on columns 73 through 80. Cards that are blank in columns 73 through 80 are inserted immediately following the last modification card image that is in ascending collating sequence. Update cards that do not maintain an ascending collating sequence or are not blank will terminate the JES2 generation with an update error.

All maintenance changes and your own modifications that apply to one source module must be integrated, in ascending sequence number order, into a single deck, beginning with an IEBUPDTE function statement (usually a CHANGE statement) naming the module. If more than one module is updated, the decks must be placed together so that the module names on the function statement cards are in ascending collating sequence in the same order as the source modules listed in Figure 2-1.

The last update card must be followed by a ./ENDUP control card, a /* delimiter card, or a machine-generated end-of-file. Figure 2-4 shows a composite deck of JES2 parameters and source updates in correct order.

```
Columns
1                                                                 73      80

&NUMLNES=1
&BSCCPU=YES
UPDATE                         (END if no source updates follow)
./        CHANGE NAME=HASPMISC
          .
          .
          (modifications to module HASPMISC)                      nnnnnnnn
          .
          .
          .
./        CHANGE NAME=HASPRTAM
          .
          .
          .
          (modifications to module HASPRTAM)                      nnnnnnnn
          .
          .
          .
./        ENDUP
/*
```

Figure 2-4. JES2 Parameter and Update Deck

The IEBUPDTE utility program is detailed in *OS/VS Utilities.*

Specifying the JES2 Parameters

The following conventions are used in this manual to describe the JES2 parameters:

- The JES2 parameters are discussed alphamerically; the first character is ignored if it is & or $.
- Letters and numbers in bold type must be coded as shown.
- Lowercase letters in italics represent variables for which you must substitute specific information or specific values.
- If an alternative item is underlined, it is the default value. This value will be used automatically if the parameter is not specified.

Parameter	**Value**	**Explanation**
&BSCCPU=	**NO** [✓ **YES**]	specifies the inclusion or exclusion of support for JES2 MULTI-LEAVING remote job entry (RJE) in the remote terminal access method (RTAM).
&BSC2770=	**NO** **YES**	specifies the inclusion or exclusion of RJE support in RTAM for the 2770 Data Communication System.
&BSC2780=	**NO** **YES**	specifies the inclusion or exclusion of RJE support in RTAM for the 2780 Data Transmission Terminal.
&BSC3780=	**NO** [✓ **YES**]	specifies the inclusion or exclusion of RJE support in RTAM for the 3780 Data Communication Terminal.
&BSHPRES=	**NO** [✓ **YES**]	specifies the inclusion or exclusion of RJE support in RTAM for the space compression/expansion feature of 2770 and 3780 terminals. This parameter must be specified if any 2770 or 3780 terminal is to transmit to JES2 using the space compression/expansion feature. Use of this support for output to any terminal is controlled by specification in the RMT*nnn* parameter during JES2 initialization.
&BSHTAB=	**NO** **YES**	specifies the inclusion or exclusion of support in RTAM for the printer horizontal format control feature for 2770, 2780, or 3780 terminals. Use of this support for output to any 2770, 2780, or 3780 terminal is controlled by specification in the RMT*nnn* parameter during JES2 initialization.
$BSPACE=	*hexadecimal number* **5F**	specifies the character that will be interpreted as the machine-defined backspace character X'16'. You specify this parameter by coding the two hexadecimal digit representation of the EBCDIC character. When the character specified by this parameter is entered from any operator console, it will be removed from the command text along with the previously entered character (if any). Characters

Parameter	Value	Explanation
$BSPACE= (continued)		following the character will be shifted left to replace the removed characters. This function applies to all commands entered by means of operator command input sources, regardless of their position in the text of the data entered. This function does not apply to a JES2 card reader or remote work station sources. The default value indicates that the EBCDIC character "¬" is to be used for backspace command entry on operator consoles. The character selected for the backspace function must be chosen with caution since it eliminates the use of that character (except as a backspace operation) in all commands and replies to WTORs.
&BSVBOPT=	NO YES	specifies the inclusion or exclusion of code in RTAM to recognize an EM (end of media) punch in card images transmitted nontransparently by the 2780 Data Transmission Terminal.
&BUFSIZE=	*value* 1960	specifies the size, in bytes, of each JES2 buffer. If the value specified is not a multiple of eight, it will be adjusted upward to a multiple of eight. The value specified must be an integer not larger than 4008 and not smaller than **600+(&NUMDA(&NUMTGV/8))** The default value is the maximum size that will allow two buffers per page of virtual storage and good utilization of 2314 and 3330 track capacities (three or six records per track, respectively). The maximum value that can be specified (4008) allows three records per track on a 3330. Each JES2 buffer is allocated to virtual storage, so the input/output block (IOB), which is 88 bytes, and the data area (the number of bytes in &BUFSIZE) are always contained in a single 4K page.
&CCOMCHR=	*character* $	specifies the character that will be used to identify JES2 commands from local consoles. If a command from a local console begins with the character specified by this option, JES2 will assume that the command is a JES2 command and will attempt to process the command. The value you specify should be a special character that is not used as the first character of any command of any other subsystem that may be operated concurrently with JES2. The specification must not be a letter, a number, a comma or an apostrophe, and must not be the character specified by the $BSPACE parameter.

Parameter	Value	Explanation
&CCOMCHR= (continued)		The value of this option may be overriden by specifying the &CCOMCHR parameter at JES2 initialization. If this parameter is changed to other than its default value, the commands will vary from their documented format in *Operator's Library: OS/VS2 Reference (JES2).*
$CKPTIME=	*number* 60	specifies, in seconds, the interval at which certain JES2 information will be checkpointed for warm start. (Checkpoints are also taken when a job changes its status in the JES2 job queue.) The time interval specified is a maximum checkpoint time for a non-shared JES2 system (one CPU or MP). For JES2 systems sharing spool and checkpoint volumes, the time interval specified is used for print/punch checkpointing only. For these systems, a value of 120 or greater is recommended. The JES2 initialization parameter, QCONTROL, specifies other checkpoint intervals for shared systems.
&DEBUG=	NO YES	specifies the inclusion or exclusion of debugging code in JES2.
$DELAYTM=	*number* 100	specifies, in microseconds, the delay time to be applied by RTAM when transmitting to either a MULTI-LEAVING System/360 Model 20, submodel 2, 4, or 6, or to a 2922 remote terminal over a high-speed (19,200 baud or greater) teleprocessing line. This delay time avoids the possibility of certain line errors. If data overrun line errors occur at the work station when the default value is used, the value should be increased.
&DMNDSET=	NO YES	specifies whether inline printer setup will be allowed for data sets whose SYSOUT class matches the job message class. If the default value is not used, all SYSOUT data sets that are not specified for special processing .in any other way (for example, HOLD) and whose class matches the message class, will be printed on one printer with appropriate setup messages to the operator as the data sets are printed. If &DMNDSET=NO is specified or if the SYSOUT class does not match the message class, separate class work queues will be created for each unique setup required. Thus, data sets

Parameter	Value	Explanation
&DMNDSET= (continued)		can be printed simultaneously on all printers available.
$ESTIME=	*number* 2	is an integer greater than 0 that specifies, in minutes, the estimated execution time for a job. This value will be used if you do not specify a value for estimated execution time in the accounting field of your JOB card or on a /*JOBPARM control card.
$ESTLNCT=	*number* 2 12	is an integer greater than 0 that specifies, in thousands of lines, the estimated print line count for a job. This value will be used if you do not specify a value for the estimated print line count in the accounting field of your JOB card or on a /*JOBPARM control card.
$ESTPUN=	*number* 100	is an integer greater than 0 that specifies, in number of cards, the estimated punched card output for a job. This value will be used if you do not specify a value for the estimated card count in the accounting field of your JOB card or on a /*JOBPARM control card.
&JCOPYLM=	*number* 3	is an integer from 1 to 255 that specifies the maximum number of job output copies that can be requested in the accounting field of your JOB card or on a /*JOBPARM control card. If the number of copies requested is greater than the value of &JCOPYLM, the request is reduced to the value of &JCOPYLM. No error message is produced. The setting of this parameter does not affect requests for multiple copies of data sets via an /*OUTPUT control card.
$LINECT=	*number* 61	specifies the maximum number of lines to be printed per page of a job's printed output. This value is used if you do not specify a value for line count in the accounting field of your JOB card or on a /*JOBPARM control card. $LINECT=0 causes automatic page overflow, normally standard in JES2, to be suppressed unless overridden by the JOB card accounting parameter or a /*JOBPARM control card specification.

Parameter	Value	Explanation
$LINECT= (continued)		If a print data set is generated without any ejects (no skips to any channel in the carriage tape), and if 0 is specified in either this parameter, the JOB card accounting field, or a /*JOBPARM control card, the data set is treated as one page, when forward-spaced, backspaced, interrupted, or warm started while printing.
&MAXCLAS=	*number* 8	specifies the maximum number of job classes that may be specified through the JES2 operator command $TI*nn*. Since there are thirty-six unique job classes, the maximum value that can be specified is 36.
&MAXJOBS=	*number* 100	is an integer from 10 to 8000 that specifies the maximum number of jobs that can be in JES2 at any given time. The value specified does not affect the range of JES2 job numbers, 1 to 9999.
&MAXPART=	*number* 3	is an integer from 1 to 99 that specifies the number of JES2 batch initiators to be defined.
&MINJOES=	*number*	specifies the minimum number of free job output elements. When the free count drops below this value, no new work is added to the in-storage queues until the termination of a print or punch activity raises the free count. The default value is calculated as follows: **&NUMJOES/5** If the job output element free count is allowed to go to zero via operator use of the $I command in a congested system, output devices may become interlocked waiting for resources.
&MLBFSIZ=	*number* 400	specifies the size, in bytes, of each JES2 MULTI-LEAVING buffer. The specification for this parameter must be a positive integer that is no greater than the value specified in the &TPBFSIZ parameter. The value specified for this parameter automatically becomes the MULTI-LEAVING buffer size in each JES2 MULTI-LEAVING RTP program. Satisfactory support of one device of each type (reader, printer, punch, console) on 8K terminal CPUs is based on the assumption that &MLBFSIZ is 400 or less. If the supported terminals include any 8K CPUs, it is recommended that &MLBFSIZ not be increased above 400, even if support of a nonprogrammable terminal requires increasing the value specified in the &TPBFSIZ parameter to 516.

Parameter	Value	Explanation
&MSGID=	**YES** **NO**	specifies whether or not the eight-character message identifier (HASP*nnn*ƀ) should be appended to the beginning of each JES2 operator message. If the default value is not used, the operator messages produced by JES2 will vary from their documentation in *OS/VS Message Library: VS2 System Messages.*
&NOPRCCW=	*number* 30	specifies the maximum number of channel command words per channel program for local printers. The recommended value for this parameter can be determined by the following formula: **&NOPRCCW=&BUFSIZE**/*average print line length* The average length of the print line should be estimated after allowing for truncation of trailing blanks by JES2.
&NOPUCCW=	*number* 30	specifies the maximum number of channel command words per channel program for local punches. The recommended value for this parameter can be determined by the following formula: **&NOPUCCW=&BUFSIZE**/*average card length* The average card length should be estimated after allowing for truncation of trailing blanks by JES2.
&NUMACE	*number* 20	is an integer between 2 and 9999 that specifies the number of automatic commands that can be concurrently active in JES2. The value should be sufficiently large to permit operators to leave a JES2 dynamic display in each "out of line" area of all graphic display consoles as well as one on each printer console controlled by VS2. For additional information, see the description of the $TA command in *Operator's Library: OS/VS2 Reference (JES2).*
&NUMBUF=	*number* 24	specifies the number of I/O buffers to be included in the JES2 load module. The value specified should reflect the total number of buffers required for proper operation of JES2. Because all JES2 buffers are maintained in a dynamic pool until required by an active function, the appropriate number of buffers should be determined, based on the predicted simultaneity of the various functions required at the installation. The following list indicates the number of buffers

Parameter	Value	Explanation
&NUMBUF= (continued)		required for each logical function. (A defined function which is inactive requires no buffers.) 6 for normal system processing 5 for each local input function 4 for each internal reader 4 for each remote input function 2 for each local print function (1 if $PRTBOPT=1) 1 for each remote print function (2 if $RPRBOPT=2) 1 for each punch function (2 if $PUNBOPT=2) 1 for each remote punch function (2 if $RPUBOPT=2) To avoid a complete system failure caused by a buffer "lockout" condition, the number of available buffers must never be less than the value determined by the following formula: **&NUMBUF=5+(4(&NUMRDRS))** **+(3(&NUMINRS))** **+3(&NUMTPRD)** **+(&NUMPRTS($PRTBOPT–1))** **+(&NUMPUNS($PUNBOPT–1))** **+(&NUMTPPR($RPRBOPT–1))** **+(&NUMTPPU($RPUBOPT–1))** This value may be increased by specifying the &NUMBUF parameter at JES2 initialization.
&NUMCLAS=	*number* 8	specifies the maximum number of classes for which a printer or a punch may be simultaneously started. Since there are thirty-six unique SYSOUT classes, the maximum allowable value that can be specified is 36.
&NUMDA=	*number* 2	is an integer greater than 0 that specifies the maximum number of direct-access volumes that can be mounted concurrently as spool volumes. All direct-access devices supported in OS/VS2 are eligible for use as spooling devices. Specifying a large number for &NUMDA may require increasing the value of &BUFSIZE. Refer to the information on the &NUMTGV and &SPOOL for related information.
8 &NUMINRS=	*number* 2	specifies the number of internal readers to be part of JES2.
&NUMJOES=	*number*	is an integer from 10 to 5000 that specifies the number of job output elements to be generated for printers and punches. The default value is calculated as follows: **&NUMJOES=10(&NUMPRTS+&NUMPUNS** **+&NUMTPPR+&NUMTPPU)**

Parameter	Value	Explanation
&NUMJOES= (continued)		Although a value as small as 10 will be accepted, performance will be degraded if a value smaller than the default is specified. One job output element is required for: • Each unique SYSOUT class that appears in a job that is queued for output • Each active printer or punch • Each interrupted or restarted job that is not currently active on a printer or punch • Each unique combination of Forms ID, UCS ID, and FCB ID for all jobs currently queued for output • Each job that was interrupted by a system failure while being printed or punched and has not yet been warm started on an output device. Specifying too small a value results in jobs waiting for in-storage queueing in order to complete active print or punch work. Additional information on estimating the value for &NUMJOES to optimize JES2 performance is provided in the chapter “JES2 Performance.”
4 &NUMLNES=	*number* 0	is a number from 0 to 255 that specifies the largest teleprocessing line identification number, thus the number of line definitions, to be used by JES2. The value specified becomes the specification for &NUMRJE, unless &NUMRJE is specified explicitly.
&NUMPRTS= ?	*number* 2	is a number from 0 to 99 that specifies the maximum number of local printers JES2 can use to print job output. If more printers are specified during system generation than are specified in this parameter, a message is written to the operator. JES2 initialization continues normally but only the printers specified or those with the lowest unit addresses are used.
&NUMPUNS=	*number* 1	is a value from 0 to 99 that specifies the maximum number of local punches JES2 can use to punch job output. JES2 supports 2520, 2540, and 3525 card punches. If more punches are specified during system generation than are specified in this parameter, a message is written to the operator. JES2 initialization continues normally but only the punches specified or those with the lowest unit addresses are used.

Parameter	Value	Explanation
&NUMRDRS=	*number* 1	is a value from 0 to 99 that specifies the maximum number of local card readers JES2 can use to read jobs. JES2 supports 2501, 2540, and 3505 card readers. If more card readers are specified during system generation than are specified in this parameter, a message is written to the operator. JES2 initialization continues normally but only the card readers specified or those with the lowest unit addresses are used.
&NUMRJE=	*number* &NUMLNES	is a value from 0 to 255 that specifies the number of remote terminal definitions to be used by JES2. If this parameter is not specified, the value specified in the &NUMLNES parameter is used as the default value.
&NUMSMFB=	*number* 5	is a value greater than or equal to 0 that specifies the number of JES2 system management facility (SMF) buffers to be assembled into JES2. If the value specified is less than 2, JES2 will neither produce SMF records nor take the SMF exit, IEFUJP. For additional information on specifying this parameter see the &SMFRSIZ parameter.
&NUMTGV=	*number* 400	specifies the number of units (track groups) into which each spool volume is divided for JES2 allocation purposes. The specification must be an integer no greater than the number of tracks on the spool device with the fewest tracks, subject to the following limitation: $\&NUMTGV=8\,\frac{\&BUFSIZE-600}{\&NUMDA}$ You should decide upon the number of tracks required in a track group and then divide by that number the total number of tracks (except alternate tracks) on a typical spool device at your installation. For example, to obtain a track group size of ten tracks on a 2314, you would specify a value of 400 in this parameter. If, at some time, your installation uses a 3330 as a spool device, the track group size for the 3330 would automatically become 19 tracks. For each spool volume found during initialization, JES2 calculates the number of tracks per group by dividing the total number of tracks on the volume by the value specified in this parameter. It then marks as unavailable for JES2 spooling all track groups that lie partially or wholly outside the first extent of data set SYS1.HASPACE on that volume.

Parameter	Value	Explanation
&NUMTGV= (continued)		Specifying a large value in this parameter may require specifying a large value in the &BUFSIZE parameter.
&NUMTPBF=	*number* &NUMLNES	is a value greater than or equal to 0 that specifies the number of teleprocessing buffers to be assembled for RJE. If this parameter is not specified, the value specified in the &NUMLNES parameter is used as the default value. Each signed-on JES2 MULTI-LEAVING terminal requires at least two JES2 teleprocessing buffers. All other signed-on terminals require at least one buffer each. If a MULTI-LEAVING terminal has more than one output function running concurrently, additional buffers can be used to increase performance. It is recommended that this value be made liberally large (for example 5X &NUMLNES) in systems that support MULTI-LEAVING terminals. For additional information, refer to the &TPBFSIZ and &MLBFSIZ parameters.
&NUMTPPR=	*number* &NUMLNES	is a value greater than or equal to 0 that specifies the maximum number of JES2 remote terminal printed output streams (including MULTI-LEAVING) that can be active simultaneously. If this parameter is not specified, the value specified in the &NUMLNES parameter is used as the default value. If any remote terminal is to receive printed output, this parameter value must not be 0.
&NUMTPPU=	*number* &NUMLNES	is a value greater than or equal to 0 that specifies the maximum number of JES2 remote terminal punched output streams (including MULTI-LEAVING) that can be active simultaneously. If this parameter is not specified, the value specified in the &NUMLNES parameter is used as the default value. If any remote terminal is to receive punched output, this parameter value must not be 0.
&NUMTPRD=	*number* &NUMLNES	is a value greater than or equal to 0 that specifies the maximum number of JES2 remote terminal input streams (including MULTI-LEAVING) that can be active simultaneously. If this parameter is not specified in the &NUMLNES parameter is used as the default value. If any remote terminal is to read cards, this parameter value must not be 0.
&NUMWTOQ=	*number* 15	is a value greater than 2 that specifies the number of message buffers to be provided for JES2. The number of buffers specified should be sufficient to accommodate all outstanding operator requests

Parameter	Value	Explanation
&NUMWTOQ= (continued)		and still allow message processing to continue. Additional information on estimating this value to optimize JES2 performance is provided in the chapter "JES2 Performance." If RJE is used, more message buffers will be needed. This is especially true with console support for MULTI-LEAVING terminals. Serious system degradation can be caused by specifying too few message buffers. During periods of high console activity, when no message buffers are available, certain noncritical JES2 messages are discarded to avoid delaying the associated process. These noncritical messages include certain RJE-oriented messages, execution time/line/card excession messages, and certain I/O error messages on JES2-controlled devices. This value may be increased by specifying the &NUMWTOQ parameter at JES2 initialization. For a MULTI-LEAVING terminal console, if more messages are queued than the number of buffers specified in this parameter, the excess messages are spooled and later printed. Normal message processing resumes after the terminal accepts those messages that were queued prior to reaching the &NUMWTOQ limit. Changing the value at JES2 initialization does not affect this limit.
&OUTPOPT=	**0** [✓ 1] 2	specifies the action to be taken when a job exceeds its estimated print lines or punched cards of output. 0 allows the job to continue execution 1 causes the job to be cancelled without a dump 2 causes the job to be cancelled with a dump Regardless of the specification for this parameter, output excession causes messages to be written to the operator (refer to the $OUTXS parameter for additional information). If 2 is specified, you should use SYSUDUMP or SYSABEND DD cards if a storage dump is desired on output excession. If 1 or 2 is specified, the job will not be cancelled if, at the time of output excession, the job task is normally or abnormally terminating.
$OUTXS=	*number* **2000**	is a value greater than 0 that specifies the interval, in print lines/punched cards, at which messages will be written to inform the operator that a job's print line count or punch card count has been exceeded.

Parameter	Value	Explanation
$OUTXS= (continued)		The first of these messages for print lines will be written to the operator when the job's estimated print line count has been exceeded. The first of these messages for punched cards will be written to the operator when the job's estimated punched card count has been exceeded.
$PRIDCT=	*number* 61	is a value greater than or equal to 0 that specifies the number of print lines to appear on each JES2 job separator page for local printers. If the specification is 0, no separator pages are produced. If the specification is 30 or greater, the first twenty-nine lines are used to produce a block-lettered job name, job number, and SYSOUT class. The equivalent parameter for remote terminal printers is $TPIDCT.
&PRIHIGH=	*number* 10	is an integer from 0 to 15 that specifies the priority to be associated with the JES2 priority aging feature. A job will not be priority-aged if its priority is (or becomes) greater than or equal to the value specified in this parameter.
&PRILOW=	*number* 5	is a value from 0 to 15 that specifies the priority to be associated with the JES2 priority aging feature. A job will not be priority-aged unless its priority is initially equal to this value. (Refer to the &PRIRATE and &PRIHIGH parameters for additional information.)
&PRIRATE=	*number* 0	is a value greater than or equal to 0 that specifies the amount by which a job's priority will be incremented in twenty-four hours by the priority aging feature. For example, if 3 is specified, a job's priority will be incremented by one for every eight hours it remains in the system. However, a job's priority will not be incremented unless it is at least equal to the value specified in the &PRILOW parameter; nor will a job's priority be incremented above the value specified in the &PRIHIGH parameter. If 0 is specified, the values specified in the &PRILOW and &PRIHIGH parameters are ignored. If a job's priority is specified on a /*PRIORITY control card, the job will be priority-aged if its priority is eligible. Refer to the &RPRT(*n*), &RPRI(*n*), &XLIN(*n*), and &XPRI(*n*) parameters for additional information.
$PRTBOPT=	1 2	specifies the printer buffering option to be used for local JES2 printers. 1 specifies single buffering. 2 specifies double buffering.

Parameter	Value	Explanation
&PRTFCB=	*name* 6	is a one to four-character name that specifies the forms buffer image or the carriage control tape that JES2 initially assumes is mounted on every printer. The forms control block (FCB) identifier can be modified by the operator for each printer.
&PRTRANS=	NO YES	specifies translation for lines of print not directed to 3211 printers. If the default value is used, each line to be printed by a local 1403 or any remote printer is first translated. Translation changes lower-case letters to upper-case and characters that are invalid on a PN train to blanks. If any print train is to be used on a JES2-controlled local 1403 or remote printer that has characters not equivalent to those on a PN train, NO must be specified. If all printers are 3211s (not 1403s or remotes), this parameter should be specified as NO.
&PRTUCS=	*image* 0	specifies the name of the print chain or print train that JES2 initially assumes is mounted on every printer. The UCS identifier can be modified by the operator individually by printer. The image identifier you specify must be an image that is present in SYS1.IMAGELIB. If 0 is specified, JES2 will bypass the UCS loading procedure until a job is processed that requires a specific UCS image. If an invalid specification is encountered, the UCS loading procedure will be bypassed and a setup message will be issued to allow specification of a valid image. Provisions for supporting other types of print chains are described in *OS/VS2 System Programming Library: Data Management.*
$PUNBOPT=	1 2	specifies the punch buffering option to be used for local JES2 punches. 1 specifies single buffering. 2 specifies double buffering.
&RCOMCHR=	*character* $	specifies the character that will be used to identify JES2 commands from JES2 input devices. If a JES2 control card is read (/* in columns 1-2) that contains this character in column 3, JES2 will assume that the card is a JES2 command statement and will attempt to process the command. The specification should be a special character other than a comma or apostrophe. The value of this parameter may be overridden by a &RCOMCHR parameter at JES2 initialization.

Parameter **Value** **Explanation**

&RCOMCHR=
(continued)

If this parameter is changed to other than its default value, the command control statement will vary from the format given in *OS/VS2 JCL*.

&RDROPSL= *character string*

specifies a 20-character parameter field to be passed to the OS/VS2 converter for TSO LOGONs (foreground jobs).

The default value is as follows:

00114400000030E00000

For a description of the parameter field refer to the &RDROPSU parameter.

The value of this parameter may be overridden by the &TSU parameter at JES2 initialization. The programmer name is not required for the &TSU parameter unless a user exit is also provided to supply it to the JOB card generated by LOGON.

&RDROPST= *character string*

specifies a 20-character parameter field to be passed to the OS/VS2 converter for console started tasks.

The default value is as follows:

01100100000000E00000

For a description of the parameter field, refer to the &RDROPSU parameter.

The value of this parameter may be overridden by the &STC parameter at JES2 initialization.

&RDROPSU= *character string*

specifies a 20-character parameter field to be passed to the OS/VS2 converter for batch (background) jobs.

The default value is as follows:

00100300012820E00001

The form of this specification is:

bppmmmmsscccrlaaaaef

b is a numeric character 0, 1, 2, or 3, that indicates whether an account number is required and whether a programmer name is required. The following chart shows the meaning of each character.

Character	Accounting Information Required	Programmer Name Required
0	No	No
1	No	Yes
2	Yes	No
3	Yes	Yes

Parameter	Value	Explanation	
&RDROPSU= (continued)		*pp*	currently unused. Code two zeros to maintain positioning within the parameter field.
		mmmmss	six numeric characters indicating the default for the maximum time that each job step may run. The first four characters indicate minutes, the last two indicate seconds. The maximum time allowed is 144000.
		ccc	three numeric characters indicating the default for the region size (specified as a number of 1024-byte blocks) assigned to each job step. This region size is assigned when no region size is specified in the JOB or EXEC statement and the job step is to be run with ADDRSPC=VIRT. It is recommended that a *ccc* value of 000 not be specified because this is the ultimate region size default when both the REGION parameter has been omitted from the JOB and EXEC statements, and the CONVPARM keyword has been omitted from the &x JES2 initialization parameter.
		r	a numeric character 0, 1, 2, or 3, that specifies the disposition of commands read from the input stream. The character has the following meanings: 0 – the OS/VS2 converter passes the command to the command scheduling routine to be executed. 1 – the OS/VS2 converter displays the command (via a WTO macro instruction), and passes it to the command scheduling routine to be executed. 2 – the OS/VS2 converter displays the command (via a WTO macro instruction), asks the operator whether the command should be executed (via a WTOR macro instruction), and passes the command to the command scheduling routine if the operator replies yes. 3 – the OS/VS2 converter ignores the command and treats it as a "no operation."

Parameter **Value** **Explanation**

&RDROPSU=
(continued)

l a numeric character 0 or 1 that specifies the bypass label processing option. The character has the following meanings:

0 – The bypass label processing parameter in the label field of a DD statement is to be ignored; the label parameter is processed as no label.

1 – Bypass label processing is not to be ignored; the label parameter is processed as it appears.

aaaa four hexadecimal numbers from 0000 to E000 indicating which operator command groups are to be executed.

Bit settings are as follows:

Byte	Bits	Bit Settings	Meaning
0	0	1	Group 1 commands
	1	1	Group 2 commands
	2	1	Group 3 commands
	3-7	00000	Reserved
1	0-7	00000000	Reserved

ef two numeric characters that specify a message level value for use when the MSGLEVEL parameter is not specified on a JOB statement. If a MSGLEVEL parameter is not specified, JCL and allocation/termination messages are recorded in the system message data set according to the value specified in this parameter. The characters have the following meanings:

e specifies the kinds of JCL listed. The character can be 0, 1, or 2, which mean:

0 – JOB statement only.

1 – input statements, cataloged procedure statements, and symbolic parameter substitution values.

2 – input statements only, including instream procedures.

Parameter | **Value** | **Explanation**

&RDROPSU=
(continued)

f specifies the kinds of allocation/termination messages listed. The character can be 0 or 1, which mean:

0 – No messages are to be listed, except in the case of an abnormal termination. (In that event, all messages are listed.)

1 – All messages are listed.

The value of this parameter may be overridden by individual job class by the *&x* parameters at JES2 initialization.

&RJOBOPT= *number*
2

is an integer from 0 to 5 that specifies the type of scan that should be performed on JOB cards that are processed by the JES2 input processor and whether an illegal JOB card is to prevent execution of the associated job. The specifications have the following meanings:

Value	**Scan JES2 parameters**	**Terminate on JES2 parameter error**	**Terminate on OS/VS2 format error**
0	Yes	Yes	Yes
1	Yes	Yes	No
[✓ 2]	Yes	No	Yes
3	Yes	No	No
4	No	–	Yes
5	No	–	No

The scan of JOB card parameters CLASS, MSGCLASS, and TYPRUN is always performed. The only JES2 JOB card parameters scanned are those included in the JOB card accounting field as defined in *OS/VS2 JCL.*

If the value specified for this parameter is 0 or 1, the JOB card must have an accounting field subparameter and the first two JES2-defined parameters must be present.

A JES2 parameter error is any violation of the requirements of the JES2 JOB card parameters. An OS/VS2 format error is any error which prevents JES2 from continuing the scan of the JOB card as defined in *OS/VS2 JCL.*

$RPRBOPT= 1
2

specifies the printer buffering option to be used for all printers at JES2 remote terminals.

1 specifies single buffering, and
2 specifies double buffering.

The specification refers to JES2 regular buffers, not to JES2 teleprocessing buffers.

Parameter | **Value** | **Explanation**

&RPRI(*n*)= | *priority* | is an integer from 0 to 15 that specifies job scheduling priorities that are associated with execution times as specified in a corresponding &RPRT(*n*) parameter. If these parameters are not specified, the following values are used as defaults:

&RPRI(1)=9
&RPRI(2)=8
&RPRI(3)=7
&RPRI(4)=6
&RPRI(5)=5
&RPRI(6)=4
&RPRI(7)=3
&RPRI(8)=2
&RPRI(9)=1

If you do not supply a /*PRIORITY control card with a job, the priority of that job is determined as follows:

The queueing priority is computed as:

priority=(**&RPRI(*n*)+&XPRI(*m*)**)/2

The subscript *n* is the smallest number for which:

t≤**&RPRT(*n*)**

The subscript *m* is the smallest number for which:

o ≤**&XLIN(*m*)**

where:

t is the estimated execution time, taken from the accounting field of the JOB card of from the /*JOBPARM control card, and

o is the sum of the estimated output lines and cards, taken from the accounting field of the JOB card or from the /*JOBPARM control card.

Refer to the &RPRT(*n*) and the &XPRI(*n*) parameters for additional information.

&RPRT(*n*)= | *execution-time* | is a value from 1 to X'FFFFFF' /60 that specifies execution times that are to be associated with job scheduling priorities as specified in a corresponding &RPRI(*n*) parameter. If these parameters are not specified, the following are used as defaults:

Parameter	Value	Explanation
&RPRT(*n*)= (continued)		&RPRT(1)=2 &RPRT(2)=5 &RPRT(3)=15 &RPRT(4)=X'FFFFFF'/60 &RPRT(5)=X'FFFFFF'/60 &RPRT(6)=X'FFFFFF'/60 &RPRT(7)=X'FFFFFF'/60 &RPRT(8)=X'FFFFFF'/60 &RPRT(9)=X'FFFFFF'/60 If a /*PRIORITY control card is specified for a job, these values are not used. Refer to the &RPRI(*n*) and &XPRI(*n*) parameters for additional information.
&RPS=	NO YES	specifies the inclusion or exclusion of rotational position sensing for JES2 channel programs directed to direct-access devices with the rotationals position sensing feature. JES2 will correctly operate with any supported direct-access device or combination of devices, regardless of the specification of this parameter.
$RPUBOPT=	1 2	specifies the punch buffering option to be used for all punches at JES2 remote terminals. 1 specifies single buffering. 2 specifies double buffering. The specification refers to JES2 regular buffers and not to JES2 teleprocessing buffers.
&SMFRSIZ=	*number* 236	is an integer greater than or equal to 236 that specifies the size of the largest SMF record to be written by JES2 or the size of one SMF common exit parameter area, whichever is greater. If &NUMSMFB is specified less than 2, this parameter is ignored. Otherwise, JES2 will generate SMF records and the value specified in this parameter will be the maximum size of the SMF records written by JES2. Refer to &NUMSMFB for additional information.
&SPOLMSG=	*number* **4096/&BUFSIZE(6)**	is a value greater than or equal to 0 that specifies the number of physical records (in the first extent of SYS1.HASPACE on the primary spool volume) to be reserved for operator messages and JES2 messages for each JES2 remote terminal. Total space reserved is &SPOLMSG*&NUMRJE. Each physical record is capable of holding one or more messages for a single remote terminal. Messages are held if they are directed to: • Any terminal not signed on, or

Parameter	Value	Explanation
&SPOLMSG= (continued)		• Any signed-on computer terminal that is not a MULTI-LEAVING terminal with a console. Space for each terminal is separate from that of other terminals. If a message is to be held and the terminal's space is filled, the oldest held messages for that terminal are discarded without operator notification. Each message to a terminal (except to a MULTI-LEAVING remote terminal with a console) is held until it can be printed, or until JES2 is cold started. Only the $DM command can generate messages to a terminal that is not signed on. For signed-on terminals, messages are generated for job-on reader, by the $DM command, and as responses to commands from the terminal. The maximum value that can be specified for this parameter is 256.
&SPOOL=	*volser* **SPOOL1**	specifies the volume serial number of the direct-access volume that is to be used as the primary spool volume by JES2. The specification must be six characters and must be valid as a volume serial number. When you define the spool volumes, the first five characters of the volume serial number of each volume must be identical to the first five characters specified in this parameter. The sixth character can be any character that is valid in a volume serial number. One volume must be designated as the primary spool volume. All six characters of its volume serial number must agree with the six characters specified in this parameter. The value of this parameter may be overridden by the JES2 initialization parameter, &SPOOL. Refer to the information about spool data sets in "Defining the Data Sets for Generation." If a value other than SPOOL1 is specified, certain messages will vary from their documentation in *OS/VS Message Library: VS2 System Messages.*
&STDFORM=	*Forms*-ID **STD**	is a 4-character alphameric value that will be used as a default Forms ID when a Forms ID is not specified. Also, this parameter specifies the default initial setup of all printers and punches at JES2 initialization.

Parameter	Value	Explanation
&SYNCTOL=	*number* 120	specifies in seconds, the time interval which must elapse before one JES2 system in a multiaccess spool environment is assumed to be not operating. This parameter allows for imprecise synchronization of TOD clocks in a multiaccess spool environment, which must be accomplished by human intervention. Actions such as a cold start, warm start, or $ESYS operator command is rejected unless the time stamps of affected systems in the shared checkpoint record are less than the current time minus this parameter.
&TGWARN=	*number* 80	is a number that specifies the threshold percentage use of spool space that will cause the message $HASP093 *xxx%* SPOOL UTILIZATION to be issued by JES2. You specify a value from 0 through 101. If you specify 0, every use of spool space will be reported. If you specify 101, no message will be issued.
&TIMEOPT=	NO [✓ YES]	specifies the inclusion or exclusion of the elapsed time job monitor feature of JES2. If YES is specified, the operator will be informed when a job's estimated execution time is exceeded and notified periodically thereafter. Refer to the $ESTIME and $TIMEXS parameters for additional information.
$TIMEXS=	*number* 1	is a value greater than 0 that specifies, in minutes, the interval at which messages will be written to the operator informing him that a job's execution time is exceeded. The first of these messages is written to the operator when the job's estimated execution time has been exceeded. If the &TIMEOPT parameter is specified as NO, this parameter is ignored.
&TPBFSIZ=	*number* 400	is an integer less than or equal to 3976 that specifies the maximum size of the JES2 teleprocessing buffers. The value specified must be greater than the value specified in the &MLBFSIZ parameter. The information specified in this parameter is conveyed automatically to the requisite RTP programs by the JESIIGEN utility.
$TPIDCT=	*number* 6	is a value greater than 0 that specifies the number of print lines that are to appear on each JES2 job separator page for printed output for remote terminal printers. If the specification is 30 or greater, the first twenty-nine lines will be used to

Parameter	Value	Explanation
$TPIDCT= (continued)		produce a block-lettered job name, job number, and SYSOUT class. The equivalent parameter for local terminal printers is $PRIDCT.
&USASCII=	NO YES	specifies the inclusion or exclusion in RTAM of the support to use ASCII line-control characters, as well as EBCDIC line-control characters.
$WAITIME=	*number* 1	is a value greater than 0 that specifies, in seconds, the length of time that RTAM should wait at the completion of the processing of any input stream, printed output stream, or punched output stream, to allow the operator to alter the normal sequence of RJE operations. For example, the operator may want to transmit another job to JES2 after a previous job has finished printing, rather than wait until all queued output has finished processing.
&XBATCH=	NO [✓ YES]	specifies the inclusion or exclusion of support for the JES2 execution batch scheduling feature.
&XBATCHN=	*name* [✓ $$$$$] FAST $	specifies the first five characters of the name of each OS/VS2 program or procedure to be started internally by JES2 when required to execute a user-job under the execution batch scheduling feature. The specification must be a five-character string, of which the first character is alphabetic or national and the remaining four are alphameric or national. If the &BATCH parameter is specified, JES2 will reject all user-submitted jobs whose jobnames begin with the five characters specified in this parameter.
&XLIN(*n*)=	*number*	is a value from 1 to 16,777,215 that specifies the output record counts that are associated with the priorities as specified in the &XPRI(*n*) parameter. If this parameter is not specified, the following are used as defaults: **&XLIN(1)=2000** **&XLIN(2)=5000** **&XLIN(3)=15000** **&XLIN(4)=X'FFFFFF'** **&XLIN(5)=X'FFFFFF'** **&XLIN(6)=X'FFFFFF'** **&XLIN(7)=X'FFFFFF'** **&XLIN(8)=X'FFFFFF'** **&XLIN(9)=X'FFFFFF'** If a /*PRIORITY control card is specified, these values are not used. Refer to the &RPRT(*n*) and &XPRI(*n*) parameters for additional information.

Parameter	Value	Explanation
&XPRI(*n*)=	*number*	is a value from 0 to 15 that specifies output priorities, corresponding to processing intervals as defined in the &XLIN(*n*) parameter. If you do not supply a /*PRIORITY control card with your job, the job scheduling priority is recomputed after execution, based on the actual number of print and punch records it produced. If the job produced *p* print lines and *c* punched cards, its output priority will become &XPRI(*n*), where *n* is the smallest number for which $p+c \leq$&XLIN(*n*) If this parameter is not specified, the following are used as defaults: &XPRI(1)=9 &XPRI(2)=8 &XPRI(3)=7 &XPRI(4)=6 &XPRI(5)=5 &XPRI(6)=4 &XPRI(7)=3 &XPRI(8)=2 &XPRI(9)=1 Refer to the &RPRT(*n*) and &XLIN(*n*) for additional information.

The following parameters must be specified the same in all JES2 systems which share the same spool and checkpoint volumes.

&BUFSIZE
&MAXJOBS
&NUMDA
&NUMJOES
&NUMRJE
&NUMTGV
&SPOLMSG
&SPOOL

It is recommended (but not required) that the following parameters be specified the same in all JES2 systems (which share the same spool and checkpoint volumes) to facilitate backup and operational consistency:

&MINJOES
&NUMLNES
&NUMPRTS
&NUMPUNS
&NUMRDRS
&TGWARN
&XBATCH
&XBATCHN

See the chapter "JES2 Initialization" for further recommendations about the JES2 multi-access spool support.

Procedures for Installing JES2

A JES2 generation occurs in two parts. The first part, called JES2GEN, may be performed concurrently with the Stage II part of system generation or after the VS2 system control program has been installed. The second part, called JES2BLD, can only occur after Stage II of system generation has been completed.

The following is a list of procedures you should follow to install JES2.

1. On the console typewriter, enter the command

   $p rd*n*

   where *n* is the identification number of your card reader.
2. Enter the command

   s jes2gen
3. The following messages will be written on the console typewriter:

   $HASP373 JES2GEN STARTED
   GENIN ALLOCATED TO *xx*

   where *xxx* is the unit address of the card reader.
4. Ready the JES2 parameter (and update) card deck and place it in the card reader.
5. The following message will be written on the console typewriter:

   $HASP900 ENTER JES2 GENERATION OPTION CHANGES(option=value),
   CARDS,UPDATE, OR END
6. Reply by typing the following:

   r *nn*,cards

   where *nn* is the corresponding reply number.

   The cards will be read in and processing will begin.

   A response to the WTOR message other than "cards" may be used. Individual JES2 parameters may be entered with a reply text of *option=value*. Lower case letters may be used, but no blanks or comments are allowed. Each JES2 parameter entered from the console is acknowledged by a message if correct or by a diagnostic, allowing you to reenter a correct form. The same parameter may be entered more than once but only the last value entered will be used. The CARDS reply may be entered at any time to enable further parameter reading from the card reader. If all parameters are entered from the console, but updates to JES2 modules are to be entered from the card reader, a reply of UPDATE may be entered to enable reading of the update deck. If all parameters are entered from the console, and if there are no update cards, a reply of END is used to terminate execution of the JESIIGEN utility program.
7. When JES2GEN and Stage II of system generation have completed, the object modules from SYS1.HASPOBJ can be link-edited into SYS1.LINKLIB and SYS1.LPALIB by entering the following command:

   s jes2bld,linkvol=*volser*,lpavol=*volser*

   where *volser* is the volume serial numbers of the volume(s) containing SYS1.LINKLIB and SYS1.LPALIB, respectively. If these values are not specified, the system residence volume is assumed.

Processing

The JES2 generation jobs are executed in a sequential order using a single initiator. The job control language required to execute the programs is a cataloged procedure in the generating system's SYS1.PROCLIB. During JES2 generation, the following occurs:

1. The JESIIGEN utility program is executed. This utility program reads in source modules from the SYS1.AOSH2 distribution library and places them in the SYS1.HASPSRC data set. It then reads the JES2 parameter cards and applies the changes to the source modules in SYS1.HASPSRC.
2. The source code in SYS1.HASPSRC is assembled. The object modules from these assemblies are placed in the SYS1.HASPOBJ data set.
3. If any JES2 remote MULTI-LEAVING terminals are included in the generation, the RMT generation will occur at this point. (Refer to the chapter "Remote Job Entry" for a description of RMT parameters and the processing of the RMT generation.) Even if terminal programs are not generated, a series of allocation-recovery messages will be written to the system console. Any response to these messages is acceptable.
4. The object modules from SYS1.HASPOBJ are link-edited into either SYS1.LINKLIB or SYS1.LPALIB.

Completion Codes

During both the JES2 and RMT generations, the success of the generation process is determined and a completion code is returned as follows:

Decimal Completion Code	Meaning
0	No errors, were detected and all members of the SYS1.HASPSRC data set were successfully constructed.
24	An unrecoverable error, which prohibited the successful construction of the SYS1.HASPSRC data set, was detected. An accompanying message gives further indication of the error. This completion code without any message indicates a JES2 parameter error (for example, the END card was omitted).

Refer to *OS/VS Message Library: VS2 System Codes* and *OS/VS Message Library: VS2 System Messages* for a more detailed discussion of the completion codes that are returned by the system.

Output

The output from the JES2 generation is the JES2 job entry subsystem. In addition, the JESIIGEN utility prints an information listing, the JES2 parameter default values, and the parameter values you specified. Also produced is a listing of each assembly and link-edit.

Modifying JES2

A partial JES2 generation may be done under a production batch system if only minor changes need to be made to the JES2 parameters or only a small number of modules need to be modified (a complete JES2 generation may also be done using this method). Execution of a partial JES2 generation invokes only the JESIIGEN utility. This utility merges member HASPPTF (located in SYS1.AOSH2) into SYS1.HASPSRC before any assemblies are done. You must specify the modules that are to be reassembled and link-edited. Figure 2-5 shows an example of the jobs that need to be executed.

```
//HASPGEN     JOB      . . .
//GEN         EXEC     HASPGEN
//HASPGEN.OPTIONS DD *
   (deck as in Figure 2-4)
/*
//HASMNUC     JOB      . . .
//NUC         EXEC     HASPASM,MODULE=HASPNUC
//HASPINIT    JOB      . . .
//UNIT        EXEC     HASPASM,MODULE=HASPINIT
```

Figure 2-5. Sample Batch JES2 Generation Jobs

A module must be reassembled if a JES2 parameter(s) is changed from the way it was specified in the last JES2 generation and the module depends on that parameter. Figure 2-6 shows the module dependencies. If a parameter that has been changed is used as a default value for another parameter, that parameter is changed also and all modules that depend on it must be reassembled. For example, if &NUMTPPR is allowed to default, it will default to the value specified in the &NUMLNES parameter. If &NUMLNES is changed, then the value of &NUMTPPR is automatically changed. If source modules are updated from a previous generation using update cards, the updated module must be reassembled. When you are not sure whether a reassembly is necessary – for example, if a source module in SYS1.HASPSRC other than one of the twelve assembly modules is updated – then all twelve modules must be reassembled.

The module HASPDOC does not actually depend on any JES2 parameters. However, it contains the most complete documentation of all JES2 control blocks. Therefore, HASPDOC should be reassembled periodically to provide listing documentation current with the operational JES2.

To perform a partial JES2 generation, do the following:

1. Mount the volumes containing the SYS1.AOSH1 and SYS1.AOSH2 distribution libraries and the JES2 data sets.
2. Scratch the SYS1.HASPSRC data set and reallocate it.
3. Place the JES2 parameter (and update) deck in the card reader and execute the JESIIGEN utility.
4. Execute the HASPASM procedure to reassemble the modules into SYS1.HASPOBJ (see Figure 2-5). If all of the modules are to be reassembled, SYS1.HASPOBJ should be scratched and space should be reallocated prior to executing the assemblies.
5. Execute the HASPLNK or HASPLPA procedures (or both) to link-edit the modules into either SYS1.LINKLIB or SYS1.LPALIB.

Input Deck for a JES2 Generation

Figure 2-7 illustrates a JES2 generation. During the JES2 generation process, modifications to the JES2 module HASPMISC are to be made.

Parameter	Module(s)	Parameter	Module(s)	Parameter	Module(s)
&BSCCPU	PM	&NUMACE	N	&PRTUCS	N
&BSC2770	M	&NUMBUF	N	$PUNBOPT	P
&BSC2780	M	&NUMCLAS	PCN	&RCOMCHR	H
&BSC3780	M	&NUMDA	HRXPVSN	&RDROPSL	N
&BSHPRES	M	&NUMINRS	HN	&RDROPST	N
&BSHTAB	M	&NUMJOES	VN	&RDROPSU	N
$BSPACE	S	&NUMLNES	HRXPMWCN	&RJOBOPT	R
&BSVBOPT	M	&NUMPRTS	HRPCN	$RPRBOPT	P
&BUFSIZE	HRXPVSMN	&NUMPUNS	HPN	&RPRI(*n*)	R
&CCOMCHR	N	&NUMRDRS	HN	&RPRT(*n*)	R
$CKPTIME	PV	&NUMRJE	RXVMCN	&RPS	HN
&DEBUG	HVN	&NUMSMFB	HPAVMN	$RPUBOPT	P
$DELAYTM	M	&NUMTGV	HRXPVSN	&SMFRSIZ	HN
&DMNDSET	P	&NUMTPBF	N	&SPOLMSG	PMN
$ESTIME	R	&NUMTPPR	HP	&SPOOL	N
$ESTLNCT	R	&NUMTPPU	HPM	&STDFORM	RPN
$ESTPUN	R	&NUMTPRD	H	$SYNCTOL	VCN
&JCOPYLM	P	&NUMWTOQ	MN	&TGWARN	HS
$LINECT	R	&OUTPOPT	N	&TIMEOPT	HXN
&MAXCLAS	XCN	$OUTXS	N	$TIMEXS	X
&MAXJOBS	HXVN	$PRIDCT	P	&TPBFSIZ	MN
&MAXPART	N	&PRIHIGH	V	$TPIDCT	P
&MINJOES	N	&PRILOW	V	&USASCII	M
&MLBFSIZ	M	&PRIRATE	HV	$WAITIME	M
&MSGID	HXPSMWCN	$PRTBOPT	P	&XBATCH	RX
&NOPRCCW	HP	&PRTFCB	N	&XBATCHN	RX
&NOPUCCW	HP	&PRTRANS	PM	&XLIN(*n*)	RS
				&XPRI(*n*)	RS

Key:		
	H=HASPNUC	S=HASPSSSM
	R=HASPRDR	M=HASPRTAM
	X=HASPXEQ	W=HASPCON
	P=HASPPRPU	C=HASPCOMM
	A=HASPACCT	N=HASPINIT
	V=HASPMISC	

Figure 2-6. Module Dependencies on JES2 Parameters

```
Column
1                                                              73        80

&NUMLNES=1
&BSCCPU=YES
UPDATE
./          CHANGE NAME=HASPMISC
            .
            .  modifications to module HASPMISC                nnnnnnnn
            .
./
/*
```

Figure 2-7. Input Deck for a JES2 Generation

CHAPTER 3. JES2 INITIALIZATION

JES2 initialization is the series of operations JES2 performs each time it is started in order to ready itself for job processing. During each initialization, JES2:

- Loads the JES2 routines and initializes buffer queues
- Locates and initializes all external devices and spool volumes
- Validates a Multi-Access Spool configuration
- Initializes internal readers and logical initiators
- Initializes internal tables and the subsystem interface

The way JES2 initializes depends upon a set of initialization options that are processed when JES2 is started and a set of initialization parameters (defined as a data set in the JES2 procedure) which JES2 reads during its execution.

The initialization *options* define how JES2 will perform initialization by specifying:

- JES2 cold start or warm start
- The data set containing the initialization parameters
- A printout of the initialization data set
- Forced formatting of the spool volumes
- Automatic start of JES2 processing or operator start of JES2 processing

The initialization *parameters* define which of the JES2 functions and devices defined at JES2 generation are to be initialized. The parameters specify:

- Logical initiator characteristics
- Internal reader characteristics
- Local and remote device characteristics
- Default job and SYSOUT class characteristics
- Multi-Access Spool control parameters
- Changes to certain JES2 generation parameters

Your installation can control how JES2 schedules jobs by the way you specify these options and parameters during JES2 initialization. Furthermore, you can respecify these options and parameters to reflect changes in your system's configuration and workload each time JES2 is started.

This chapter explains how to specify the options and parameters and how JES2 performs initialization under different starting conditions.

How to Control JES2 Initialization

JES2 initialization is performed after JES2 is started and before JES2 starts to process jobs. To control how JES2 initializes, your installation can do three things:

- Create a data set containing the initialization parameters
- Update the JES2 procedure to include definitions of the initialization data set and (optionally) other procedure libraries
- Specify the initialization options

Directions for these steps are contained in the following sections.

Creating an Initialization Data Set

An initialization data set contains the initialization parameters and, optionally, JES2 control statements and operator commands. All of the parameters, control statements, and commands are coded on punched cards and entered as a data set into a system library by one of the IBM utility programs, such as IEBUPDTE.

The initialization parameters allow you to specify the functions and device characteristics JES2 will use during its current execution. The parameters and their functions are summarized in Figure 3-1 and they are fully described at the end of this chapter. HASP users will recognize many of these parameters as former HASP generation parameters. The purpose of moving these parameters from the generation process to the initialization process was to give the installation more flexibility in controlling the system.

JES2 Initialization Parameters

Parameter	Function
&CCOMCHR	changes the JES2 identifier character that precedes each command entered at the operator console from the character that was specified at JES2 generation.
&CHKPT	specifies the volume serial ID of the volume containing the SYS1.HASPCKPT data set.
Innn	specifies the characteristics of a logical initiator.
INTRDR	specifies the characteristics of the internal readers.
LINEnnn	specifies the characteristics of a teleprocessing line.
&NUMBUF	increases the number of I/O buffers from the number specified at JES2 generation.
&NUMWTOQ	increases the number of console message buffers from the number specified at JES2 generation.
PRINTERnn	specifies the characteristics of a local printer.
PUNCHnn	specifies the characteristics of a local card punch.
QCONTROL	specifies job queue control parameters for a Multi-Access Spool configuration.
&RCOMCHR	changes the JES2 identifier character that precedes commands entered from local and remote card readers from the character that was specified at JES2 generation.
READERnn	specifies the characteristics of a local card reader.
RMTnn	specifies the characteristics of a remote terminal.
Rnnn.RDm	specifies the characteristics of a remote card reader.
Rnnn.PRm	specifies the characteristics of a remote printer.
Rnnn.PUm	specifies the characteristics of a remote card punch.
Sn	identifies each individual system in a Multi-Access Spool configuration.
&SPOOL	changes the volume serial ID of the primary spool volume from the ID that was specified at JES2 generation.
STCMCLAS	defines the message class for all started tasks.
&STC,&TSU,&x	specifies the characteristics of a job class.
TSUMCLAS	specifies the message class for all time-sharing foreground jobs.
$$x	specifies the characteristics of a SYSOUT class.

Figure 3-1. The JES2 Initialization Parameters and Their Functions

With JES2, the generation parameters specify the support for and the limitations of the JES2 functions while the initialization parameters specify the characteristics and use of each function and device. Consequently, many of the initialization parameters depend upon specifications of related JES2 generation parameters. For example, the PRINTERnn initialization parameter defines the characteristics of each printer as it will operate in the current configuration. However, PRINTERnn can be specified for only the number of printers that were supported at JES2 generation by the &NUMPRTS parameter. Because of this and similar dependencies between the two sets of parameters, you may find it practical to code both sets in one session. (At the end of this chapter, there is a table that lists and correlates both the generation and initialization parameters.)

The initialization parameters are not required for local single system configurations. However, if you have specified RMT generations for remote terminals, you will have to code the RMTnnn initialization parameters to initialize these terminals. If you choose not to specify the parameters for local devices and JES2 functions, JES2 provides default values (using values specified by system generation and JES2 generation parameters). For instance, for local devices, JES2 checks all the unit control blocks (UCBs) built during system generation and, when initialization is complete, starts all physically-connected devices that are ready. By specifying initialization parameters for all local devices, you can choose, for example, to drain the devices you will not want to use right away.

If you are operating a Multi-Access Spool configuration, you must define an initialization data set for each system in the configuration. Each data set must include the system identifers (specified by the Sn initialization parameter) of all systems in the configuration. In addition, the initialization data set for each system should be set up so that each unit-record device name is assigned to only one physical device in the Multi-Access Spool configuration. For example, if three readers are generated for a configuration of three systems, they should be initialized as READER1, READER2, and READER3 among the data sets and not as READER1 in each system's data set. Then the readers can easily be reassigned in subsequent system initializations. (A device that is not to be attached to a particular system can be forced into an undefined status by assigning it a nonexistent unit address, such as UNIT=FFF.)

In addition to the initialization parameters, an initialization data set can also contain:

- Patch and SUPERZAP statements
- JES2 operator commands
- JES2 initialization control statements

The operator commands, Patch, SUPERZAP, and initialization control statements can be mixed among the initialization parameters without any special coding requirements.

Patch and SUPERZAP statements can be used to make minor and temporary modifications to the JES2 source code for the duration of an IPL by directly replacing the changed code. Directions for using these statements are provided in the chapter "Miscellaneous JES2 Facilities." JES2 processes the Patch and SUPERZAP statements as they are read within the initialization data set.

JES2 operator commands can be used to control the initial status of devices. For instance, operator commands can be used to start RJE lines during initialization. (RJE lines, unlike other devices, cannot be started automatically by an initialization parameter.) Or, the $VS operator command can be used to enter VS commands such as those to VARY devices on and off line before JES2 starts processing. JES2 operator commands are described in the *Operator's Library: OS/VS2 Reference (JES2).*

The number of operator commands you can specify in an initialization data set is limited to the number of message buffers you specified in the JES2 &NUMWTOQ generation parameter. During initialization, JES2 stores the operator commands in these message buffers. Then, when initialization is complete, JES2 processes the commands. To ensure that operator commands are completely processed before JES2 starts processing jobs, you should use the REQ initialization option which lets the operator start JES2 processing. Or, the $S command can be included as the last operator command in the initialization data set to eliminate the need for operator intervention.

JES2 initialization control statements can be used to format the listing of the data set when it is printed during initialization. There are three of these control statements:

NOLIST–which tells JES2 to stop or discontinue listing of the data set from this point on

LIST–which tells JES2 to resume listing of the data set from this point on

*–which tells JES2 this is a comment statement

LIST and NOLIST provide a convenient way to protect portions of your data set (such as passwords) during printout of the data set. Comment (*) statements can be used to provide spacing and headings within the data set. All three types of statements are processed at the point where they occur in the data set. They are ignored if you specify the NOLIST initialization option.

The operator commands, Patch, SUPERZAP, and initialization control statements can be mixed among the initialization parameters without any special coding requirements. Figure 3-2 shows an example of an initialization data set that contains operator commands and initialization control statements. (The initialization parameters are described later in this chapter.)

After you have coded your data set, it should be transferred to a direct-access volume by using one of the IBM utility programs, such as IEBUPDTE. The data set should be entered as a member of a blocked system library such as SYS1.PROCLIB or as a member of a blocked user library. This member must then be defined in the JES2 procedure so that when JES2 executes, it can locate and read the initialization data set. Directions for updating the JES2 procedure are contained in the following section.

Updating the JES2 Procedure

The basic JES2 procedure (Figure 3-3) provided with the VS system contains an EXEC statement and three data definition (DD) statements named PROC00, HASPLIST, and HASPPARM. PROC00 defines a default procedure library to be used for converting the JCL of user jobs, time-sharing logons, and system tasks. HASPLIST defines what is normally a dummy output data set. HASPPARM defines a member in SYS1.PARMLIB that contains a null initialization data set. (See Figure 3-4)

The JES2 procedure can be updated by entering update cards with the IBM IEBUPDTE utility program. You can add DD statements to the JES2 procedure to define:

- Other cataloged procedure libraries that are associated with job classes by the &STC, &TSU, and &x initialization parameters or by the JOBPARM control statement. Each library should be defined by a PROCnn DD statement. For instance, to specify a special library for TSO logons, code PROCLIB=*nn* (where nn corresponds to the associated PROCnn DD statement) in the &TSU initialization parameter.

```
  *
  *                SAMPLE JES2 PARAMETER LIBRARY LISTING
  *
  &CHKPT=IPLVOL
  READER1     UNIT=00C
  READER2     UNIT=00B,PRIOLIM=9,CLASS=X,AUTH=7,PRDEST=1003
  PRINTER1    UNIT=002,CLASS=AJH,UCS=P11
  PRINTER2    UNIT=00E,CLASS=AJH,UCS=PN,AUTO
  PRINTER3    UNIT=00F,CLASS=A,ROUTECDE=1003,UCS=HN
  PUNCH1      UNIT=00D,PAUSE
1 INTRDR      PRIOLIM=9,AUTH=7
  I1          CLASS=AFJKE                INITIATOR 1
  I2          CLASS=BCDEF                INITIATOR 2
  I3          CLASS=DEFGH                INITIATOR 3
  I4          CLASS=XKH                  INITIATOR 4
  I5          CLASS=JKEBF                INITIATOR 5
  I6          DRAIN                      SPARE INITIATOR
  I7          DRAIN                      SPARE INITIATOR
  I8          DRAIN                      SPARE INITIATOR
  &STC        NOJOURN,NOLOG,NOOUTPUT     STARTED TASK DEFINITIONS
  &TSU        CONVPARM=00014400005030E00011
  &S          PROCLIB=03,HOLD            SYSTEM PROGRAMMER CLASS
  &X          PERFORM=3,XBATCH           EXECUTION BATCHING CLASS
  $$H         HOLD                       SYSOUT CLASS HELD FOR OUTPUT PROCESS
  $$X         DUMMY                      THROWAWAY CLASS
  *
  *                RJE INITIALIZATION PARAMETERS
  *
  LINE1       UNIT=040,FDUPLEX,TRANSP
  LINE2       UNIT=041,TRANSP,PASSWORD=SECRET
  LINE3       UNIT=042,TRANSP,PASSWORD=SECRET
  LINE4       UNIT=043,TRANSP,PASSWORD=SECRET
  LINE5       UNIT=044,TRANSP,PASSWORD=SECRET
  RMT1        3780,LINE=1,NUMPU=1,TRANSP,ABUFEX,COMP
2 R1.PR1      PRWIDTH=144
  RMT2        2922,NUMPU=1,CONSOLE,MULTI,TRANSP
  R2.PR1      PRWIDTH=132,AUTO
  RMT3        S370,NUMPR=2,CONSOLE,MULTI,TRANSP
  R3.PR1      PRWIDTH=150,FCBLOAD
  R3.PR2      PRWIDTH=132
  RMT4        1130,CONSOLE,MULTI,NUMPU=1
  R4.PR1      AUTO
  R4.PU1      DRAIN
  RMT5        SYSTEM3,NUMRD=3,NUMPU=2,CONSOLE,MULTI
  R5.PR1      PRWIDTH=132,AUTO
  RMT6        2780,NUMPU=1,TRANSP,MRF,TABS
  R6.PR1      PRWIDTH=144
3 NOLIST
  *
  *                REMOTE PASSWORDS
  *
```

1 All JES2 internal readers are defined by one INTRDR parameter.

2 Parameters that specify remote devices do not have to follow their associated RMTnnn parameters; they may be put anywhere in the card deck.

3 Shaded area will not appear on a printout.

Figure 3-2.(Part 1 of 2). Example of a JES2 Initialization Data Set

```
4 RMT1                 PASSWORD=CHICAGO
  RMT2                 PASSWORD=BOARDWLK
  RMT3                 PASSWORD=ALBANY
  RMT4                 PASSWORD=LACKLSTR
  RMT5                 PASSWORD=KALAMAZO
  RMT6                 PASSWORD=UNIQUE
  LIST
  *
  *                                JES2 GENERATION PARAMETER OVERRIDES
  *
  &NUMBUF=40
  &NUMWTOQ=50
  TSUMCLAS=H
  *
  *                                MULTI-ACCESS SPOOL CONFIGURATION PARAMETERS
  *
5 S1                   SID=L158
  S2                   SID=K168
6 QCONTROL             HOLD=150
  *
  *                                OPERATOR COMMANDS
  *
  $S LNE1
  $S LNE2
  $S LNE3
  $S LNE4
  $S LNE5
7 $VS,'V(234,235,236,237),OFFLINE'
  *
  *                                END OF JES2 PARAMETER LIBRARY LISTING
  *
```

4 Parameters can be coded more than once to incorporate additional subparameters. (When the same subparameter is repeated for a parameter, the value specified last is the one that is used.)

5 Assuming this initialization data set is for a system whose SMF identifier is L158. Both of these parameters must also be included in the initialization data set for the system whose identifier is K168.

6 HOLD is set higher than its default value to accommodate the slower processing time of the Model 158. (In the initialization data set for the Model 168, QCONTROL would probably not be specified and the default values would be used.)

7 An operator command is used to ensure these devices are varied offline regardless of their initial status.

Figure 3-2 (Part 2 of 2). Example of a JES2 Initialization Data Set

```
//JES2        PROC MEMBER=JES2PARM
//IEFPROC     EXEC PGM=HASJES20,DPRTY=(15,15),TIME=1440
//PROC00      DD DSN=SYS1.PROCLIB,DISP=SHR
//HASPPARM    DD DSN=SYS1.PARMLIB(&MEMBER),DISP=SHR
//HASPLIST    DD DDNAME=IEFRDER
```

Figure 3-3. The Basic JES2 Procedure

```
*
*
*
*                     JES2 INITIALIZATION DATA SET
*
*
*         JES2 INITIALIZATION PARAMETERS,CONTROL STATEMENTS
*         AND OPERATOR COMMANDS SHOULD BE MERGED WITH THIS
*         MEMBER.
*
*         SYSTEM PERFORMANCE MAY BE IMPROVED BY MOVING THIS
*         MEMBER TO A BLOCKED SYSTEM LIBRARY SUCH AS SYS1.PROCLIB
*         OR BLOCKED USER LIBRARY AND CHANGING THE HASPPARM
*         DD STATEMENT IN THE JES2 PROCEDURE TO REFLECT
*         THIS CHANGE.
*
*
*
```

Figure 3-4. The HASPPARM Initialization Data Set

```
//JES2        PROC CONFIG=HASPPARM
//IEFPROC     EXEC      PGM=HASJES20,PARM='NOREQ,HASPPARM=&CONFIG',
//                      TIME=1440,DPRTY=(15,15)
//PROC00      DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC01      DD DSN=SYS1.USERLIB,DISP=SHR
//PROC02      DD DSN=SYS1.USERLIB,DISP=SHR
//            DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC03      DD DSN=SYS1.SYSPLIB,DISP=SHR
//            DD DSN=SYS1.PROCLIB,DISP=SHR
//PROCLIB     DD DSN=SYS1.PROCLIB,DISP=SHR
//            DD DSN=SYS1.USERLIB,DISP=SHR
//HASPPARM    DD DSN=SYS1.PARMLIB(JES2PARM),DISP=SHR
//BATCH       DD DSN=SYS1.PROCLIB(BATCHP),DISP=SHR
//RJE         DD DSN=SYS1.PROCLIB(RJEPARMS),DISP=SHR
//TSU         DD DSN=SYS1.PROCLIB(TSUPARMS),DISP=SHR
//HASPLIST    DD DDNAME=IEFRDER
```

Figure 3-5. Example of an Updated JES2 Procedure

- One or more initialization data sets that define different operating conditions and types of workloads. The example of an updated JES2 procedure in Figure 3-5 defines three user data sets:

  RJE–for evening shifts with a lot of remote activity
  TSU–for day shifts with many TSO users
  BATCH–for midnight shifts with mostly background jobs

  If you wish, you can use the HASPPARM DD statement to define an initialization data set by replacing the IBM-supplied null data set. Alternately, the initialization data set may be created separately and defined by its own DD statement in the procedure.

To make the HASPLIST DD statement effective, IEFRDER must be defined with the address of an output device in your system. This permits the HASPLIST data set to be written to that output device for a listing of the initialization data set.

DD statements added to the JES2 procedure must refer to data sets that are cataloged in the master catalog or that are specified by volume serial number. You cannot add DD * or DD DATA statements to the JES2 procedure. If SYSOUT DD statements are added, they will be ignored.

Specifying the Initialization Options

When JES2 is started, it uses five initialization options to determine how it will perform the current initialization. The initialization options, which are described in Figure 3-6, may be specified in either of two ways: as parameters on the EXEC statement in the JES2 procedure or as options specified at the console.

If the options are not specified on the EXEC statement, JES2 requests them from the operator by issuing the following WTOR:

```
$HASP426 SPECIFY OPTIONS–HASP-II, VERSION JES2.0
```

The operator then enters the options using the standard reply format as described in *Operator's Library: OS/VS2 Reference (JES2).* Options may be entered in upper or lower case and must be separated by commas. If conflicting options (for example, WARM, COLD) are entered, the latter option overrides the former one.

If the options are specified on the EXEC statement, JES2 suppresses the SPECIFY OPTIONS WTOR and completes initialization without operator intervention. The EXEC statement in the JES2 procedure in Figure 3-5 shows how the options can be coded as parameters.

After it has accepted the options, JES2 reads the specified initialization data set. When initialization is complete, JES2 is ready to start processing jobs. JES2 will start processing automatically if you specified the NOREQ option. Otherwise, it will issue the following message to request the operator to issue the $S command to start JES2 processing:

```
$HASP400 ENTER REQUESTS
```

The operator can also respond to this message with other commands to change the initial status of initialized devices before JES2 starts to process.

How JES2 Performs Initialization

JES2 performs an initialization the first time it is started (as part of the IPL procedure) and every time it is restarted after a normal system shutdown or after a system failure. In a Multi-Access Spool configuration, JES2 must be started and initialized in each system in the configuration. The way JES2 performs initialization during these start situations is described below.

Initialization Options

Option	Explanation
FORMAT NOFMT	FORMAT specifies that JES2 is to format all existing spool volumes. If unformatted spool volumes are added, JES2 automatically formats them whether FORMAT is specified or not. When FORMAT is specified, JES2 will automatically be cold started. *Note:* The FORMAT option is denied if this is a Multi-Access Spool configuration and JES2 is processing in one or more of the other systems. *Default:* NOFMT specifies that JES2 is not to format existing spool volumes unless JES2 determines that formatting is required.
COLD WARM	COLD specifies that JES2 is to be cold started. All jobs in the system will be purged and all job data on the spool volumes will be scratched. *Note:* The COLD option is denied if this is a Multi-Access Spool configuration and JES2 is processing in one or more of the other systems. *Default:* WARM specifies that JES2 is to be warm started. JES2 will continue processing jobs from where they were stopped. *Note:* If the system to be warm started is in a Multi-Access Spool configuration with any other active systems, only this system is warm started. If there are no other active systems, JES2 requests the operator to verify that no other systems are active and, when verified, proceeds to warm start all jobs.
NOREQ REQ	NOREQ specifies that the $HASP400 ENTER REQUESTS message is to be suppressed and that JES2 is to automatically start processing when initialization is complete. *Default:* REQ specifies that the $HASP400 ENTER REQUESTS message is to be written at the console. This message allows the operator to start JES2 processing with the $S command.
NOLIST LIST	NOLIST specifies that JES2 is not to print the contents of the initialization data set or any error flags that occur during initialization. If NOLIST is specified, any LIST control statements in the initialization data set will be ignored. *Default:* LIST specifies that JES2 is to print all the statements in the initialization data set and any error flags that occur during initialization. (JES2 prints these statements if a printer is defined for that purpose when JES2 is started.) LIST will not print any statements that follow a NOLIST control statement in the initialization data set.

HASPPARM=ddname
HASPPARM=HASPPARM
	ddname specifies the name of the data definition (DD) statement that defines the data set containing the initialization parameters that JES2 is to use for this initialization.
	HASPPARM specifies that JES2 is to initialize using the initialization parameters in the data set defined by the HASPPARM DD statement in the JES2 procedure.

NONE U N null	NONE, U, N, or a null specifies that JES2 is to use all the default initialization options.

Figure 3-6. The JES2 Initialization Options

Starting JES2 for the First Time

JES2 is automatically started* via the JES2 procedure (refer to Figure 3-3) in SYS1.PROCLIB. As soon as JES2 is started, it issues the SPECIFY OPTIONS WTOR. The operator should specify the COLD (or FORMAT) option. If COLD (or FORMAT) is not specified, JES2 will issue the following messages:

```
$HASP434 WARM START DENIED–INVALID CHECKPOINT RECORD
$HASP420 PERM I/O ERROR READING JES2 CKPT
$HASP428 CORRECT THE ABOVE PROBLEMS AND RESTART JES2
```

You will have to restart JES2 and then specify COLD (or FORMAT) in response to the SPECIFY OPTIONS WTOR.

JES2 initializes its functions and devices according to the default values of the initialization parameters (since HASPPARM points to a null initialization data set). When initialization is complete, JES2 can start processing jobs. Your first job can be the utility programs that create your initialization data set and update the JES2 procedure. Before you can use your data set, you must stop and restart JES2 with the name of the data set specified as an initialization option. The way to do this is described in the section, "Restarting JES2 After an Orderly Shutdown."

Note: *If the primary JES2 spool volume (specified by the &SPOOL initialization parameter) is not mounted and ready when JES2 is started, JES2 will inform the operator that the spool volume must be mounted and will then terminate. This mount cannot be done since MOUNT command processing requires the as-yet-uninitialized JES2. The only thing the operator can do is to ready the spool volume and IPL again. A way to avoid this situation is to include in the VATLSTnn parmlib member an entry specifying the required spool volume without suppressing the mount option. VATLST processing will then request that the volume be mounted during the IPL process itself. An example would be:*

```
SPOOL1,0,2,330
```

Starting JES2 in a Multi-Access Spool Configuration

Whenever JES2 is started, it checks to see if JES2 has already been started in another system that was generated with the current system as a Multi-Access Spool configuration. JES2 determines this by reading the JES2 checkpoint record (SYS1.HASPCKPT) and checking the last time stamp recorded by each system. If a time stamp for any system is less than the time-of-day (TOD) clock plus the time interval specified by the JES2 generation parameter &SYNCTOL, JES2 is assumed to be processing in the associated system. In such a case, JES2 will reject the COLD (or FORMAT) option and will request the current system to perform a WARM start.

If all time stamps are older than the TOD clock plus the &SYNCTOL interval, JES2 is assumed not to be operating in any other system. Therefore, the current system can either be cold started or warm started following an orderly shutdown or a system failure. A WARM start will reallocate the spool volumes that were in use, in order to recover any direct-access space that might have been lost during a system failure.

*The START JES2 command is automatically placed in MSTRJCL during system generation. If an installation wants the option of manually starting JES2, the MSTRJCL should be changed as follows:

```
Name   MSTRJCL                              (SYS1.LINKLIB)
VER    0320  616140E2,E3C1D9E3,40D1C5E2F2
REP    0320  61614040,40404040,4040404040
```

It is necessary to write blanks (X'40's) over the //START JES2 command.

Restarting JES2 After an Orderly Shutdown

JES2 can be stopped and restarted in a system at any time by operator commands. This capability allows you to:

- Quiesce job processing in preparation for an orderly system shutdown
- Restart JES2 to perform an initialization with a different initialization data set

For both situations, the operator first issues the $p command to drain the JES2 queues. When all JES2 logical initiators, printers, and punches complete their current activities and become inactive, JES2 notifies the operator with the following message:

```
$HASP099 ALL AVAILABLE FUNCTIONS COMPLETE
```

The operator can then enter the $p JES2 command which stops JES2 and removes it from the system. When you are halting the system at end-of-day or end-of-work shift, the operator should also enter the HALT EOD command. When the system is re-IPLed, JES2 will be automatically started. (You can, of course, specify a different initialization data set as an initialization option at that time by responding with HASPPARM=ddname when JES2 issues its SPECIFY OPTIONS message.) When JES2 completes initialization, it will either begin to process jobs automatically or wait for the operator to enter the S command in response to the ENTER REQUESTS message.

If you do not halt the system after you stop JES2, you can restart JES2 with the S JES2 command. As part of this start command, you can specify your initialization options as parameters. For example,

```
S JES2,PARM='WARM,HASPPARM=RJE,NOREQ'
```

When you specify the options on the START command this way, the SPECIFY OPTIONS WTOR is suppressed just as it is when you specify the options as parameters on the EXEC statement within the JES2 procedure.

You can also use the START command to specify a printer address for the HASPLIST DD statement in the JES2 procedure. For example,

```
S JES2,00E
```

If you specify both a printer address and initialization options, the printer address must be specified first:

```
S JES2,00E,PARM='WARM'
```

You may also use the START command to start JES2 manually at IPL. (Use a SUPERZAP statement to remove the start command for JES2 that is contained in the MSTRJCL member of SYS1.LINKLIB.)

Restarting JES2 After a System Failure

JES2 is automatically restarted in a VS2 system whenever that system is re-IPLed following a system failure. The WARM initialization option allows you to warm start JES2 and continue job processing from the point of the last checkpoint before the system failure.

During a warm-start initialization, JES2 reads through its job queues and handles each job according to its status:

1. Jobs in input readers are lost and must be reentered.
2. Jobs in output (print/punch) are restarted at the last JES2 checkpoint.
3. Jobs waiting for JES2 functions remain on the JES2 queues.
4. Jobs in execution are scheduled for warm start processing.

Before JES2 schedules any job for warm start processing, it ensures that the job has a valid job journal. Any job that doesn't have one is terminated with an appropriate diagnostic message. JES2 routes each job with a valid journal to the initiator for warm start processing.

The initiator/terminator purges existing job tables and messages for the job and then checks that each job is authorized for warm start processing:

1. The job must have a valid restart definition (RD) code in the RD parameter on its JOB card.
2. The operator must authorize the job to be restarted. (Each eligible job is presented to the operator asking whether the job is to be restarted.)

If either of these authorizations is not made, the job is not eligible to be restarted and it is returned to JES2 for termination. Eligible jobs are returned to JES2 which processes them according to the operator's response to the restart query:

YES–JES2 puts the job back on the execution queue for immediate selection for execution.
NO–JES2 puts the job on the output queue.
HOLD–JES2 puts the job back on the execution queue in hold status.

Jobs can be presented to the system for warm start at any time in any order. This allows important jobs and system functions to be executed ahead of less important jobs scheduled for execution or termination. New jobs can be presented to the system and processed according to their priority–ahead of lower-priority warm start jobs.

When JES2 is warm started, all spool volumes (and the checkpoint data set, if different) that were up during the last execution of JES2 must be present and available, although it is not necessary that they be mounted on the same drives. If all the spool volumes are not up, JES2 notifies the operator. If you add a new spool volume, JES2 will add it to the list of spool volumes; if it is unformatted, JES2 will automatically format it.

During warm start initialization, JES2 will list the activity in process for each job at the time JES2 was stopped. Then, when the ENTER REQUESTS message is issued (unless the NOREQ initialization option was specified), the operator can enter requests to modify or delete each activity. This message also allows the operator to examine the status of output devices (such as UCS and FCB settings) to determine what action to take prior to restarting the output process, to change the status of logical initiators and devices, and to modify the status of jobs on the JES2 queues. (Note that the status and activity of each device will revert to the specifications of the initialization parameters. If you specify NOREQ, JES2 begins processing each job according to the specifications of the initialization parameters as soon as resources to process each job become available.)

How to Correct Initialization Errors

During JES2 initialization, two kinds of errors can occur. The first can occur when JES2 cannot open the data set containing the initialization parameters. This will happen, for instance, when a DD statement is named in the HASPPARM=ddname initialization option, but the named DD statement is not defined in the JES2 procedure. JES2 acknowledges the error with the following message:

```
$HASP450 OPEN FAILED FOR JES2 PARAMETER LIBRARY
```

This message requires no action since JES2 can initialize without specified parameters. However, it allows the operator to stop JES2 and restart it with a correctly defined data set.

The second error can occur when JES2 encounters a user error in the statements in the initialization data set. JES2 flags each statement in error and reads the next one in the data set. At the end of initialization, JES2 issues the following message:

$HASP451 ERROR ON JES2 PARAMETER LIBRARY

If the data set is printed out, each statement that was in error will have an error flag printed beside it. These error flags and their explanations follow.

COMMAND LIMIT EXCEEDED

This operator command exceeds the maximum number of commands allowed in the JES2 parameter library as specified by the &NUMWTOQ generation parameter.

CONTINUATION CARD EXPECTED

The statement ahead of this one was not a complete one and JES2 is expecting a continuation card for it. For example, a previous statement of "I8 DRAIN," followed by an end of file would cause JES2 to expect more subparameters for this statement.

DATA OR FORMAT ERROR

You specified this parameter incorrectly and JES2 cannot classify the error. For example, specifying B instead of &B as a job class parameter.

ILLEGAL DECIMAL VALUE

A decimal value is required for this parameter. The value you specified contains one or more non-decimal symbols. For example, NUMBUF=1A2 instead of NUMBUF=12.

ILLEGAL keyword VALUE

The value you specified for this subparameter keyword does not meet the required value range or specifications. For example, ROUTECDE=100B instead of ROUTECDE=1002.

INVALID CHARACTER VALUE

The value specified for this parameter contains one or more invalid characters. For example, STCMCLAS=% instead of STCMCLAS=H.

INVALID DEVICE NAME

The number you specified for this device exceeds the number of devices supported at JES2 generation. For instance, specifying PRINTER6 when &NUMPRTS=5 was specified at JES2 generation. Or, specifying R5.RD1 when &NUMRJE=4.

INVALID HASPPARM STATEMENT

You specified this parameter incorrectly and JES2 cannot classify the error. For example, READR1 instead of READER1.

INVALID INITIATOR NUMBER

The number you specified for this logical initiator either exceeds the number allowed in the &MAXPART generation or it contains an invalid character.

INVALID KEYWORD–keyword

This subparameter keyword is misspelled or is not valid for this parameter.

INVALID PARAMETER VALUE

The value you specified for a subparameter exceeds the range limit. For example, PRIOINC=16 when the maximum allowed is 15.

Note: *When a parameter statement contains an error, JES2 honors all values specified up to the error and ignores the remainder of the statement. The one exception to this is when a parameter statement contains a subparameter that exceeds its range limit. When this error occurs, JES2 ignores the entire statement and uses instead the default (or any previously assigned) values for that parameter. The error flag (INVALID PARAMETER VALUE) is then written and processing continues with the next statement.*

Alternate Subsystem Options with JES2

MVS allows more than one subsystem to operate at a time as long as one subsystem is designated as the primary subsystem and others are identified as secondary subsystems. Secondary JES2s can be useful in testing user modifications while the primary JES2 is being used for production. The SCHEDULR sysgen macro must name all subsystems that are allowed.

It is therefore possible to run more than one JES2 at a time with certain restrictions applying to the secondary JES2. The secondary JES2 cannot interface with TSO or started tasks. When running more than one JES2 at a time, it is necessary to assign (by the &CCOMCHR JES2 initialization parameter) a unique operator command character to each JES2 and to have a unique SPOOL direct access device for each JES2. Also, the JES2 parameter must be used to stop whatever subsystem is running regardless of the name on the START command; for example:

```
S JES2, $P JES2; S JSS, *P JES2.
```

If an alternate Subsystem Support Module (HASPSSSM) is to be used, it must be linked into SYS1.LPALIB under an alternate name (for example, TESTSSSM). The initialization deck for the subsystem using this module must contain a statement that names the alternate HASPSSSM. This statement takes the form HASPSSSM=altname (for example, HASPSSSM=TESTSSSM).

In place of the primary job entry subsystem, an alternate subsystem can be executed as the primary subsystem. This alternate subsystem must be in SYS1.PROCLIB and named on the START command.

If an installation specified no secondary subsystem during system generation, or wants to change a previously-specified secondary subsystem, the JES Subsystem Names Table must be changed. For example, assuming JES2 is the primary subsystem and xxxxxxxx is the desired alternate:

```
Name        IEEVIPL          IEFJESNM
VER    0000        D1C5E2F2,D4E2E4D9
REP    0000        D1C5E2F2,D4E2E3D9,xxxxxxxx
```

The JES2 generation/initialization parameter &CCOMCHR affects the identification assigned to JES2-initiated messages to the operator. Normally JES2-initiated messages are tagged with a "$" at the beginnning of the text. However, the "$" character is taken from the value of &CCOMCHR; therefore a different specification of this value for each subsystem would allow the origin of the message to be uniquely identified.

Initialization Parameter Descriptions

This section describes the JES2 initialization parameters, their functions, formats, and default values. The parameters are described in alphabetical order (excluding the first characters if & or $). The following conventions are used in the parameter descriptions:

- Numbers and upper-case letters must be coded exactly as shown.

- Lowercase letters represent variables for which you must substitute specific information or specific values.
- Paired subparameters (for example, HOLD/NOHOLD) indicate that you may choose one or the other. If you specify neither, the underlined one will be used as the default value.

The following syntax rules apply to the coding of most of the parameters. (Exceptions are contained within the parameter descriptions.) Refer to Figure 3-2 for examples of coded parameters.

- A parameter is separated from its subparameters by at least one blank; subparameters are separated from each other by commas.
- Any columns between 1 and 71 can contain data; column 72 is used for continuation; columns 73-80 are ignored.
- Parameter statements can be continued on successive cards; continuation is indicated by a comma followed by a blank. (If the last subparameter on a card is not followed by a comma and column 72 is not blank, then the next card is considered to contain only comments.)
- Parameters cannot contain embedded blanks. The first blank terminates the parameter statement and the rest of the card is considered to contain comments.
- Only one parameter can be coded per card although several subparameters can be coded on the same card.
- Leading zeros cannot be used in parameter values (except as noted for specific parameters). This is especially true for device names: READER01 is an error; READER1 is correct.
- A parameter deck is terminated by an end of file or any card that contains a /* in columns 1 and 2.

Parameter cards can be put in the card deck in any order. Subparameters may also be specified in any order. If the same subparameter occurs more than once for a parameter, JES2 will use the value from the last one it reads.

&CCOMCHR (Local Operator Command Identifier)

&CCOMCHR=c

The &CCOMCHR parameter specifies the character that precedes and identifies all JES2 operator commands entered from a local console. Use it only to change the character specified by the &CCOMCHR generation parameter.

c

can be any special character except a comma, an apostrophe, or the character specified in the `$BSPACE generation parameter. This character must not be used as the first character of commands of any other subsystem that operates concurrently with JES2. If it is, JES2 will assume the command is a JES2 command and attempt to process it.

Note: *The JES2 generation/initialization parameter &CCOMCHR affects the identification assigned to JES2-initiated messages to the operator. Normally JES2-initiated messages are tagged with a "$"at the beginning of the text. However, the "$" character is taken from the value of &CCOMCHR; therefore, a different specification of this value for each subsystem would allow the origin of the message to be uniquely identified.*

Default: The character specified by the &CCOMCHR generation parameter.

&CHKPT (JES2 Checkpoint Volume ID)

&CHKPT=cccccc

The &CHKPT parameter specifies the volume serial number of the volume that contains the JES2 checkpoint data set, SYS1.HASPCKPT. (Space for this data set is allocated at JES2 generation as described in the chapter "Installing JES2.")

Note: *The checkpoint data set is frequently referenced especially in Multi-Access Spool configurations. Therefore, only low-usage data sets (if any) should be allocated on the same volume as the checkpoint data set. Otherwise, JES2 performance could be seriously degraded.*

cccccc

specifies the volume serial number of the volume containing JES2 checkpoint data set. From one to six characters that define a valid volume serial number can be used.

Note: *When this parameter is changed to other than SPOOL1 (the generation default for &SPOOL), certain messages will vary from their documentation in* OS/VS Message Library: VS2 System Messages.

Default: The volume serial number specified in the &SPOOL generation or initialization parameter.

Innn (Logical Initiator)

Innn CLASS=$c_1...c_n$
DRAIN/START
NAME=cc

The Innn parameter specifies the characteristics of one logical initiator. Initiators are numbered consecutively (I1-I999) for the number of initiators specified by the &MAXPART generation parameter. Initiator characteristics are specified by the following subparameters.

CLASS=$c_1...c_n$

specifies the job classes (in order of their priority) from which this initiator will schedule jobs. You can specify any number of job classes (A-Z,0-9) up to the number specified by the &MAXCLAS generation parameter.

Default: If not specified, JES2 assigns job classes in the following manner:

For Logical Initiator	Default Classes Are
I1	A
I2	BA
I3	CBA
.	.
.	.
.	.
I26	ZYX...CBA
I27	0ZYX...CBA
I28	10ZYX...CBA
.	.
.	.
.	.
I36	987...CBA
I37	987...CBA
.	.
.	.
.	.
I99	987...CBA

Note: *When &MAXCLAS specifies a number less than 36, JES2 determines the default job classes as shown above, but recognizes only the number of them allowed by the &MAXCLAS value. For instance, when I8 is assigned default classes HGFEDCBA, but &MAXCLASS=5, JES2 recognizes only the first five (HGFED) job classes as default classes for I8.*

DRAIN/START

DRAIN specifies that this initiator will be started by operator command.

Default: START specifies that this initiator is to be started automatically when JES2 starts processing.

NAME=cc

specifies a unique name that the operator can use to refer to this initiator. *cc* may be 1 or 2 characters (A-Z,0-9).

Default: *nnn* of the Innn specification. (Beyond I99, you must specify a name.)

INTRDR (Internal Readers)

```
INTRDR   AUTH=n
         CLASS=c
         HOLD/NOHOLD
         PRIOINC=nn
         PRIOLIM=nn
```

The INTRDR parameter specifies the characteristics of *all* JES2 internal readers defined by the &NUMINRS generation parameter. An internal reader is a special SYSOUT data set that other programs can use to submit jobs, control statements, and commands to JES2. External readers (for example, RDR) and time-sharing users use the internal readers to submit jobs from direct access devices or tapes. Internal readers are treated like physical input devices. Internal reader characteristics are specified by the following subparameters.

AUTH=n

specifies the command authority number for internal readers. This number authorizes certain JES2 commands to be submitted through an internal reader. *n* is a number from 0-7 defining the kind of commands that can be entered:

7–display only
6–system authority
5–device authority
4–system and device authority
3–job authority
2–system and job authority
1–device and job authority
0–system, device, and job authority

This command authority can be changed at any time by the operator.

Note: *The numbers defining these command authorities are intentionally opposite to the numbers of the command authorities defined for the $T operator command.*

Default: 0

CLASS=c

specifies the default job class to be assigned to all jobs submitted through an internal reader that do not specify a job class in the CLASS operand of their JOB statements. *c* can be any character A-Z,0-9.

Default: A

HOLD/NOHOLD

HOLD specifies that all jobs submitted through an internal reader are to be held until they are released for execution by the operator.

Note: *Because all internal readers are treated as a single facility, if one internal reader is held, all internal readers are held. This can be particularly troublesome if TSO users are submitting jobs and the central operator has held the internal readers. This can be overcome by several operating techniques:*

- All jobs submitted via an internal reader can be assigned a class and that class can be held via a JES2 parameter library entry or the $HQn operator command.
- Jobs submitted via the internal reader can use the TYPRUN=HOLD parameter on the JOB card.
- Jobs submitted via an internal reader can be individually held with the $HJ operator command.

Default: NOHOLD, which specifies that jobs submitted through an internal reader are to be queued as usual.

PRIOINC=nn

specifies a number (0-15) to be *added* to the selection priorities of all jobs submitted through internal readers. If the total of this number and a job's priority exceeds the value specified by PRIOLIM, JES2 will assume the priority specified by PRIOLIM.

Default: 0

PRIOLIM=nn

specifies the maximum priority level (0-15) that can be assigned to jobs submitted through an internal reader. If a job's priority (with or without the increment specified by PRIOINC) exceeds this level, it will be reduced to this level.

Default: 15

LINEnnn (RJE Lines)

```
LINEnnn  CODEB/CODEA
         FDUPLEX/HDUPLEX
         HISPEED/LOWSPEED
         IFACEB/IFACEA
         NOADISC/ADISCON
         PASSWORD=cccccccc
         TRANSP/NOTRANSP
         UNIT=cau
         USASCII/EBCDIC
```

The LINEnnn parameter specifies the characteristics of one teleprocessing line to be used during remote job entry. This parameter should be specified for each teleprocessing line. Lines are numbered consecutively (LINE1–LINE255) for the number of lines specified by the &NUMLNES generation parameter. Line characteristics are specified by the following subparameters.

CODEB/CODEA

CODEB specifies code B for this line. Code B refers to the second code in a BSC Adapter that has the Dual Code feature. If the Dual Code feature is not present, CODEB should not be specified.

Default: CODEA which specifies code A for this line.

FDUPLEX/HDUPLEX

FDUPLEX specifies that this is a full-duplex line.

Default: HDUPLEX, which specifies that this is a half-duplex line.

HISPEED/LOWSPEED

HISPEED specifies that this is a high-speed (greater than 9600 baud) line.

Default: LOWSPEED, which specifies that this is a low-speed line.

IFACEB/IFACEA

IFACEB specifies interface B for this line. Interface B refers to the second interface in a BSC Adapter that has the Dual Communications Interface feature. If the adapter for this line doės not have the Dual Communications Interface feature, IFACEB should not be specified.

Default: IFACEA, which specifies interface A for this line.

NOADISC/ADISCON

NOADISC specifies that this line is not to be automatically disconnected from a terminal when the local modem disconnects.

Default: ADISCON, which specifies that this line will be automatically disconnected when the local modem disconnects.

PASSWORD=cccccccc

specifies a security password (1-8 characters) to prevent unauthorized terminals from using this line.

Default: No password.

TRANSP/NOTRANSP

TRANSP specifies that the Text Transparency feature of the BSC Adapter is present on this line.

Default: NOTRANSP, which specifies that the Text Transparency feature of the BSC Adapter is not present on this line.

UNIT=cau

specifies the unit address of this teleprocessing line.

Default: If not specified, JES2 assigns the first available BSC line address.

Note: *The same unit address may be specified for more than one line to allow use of different interfaces or codes available in a single BSC Adapter. JES2 will allow only one of these lines to be started by the operator at any one time.*

USASCII/EBCDIC

USASCII specifies that the BSC Adapter is configured for ASCII line-control characters. When USASCII is specified, this line must be used with a 2770, 2780, or 3780 USASCII terminal.

Note: *Support for this option must be specified by the &USASCII generation parameter.*

Default: EBCDIC, which speciifes that the BSC Adapter is configured for EBCDIC line-control characters.

&NUMBUF (JES2 Working Buffers)

&NUMBUF=nnn

The &NUMBUF parameter specifies the total number of work buffers required for JES2 operations. Use it only to increase the number of buffers from the number specified by the &NUMBUF generation parameter.

nnn

can be any number between the number specified by the &NUMBUF generation parameter and 999. If you specify a number less than or equal to the &NUMBUF generation value, this parameter is ignored.

For a list of required buffers for each JES2 logical function, see the description of the &NUMBUF parameter in the chapter "Installing JES2."

Default: The number specified in the &NUMBUF generation parameter.

&NUMWTOQ (Console Message Buffers)

&NUMWTOQ=nnn

The &NUMWTOQ parameter specifies the number of console message buffers required for JES2 operations. Use it only to increase the number of buffers from the number specified by the &NUMWTOQ generation parameter.

nnn

can be any number between the number specified by the &NUMWTOQ generation parameter and 999. If you specify a number less than or equal to the &NUMBUF generation value, this parameter is ignored.

Message buffers are allocated from the Common Storage Area so care should be used in determining this number. When RJE is used, more message buffers are usually needed. This is especially true with console support for MULTI-LEAVING terminals. Also, serious system degradation can be caused by specifying too few message buffers.

Note: *The number you specify for this parameter does not change the limit for operator commands allowed in the JES2 initialization data set. Nor does it change the remote console queuing limit. Both of these numbers will remain the same as the numbers specified by the &NUMWTOQ generation parameter. Also, this parameter has no correlation to the WTOBFRS parameter in the IEASYSxx parmlib member which specifies buffers for system messages.*

Default: The number specified by the &NUMWTOQ generation parameter.

PRINTERnn (Local Printer)

PRINTERnn CLASS=c_1...c_n
DRAIN/START
FCB=cccc
FORMS=cccc
NOSEP/SEP
OPERATOR/AUTO
PAUSE/NOPAUSE
ROUTECDE=nnnn
UCS=cccc
UNIT=cau

The PRINTERnn parameter specifies the characteristics of one local printer. Printers are numbered consecutively (PRINTER1-PRINTER99) for the number of printers specified by the &NUMPRTS generation parameter. Printer characteristics are defined by the following subparameters.

CLASS=c_1...c_n

specifies the output classes, in priority sequence, to be processed initially by this printer. You can specify any number of classes (A-Z,0-9) up to the number of classes specified by the &NUMCLAS generation parameter.

Default: AJ

DRAIN/START

DRAIN specifies that this printer is to be started by operator command.

Default: START, which specifies that this printer (if it is ready) is to be automatically started when JES2 starts processing.

FCB=cccc

specifies the forms buffer image or the carriage control tape that is to be initially mounted on this printer. cccc is the forms control buffer (FCB) identifier (1 to 4 alphameric characters) that resides in SYS1.IMAGELIB. (Refer to *OS/VS2 System Programming Library: Data Management* for information on how to add FCBs to SYS1.IMAGELIB.)

Default: The identifier specified by the &PRTFCB generation parameter.

FORMS=cccc

specifies the forms identifier (1 to 4 alphameric characters) of the forms that are to be loaded initially in this printer.

Default: The forms identifier specified by the &STDFORM generation parameter.

NOSEP/SEP

NOSEP specifies that separator pages are not to be initially provided between data set groups. (Separator pages can be specified later by the $T command.)

Default: SEP, which specifies that separator pages are to be initially provided between data set groups.

Note: *If a zero number of print lines was specified for the $PRIDCT generation parameter, separator pages will not be produced even if SEP is specified.*

OPERATOR/AUTO

OPERATOR specifies that this printer is to operate initially in operator-controlled (forms) mode.

Default: AUTO, which specifies that this printer is to operate initially in automatic (demand) forms mode.

PAUSE/NOPAUSE

PAUSE specifies that this printer is to pause between data set groups.

Default: NOPAUSE, which specifies that this printer is not to pause between data set groups.

ROUTECDE=nnnn

specifies the internal route code to be assigned to this printer. A route code indicates that this printer is to be eligible for special print routing. *nnnn* may be 0 or any value between 1001 and 1099.

Note: *Route codes for local devices should be used cautiously. Once a printer has been assigned a route code, it will only be considered as an available output device for a job that requests printed output via the ROUTE control statement, the DEST keyword on the OUTPUT control statement, or by operator command.*

Default: 0, which indicates no special routing.

UCS=cccc

specifies the print train (or print chain) that is mounted on this printer. *cccc* is the identifier (1 to 4 characters) of a universal character set (UCS) image that resides in SYS1.IMAGELIB. (Refer to *OS/VS2 System Programming Library: Data Management* for information on how to add a UCS image to SYS1.IMAGELIB.)

If you specify an invalid identifier, JES2 bypasses the UCS loading procedure and issues a setup message so the operator can specify a valid image.

Note: *This subparameter is only valid for a 3211 or a 1403 printer that has the UCS feature. If you specify UCS=0 (or if a zero value was specified for the &PRTUCS generation parameter), JES2 will not load the UCS buffer.*

Default: The identifier specified by the &PRTUCS generation parameter.

UNIT=cau

specifies the unit address of this printer.

Default: If not specified, JES2 assigns the first available printer address that is not already assigned.

PUNCHnn (Local Card Punch)*

```
PUNCHnn   CLASS=c1...cn
          DRAIN/START
          FORMS=cccc
          NOSEP/SEP
          OPERATOR/AUTO
          PAUSE/NOPAUSE
          ROUTECDE=nnnn
          UNIT=cau
```

The PUNCHnn parameter specifies the characteristics of one local card punch. Punches are numbered consecutively (PUNCH1-PUNCH99) for the number of punches specified by the &NUMPUNS generation parameter. Punch characteristics are specified by the following subparameters.

*The dual reader/punch feature is supported by JES2 as shown in the following example. Assume that a 3525 with the read feature has a unit address of 013. In the JES2 initialization data set, the following two items appear:

```
READER   UNIT=013,DRAIN
PUNCH1   UNIT=013
```

When JES2 is started, the reader will be drained and the punch feature will be activated. If the operator later wishes to read data from the 3525, he can drain punch 1 and start reader 2 with operator commands.

CLASS=c_1...c_n

specifies the output classes, in priority sequence, to be processed initially by this card punch. You can specify any number of classes (A-Z,0-9) up to the maximum number of classes specified by the &NUMCLAS generation parameter.

Default: BK

DRAIN/START

DRAIN specifies that this card punch is to be started by operator command.

Default: START, which specifies that this card punch (if it is ready) is to be automatically started when JES2 starts processing.

FORMS=cccc

specifies the forms identifier (1 to 4 alphameric characters) of the forms that are to be loaded initially in this punch.

Default: The forms identifier specified by the &STDFORM generation parameter.

NOSEP/SEP

NOSEP specifies that separator cards are not to be initially provided between data set groups. (Separator cards can be specified later by the $T operator command.)

Default: SEP, which specifies that separator cards are to be initially provided between data set groups.

OPERATOR/AUTO

OPERATOR specifies that this card punch is to operate initially in operator-controlled (forms) mode.

Default: AUTO, which specifies that this card punch is operate initially in automatic (demand) forms mode.

PAUSE/NOPAUSE

PAUSE specifies that this card punch is to pause between data set groups.

Default: NOPAUSE, which specifies that this card punch is not to pause between data set groups.

ROUTECDE=nnnn

specifies the internal route code to be assigned to this card punch. A route code indicates that this card punch is to be eligible for special punch routing. *nnnn* may be 0 or any value between 1001 and 1099.

Note: *Route codes for local devices should be used cautiously. Once a card punch has been assigned a route code, it will only be considered as an available output device for a job that requests punched output via the DEST keyword on the OUTPUT control statement, the ROUTE control statement, or by operator command.*

Default: 0, which means no route code is to be assigned.

UNIT=cau

specifies the unit address of this card punch.

Default: If not specified, JES2 assigns the first available card punch address that is not already assigned.

QCONTROL (Job Queue Control Variables)

```
QCONTROL     HOLD=nnn
             MINDORM=nnn
             MAXDORM=nnn
             WARN=nnn
```

The QCONTROL parameter is used only for systems in a Multi-Access Spool configuration. It defines each system's accessibility to the job queue by the following subparameters.

HOLD=nnn

specifies the minimum length of time (in hundredths of seconds) that the job queues will be held whenever it is acquired by this system. nnn can be any number from 0 to 2,147,483,647 (more than six months).

Default: 100 (one second).

MINDORM=nnn

specifies the minimum length of time (in hundredths of seconds) that this system must wait before it attempts to re-acquire the job queue after it has released it. *nnn* can be any number from 0 to 2,147,483,647.

Default: 100 (one second).

MAXDORM=nnn

specifies the maximum time (in hundredths of seconds) that this system can wait before it must attempt to re-acquire the job queue after it last released it. *nnn* can be any number from 1 to 2,147,483,647.

Note: *If MAXDORM is specified too small, excessive system time could be expended in unnecessary attempts to re-acquire the queue. However, if MAXDORM is specified too large, the start of certain functions and the responses to certain job-oriented SET commands may be delayed.*

Default: 500 (five seconds).

WARN=nnn

specifies the length of time (in hundredths of seconds) after which JES2 will warn the operator of a possible lock-out situation. JES2 issues the HASP260 message and resets the timer interval to the WARN value. nnn can be any number from 1 to 2,147,483,647.

Default: 1000 (ten seconds)

QCONTROL Coding Considerations: The purpose of QCONTROL is to maintain a balanced workload among the JES2 systems by allowing each system adequate access to the shared job queue. The defaults for the QCONTROL subparameters should generally provide this balance among similar JES2 systems. The purpose of requiring one system to hold the job queue when once acquired is to allow multiple updates per access, thereby reducing the processing overhead on that system.

However, in configurations that consist of unequal systems, time values should favor the slower systems. For example, in a configuration with a System/370 Model 168 and a Model 155, HOLD might be raised to 150 for the Model 155 to reflect that system's slower job processing rate. (By comparison, the Model 168 could be assigned the HOLD default value with no consequent performance degradation.) By setting both HOLD and MINDORM to 0, you will have a full contention system.

Another factor to consider is which other data sets may be allocated to the same volume as the job queue. It is best to allocate only seldom used data sets to the same volume as the job queue, then set the HOLD and MINDORM values so that the volume's data sets can be reasonably shared, as in the following example.

> If HOLD is small–25, for example–and MINDORM is 100 for both systems in a two system case, you are guarenteed that the volume will be available at least 50% of the time for processing other data sets placed on that volume.

&RCOMCHR (Instream Command Identifier)

&RCOMCHR=c

The &RCOMCHR parameter specifies the character that will be used to identify all JES2 operator commands entered from a local or remote card reader. Use it only to change the character specified by the &RCOMCHR generation parameter.

c

can be any special character except a comma or an apostrophe. (It can be the same as the character specified by the $BSPACE generation parameter.)

Default: The character specified by the &RCOMCHR generation parameter.

Note: *This character may be the same as the &CCOMCHR character.*

READERnn (Local Card Reader)*

```
READERnn  AUTH=n
          CLASS=c
          DRAIN/AUTO
          HOLD/NOHOLD
          MSGCLASS=c
          PRDEST=nnnn
          PRIOINC=nn
          PRIOLIM=nn
          PUDEST=nnnn
          UNIT=cau
```

The READERnn parameter specifies the characteristics of one local card reader. Readers are numbered consecutively (READER1-READER99) for the number of card readers specified by the &NUMRDRS generation parameter. Reader characteristics are specified by the following subparameters.

AUTH=n

specifies the command authority number for this card reader. This number authorizes certain JES2 commands to be entered at this card reader. *n* is a number from 0-7 defining the kind of commands that can be entered.

7–display only	3–job authority
6 - system authority	2–system and job authority
5–device authority	1–device and job authority
4–system and device authority	0–system, device, and job authority

*The dual reader/punch feature is supported by JES2 as shown in the following example. Assume that a 3525 with the read feature has a unit address of 013. In the JES2 initialization data set, the following two items appear:

```
READER2   UNIT=013,DRAIN
PUNCH1    UNIT=013
```

When JES2 is started, the reader will be drained and the punch feature will be activated. If the operator later wishes to read data from the 3525, he can drain punch 1 and start reader 2 with operator commands.

This command authority can be changed at any time by the operator.

Note: *The numbers defining these command authorities are intentionally opposite to the numbers of the command authorities defined by the $T operator command.*

Default: 0

CLASS=c

specifies the default job class to be assigned to all jobs entered at this card reader that do not specify a job class in the CLASS operand of their JOB statements. c can be any class A-Z,0-9.

Default: A

DRAIN/AUTO

DRAIN specifies that this card reader is to be started by operator command.

Default: AUTO, which specifies that this card reader (if it is ready) is to start automatically when JES2 starts processing.

HOLD/NOHOLD

HOLD specifies that all jobs entered at this card reader are to be held until they are released for execution by the operator.

Default: NOHOLD, which specifies that all jobs entered at this card reader are to be queued as usual.

MSGCLASS=c

specifies the default message class to be assigned to jobs entered at this card reader that do not specify a MSGCLASS operand in their JOB statements. *c* can be any class A-Z,0-9.

Default: A

PRDEST=nnnn

specifies the default printer destination for the print output from all jobs that are entered at this card reader that do not have a ROUTE statement or DEST parameter. *nnnn* can be the route code of a local printer (0,1001-1099) as specified by the PRINTERnn initialization parameter or it can be the route code of a remote printer (1-9999) as specified by the RMTnnn initialization parameter.

Default: 0, which specifies that job output will be printed at any available local printer that was not assigned a route code by the PRINTERnn initialization parameter.

Default: 0

PRIOINC=nn

specifies a number (0-15) to be added to the selection priority of each job entered at this card reader. If the total of this number and a job's priority exceeds the priority level specified by PRIOLIM (described below), JES2 will use the priority level specified by PRIOLIM.

Default: 0

PRIOLIM=nn

specifies the maximum priority level (0-15) that can be assigned to jobs entered at this card reader. If a job's priority (with or without the increment specified by PRIOINC) exceeds this level, it will be reduced to this level.

Default: 15

PUDEST=nnnn

specifies the default card punch destination for the punch output from jobs entered at this card reader that do not have a ROUTE statement or DEST parameter. *nnnn* can be the route code of a local card punch (0,1001-1099) as specified by the PUNCHnn initialization parameter or it can be the route code of a remote card punch (1-999) as specified by the RMTnnn initialization parameter.

Default: 0, which specifies that job output will be printed at any available local card punch that was not assigned a route code by the PUNCHnn initialization parameter.

UNIT=cau

specifies the unit address of this card reader.

Default: If not specified, JES2 assigns the first available card reader address that is not already assigned.

RMTnnn (Remote Terminal)

RMTnnn terminal type
ABUFEX/NOABUFEX
BUFEX/NOBUFEX
COMP/NOCOMP
CONSOLE/NOCON
DISCINTV=n
HXED/VARIABLE
LINE=nnn
MRF/NOMRF
MULTI/HARDWARE
NUMPR=n
NUMPU=n
NUMRD=n
PASSWORD=cccccccc
ROUTECDE=nnn
TABS/NOTABS
TRANSP/NOTRANSP
UNBLOCK/BLOCKED

The RMTnnn parameter specifies the characteristics of one remote terminal. It should be specified for each remote terminal. If not specified, JES2 will assume this is a basic 2770 terminal with no features. Remote terminals are numbered consecutively (RMT1-RMT255) for the number of remote terminals specified by the &NUMRJE generation parameter.

If a remote terminal has attached devices, use the following initialization parameters to describe their characteristics.

Rnnn.PRm–specifies remote printer characteristics
Rnnn.PUm–specifies remote card punch characteristics
Rnnn.RDm–specifies remote card reader characteristics

JES2 associates devices to a remote terminal by correlating the nnn in the above parameters to the nnn in an RMTnnn parameter.

Remote terminal characteristics are specified by the following subparameters.

terminal type
2770
2789
3780
2922
M20-2
M20-4
M20-5
M20-6
S/360
S/370
1130
System/3

specifies the type of terminal or CPU at this remote location. If this terminal is a 2770, 2780, or 3780, it must be supported respectively, by the &BSC2770, &BSC2780, or &BSC3780 generation parameters. MULTI-LEAVING terminals must be supported by the &BSCCPU generation parameter.

Default: 2770

ABUFEX/NOABUFEX

ABUFEX specifies that this (2770) terminal has the additional buffer expansion feature. ABUFEX can be specified even when BUFEX has not been specified for this terminal.

Default: NOABUFEX, which specifies that this terminal does not have the additional buffer expansion feature.

BUFEX/NOBUFEX

BUFEX specifies that this (2770) terminal has the buffer expansion feature.

Default: NOBUFEX, which specifies that this terminal does not have the buffer expansion feature.

COMP/NOCOMP

COMP specifies that this (2770 or 3780) terminal has the compression/expansion feature. Support for this feature must be specified by the &BSHPRES generation parameter.

Default: NOCOMP, which specifies that this terminal does not have the compression/ expansion feature.

CONSOLE/NOCON

CONSOLE specifies that this MULTI-LEAVING terminal has an operator console or that it is simulating a console as specified by the &PRTCONS (RMT) generation parameter.

Default: NOCON, which specifies that this terminal has no operator console.

DISCINTV=nnnn

specifies the interval (in seconds) after which, if there is no successful text transmission in either direction, this terminal will be disconnected from the central processor. Error recovery tries and idle time are not counted as successful text transmission. *nnnn* may be from 1 to 8192 seconds; JES2 rounds this value to the next highest multiple of 30.

Default: 0, which indicates that this terminal is not to be disconnected.

FIXED/VARIABLE

FIXED specifies a fixed data record length for this terminal.

Default: VARIABLE, which specifies a variable data record length.

LINE=nnn

specifies the number of the teleprocessing line that is connected (and dedicated) to this terminal. (The number of this line can not exceed the number of lines specified in the &NUMLNES generation parameter.)

Default: If no line number is specified, JES2 assumes this is a non-dedicated (SIGNON) line.

MRF/<u>NOMRF</u>

MRF specifies that this 2780 terminal has the multiple record feature.

Default: NOMRF, which specifies that this terminal does n t have the multiple record feature.

MULTI/<u>HARDWARE</u>

MULTI specifies that this terminal will use the BSC (binary synchronous) MULTI-LEAVING interfaces.

Default: HARDWARE, which specifies that this terminal will not use the BSC MULTI-LEAVING interfaces.

NUMPR=n

specifies the number (1-7) of printers at this remote terminal. Use the Rnnn.PRm initialization parameter to specify the characteristics of each printer.

Default: 1 (If zero is specified, one will be assumed.)

NUMPU=n

specifies the number (0-7) of card punches at this terminal. Use the Rnnn.PUm initialization parameter to specify the characteristics of each card punch.

Default: 0

NUMRD=n

specifies the number (1-7) of card readers at this remote terminal. Use the Rnnn.RDm initialization parameter to specify the characteristics of each reader.

Default: 1 (If zero is specified, one will be assumed.)

PASSWORD=cccccccc

specifies a security password (1-8 characters) to prevent unauthorized terminals from using this remote terminal's resources.

Default: No password.

ROUTECDE=nnn

specifies the route code to be assigned to this terminal and its associated printers, punches and readers. *nnn* may be any number from 1-999.

Default: If a route code is not specified, JES2 assigns the number of this terminal (RMTnnn) as its route code.

TABS/NOTABS

TABS specifies that this (2770, 2780, or 3780) terminal has the horizontal format control feature. Support for this feature must be specified by the &BSHTAB generation parameter.

Default: NOTABS, which specifies that this terminal does not have the horizontal format control feature.

TRANSP/NOTRANSP

TRANSP specifies that this terminal has the text transparency feature. To be effective, the TRANSP subparameter must also be specified for the line (LINEnnn initialization parameter) connected to this terminal.

Default: NOTRANSP, which specifies that this terminal does not have the text transparency feature.

UNBLOCK/BLOCKED

UNBLOCK specifies an unblocked data record format for this terminal.

Default: BLOCKED, which specifies a blocked data record format.

Rnnn.PRm (Remote Printer)

```
Rnnn.PRm    AUTO/OPERATOR
            CLASS=c1...cn
            DRAIN/START
            FCB=cccc
            FCBLOAD/NOFCBLOD
            FORMS=cccc
            NOSEP/SEP
            NOSUSPND/SUSPEND
            PRWIDTH=nnn
            UCS=cccc
```

The Rnnn.PRm parameter specifies the characteristics of one printer at a remote terminal. *nnn* is the number of a remote terminal as specified in the RMTnnn parameter and *m* is the number of this printer. Printers are numbered consecutively (Rnnn.PR1 to Rnnn.PR7) for the number of printers specified (NUMPR=n in the RMTnnn parameter) for this remote terminal. For example, if there are three printers attached to remote terminal number 28, the printers are numbered RMT28.PR1, RMT28.PR2, and RMT28.PR3.

Characteristics for remote printers are specified by the following subparameters.

AUTO/OPERATOR

AUTO specifies that this printer is initially to operate in automatic (demand) forms mode when JES2 starts processing.

Note: *Printers connected to terminals without MULTI-LEAVING should be operated in operator-controlled mode.*

Default: OPERATOR, which specifies that this printer is initially to operate in operator-controlled (forms) mode.

CLASS=c_1...c_n

specifies the output classes, in priority sequence, to be processed by this printer. You can specify any number of classes (A-Z,0-9) up to the maximum number of classes specified by the &NUMCLAS generation parameter.

Default: AJ

DRAIN/START

DRAIN specifies that this printer is to be started by operator command.

Default: START, which specifies that this if it is ready) is to be automatically started when JES2 starts processing.

FCB=cccc

specifies the forms buffer image or the carriage control tape that is to be initially mounted on this printer. *cccc* is the forms control buffer (FCB) identifier (1 to 4 alphameric characters) that resides in SYS1.IMAGELIB. (Refer to *OS/VS2 System Programming Library: Data Mangement* for information on how to add FCBs to SYS1.IMAGELIB.)

Default: The identifier specified by the &PRTFCB generation parameter.

FCBLOAD/NOFCBLOD

FCBLOAD specifies that FCB support is to be provided for this printer.

Note: *FCBLOAD is effective only if this is a 3211 printer attached to a S/360 or S/370 (not including M/20) CPU which has the text transparency feature (TRANSP specified for both the LINEnnn and the RMTnnn parameters). Also, the length of FCB images that can be used for this printer cannot exceed the line length specified for this printer (PRWIDTH) minus two.*

Default: NOFCBLOD, which specifies that no FCB support is to be provided for this printer.

FORMS=cccc

specifies the forms identifier (1 to 4 alphameric characters) of the forms that are to be loaded initially in this printer.

Default: If no value is specified, JES2 will use the forms identifier specified by the &STDFORM generation parameter.

NOSEP/SEP

NOSEP specifies that separator pages are not to be provided initially between data set groups. (Separator pages can be specified later by operator command.) NOSEP also suppresses printout of operator messages.

Default: SEP, which specifies that separator pages are to be provided initially between data set groups.

Note: *If a zero number of print lines was specified by the $TPIDCT generation parameter, separator pages will not be produced even if SEP is specified.*

NOSUSPND/SUSPEND

NOSUSPND specifies that this printer is not to use the printer interrupt feature. This feature allows the remote operator to interrupt printing to transmit jobs or JES2 commands to JES2.

Default: SUSPEND, which specifies that this printer is to use the printer interrupt feature.

Note: *This parameter is ignored for printers that are connected to terminals without MULTI-LEAVING.*

PRWIDTH=nnn

specifies the maximum number of characters to be printed on one line.

Note: *This value must not exceed the printer width you specified during RMT generation of this MULTI-LEAVING terminal (via the &PRTSIZE parameter for Mod 20, S/360, and S/370 terminals and via the &PRFOTLW parameter for 1130 terminals.*

Default: 120

UCS=cccc

specifies the print train (or print chain) that is initially mounted on this printer.

Default: The identifier specified by the &PRTUCS generation parameter.

Rnnn.PUm (Remote Card Punch)

```
Rnnn.PUm    AUTO/OPERATOR
            CLASS=c1...cn
            DRAIN/START
            FORMS=cccc
            NOSEP/SEP
            NOSUSPND/SUSPEND
```

The Rnn.PUm parameter specifies the characteristics of one card punch at a remote terminal. *nnn* is the number of a remote terminal as specified in the RMTnnn parameter and m is the number of this card punch. Card punches are numbered consecutively (Rnnn.PU1 to Rnnn.PU7) for the number of card punches specified (NUMPU=n in the RMTnnn parameter) for this remote terminal. For example, if there are two punches attached to remote terminal number 14, the punches are numbered RMT14.PU1 and RMT14.PU2.

Characteristics for remote punches are specified by the following subparameters.

AUTO/OPERATOR

AUTO specifies that this card punch is to initially operate in automatic (demand) forms mode when JES2 starts processing.

Note: *Punches connected to terminals without MULTI-LEAVING must be operated in operator controlled mode.*

Default: OPERATOR, which specifies that this punch is initially to operate in operator-controlled (forms) mode.

CLASS=$c_1 ... c_n$

specifies the output classes, in priority sequence, to be processed initially by this card punch. You can specify any number of classes (A-Z,0-9) up to the maximum number of classes specified by the &NUMCLAS generation parameter.

Default: BK

DRAIN/START

DRAIN specifies that this card punch is to be started by operator command.

Default: START, which specifies that this punch (if it is ready) is to be automatically started when JES2 starts processing.

FORMS=cccc

specifies the forms identifier (1 to 4 alphameric characters) of the forms that are to be loaded initially in this card punch.

Default: JES2 will use the identifier specified by the &STDFORM generation parameter.

NOSEP/SEP

NOSEP specifies that separator cards are not initially to be provided between data set groups. (Separator cards can be specified later by operator command.)

Default: SEP, which specifies that separator cards are to be provided initially between data set groups.

NOSUSPND/SUSPEND

NOSUSPND specifies that this card punch is not to use the punch interrupt feature. This feature allows the remote terminal operator to interrupt punching to transmit jobs or JES2 commands to JES2.

Default: SUSPEND, which specifies that this card punch is to use the punch interrupt feature.

Note: *This parameter is ignored for punches that are attached to terminals without MULTI-LEAVING.*

Rnnn.RDm (Remote Card Reader)

```
Rnnn.RDm    CLASS=c
            DRAIN/START
            HOLD/NOHOLD
            MSGCLASS=c
            PRDEST=nnnn
            PRIOINC=nn
            PRIOLIM=nn
            PUDEST=nnnn
```

The Rnnn.RDm parameter specifies the characteristics of one card reader at a remote terminal. *nnn* is the number of the remote terminal as specified in the RMTnnn parameter and m is the number of this reader. Readers are numbered consecutively (Rnnn.RD1 to Rnnn.RD7) for the number of readers (NUMRD=n in the RMTnnn parameter) specified for this remote terminal. For example, if there are three card readers attached to remote terminal 2, the readers are numbered RMT2.RD1, RMT2.RD2, and RMT2.RD3.

Characteristics for remote readers are specified by the following subparameters.

CLASS=c

specifies the default job class to be assigned to all jobs entered at this card reader that do not specify a job class in the CLASS operand of their JOB statements. *c* can be any class A-Z,0-9.

Default: A

DRAIN/START

DRAIN specifies that this card reader is to be started by operator command.

Default: START, which specifies that this card reader is to start automatically when JES2 starts processing.

HOLD/NOHOLD

HOLD specifies that all jobs entered at this card reader are to be held until they are released for execution by the operator.

Default: NOHOLD, which specifies that jobs entered at this card reader are to be queued as usual.

MSGCLASS=c

specifies the default message class to be assigned to jobs entered at this card reader that do not specify a MSGCLASS operand in their JOB statements. *c* can be any class A-Z,0-9.

Default: A

PRDEST=nnnn

specifies the default printer destination for the print output from all jobs that are entered at this card reader that do not have a ROUTE statement or DEST parameter. *nnnn* is the route code (0, 1001-1099 for a local printer or 1-999 for a remote printer) as specified by the PRINTERnn or RMTnnn initialization parameter. When 0 is specified, job output will be printed at any available local printer that was not assigned a route code by the PRINTERnn initialization parameter.

Default: The route code (ROUTECDE) specified in the RMTnnn parameter for this remote terminal.

PRIOINC=nn

specifies a number (0 to 15) to be added to the selection priority of each job entered at this card reader. If the total of this number and a job's priority exceeds the priority level specified by PRIOLIM, JES2 will use the priority level specified by PRIOLIM.

Default: 0

PRIOLIM=nn

specifies the maximum priority level (0 to 15) that can be assigned to jobs entered at this card reader. If a job's priority (with or without the increment specified by PRIOINC) exceeds this level, it will be reduced to this level.

Default: 15

PUDEST=nnnn

specifies the default card punch destination for the punch output from jobs entered at this card reader that do not have a ROUTE statement or a DEST parameter. *nnnn* is the route code (0, 1001-1099 for a local card punch or 1-999 for a remote punch) as specified by the PUNCHnn or RMTnnn initialization parameter. When 0 is specified, job output will be printed at any available local card punch that was not assigned a route code by the PUNCHnn parameter.

Default: PUDEST=0

Sn (System Identifier)

Sn SID=cccc

The Sn parameter is required to identify each system in a Multi-Access Spool configuration. The initialization data set for each system must contain the Sn parameters of all the systems in the configuration. That is, if there are three systems (K158, L158, L168) in the configuration, the initialization data set for each system would contain the following parameters:

S1 SID=K158
S2 SID=L158
S3 SID=L168

Systems are numbered consecutively from one to seven (S1-S7). The system identifier is specified by the following subparameter.

SID=cccc

specifies the system identifer. *cccc* is the four-character alphameric name that was generated as the System Management Facility (SMF) system ID for this system.

Default: For a single system configuration, the system identifier for S1 defaults to the generated SMF system ID. For Multi-Access Spool configurations, the system IDs must be specified for each system and no default is permitted.

&SPOOL (Spool Volume ID)

&SPOOL=cccccc

The &SPOOL parameter changes the volume serial number of the primary spool volume from the one specified by the &SPOOL generation parameter.

cccccc

specifies the volume serial number of the primary spool volume. Any six characters that define a valid volume serial number may be used. (This is a change from HASP where you specified only the first five characters of the primary spool volume ID.)

If you change more than the last character of this number from the number specified by the generation parameter, you also change the volume serial numbers of all the spooling volumes. (The first five characters of this number define the first five characters of the volume serial numbers of all the JES2 spooling volumes.)

Note: *If this parameter is changed to other than SPOOL1 (the generation default), certain messages will vary from their documentation in* OS/VS Message Library: VS2 System Messages.

Default: the number specified in the &SPOOL generation parameter.

&STC,&TSU,&x (Job Class)

```
( &STC )   CONVPARM=bppmmmmssccrlaaaaef
{ &TSU }   COPY/NOCOPY
( &x   )   HOLD/NOHOLD
           NOJOURN/JOURNAL
           NOLOG/LOG
           NOOUTPUT/OUTPUT
           NOTYPE6/TYPE6
           NOTYPE26/TYPE26
           NOUJP/IEFUJP
           NOUSO/IEFUSO
           PERFORM=nnn
           PROCLIB=nn
           SCAN/NOSCAN
           XBATCH/NOXBATCH
```

The &STC,&TSU and &x parameters specify the characteristics to be associated with one job class. Job classes are specified as follows:

&STC—defines the job class for started tasks
&TSU—defines the job class for time sharing users
&x—defines any batch class A-Z,0-9

Job class characteristics are specified by the following subparameters.

CONVPARM=bppmmmmssccrlaaaaef

specifies the default parameters to be used by the OS/VS2 Converter to process jobs in this job class. The parameters are specified by 20 hexadecimal characters that must be coded in order, as follows:

b

specifies whether an account number and/or programmer name is required for this job, as follows:

0—no account number, no programmer name
1—no account number, programmer name required
2—account number required, no programmer name
3—account number required, programmer name required

pp

currently unused. Code two zeros to maintain positioning within the parameter field.

mmmmss

six numeric characters indicating the default for the maximum time that each job step may execute. (When a job step exceeds this limit, it is cancelled.) The first four characters indicate minutes; the last two indicate seconds. This time limit is assigned to the job step when no time limit is specified in the JOB or EXEC statement. The value specified is subject to the limits described for the TIME parameter in *OS/VS2 JCL.*

ccc

three numeric characters indicating the default for the region size (specified as a number of 1024 byte blocks) assigned to each job step. This region size is assigned when no region size is specified in the JOB or EXEC statement and the job step is to be run with ADDRSPC=VIRT.

Note: *Do not specify a* ccc *value of 000 with &x. Specifying 000 in conjunction with no region size specification on the JOB or EXEC statement and a* ccc *value specification of 000 in the &RDROPSU generation parameter may yield unpredictable results (see* OS/VS2 SPL: Job Management—*"Limiting User Region Size").*

r

a numeric character from 0 to 3 that specifies the disposition of commands read from the input stream as follows:

0—The OS/VS2 Converter passes the command to the scheduling routine to be executed.

1—The OS/VS2 Converter displays the command (via a WTO macro instruction), and passes it to the command scheduling routine to be executed.

2—The OS/VS2 Converter displays the command (via a WTO macro instruction), asks the operator whether the command should be executed (via a WTOR macro instruction), and passes the command to the command scheduling routine if the operator replies yes.

3—The OS/VS2 Converter ignores the command.

l
a numeric character that specifies the bypass label processing (BLP) option as follows:

0–The bypass label processing parameter in the label field of a DD statement is to be ignored; the label parameter is processed as no label.

1–Bypass label processing is not to be ignored; the label parameter is processed as it appears.

aaaa
four hexadecimal numbers from 0000 to E000 indicating which operator command groups are to be executed as follows:

Byte	Bits	Bit Settings	Meaning
0	0	1	Group 1 commands
	1	1	Group 2 commands
	2	1	Group 3 commands
	3-7	00000	Reserved
1	0-7	00000000	Reserved

ef
two numeric characters that specify a message level for use when the MSGLEVEL parameter is not specified on a JOB statement. If a MSGLEVEL parameter is not specified, JCL and allocation/termination messages are recorded in the system message data set according to the following values:

e
specifies the kinds of JCL listed as follows:

0–JOB statement only.
1–Input statements, cataloged procedure statements, and symbolic parameter substitution values.
2–Input statements only, including instream procedures.

f
specifies the kinds of allocation/termination messages listed as follows:

0–No messages are to be listed except in the case of an abnormal termination. (In that event, all messages are listed.)
1–All messages are listed.

Default: If this subparameter is not specified, the Converter will use the parameters specified in the following JES2 generation parameters:

&RDROPSL, for the time sharing user job class
&RDROPST, for the started task job class
&RDROPSU, for background job classes.

COPY/<u>NOCOPY</u>
COPY specifies that jobs in this job class are to be queued for output processing as though TYPRUN=COPY was specified on the JOB statement for these jobs.

Default: NOCOPY, which specifies that jobs in this job class are to be queued as usual. NOCOPY will be ignored if the TYPRUN=COPY parameter is specified on the JOB statement for a job.

HOLD/<u>NOHOLD</u>
HOLD specifies that jobs in this job class are to be held until a RELEASE command is issued by the operator.

Default: NOHOLD, which specifies that jobs in this job class are to be queued as usual. NOHOLD will be ignored if the TYPRUN=HOLD parameter is specified on the JOB statement for a job (or if the job is held for other reasons).

NOJOURN/JOURNAL

NOJOURN specifies that information for the job journal is not to be processed for this job class. Note that the job journal contains checkpoint/restart information. Jobs that are not recorded in the job journal cannot be automatically restarted or warmstarted.

Default: JOURNAL, which specifies that information for the job journal is to be processed for this job class.

NOLOG/LOG

NOLOG specifies that the JES2 job log is not to be printed for this job class. The JES2 job log contains the user's console messages and replies to WTORs issued during the processing of the job. When NOLOG is specified, JES2 statistics information, normally printed with the job, is also suppressed.

Default: LOG, which specifies that the job log is to be printed for this job class. Even when LOG is specified, the job log may be suppressed on an individual job basis via a parameter in the accounting field of the JOB card or via a parameter on a JOBPARM control card.

NOOUTPUT/OUTPUT

NOOUTPUT specifies that no SYSOUT data is to be written for jobs executed in this job class.

Default: OUTPUT, which specifies that SYSOUT data is to be written for jobs executed in this job class.

NOTYPE6/TYPE6

NOTYPE6 specifies that JES2 is not to produce type 6 SMF (external writer) records for jobs in this job class. Type 6 SMF records are written for each group of job-related data sets and each spinoff data set that is processed. Type 6 records are described in *OS/VS System Management Facilities (SMF).*

Default: TYPE6, which specifies that JES2 is to produce type 6 SMF records for this job class. When type 6 records are to be produced, the &NUMSMFB generation parameter must specify two or more SMF buffers. The maximum size of the records is set by the &SMFRSIZ generation parameter.

Note: *Specifying TYPE6 has no effect for the &STC parameter which assumes the NOTYPE6 option.*

NOTYPE26/TYPE26

NOTYPE26 specifies that JES2 is not to produce type 26 (job summary) SMF records for jobs in this job class. Type 26 records are described in *OS/VS System Management Facilities (SMF).*

Default: TYPE26, which specifies that JES2 is to produce type 26 SMF records for jobs in this job class. When type 26 records are to be produced, the &NUMSMFB generation parameter must specify two or more SMF buffers. The maximum size of the records is set by the &SMFRSIZ generation parameter.

Note: *Specifying TYPE26 has no effect on the &STC parameter which assumes the NOTYPE26 option.*

NOUJP/IEFUJP

NOUJP specifies that the IEFUJP exit is not to be taken when a job is purged. IEFUJP receives control when a job is ready to be purged from the system; that is, after the job has been terminated and all the SYSOUT output that pertains to the job has been processed.

Default: IEFUJP, which specifies that the IEFUJP exit is to be taken when a job is purged.

Note: *Specifying IEFUJP has no effect on the &STC parameter which assumes the NOUJP option.*

NOUSO/IEFUSO

NOUSO specifies that the IEFUSO user exit is not to be taken when the SYSOUT limit is reached for a job in this job class. The SYSOUT limit, which is specified by the OUTLIM parameter on the JOB card, defines the maximum number of physical records to be written to the associated SYSOUT data set. When the OUTLIM value is exceeded, JES2 normally calls the IEFUSO SMF exit routine to either increase the SYSOUT limit or to terminate the job. When NOUSO is specified and OUTLIM is exceeded, JES2 abnormally terminates the job.

Default: IEFUSO, which specifies that the IEFUSO user exit is to be taken when the SYSOUT limit is reached for a job in this job class.

Note: *IEFUSO has no effect on the &STC parameter which assumes the NOUSO option.*

PERFORM=nnn

specifies the default performance group number for this job class. This number is used when a performance group number is not specified on the JOB or EXEC control statement for a job of this job class. Any decimal number form 1-255 can be specified.

Default: 0, which indicates that no performance group processing will be performed by JES2 and causes the Workload Manager to assign a default value (1 for background jobs and 2 for foreground jobs) and issue an error message.

PROCLIB=nn

specifies the default procedure library number which is to be used for this job class. It allows you to specify different procedure libraries for different job classes. In the JES2 procedure, one DD statement must be named PROC00. If you specify additional procedure libraries (01-99) at that time, you may associate these libraries to a job class by replacing the nn of this subparameter with the appropriate procedure library number.

Note: *All cataloged procedure libraries to be used by jobs, time sharing users or system tasks must be defined in the JES2 procedure.*

Default: 00

SCAN/NOSCAN

SCAN specifies that jobs in this job class are to be queued for output processing immediately after JCL conversion as though TYPRUN=SCAN was specified on the JOB statement for these jobs.

Default: NOSCAN, which specifies that jobs in this job class are to be queued as usual. NOSCAN will be ignored if the TYPRUN=SCAN parameter is specified on the JOB statement for a job.

XBATCH/NOXBATCH

XBATCH specifies that the Execution Batch Scheduling feature is to be associated with this job class. Support for this feature must have been specified by the &XBATCH generation parameter. (This feature is described for JES2 in the chapter "JES2 Processing.")

Default: NOXBATCH, which specifies that the Execution Batch Scheduling feature is not to be associated with this job class.

STCMCLAS (Message Class for Started Tasks)

STCMCLAS=c

The STCMCLAS parameter specifies the message class for all started tasks. All job control statements and system messages will be assigned to this class.

c

can be any valid message class (A-Z,0-9).

Default: A

TSUMCLAS (TSO Message Class)

TSUMCLAS=c

The TSUMCLAS parameter specifies the message class for all time-sharing foreground jobs. It also specifies the default message class for all background jobs submitted from a time-sharing session.

c

can be any valid message class (A-Z,0-9).

Default: A

$$x (SYSOUT Class)

```
$$x    DUMMY/SYSOUT
       HOLD/NOHOLD
       PUNCH/PRINT
```

This parameter specifies the SYSOUT class characteristics for one output class. *x* can be any one of the 36 (A-Z,0-9) possible output classes.

DUMMY/SYSOUT

DUMMY specifies that JES2 is to process this output class as a dummy data set (like a DD DUMMY statement).

Default: SYSOUT specifies that JES2 is to process this output class as SYSOUT.

HOLD/NOHOLD

HOLD specifies that data sets specifying this SYSOUT class are to be eligible to be held for TSO SYSOUT processing. The class specified on the MSGCLASS parameter should be either the same class or another class that is also defined with $$x HOLD.

Note: *The only other way to specify that a data set is to be held is by specifying the HOLD=YES parameter on the data definition (DD) control statement that defines the SYSOUT data set. In this case, the data set will be held even though NOHOLD is specified for this subparameter.*

Default: NOHOLD, which specifies that these data sets are not to be held for TSO user processing.

PUNCH/PRINT

PUNCH specifies that this output class is normally to be punched.

Default: PRINT, which specifies that this output class is normally to be printed. For classes B and K, the default is PUNCH.

Note: *Any class may be specified for either print or punch. The purpose of this subparameter is to define the installation's standard for output classes so that appropriate print and punch accounting can be maintained.*

Figure 3-7 (on the following pages) lists all of the JES2 initialization parameters and subparameters with related JES2 generation parameters and operator commands.

Specified at JES2 Initialization	JES2 Initialization Parameters and Subparameters	Relation to JES2 Generation Parms		Operator Commands that Change Values Specified by Init Parms
		Generation Parm	Relation*	
Local operator command identifier	&CCOMCHR=c	&CCOMCHR	Overrides	
Volume serial number of the SYS1.HASPCKPT volume	&CHKPT=cccccc	&SPOOL	Overrides	
Characteristics of logical initiator nnn	Innn	&MAXPART	Depends on	
assigned job classes in priority sequence	CLASS=$c_1 \ldots c_n$	&MAXCLAS	Depends on	$T In, classes
operator start or automatic start	DRAIN/START			
unique name for reference	NAME=cc			
Characteristics of all JES2 internal readers	INTRDR	&NUMINRS	Depends on	
command authority number	AUTH=n			$T RDI, A=n
default job class when not specified with job	CLASS=c			$T RDI, C=class
hold or queue jobs	HOLD/NOHOLD			&T RDI, H/R
increment added to job selection priorities	PRIOINC=nn			
maximum job priority level	PRIOLIM=nn			
Characteristics of remote line nnn	LINEnnn	&NUMLNES	Depends on	
code B or code A	CODEB/CODEA			
full-duplex or half-duplex line	FDUPLEX/HDUPLEX			
high-speed or low-speed line	HISPEED/LOWSPEED			
interface B or interface A	IFACEB/IFACEA			
automatic disconnect when terminal disconnects	NOADISC/ADISC			
security password	PASSWORD=cccccccc			$T LNEn, P=[password]
text transparency feature	TRANSP/NOTRANSP			
unit address	UNIT=cau			
ASCII or EBCDIC line-control characters	USASCII/EBCDIC	&USASCII	Depends on	
Total number of JES2 work buffers	&NUMBUF=nnn	&NUMBUF	Overrides	
Total number of JES2 console message buffers	&NUMWTOQ=nnn	&NUMWTOQ	Overrides	
Characteristics of local printer nn	PRINTERnn	&NUMPRTS	Depends on	
assigned output classes, in priority sequence	CLASS=$c_1 \ldots c_n$	&NUMCLAS	Depends on	$T PRTn, Q=classes
operator start or automatic start	DRAIN/START			
initial forms buffer image for carriage control tape	FCB=cccc	&PRTFCB	Overrides	$T PRTn, C=id
forms identifier of initial forms	FORMS=cccc	&STDFORM	Overrides	$T PRTn, F=form
separator pages between data set groups	NOSEP/SEP	&PRIDCT	Might depend	$T PRTn, S=Y/N
operator-controlled or automatic form mode	OPERATOR/AUTO			$T PRTn, F=form/AUTOM
pause between data set groups	PAUSE/NOPAUSE			$T PRTn, P=Y/N
internal route code	ROUTECDE=nnnn			
print train or chain identifier	USC=cccc	&PRTUCS	Overrides, Might depend	$T PRTn, T=id
unit address	UNIT=cau			

Figure 3-7 (Part 1 of 4). JES2 Initialization Parameters With Related Generation Parameters and Operator Commands

Specified at JES2 Initialization	JES2 Initialization Parameters and Subparameters	Relation to JES2 Generation Parms		Operator Commands that Change Values Specified by Init Parms
		Generation Parm	Relation*	
Characteristics of local punch nn	PUNCHnn	&NUMPUNS	Depends on	$T PUNn, Q=classes
assigned output classes, in priority sequence	CLASS=$c_1 \dots c_n$	&NUMCLAS	Depends on	
operator start or automatic start	DRAIN/START			
forms identifier of initial forms	FORMS=cccc	&STDFORM	Overrides	$T PUNn, F=form
separator cards between data set groups	NOSEP/SEP			$T PUNn, S=Y/N
operator-controlled or automatic forms mode	OPERATOR/AUTO			$T PUNn, F=form/AUTOM
pause between data set groups	PAUSE/NOPAUSE			$T PUNn, P=Y/N
internal route code	ROUTECDE=nnnn			
unit address	UNIT=cau			
Control variables for shared job queue	QCONTROL			
queue hold time	HOLD=nnn			
minimum dormant time	MINDORM=nnn			
maximum dormant time	MAXDORM=nnn			
lockout warning time	WARN=nnn			
Identifier for instream commands	&RCOMCHR=c	&RCOMCHR	Overrides	
Characteristics of local card reader nn	READERnn	&NUMRDRS	Depends on	
command authority number	AUTH=n			$T RDRn, A=n
default job class when not specified with job	CLASS=c			$T RDRn, C=class
operator start or automatic start	DRAIN/AUTO			
hold or queue jobs	HOLD/NOHOLD			$T RDRn, H/R
default message class when not specified with job	MSGCLASS=c			$T RDRn, Q=class
default printer destination when not specified with job	PRDEST=nnnn			
increment added to job selection priorities	PRIOINC=nn			
maximum job priority level	PRIOLIM=nn			
default card punch destination when not specified with job	PUDEST=nnnn			
unit address	UNIT=cau			

Figure 3-7 (Part 2 of 4). JES2 Initialization Parameters With Related Generation Parameters and Operator Commands

Specified at JES2 Initialization	JES2 Initialization Parameters and Subparameters	Relation to JES2 Generation Parms		Operator Commands that Change Values Specified by Init Parms
		Generation Parm	Relation*	
Characteristics of remote terminal nnn	RMTnnn	&NUMRJE	Depends on	
type of terminal	terminal type	&BSCCPU	Depends on --	--or &BSC2770 &BSC2780 &BSC3780
additional buffer expansion feature	ABUFEX/NOABUFEX			
buffer expansion feature	BUFEX/NOBUFEX			
compression/expansion feature	COMP/NOCOMP	&BSHPRES	Depends on	
operator console	CONSOLE/NOCON	&PRTCONS**	Depends on	
maximum interval between successful text transmissions	DISCINTV=n			
fixed or variable data record length	FIXED/VARIABLE			
number of the remote line connected to this terminal	LINE=nnnn	&NUMLNES	Depends on	
multiple record feature	MRF/NOMRF			
BSC MULTI-LEAVING interfaces	MULTI/HARDWARE			
number of attached printers	NUMPR=n			
number of attached card punches	NUMPU=n			
number of attached card readers	NUMRD=n			
security password	PASSWORD=cccccccc			
route code for terminal and all attached devices	ROUTECDE=nnnn			
horizontal format control feature	TABS/NOTABS	&BSHTAB	Depends on	
text transparency feature	TRANSP/NOTRANSP			
unblocked or blocked data record format	UNBLOCK/BLOCKED			
Characteristics of printer m at remote terminal nnn	Rnnn.PRm			
automatic forms or operator-controlled mode	AUTO/OPERATOR			$T Rn.PRn, F=form/AUTOM
assigned output classes, in priority sequence	CLASS=$c_1 \dots c_n$	&NUMCLAS	Depends on	$T Rn.PRn, Q=classes
operator start or automatic start	DRAIN/START			
initial forms buffer image for carriage control tape	FCB=cccc	&PRTFCB	Overrides	$T Rn.PRn, C=id
FCB support	FCBLOAD/NOFCBLOD			
forms identifier of initial forms	FORMS=cccc	&STDFORM	Overrides	$T Rn.PRn, F=form
separator pages between data set groups	NOSEP/SEP	$TPIDCT	Might depend	$T Rn.PRn, S=Y/N
printer interrupt feature	NOSUSPND/SUSPEND			

characters per print line	PRWIDTH=nnn	{ &PRTSIZE** }
		{ &PRFOTLW** }
			Depends on	

print train or chain identifier	UCS=cccc	&PRTUCS	Overrides	$T Rn.PRn, T=id
Characteristics of card punch m at remote terminal nnn	Rnnn.PUm			
automatic forms or operator-controlled mode	AUTO/OPERATOR			$T Rn.PUn, F=form/AUTOM
assigned output classes, in priority sequence	CLASS=$c_1 \dots c_n$	&NUMCLAS	Depends on	$T Rn.PUn, Q=classes
operator start or automatic start	DRAIN/START			
forms identifier of initial forms	FORMS=cccc	&STDFORM	Overrides	$T Rn.PUn, F=form
separator cards between data set groups	NOSEP/SEP			$T Rn.PUn, S=Y/N
punch interrupt feature	NOSUSPND/SUSPEND			

Figure 3-7 (Part 3 of 4). JES2 Initialization Parameters With Related Generation Parameters and Operator Commands

Specified at JES2 Initialization	JES2 Initialization Parameters and Subparameters	Relation to JES2 Generation Parms		Operator Commands that Change Values Specified by Init Parms
		Generation Parm	Relation*	
Characteristics of card reader m at remote terminai nnn	Rnnn.RDm			
default job class when not specified with job	CLASS=c			$T Rn.RDn, C=class
operator start or automatic start	DRAIN/START			
hold or queue jobs	HOLD/NOHOLD			$T Rn.RDn, H/R
default message class when not specified with job	MSGCLASS=c			$T Rn.RDn, Q=class
default printer destination when not specified with job	PRDEST=nnnn			
increment added to job selection priorities	PRIOINC=nn			
maximum job priority level	PRIOLIM=nn			
default card punch destination when not specified with job	PUDEST=nnnn			
System identifier	Sn			
system identifier name	SID=cccc			
Volume serial number of primary spool volume	&SPOOL=cccccc	&SPOOL	Overrides	
Characteristics of a job class	&STC, &TSU, &x			
default parameters for OS/VS Converter	CONVPARM=parmchars	&RDROPSL &RDROPST &RDROPSU	Overrides	
hold or queue jobs	HOLD/NOHOLD			
output or queue jobs	COPY/NOCOPY			
maintain job journal	NOJOURN/JOURNAL			
print job log	NOLOG/LOG			
write SYSOUT data	NOOUTPUT/OUTPUT			
produce type 6 (external writer) SMF records	NOTYPE6/TYPE6	&NUMSMFB	Might depend	
produce type 26 (job summary) SMF records	NOTYPE26/TYPE26	&NUMSMFB	Might depend	
take IEFUJP exit when a job is purged	NOUJP/IEFUJP			
take IEFUSO user exit at SYSOUT limit	NOUSO/IEFUSO			
default performance group when not specified with job	PERFORM=nnn			
default procedure library when not specified with job	PROCLIB=nn			
scan or queue jobs	SCAN/NOSCAN			
execution batch scheduling feature	XBATCH/NOXBATCH	&XBATCH	Depends on	
Message class for all started tasks	STCMCLAS			
Message class for all time-sharing foreground jobs	TSUMCLAS			
SYSOUT class characteristics for output class x	$$x			
process as dummy or as SYSOUT data set	DUMMY/SYSOUT			
hold data sets for TSO SYSOUT processing	HOLD/NOHOLD			
punch or print output class	PUNCH/PRINT			

* Relationships of initialization parameters to generation parameters as follows:

Depends on — Initialization parameter depends on value of this generation parameter.
Overrides — Initialization parameter overrides value of this generation parameter.
Might depend — Initialization parameter might be affected by value of this generation parameter.

** RMT generation parameter

Figure 3-7 (Part 4 of 4). JES2 Initialization Parameters With Related Generation Parameters and Operator Commands

CHAPTER 4. JES2 PROCESSING

By means of JES2 and RMT generation and JES2 initialization, you can define and control the configuration of job entry sources and job output destinations. You can also indicate how JES2 is to control certain aspects of job input, queuing, and output.

This chapter describes the aspects of JES2 processing that you can affect by means of generation and initialization parameters and operator commands. For specific information about coding generation and initialization parameters, refer to the chapters "Installing JES2" and "JES2 Initialization," respectively. For information about the operator commands, see *Operator's Library: OS/VS2 Reference (JES2).*

Configuration

During JES2 generation and initialization, the system programmer can specify the configuration of JES2 local devices, the JES2 internal reader facility, the JES2 remote lines and devices, and the JES2 spool volumes.

Local Device Configuration

Local devices refer to card readers, printers, and card punches which are attached to the MVS system and which are used for reading jobs and writing output.

During JES2 generation the system programmer specifies the number of readers, printers and punches to be controlled by JES2 via the &NUMRDRS, &NUMPRTS and &NUMPUNS parameters. It is not possible to assign to JES2 more than the number of devices specified without regenerating the JES2 system.

During JES2 initialization, the system programmer can identify the devices which are to be used by JES2 via the READERnn, PRINTERnn and PUNCHnn parameters. The system programmer can also specify JES2 processing parameters to be associated with each device and indicate whether the device is to be considered active or inactive upon completion of JES2 initialization. An active device is dynamically allocated during JES2 initialization, and processing on that device begins as soon as work is available. An inactive device must be activated by the operator via the JES2 START command (S).

If, during JES2 initialization, the system programmer does not identify as many devices as were specified during JES2 generation, JES2 selects devices and dynamically allocates them. Devices are selected according to lowest device address for each type of device (reader, printer, punch), until the number specified during JES2 generation is obtained or no devices of that type remain. For a device to be selected, it must be physically attached to the system. For devices not identified to JES2 during JES2 initialization, default parameters established during JES2 generation and initialization are used.

During JES2 processing, devices can be activated via the JES2 START command (S) and deactivated via the JES2 STOP command (P), resulting in dynamic allocation or deallocation of the device.

Internal Reader

During JES2 generation, the maximum number of jobstreams that can be simultaneously entered through the internal reader facility is specified (&NUMINRS). To JES2 it appears that this number of devices is attached. Actually, there is one internal reader facility. Refer to "The Internal Reader Facility" in the section describing job submission.

During JES2 initialization, the internal reader facility is defined with the INTRDR parameter.

The maximum specified by the &NUMINRS parameter does not apply to the system internal readers associated with the time-sharing LOGONs (TSUINRDR) and started tasks (STCINRDR).

Remote Line and Device Configuration

Configuring RJE devices consists of defining both teleprocessing lines and remote station facilities. The remote facility can range from one remote terminal (2270,3780), to a remote workstation consisting of a computer operating many devices including (optionally) a remote console under the control of a JES2 remote terminal program and communicating with JES2 via the MULTI-LEAVING technique.

Teleprocessing lines are either dedicated (permanently attached) to a single station, or nondedicated which implies that multiple stations can use the line. JES2 does not support multiple active remote stations on one line.

During JES2 generation, the system programmer:

- Specifies the maximum number of teleprocessing lines supported by JES2. &NUMLNES parameter.
- Specifies the maximum number of remote stations supported by JES2. &NUMRJE parameter.
- Specifies the maximum number of remote card readers, remote printers, and remote punches simultaneously supported by JES2. This includes devices that make up remote terminals and devices attached to remote work stations. &NUMTPRD, &NUMTPPR, &NUMTPPU parameters.
- Specifies inclusion of support in JES2 for any remote stations supported with the &BSCCPU, &BSC2770, &BSC2780, and &BSC3780 parameters.
- Generates the remote terminal programs by specifying RMTGEN parameters for each remote workstation.

During JES2 initialization, the system programmer can identify and specify characteristics for each line, remote station, and remote device with the LINEnnn, RMTnnn, Rnnn.RDm, Rnnn.PRm, and Rnnn.PUm parameters. A line is dedicated to a remote station by designating the line number in the RMTnnn parameter. Any line not designated by any RMTnnn parameter is a nondedicated line.

The remote station operator can control the remote station and the remote station devices, jobs submitted through the remote station, or data routed to it. This control can be effected through the remote console or the remote card reader. Each remote station is considered an extension of the local JES2 facility.

For more detailed information about remote devices controlled by JES2, refer to the chapter "Remote Job Entry."

Spool Configuration

JES2 uses the SYS1.HASPACE data set on each volume identified as a spool volume to store all job input, job output, JES2 control blocks, and system data such as the job journal. Spool volumes are identified to JES2 by volume serial number. A six-character name identifying the primary spool volume is specified in the &SPOOL parameter during JES2 generation. An &SPOOL parameter can be used during JES2 initialization to override the JES2 generation parameter. The primary spool volume must exist during JES2 initialization.

Each volume with a volume serial number matching the first five characters of the &SPOOL parameter is considered a spool volume by JES2, and is searched for a SYS1.HASPACE data set. The maximum number of volumes that can be used as spool volumes is specified during JES2 generation with the &NUMDA parameter.

The system programmer also specifies the manner in which the tracks of the volumes are allocated and subdivided into physical records, by specifying the &NUMTGV and &BUFSIZE parameters. These parameters can be specified only during JES generation.

For further discussion of the JES2 spool volumes, refer to the chapter "JES2 Performance."

JES2 also requires one SYS1.HASPCKPT data set on a direct access volume to store a copy of the JES2 queue and other information needed for warm start. This data set may be on the primary spool volume or on another volume as specified by the &CHKPT parameter during JES2 initialization. See the chapter "Installing JES2" for a description of how to allocate this data set and the SYS1.HASPACE data sets.

Starting or Stopping JES2

JES2 and the parameters that define its operation are selected during job entry subsystem initialization. Initialization of VS2 and of the job entry subsystem JES2 are two distinct processes. Basically this means that those processes associated with initialization (coldstart, warmstart) are specified separately for VS2 (reloading the LPA, clearing the VIO data sets) and for JES2 (initializing the job queues). Although for any given IPL it is possible to coldstart one (VS2 or JES2) and warmstart the other, the usual procedure is a warmstart for both. A detailed description of the process and options for an IPL generally, and for starting and stopping JES2 particularly, is located in the chapter "JES2 Initialization."

Controlling Job Submission and Queuing

Jobs are submitted through the job entry subsystem and queued in priority order. The system programmer can use various parameters and facilities to control input streams, to control the specificiation of job classes and generation of priorities for jobs, to hold or release jobs, to set the default performance group for a job, and to change these specifications by changing entries in the JES2 job control table using the JES2 job statement accounting field scan exit.

Submitting Jobs

Jobs are submitted to JES2 in three ways:

- through card readers allocated to JES2
- through RJE devices allocated to JES2
- through a JES2 internal reader facility

Local Device Submission

Local card readers can be supported via the JES2 automatic starting facility by specifying AUTO on the READERnn JES2 initialization parameter. Jobstreams can then be entered simply by readying the card reader. No other operator action is necessary. The card reader is deactivated and deallocated from JES2 with the JES2 stop ($P) command.

Remote Job Submission

The RJE support is described in a separate chapter ("Remote Job Entry"). It should be noted that JES2 processes remote jobs no differently from those received from local card readers or the internal reader facility.

The Internal Reader Facility

An internal reader jobstream is identified to JES2 by the fact that an output data set specifying a special user writer (INTRDR) has been allocated dynamically, or via SYSOUT=(x,INTRDR) coded on a DD card. JES2 recognizes such data sets and places them in the input stream, thus allowing jobs and system tasks to enter jobs in the input stream.

A job entered through an internal reader is delimited beginning with a //JOB statement and ending with the next //JOB statement, a /*EOF statement, or the closing of the internal reader data set. Abnormal closing or closing after a WRITE error causes deletion of the last job. A /*DEL statement may be used to explicitly delete the last job.

The class to which the internal reader data set is allocated (for example, class X if SYSOUT=(X,INTRDR)) becomes the MSGCLASS for the submitted job unless the JOB statement contains a MSGCLASS parameter. If the internal reader data set is dynamically allocated without a class specified, the MSGCLASS of the submitting job or TSO user becomes the default. Two exceptions to this are time sharing LOGONS and started tasks. These are assigned the TSUMCLASS and STCMCLASS JES2 initialization parameter values. The DEST parameter, if specified for the internal reader allocation, becomes the default print data destination for all jobs submitted via that internal reader.

JES2 provides the capability of receiving multiple jobs simultaneously via the internal reader facility. The system uses it to pass started tasks, TSO logon, and TSO background jobs to JES2. Also, jobstreams can be read from tape and disk (any QSAM-supported device) and submitted through the internal reader via the RDR procedure, and any job executing in MVS can use the internal reader facility to pass a jobstream to JES2.

Controlling the Internal Reader Facility: Although the internal reader facility appears to JES2 logically as multiple input devices (maximum specified the &NUMINRS parameter during JES2 generation), the facility is controlled as one entity. The number (&NUMINRS) of internal readers is the number of jobs that can be received simultaneously through this facility.

Characteristics of the facility are specified during JES2 initialization as subparameters of the INTRDR parameter.

Using the RDR Procedure: The procedure supplied by IBM for using the internal reader facility to read jobs from tape or disk is named RDR. The starter system provides the RDR procedure in SYS1.PROCLIB to allow the operator to start JES2 generation (see Figure 4-1). Basically the same procedure can be used to read a jobstream from any QSAM-supported device. The operator uses the RDR procedure as follows:

- To read a jobstream from the second file of a tape named JOBTAP on device 180:

  STARTRDR,180,JOBTAP,LABEL=2,DSN=JOBS

- To read a jobstream from a cataloged library of jobs:

  STARTRDR,3330,DSN=PRODUCTN(PAYROLL)

- To read a jobstream starting with a specific job on a tape named JOBTAP, the operator must submit a job to JES2:

```
//READJOBx JOB      ......
//         EXEC     RDR
//IEFRDER  DD       DSN=JOBS,VOL=SER=JOBTAP,UNIT=3400,DISP=OLD
//SYSIN    DD       *
           EDIT     START=JOBx
/*
```

```
//IEFPROC     EXEC    PGM = IEBEDIT
//SYSUT1      DD      DDNAME = IEFRDER
//IEFRDER     DD      DSN = NULLFILE,DISP = OLD
//SYSUT2      DD      SYSOUT = (A,INTRDR)
//SYSPRINT    DD      SYSOUT = A
//SYSIN       DD      DUMMY
```

Figure 4-1. The RDR Procedure

The system programmer can define internal readers on EXEC statements in such a manner that they are started conditionally. This allows the formation of a set of dependent jobs that can execute without operator intervention. For example:

- To submit JOBB and JOBC if the first four steps of JOBA complete successfully.

```
//JOBA       JOB      ...
//STEP1      EXEC     ...
             –
             –
             –
             –
//STEP5      EXEC     RDR,COND=(8,LE)
//IEFRDER    DD       DSN=JOBS(JOBB),DISP=SHR
//           DD       DSN=JOBS(JOBC),DISP=SHR
```

- To submit JOBB if JOBA terminates normally, JOBC if it terminates abnormally, and JOBD in either case.

```
//JOBA       JOB      .....
//STEP1      EXEC     ....
             –
             –
             –
//STEPN      EXEC     RDR
//IEFRDER    DD       DSN=JOBS(JOBB),DISP=SHR
//STEPN1     EXEC     RDR,COND=ONLY
//IEFRDER    DD       DSN=JOBS(JOBC),DISP=SHR
//STEPN2     EXEC     RDR,COND=EVEN
//IEFRDER    DD       DSN=JOBS(JOBD),DISP=SHR
```

Installation-written procedures and programs can further exploit the internal reader facility to select particular jobs, to generate special job streams, and to allow operator submission of production jobstreams.

Controlling Job Enqueuing

In JES2, jobs are enqueued by priority sequence and, when ready for execution, within individual classes. The system programmer can control job selection through determining job class and priority.

The JES2 Queue

A job received by JES2 is enqueued in job priority order on the JES2 queue residing in the JES2 address space in main storage. A job is considered received by JES2 when it has been totally read in and the JES2 control blocks placed on the spool.

The queue entry for each job contains the job name, job priority, a flag to indicate the job is held, pointers to JES2 control blocks on the spool, and the JES2 process (JCL conversion, execution, output processing, purging) for which the job is next eligible. Jobs are selected in priority order for each JES2 process. Logically, the one JES2 job queue has queues for each process, plus 38 (one for each class) within the execution process.

A job which is held is not removed from the queue; instead it is made ineligible to be selected for any JES2 processing. A job can be held at any time. Thus a job in execution that is held by the operator, is not eligible for output processing until released. The holding of a job that is read for execution can occur by job, by class, or by holding all jobs.

Job Class

There are 38 classes of jobs possible under JES2. Two are used by the system: STC for started task control, TSU for time sharing logon. The other 36 classes, A-Z and 0-9, are for normal jobs and can be used to help control the job mix.

The job class is specified on the JOB statement (CLASS=jobclass). If not specified, a default based on the particular device through which the job is entered into JES2 is assigned. All jobs entered through the internal reader facility are considered to be entered through a device described by the INTRDR parameter.

There are no absolute rules for assigning job classes, and some experimentation is necessary. Generally, jobs of similar characteristics and specifying identical processing requirements should be assigned to the same class. For example, if several jobs are time-dependent and execute in nonpageable dynamic storage, it may not be desirable to tie up all of nonpageable dynamic storage by having these jobs running concurrently. These jobs may all be assigned to class B (or C or D—class names have no inherent meaning); then, if only one initiator is started that can handle class B jobs, there will never be more than one of these jobs executing at once.

Suppose the following assignments are made:

Class B = jobs that are time-dependent.
Class C = jobs with high CPU requirements.
Class D = jobs with high I/O requirements.

The system programmer can specify initiator parameters such as:

I1	CLASS=BCD
I2	CLASS=CDB
I3	CLASS=DCB

If the three initiators are processing jobs with the same priority and all necessary resources (for example, I/O devices and data sets) are available, then three jobs, one from each of the three different classes, run concurrently. If a job within one of the classes has higher priority than the others in the class, it will be initiated first.

During JES2 initialization, the system programmer can assign job characteristics to jobs enqueued in each class. Characteristics that can be assigned are:

- A default performance group for each job.
- JCL conversion parameters.
- Whether a JES2 job log is to be produced for jobs in this class. The JES2 job log is a list of all messages and replies issued by, or on behalf of, a job.
- Whether a system journal is to be saved for this job. If it is not saved, the overhead is avoided but the job may not be automatically restarted in case of job failure or system restart.
- Whether this class is reserved for the execution batch scheduling facility. (See Execution Batch Scheduling.)
- Whether output is suppressed for jobs in this class (for example, started tasks).
- Define the procedure library (PROCnn).
- SMF options.
- Whether the job is held.

- Whether the job is to be simply copied to message class output or converted but not executed.

The system programmer should assign separate job classes to jobs that are to be assigned separate characteristics, and to jobs that have different execution characteristics such as:

- rate of CPU to I/O processing
- use of special devices
- number of devices used
- use of real storage

JES2 Job Scheduling Priority

Job priority is determined through use of the PRIORITY statement, or by an algorithm that uses programmer-supplied or generation-defaulted job characteristic data.

In addition an increment can be added to the priority and a priority limit enforced, depending on the device through which the job entered JES2. Those parameters are associated with the input device during JES2 initialization.

Specifying Priority: Priority is specified on the /*PRIORITY statement. If specified, it must immediately precede the JOB statement, or the input stream is flushed until another JOB or PRIORITY statement is found.

Calculating Priority: When the scheduling priority is not specified on the /*PRIORITY statement, priority is calculated using estimated execution time and the estimated number of output lines and cards. These values can be entered on the JOBPARM statement, in the accounting information on the JOB card, or can be entered or defaulted with JES2 generation parameters ($ESTIME, $ESTLNCT, $ESTPUN).

To calculate priority, these estimates are used in conjunction with four JES2 generation parameters, each of which is supplied (or defaulted) as a table of values. These are the &XLIN(m), &RPRT(n), &RPRI(n), &XPRI(m) parameters. The default values for these tables are used for the examples of priority calculation that follow. The default values are:

For &XLIN:
(&XLIN values are estimates of the number of output lines and cards for a job.)

&XLIN(m)	**m**
2000	1
5000	2
15000	3
2^{24}-1	4
⋮	⋮
2^{24}-1	9

For &RPRT:
(&RPRT values are estimates of the number of minutes a job will take to run.)

&RPRT(n)	**n**
2	1
5	2
15	3
2^{24}-1	4
⋮	⋮
2^{24}-1	9

For both &XPRI and &RPRI:
(The m and n values are determined from the &XLIN and &RPRT tables.) Note that the &XPRI and &RPRI tables can have two different sets of values. The values described here are the default values.

m or n	**&XPRI(m) or &RPRI(n)**
1	9
2	8
3	7
⋮	⋮
9	1

Priority is calculated by using the values specified for &XLIN and &RPRT to determine *m* and *n* from the table. These *m* and *n* values are then used with the &XPRI and &XRPRI tables to determine values for &XPRI(m) and &RPRI(n). Priority is then calculated as:

PRIORITY=[&XPRI(m)+&RPRI(n)] /2

The following examples illustrate the use of the various parameters in this calculation.

Example 1. The programmer estimates 2000 lines plus cards, 2 minutes execution time.

From &XLIN, 2000 lines implies m=1
From &RPRT, 2 minutes implies n=1
From &XPRI, m=1 implies &XPRI=9
From &RPRI, n=1 implies &RPRI=9

Therefore PRIORITY=(9+9)/2=9

Example 2. The programmer estimates 4000 lines plus cards, 15 minutes execution time.

From &XLIN, 4000 lines implies m=2
From &RPRT, 15 minutes implies n=3
From &XPRI, m=2 implies &XPRI=8
From &RPRI, n=3 implies &RPRI=7

Therefore PRIORITY=(8+7)/2=7

(The fraction is ignored; only the integer value is used.)

Example 3. The programmer estimates 15,000 lines plus cards, 10 minutes execution time.

From &XLIN, 15,000 lines implies m=3
From &RPRT, 10 minutes implies n=3
From &XPRI, m=3 implies &XPRI=7
From &RPRI, n=3 implies &RPRI=7

Therefore PRIORITY=(7+7)/2=7

If priority is computed for the job during input, it is recomputed for output. Output priority is based on the count of lines and cards (&XLIN) actually produced during job execution. The execution time parameter &RPRT is ignored.

The system programmer, by specifying other values for the tables during JES2 generation, can more closely control priority specification. Values specified on the JOBPARM statement supercede those in the account field of the JOB statement. During JES2 generation the &RJOBOPT parameter can be specified to ignore the account field on the JOB card.

Priority Aging

The priority of a job can be increased as a function of the length of time that it has been in the system. The &PRIHIGH, &PRILOW, and &PRIRATE JES2 generation parameters specify respectively a limit above which there is no incrementing, a limit below which there is no incrementing, and an integer representing the number of times that the priority is incremented in a 24-hour period—subject to the upper limit. The default of zero for the &PIRRATE parameter specifies no priority aging.

The PRIRATE parameter specifies whether the feature is used and, if so, how many times in a 24 hour period the priority is incremented. For example, PRIRATE=48 specifies a priority increment of one unit every 30 minutes. The PRIHIGH parameter specifies the upper limit; a priority lower than the value of the PRILOW parameter specifies that the job is not subject to priority aging.

The JOB Statement Accounting Field Scan

During JES2 generation the &RJOBOPT parameter is specified to determine whether JES2 is to scan the account field of the JOB statement, and to set the conditions under which JCL scan errors cause job termination prior to JCL conversion. Figure 4-2 describes the &RJOBOPT options.

The account field is considered valid if its format is that specified for HASP II. The format is assumed to be as follows:

(pano,room,time,lines,cards,forms,copies,log,linect)

pano

programmmer's accounting number. One to four alphameric characters.

room

programmer's room number. One to four alphameric characters.

time

time estimated execution time in minutes. Up to four numeric digits (example: ",30" for 30 minutes). If omitted, a standard value is assumed.

lines

estimated line count in thousands of lines. Up to four numeric digits (example: ",5" for 500 lines). If omitted, a standard number of lines is assumed.

cards

estimated number of cards to be punched. Up to four numeric digits. If omitted, a standard number of cards is assumed.

forms

special forms for printing entire job. From one to four alphameric characters (example: ",5" for 5-part forms). If omitted, standard forms are assumed.

copies

number of times output is to be printed or punched. Up to three numeric digits not exceeding an installation-specified limit (maximum 255) (example: ",2" for two copies). If omitted, one copy is assumed.

&RJOBOPT	Scan Account Field	Terminate if Account Field Invalid	Terminate if JCL Invalid
0	Yes	Yes[1]	Yes
1	Yes	Yes[1]	No
2	Yes	No	Yes
3	Yes	No	No
4	No	-	Yes
5	No	-	No

[1] If these values are indicated, the first two subfields of the account field (pano and room) are required.

Figure 4-2. Job Statement Accounting Field Scan Exit

log

Job Log option. This subfield should consist of one character. If this character is an "N", the HASP Job Log is not produced. If any other character, or omitted, the log is produced.

linect

to be printed per page. Up to three numeric digits not exceeding 255. If coded as zero, no automatic overflow is produced. If omitted, a standard value is assumed.

The JCL scan is not exhaustive; only JOB, DD *, and DD DATA statements are scanned. Job termination on a JCL error at this point does not guarantee that all JCL errors have been found. If the job is not terminated on a JCL error at this point in the process, it can still fail during JCL conversion when all JCL is scanned.

Job Statement Accounting Field Scan Exit: A routine can be written that allows the installation to control a job by modifying data in the JES2 Job Control Table (JCT) immediately following the scan of the user's JOB statement. (Figure 4-3 shows selected fields from the JCT.)

The name of the CSECT must be HASPRSCN and must replace the existing CSECT of that name in the HASJES20 load module. This routine receives control from JES2 with registers set as follows:

R0 A binary number giving the length (in bytes) of the accounting field
R1 Address of the accounting field from the JOB card
R2 Address of the SMF job management record (as defined in the description of the VS2 Common Exit Parameter Area, *OS/VS System Management Facilities*
R8 Addressability
R10 Address of the (JES2) JCT
R13 Save area (18 words)
R14 Return address

Registers 3-14 must be restored when control is returned to JES2. The save area addressed by R13 can be used to store registers.

There are no return codes necessary.

Programming Notes: Parameters of the JOBPARM statement or ROUTE statement can override changes made with this routine. Otherwise the change is effective for the duration of the job.

Data placed in the user identification field of the SMF record at this time, is available to the programmer at all SMF exits.

If system services that have implied WAITs (for example, WTO the SMF WTM) are used by this routine, severe system degradation may occur.

Since this routine runs under the JES2 task, if abnormal termination occurs the system must be restarted.

Controlling Conversion and Execution

The JCL for a job, logon, or started task is passed through the converter and converted into internal text. The job is then available for execution, which occurs as soon as an initiator eligible to process the job is available.

Displacement (hex) in JCT	Length (bytes)	Notes	Field	
8D	1		SMF Flags:	
		-	bit 0-1	Reserved
		1,2	2	If set, IEFUSO exit not taken
		-	3-4	Reserved
		1,2	5	If set, no Type 6 SMF records produced
		1,2	6	If set, IEFUJP exit not taken
		1,2	7	If set, no Type 26 SMF record produced
8E	1		Job Flags:	
		-	bit 0	Background job
		-	1	TSO (foreground) job
		-	2	Started task
		1,2	3	No job journalling
		1,2	4	No output
		1,2,3	5	TYPRUN=SCAN
		2,3	6	TYPRUN=COPY
		-	7	Reserved
8F	1		Job Options:	
		-	bit 0	/*PRIORITY card was read, value is in Priority field (B6)
		-	1	/*SETUP card was read
		1,2,4	2	TYPRUN=HOLD was specified
		1.2,6,8	3	No job log for this job
		1,2	4	Execution batch job
		-	5	The job was read through an Internal Reader
		-	6-7	Reserved
90	8	-	JES2 JOB identifier	
98	8	3	Job name	
A0	20	3	Programmer name	
B4	1	1,4	Message class	
B5	1	1,4	Job class	
B6	1	1,5	Priority	
.				
.				
BA	2	-	Route code of input device	
BC	8	-	Input device name	
C4	4	1,6	Account number (for HASP compatability)	
C8	4	1,6,8	Room number	
CC	4	1,6,8	Estimated real time job will run	
D0	4	1,6,8	Estimated count of output lines (in thousands)	
D4	4	1,6,8	Estimated number of output cards punched	
D8	4	1,6,8	Job Forms	
DD	1	1,6,8	Job copy count (binary)	
DF	1	1,6,8	Lines per page (binary)	
E0	2	1,7	Default print routing (binary)	
E2	2	1,7	Default punch routing (binary)	
			0	Any local device
			1-999	Remote devices
			1001-1999	Specific local devices
EC	8	1,2,8	Procedure DD name	

Notes:

1. Can be modified by installation routine.
2. Preset from $X initialization parameter according to job class.
3. Preset from JOB statement.
4. From JOB statement, if specified; otherwise according to input device as established at JES2 initialization (e.g. in READERn).
5. Preset from /*PRIORITY statement or an "*"
6. The HASPRSCN routine is used by JES2 to scan the account field of the JOB statement. If the HASPRSCN routine is replaced by an installation-written routine, the account field is empty.
7. Preset according to an input device initialization parameter (e.g. READERn). If not set at initialization the parameter defaults to the job input source value (LOCAL or RMTnnn). Can be modified by a ROUTE statement after the scan exit.
8. Can be modified by a JOBPARM statement after the scan exit.

Figure 4-3. Selected JES2 Job Control Table Fields

JCL Conversion

A job is eligible for JCL conversion as soon as it is placed on the queue. The converter is invoked separately for each job. JES2 passes to the converter, the converter parameters and a pointer to a catalog procedure library.

Converter Parameters

If not defaulted (&RDROPSU, &RDROPST, &RDROPSL JES2 generation parameters), the converter parameters are specified for each class on the CONVPARM subparameter of the &STC, &TSU, or &X parameters during JES2 initialization. Converter parameters specify defaults such as execution time estimate, and JCL and allocation MSGLEVEL options. Command disposition and authority and the bypass label options are specified. The specific parameters are described in the chapter "JES2 Initialization."

Procedure Library Selection

The JES2 Procedure is located in SYS1.PROCLIB. It defines job-related procedure libraries such as:

```
//PROC00    DD     ...
//PROC01    DD     ...
.
.
.
//PROCnn    DD     ...
//anyname   DD     ...
```

The programmer can specify any of the libraries included in the JES2 procedure on the JOBPARM statement by specifying the library DDNAME. If multiple data sets are required they must be specified as concatenations in the JES2 Cataloged Procedure.

If there is no procedure specification on the JOBPARM statement, class-related initialization parameters can specify the library as PROCnn. If the procedure is not specified or specified and not found, PROC00 is used.

Execution Control

Execution is controlled through controlling the initiators and the jobs on the queue (see Controlling Job Enqueuing), as well as by monitoring the job and issuing commands.

JES2 associates one logical initiator residing in JES2, with each system initiator interfacing with JES2. The maximum number of logical initiators is specified during JES2 generation (&MAXPART parameter). The number of active logical initiators, subject to the maximum, is controlled by the operator ($S Inn). The operator can also associate with logical initiators the order in which the classes are selected by JES2.

Classes are associated with each initiator during JES2 initialization or dynamically by the operator. During execution, the initiator selects non-held jobs in priority order within their class, and the non-held class in the order specified for that initiator. That is, the lowest priority job in the first non-empty class is selected ahead of the highest priority job of the next class—assuming neither job nor class is held.

The Initiator Cataloged Procedure

One initiator cataloged procedure (INIT) must be contained in SYS1.PROCLIB for use by JES2 in creating job address spaces into which a system initiator is initialized. JES2 uses the START command (system command) to create one system initiator for each active JES2 logical initiator. The number of active initiators must be controlled by starting and stopping JES2 logical initiators.

The standard initiator cataloged procedure supplied by IBM is named INIT. The procedure is:

```
&//IEFPROC&EXEC&PGM=IEFIIC,DPRTY=12
```

Job Monitoring

A job can be monitored by elapsed (wall clock) time, execution time, and by output in terms of lines and cards.

During JES2 generation, the &TIMEOPT parameter can be specified to cause JES2 to write a message to the operator when the elasped time specified on the JOBPARM statement is exceeded, and an additional message at each interval specified by the &TIMEXS parameter. The system programmer can use the SMF accounting exit to enforce these values, if the time was placed in the SMF userid field during the JCL scan exit. The SMF exits are described in *OS/VS2 System Management Facilities (SMF)*.

Execution time can be specified on either the EXEC statement or the JOB statement, or in the converter (CONVPARM) parameters. If the time is exceeded, the SMF exit is entered and the job can be cancelled or continued.

The &OUTXS JES2 generation parameter is used to specify the total number of printed lines and punched cards that a job can print before action is taken by JES2. The &OUTPOPT parameter specifies the action that is taken. The job can be allowed to continue after a message is written to the operator, or the job can be cancelled with or without a dump.

The installation can specify SMF output limiting by class, with the &x, &STC, and &TSU JES2 generation parameters. Output (OUTLIM DD) can be monitored for each data set by SMF.

Entering Commands in the Jobstream

JES2 commands and standard system commands are accepted at different points in a jobstream, with different types of control. A jobstream is defined as the set of jobs submitted between the physical start of a reader and physical end-of-file, or between the opening and closing of an internal reader data set. Refer to Figure 4-4 for a pictorial representation of the following paragraph.

JES2 commands are accepted in the jobstream only if they are in front of the first //JOB statement of a jobstream. The commands that are accepted from any given device are controlled by a command authority associated with the device ($T command). The command authority associated allows various combinations of display and system, job, or device control commands to be entered. JES2 commands are in the form /*$command.

System commands in the form /*$VS 'systemcommand' are accepted in front of the JOB statement, subject to the same authority described for JES2 commands.

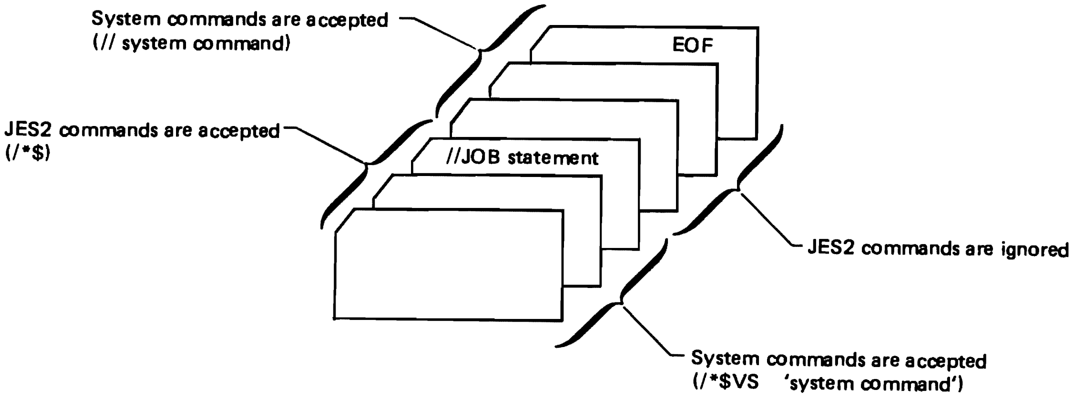
System commands are accepted
(// system command)
EOF
JES2 commands are accepted
(/*$)
//JOB statement
JES2 commands are ignored
System commands are accepted
(/*$VS 'system command')

Figure 4-4. Entering Commands in the Jobstream

JES2 commands found in the jobstream between the first JOB statement and EOF are ignored.

System commands (//system cmd) which appear in the jobstream after the first Job statement, are executed in the converter. Whether they are issued is subject to control by the converter via the converter parameters. System commands appearing before the first JOB statement are ignored.

Execution Batch Scheduling

Execution batch scheduling is an extension of normal job scheduling that may provide improved system performance. It is the process of gathering psuedo-jobs, called execution batch jobs, into a single input stream for processing by an execution batch processing program. The execution batch jobs are submitted to JES2 one at a time; they may have different input sources, and different print and punch output routing. Execution batch scheduling collects these numerous related batch jobs into a single data stream and passes them as a SYSIN data set to the user-written execution batch processing program. This reduces the overhead associated with setting up for, and processing, numerous individual jobs and/or job steps. Another advantage is that individual accounting for all but type 4 and 5 records is available.

The processing programs to be used with the execution batch scheduling feature may cover a wide variety of application areas such as:

- Compile-and-go debugging compilers
- File inquiry programs
- Hardware or software system emulators

It is desirable that the program process jobs or transactions of relatively short duration. If not, the saving in job management overhead between successive jobs may not be a large enough percentage of total job execution time to justify use of this feature.

At JES2 initialization, the installation defines the job class or classes that are to be dedicated to execution batch scheduling. One class or group of classes is assigned to each type of execution batch processing program. Subsequently, the batch user identifies the program requested by the class stated on the JOB statement.

JES2 can support more than one execution batch processing program to process various kinds of batch jobs. Each execution batch processing program must be associated with at least one JES2 initiator. The system initiators request a job, any job, and JES2 decides which job is to be processed.

To determine which jobs are to be processed by an execution batch processing program, JES2 recognizes jobs assigned to eligible classes. Instead of sending these jobs directly to an initiator, JES2 invokes an appropriate procedure from PROCLIB to initiate the execution batch processing program. The job as submitted is now considered part of the input data of the execution batch processing program.

For example, consider an order entry system that requires an inventory update and an invoice for each order. With standard processing, the normal procedure would be to batch all orders and submit them as a data stream at the end of the day to an order entry system.

However, this causes delay. Alternatively, the installation can periodically batch together orders received during a certain time period and run the job several times a day. By using the execution batch scheduling facility, orders can be processed as if the order processing program were scheduled for every order, but without the overhead of scheduling the order program for the runs.

The order entry program would become an execution batch processing program. The orders themselves would be submitted as execution batch jobs by taking the order data that would have been submitted in batch, and putting a system JOB statement in front of it. In an order entry program accustomed to reading batch jobs at the end of the day, the only programming changes would be (1) to use the ddname SYSIN for the input stream (this may be accomplished by JCL in PROCLIB), (2) to recognize the null statement as an order separator or establish other defined terminators, and (3) ignore all other JCL (//) cards. The program would have to print all related information for each order before processing the next order, to distinguish one from another. JES2 automatically schedules the order entry program when it is needed and concatenates all orders into the input stream data regardless of where they have originated.

Submitting Input to an Execution Batch Processing Program

A representative input stream follows:

```
// JOB
   JES2 control statements
      input
```

To submit data to an execution batch processing program, follow these rules:

- The first statement of each job must be a standard JOB statement that includes a CLASS=job class parameter. The job class identifies which program is to receive the input. The installation associates job classes with an execution batch facility via the procedure library. It associates job classes with initiators at initialization. The accounting field is interpreted by JES2 just as it is for normal jobs.
- All JES2 control statements are effective with batching jobs except /*OUTPUT, which is ignored.
- No other JCL is used. All other statements are input to the execution batch processing program. These can be read just as if they had been placed in a DD DATA data set and the execution batch program has been invoked by standard JCL. If the execution batch program requires it, each transaction can be terminated by a statement with in columns 1 and 2.

In the order entry system example mentioned earlier, code the following:

```
//JOBxx     JOB       (INVO1,667),CLASS=X
/*ROUTE     PRINT     RMT47
order 1
order 2
```

The /*ROUTE statement will cause the invoice to be printed at the remote location.

Execution Batch Scheduling Operations

Special actions take place when JES2 recognizes input for an execution batch program.

If the execution batch program is not already active, JES2 submits an internal job which uses JCL from SYS1.PROCLIB to invoke the execution batch processing program when an initiator capable of processing it becomes available. JES2 control cards are converted to JCL comment statements. The entire input, plus a JCL null statement added by JES2, is allocated to the execution batch processing program as an input data set with the ddname of SYSIN.

If the execution batch program is already active and simply waiting for another job, JES2 makes the input data set allocation as above, and processing begins immediately without any use of job management.

The end of input can be detected by the execution batch program when it reads the JCL null statement added by JES2. After writing any remaining SYSOUT data for the completed job, the execution batch program attempts to read ahead in its input file for another transaction. JES2 detects this condition, temporarily forces the execution batch program into a wait state, and performs job termination actions for the execution batch job (flushes output buffers, releases input spool space, queues the job for printing, and so forth). The execution batch program remains active in the MVS address space.

When an execution batch program is waiting, JES2 job selection is altered. Instead of scanning for all classes eligible to execute in that address space, JES2 first tries to start an execution batch job which may be processed by that execution batch program. If successful, processing can begin immediately.

If no jobs of the same execution batch class are available to execute, all other job classes of the address space are scanned in order. If a job is found, JES2 internally cancels the execution batch processing program and normal scheduling, using job management, takes place.

If no jobs of the other classes are found, the address space and execution batch processing program remain idle, awaiting availability of a job in any of its classes. If a job becomes available in the class of the execution batch program still in the address space, processing begins immediately.

If an execution batch program ends (ABEND or normal return to VS), JES2 detects this as a nonbatch termination in the address space. Job management will be used to reinvoke the batch program when another job for its class is selected.

Use of the operator commands $P I or $P In will cause JES2 to cancel an execution batch processing program when it becomes idle, and then delete the address space.

In summary, an execution batch processing program must have certain characteristics:

- It must read all user input from a single sequential data set.
- It must recognize a standard OS JOB statement, or its own control statement, to determine the beginning of a job.
- It must recognize a standard OS null JCL statement (// followed by 78 blanks), or its own control statement, to determine the end of a job.
- To ensure system integrity, it should *not* use dynamic allocation of SYSOUT data sets.

The execution batch processing program will receive an end-of-file condition when a card with $$ in columns 1 and 2 is read while processing a job. The program may continue to the next logical subfile by simply resetting appropriate bits in I/O control blocks and continuing reading, or by closing and reopening the data set to continue reading at the card following the $$ card.

Preparing for Execution Batch Scheduling

The batching feature is included in JES2 by setting the &XBATCH=YES parameter during JES2 generation. Job classes are reserved for execution batch jobs with the $$x initialization parameter. The &BATCHN JES2 generation parameter may be set to define the first five characters of the catalog procedure name that contains the JCL necessary to execute an execution batch program. (See the chapter "Installing JES2" for a description of this parameter.)

Each batch class should be associated with one execution batch program. Each batch class should be made eligible to execute in the MVS address space by setting the Inn initialization parameter or by using the $T In operator command.

For each combination of batch class and initiator, there must be a procedure in SYS1.PROCLIB named "nnnnncid", where

- nnnnn are the five characters assigned to &XBATCHN.
- c is the particular batch job class set in $$x.
- id is the 1- or 2-character initiator identification, corresponding to nn of the Inn parameter

These procedures actually call the execution batch program for each class, and define all data sets other than the user input data set.

The procedures may be single step, or may have preliminary steps before the single step that invokes the execution batch program (stepname GO). The execution batch program invoked by this step must read its input from a SYSIN, or the procedure must refer to DDNAME=SYSIN on a DD statement used for input by the processing program.

If a given batch class is eligible (the Inn initialization parameter or $T In operator command defines eligible classes) to be executed by more than one initiator, the requirement for a separate procedure name for each address space/class combination may be satisfied by alias names of a single procedure, or by actual separate procedures which can specify different work fields.

The following example shows the internal job that JES2 generates to initially load a program to process batch class X jobs for Init=3, assuming the default setting for &XBATCHN.

```
//$$$$$X3    JOB        1,SYS,MSGLEVEL=1
//FAKE       EXEC       $$$$$X3
//GO.SYSIN   DD         DATA,DCB=BUFNO=1
//
```

The following is an example of a procedure that an installation might use for a simple file inquiry program that reads inquiry input from SYSIN, checks a file, and prints responses to SYSPRINT.

```
//$$$$$X3    PROC
//GO         EXEC       PGM=FINDPART
//SYSPRINT   DD         SYSOUT=A
//PARTFILE   DD         DSN=PARTFILE.MASTER,DISP=SHR
//SYSUDUMP   DD         SYSOUT=A
```

The following JCL is for the order entry system example.

```
//$$$$$X3    PROC
//MDSE       EXEC       PGM=ORDERIN
//MESSAGE    DD         SYSOUT=M
//INVOICE    DD         SYSOUT=(P,,INVC)
//INVTRY     DD         DSN=MSTRINVT,DISP=SHR
//ORDERS     DD         DSN=ORDERS,DISP=MOD
```

- //MESSAGE–the installation might identify class M as a punch class. This will allow the submitter of the execution batch job to route the invoices and messages separately, as shown in the example in "Submitting Input to an Execution Batch Processing Program".

- //INVOICE–defines the specially prepared output.
- //INVTRY–uses a master inventory list as a base; it is updated as the orders are received.
- //ORDERS–accumulates the day's orders. ORDERS has a disposition of MOD because the execution batch processing program is periodically started and stopped during the day.
- SYSOUT data sets–the messages and invoices.
- SYSIN data sets–the DD DATA input is every execution batch job that is processed by the execution batch processing program.

Controlling System Output

JES2 provides:

- Queuing levels beyond the simple output class queuing provided by the output writer.
- The ability to specify print train and either carriage tape name or forms control buffers for sysout directed to 3211 and 1403 printers plus support of the 3525 print and interpret features for sysout data sets.
- Features that minimize operator interaction due to forms, carriage tape, and print train loading.
- An external writer facility that, although possible to use for writing any sysout data, is specifically intended for writing to devices other than printers and punches and for controlling all output written by installation-written writers.

Queuing Output

The job output elements JOEs) are created during output processing, or during execution in the case of spinoff, by JES2. Each JOE represents a unit of work to JES2, and is placed in a job output table (JOT) in order of output priority. If the priority was calculated originally, it has now been recalculated with the actual number of lines and cards. See "Calculating Priority".

The JES2 writer and the external writer can select only data sets for which JOEs have been constructed. Varying the number of JOEs, (&NUMJOES JES2 generation parameter) influences the way output is processed. By specifying a large number of JOEs the output processors are given a large number of output data sets from which to choose. This minimizes the setup changes in JES2-controlled printers and punches by providing a series of data sets of the same class for the external writers. However, a given data set may wait a long time for a printer with the specified setup, an available device destination, or for an available external writer to dequeue its class. This long wait may fill spool space, since most of a job's output-related spool space is freed only when all output data sets have been processed. Specifying many JOEs tends to optimize output device utilization at the expense of throughput for a specific job.

Specifying few JOEs tends to reduce the number of jobs with output eligible for printing while processing the entire job output more nearly together. This specification may minimize the turnaround of a particular job at the expense of operational efficiency.

A job output element that does not yet describe a unit of work is said to be "free". The $MINJOE parameter specifies the number of JOEs that must be left to be used when the $I command interrupts an output data set or when a printer is started. When the building of JOEs for a job would drop the number available below the specified minimum, the job or spinoff data is forced to wait until JOEs are available.

JES2 queues output data by combinations of data set characteristics such as output class, forms, print train and forms control buffer name. (Data sets are also queued by installation writer name and destination, as discussed in the "External Writer" section.) These characteristics are obtained from the SYSOUT DD STATEMENT or the JES2 OUTPUT statement. With the exception of held data sets and spinoff data sets, a job's, started task's, or time-sharing user's output that has identical characteristics is queued together in a data set group pointed to by a job output element (JOE). (This queueing can be altered by the demand setup option.) Each held and spinoff data set is queued separately and constitutes a "group" of one data set. Each data set group is considered a processing entity with a set of processing characteristics. JES2 selects work by data set groups and will, if the separator option is specified, delimit each group with separator pages or cards.

Figure 4-5 represents how one JOE can represent one of several sysout data sets.

JOEs built for job-related output are duplicated according to the number of job copies requested on the /*JOBPARM statement. This allows the number of copies being processed for any job to be governed by the devices available for output.

Output Class Assignment

Output from a problem program is assigned to an output class which is processed by JES2. A maximum of 36 sysout classes can be named by specifying SYSOUT=x on the DD statement, where *x* is any single letter (A-Z) or digit (0-9). The names have no inherent meaning; they are simply used to group output of similar characteristics. During JES2 initialization, the fact that a class is designated as containing print or punch data is used for output limiting and job accounting purposes only and has no bearing on the actual device to which the class is assigned.

JES2 writers and external writers are assigned to process only designated classes of output. These classes can initally be assigned to JES2 writers during JES2 initialization, to external writers in the external writer cataloged procedure, or the operator can assign them either as parameters of the START ($S) or SET ($T) commands.

If output is assigned to a class for which no writer is started, it remains indefinitely on the queue.

```
  SYSOUT=(A,2PRT),UCS=PN
  SYSOUT=A,UCS=PN                      Total of four JOEs
  SYSOUT=B                             built
  SYSOUT=A
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  SYSOUT=A
  SYSOUT=A
     .                            Total of three
     .                            JOEs built
     .
  SYSOUT=B
  SYSOUT=B
     .
     .
     .
  SYSOUT=(A,,2PRT)
```

Figure 4-5. Relationship of SYSOUT Specification to Number of Job Output Elements

System Message Classes

System messages generated during the execution of a program must also be routed to an output device; if messages must appear with their program output, they should be assigned to the same message class as the output. But to guarantee that the messages and data appear together, all data sets for the job must be described to JES2 in a single job output element (JOE).

It is also possible for an installation to specify ($DMNDEST parameter) during JES2 generation, that all job output of the same class and for the same destination as its system messages (MSGCLASS), JCL statement images, and job log (if any), be placed in a single JOE. This keeps the data sets together on output listings but can cause operational inefficiencies.

The message class is assigned as a parameter of the job statement. Its format is MSGCLASS=x, where *x* is any single letter (A-Z) or digit (0-9). If no message class is specified, the default class specified for TSUMCLAS, STCMCLAS, or the device through which the job was read, is used.

Output Class Considerations

The system programmer should assign output classes in a manner that distinguishes types of output and results in the most efficient use of devices. JES2 automatically balances output scheduling which makes the assignment of classes less important than in MVT. However, classes should be assigned with the following characteristics:

- Data to be processed by standard JES2 writers should be distinguished from that to be processed by external writers.
- Data placed on different devices and data placed on similar devices but with different characteristics should be in a separate classes. It is not necessary, however, to use classes to separate data with different punch interpretaion UCS and FCB requirements if the data is processed by a JES2 writer, since JES2 handles these parameters automatically. Class should be separate if an external writer is used to print this data.
- Class should be assigned to give different priority to different types of data such as that to be printed on a different work shift.
- Classes need not be specified to give priority to short data sets, since JES2 priority calculation can be used for that purpose.

JES2 Output Selection

When assigning such things as priority classes, and form requirements for data sets, the system programmer should balance the choices against the criteria used by JES2 to select the output data set to be processed.

Setup Characteristics

As established during JES2 initialization and altered by the operator, each JES2 controlled printer and punch possesses setup characteristics and a setup mode—manual or automatic. Setup characteristics are class, destination, forms, print train, and either carriage control tape or forms control buffer. (For a 1403 printer, JES2 uses the FCB parameter as carriage control tape name. The operator is requested to mount carriage tapes by this name in a manner similar to mounting forms.) Setup characteristics determine the data set groups that are eligible for processing on this device. Each locally attached printer and punch posseses either a destination of LOCAL or a specific device-name destination. If the device posseses a specific destination name, it is eligible to process only data sets specifically routed to it by JCL or the operator. Each remotely attached printer and punch posseses a destination that is the name of the workstation to which it is attached or the installation-assigned name of a remote pool of devices. Remote devices are eligible to receive only that output directed to them by JCL or the operator.

Setup mode determines the manner in which data sets are selected, automatically or manually. In automatic mode, after all data sets with characteristics matching the setup for a particular device have been selected, JES2 requests that the operator change the setup for that device.

In manual setup mode (operator-controlled), only data set groups with characteristics matching the setup of the device are selected. Manual mode printers do not request a new setup when there is no more work in the queue. The printer becomes idle.

When a device is available for output, JES2 selects job output elements according to the following algorithm:

1. The setup priority
   - First choice is between JOEs with setup requirements exactly matching those currently on the device.
   - Second choice is a JOE specifying a setup not currently being processed by any output device.
   - Third choice is a JOE specifying the standard forms setup as described by the STDFORM, PRTUCS, and PRTFCB JES2 initialization parameters.
2. When the setup has been selected, the first class specified for this device by the operator, or during JES2 generation, is chosen.
3. When setup and class are selected, the highest output priority JOE with these characteristics is chosen.

Some implication of the setup algorithm are:

- If an output device has been set up explicitly by the operator ($T command), JES2 does not set up another device to process data specifying that setup—unless the explicit setup is the same as that for standard forms. This is true no matter how many devices are idle, unless explicitly set up by the operator.
- Output matching an existing device setup and class is processed before output with no active setup, regardless of relative priority of the jobs producing the output.
- Output with setup requirements not loaded on any device is preferred over output with the setup loaded by the device busy.
- Installations should ensure that class and setup conflicts do not cause data to be overlooked. Commands are provided to determine output backlog.

Demand Setup

For those installations wanting job-related data sets, regardless of setup requirements, to appear together on the output listing, a demand setup option (DMNDSET) can be specified during JES2 generation. The output data sets of the job, possibly with several different setup requirements, are then placed in the same data set group. The message data set is the first one in the group, therefore its characteristics are used as those of the group for setup purposes.

The operator is requested by JES2 to set up the device as different setup requirements become necessary. Responding to demand setup requests is identical to responding to automatic setup requests.

Defaults

Defaults are assumed for any data set characteristics that are not specifically requested on the SYSOUT DD statement or the JES2 OUTPUT statement. Any FCB or UCS image that is specified as default by the installation will be used to print any data set that does not specify an FCB or UCS parameter. JES2 uses the name '****' when requesting from the operator a default FCB or UCS image. The operator satisfies this request by mounting any image specified as a default. If one or more of the parameters (FCB,UCS, form) is not specified, then any default will satisfy it.

The form used for all data sets not specifying a form is identified (STDFORM) during JES2 generation. This is the standard form for both printers and punches.

Operation for Printers (Punches)

An installation generally has one or more printers (and/or punches) in manual setup (operator-controlled) mode for processing output that requires the most common combination (standard setup) of form, print train, and carriage control. The remaining printers are in automatic mode. Initially each printer is assigned setup characteristics, and a set of output classes from which to select data sets.

For each printer, data set groups are dequeued that have characteristics matching the printer setup characteristics. As automatic-mode printers exhaust the queue of data sets specifying their current setup, a data set group with a different set of characteristics is chosen and the operator notified (message HASP190) to change the setup.

An operator can respond to a request for a setup change from an automatic mode device by doing one of the following:

- Execute the request, then issue the $S command to the device.
- Allow the use of the setup only for the data set group that requested it, by issuing the $S command followed by the $P command. The $P command causes the device to become idle after printing the current data set group.
- Force an alternate setup on this data set group by issuing the $T command, followed by the $S command. The device must be set up, however, and the $T and $S commands repeated for each data set in the group. Header and trailer pages are considered data sets for this sequence.
- Cause the selection of an alternate data set group by holding ($H command) the job, then issuing the $I or $F command, which causes the data set group to be requeued in a held state. The held group must be released later by the operator.
- Delete the data set group by issuing the $C command to the device. Another data set group is then selected for the setup on this device, or another setup is requested.

The operator can also change characteristics of a manual mode device or change the mode of the device if the device is idle.

Output Routing

A user can route output to a specific local or remote device, to a specific remote station, to a remote workstation, or to a pool of remote workstations. The user can route a specific data set via the DEST parameter of the DD statement, a specific data set or group of data sets via the /*OUTPUT statement, or the entire print/punch output for the job via the /*ROUTE statement. DEST cannot be used to route a specific device.

If the destination for a data set is specifically stated on the /*OUTPUT statement, or via the DEST parameter, it is used. For data sets with no destination specified, the destination on the /*ROUTE statement or a default is used. The default print and punch destination may be specified on the reader from which the job was received. If not, the default becomes the location (LOCAL or RMTnnn) from which the job was received.

The system programmer specifies the destination number (printer or punch) for each local and remote device, and for each remote station, during JES2 initialization. If a destination number is specified for any device, that device is eligible to receive only data which is specifically routed to it.

Destination names are of the form PRINTERn or PRINTRnn, PUNCHn, RMTnn, or LOCAL. LOCAL indicates any device attached to the local CPU. The *n* or *nn* is a numeric destination ID assigned to the device or remote station during initialization. The form PRINTERn must be used if the installation has less than ten printers; the form PRINTRnn must be used if ten or more printers are specified during JES2 generation.

By assigning the same destination id to a group of remote stations or a group of devices, the system programmer can create a remote pool or device pool.

Processing Held Data Sets

A data set is explicitly held via the HOLD parameter of a DD statement, or by specifying HOLD during dynamic allocation or deallocation.

A data set can also be implicitly held if it is in a class that is held and the job's MSGCLASS is also a held class. Since for a data set to be implicitly held, both the class in which it is written and its MSGCLASS must be held classes (the X JES2 initialization parameter is used to specify held classes), a programmer can control the holding of data sets using only the MSGCLASS parameter.

For the job described in Figure 4-6, assume that output classes A and M are defined as held at JES2 initialization; then:

- Because MSGCLASS=A, the SYSUT2 and SYSPRINT data will be held.
- SYSUDUMP will not be held.
- If the MSGCLASS were changed to C, none of the data would be held.
- If this JCL is submitted through TSO, it can be held with MSGCLSS=A. The same JCL can be submitted from an RJE terminal with MSGCLASS=C and the output will be printed at the RJE station.

A held data set is enqueued in a special queue. Job output elements are not built for a held data set.

Data sets are released from the HOLD state either from a time sharing terminal or by the output operator command ($O). *Only data sets in the HOLD state can be retrieved with the TSO OUTPUT command.*

External Writers

After output is described by job output elements and queued in priority order in the job output table, the output can be written by the JES2 writer or an external writer. An external writer can be standard IBM-supplied external writer processor, or an installation-written writer name on the SYSOUT DD statement. The operator starts an external writer in a private address space, and the data is written using the QSAM access method.

For details on the external writer, see *OS/VS2 System Programming Library: Job Management.*

3540 Diskette Writers

When SYSOUT data sets are to be written on 3540 diskettes, the 3540 diskette writer program must be used. See *OS/VS2 IBM 3540 Programmer's Reference* for details.

```
//TSOUSER     JOB     name,MSGLEVEL = 1,MSGCLASS = A
//STEP        EXEC    PGM = IEBGENER
//SYSPRINT    DD      SYSOUT = A
//SYSUDUMP    DD      SYSOUT = D
//SYSUT1      DD      DSN = USERA.DSN1.ASM,DISP = OLD
//SYSUT2      DD      SYSOUT = M,DCB = (RECFM = F,LRECL = 80,BLKSIZE = 80)
//SYSIN       DD      DUMMY
```

Figure 4-6. Sample JCL for TSO-Submitted Job

Output Separation

The JES2 writer uses an output separate facility to write separation records prior to writing the output of each job. These separation records make it easy to identify and separate the various job outputs that are written contiguously on the same printer or card punch device.

For data processed by a JES2 writer, the JES2 separator pages are written before and after the writing of the output represented by one JOE.

The JES2 Print Separator

JES2 START JOB and END JOB separator pages consist of one-half page of blocked letters specifying the jobname, job id and output class; plus a single line of information duplicated as specified by each installation. All alphanumeric and all national characters are represented in blocked letter format. (The installation specifies the total number of lines on the separator page. If less than thirty, no blocked letters will appear.) The operator may request separator lines or cards via issuing a "$T device,S=Y [ES] " command. This function may be deleted by issuing a "$T device,S=N [C] " command. The default status is "S=YES" unless specified by an initialization option. An example of the information line is as follows:

Columns	Contents
1-4	asterisks(*)
5	output class
8-12	START
	CONT
	END
15-22	job id assigned by JES2
25-32	job name
35-54	programmer name from job card
57-60	ROOM
62-65	room number
68-78	time of printing the page in the form: hh.mm.ss. AM or PM
80-88	date of printing the page in the form: day month year
91-98	name of JES2 output device
101-104	SYS
105-108	system id from SMF
111-118	job id assigned by JES2
121-125	START
	CONT
	END
128	output class
129-132	asterisks(*)

The JES2 Punch Separator Card

Each job's punch output will optionally be preceded by an identification card. To make the card easy to identify, it has an 11-punch and a 12-punch punched in all 80 columns. To make the room number and job number easy to read, each digit is extended over ten columns. Alphabetic characters are converted to digits as follows:

Alphabetic Characters	Numeric Punch
A or J	1
B, K or S	2
C, L, or T	3
D, M, or U	4
E, N, or V	5
F, O, or W	6
G, P, or X	7
H, Q, or Y	8
I, I, or Z	9

CHAPTER 5. REMOTE JOB ENTRY

The remote job entry (RJE) facility of JES2 allows remote workstations to use the job entry subsystem even though they are not located at the central installation. JES2 processes remote jobs no differently from those received from local readers, printers, and punches.

This chapter describes the characteristics of the remote devices supported by JES2, and the RMT generation procedure for generating remote terminal processor (RTP) programs (including the parameters used and the processing of the generation). Some aspects of operating a remote station are also included in this chapter; for example, starting remote job entry and disconnecting remote lines.

Overview

Remote job entry is the ability to submit jobs and receive system output at remote facilities as if the jobs had been submitted at a local facility. The remote facility must be attached to JES2 by a (point-to-point) binary synchronous communication link. The remote facility becomes a logical extension of the local computer facility and is expected by JES2 to be under the control of a person called a remote operator.

There are two types of remote job entry stations. The first type is the *remote terminal*, that does not have a CPU. A remote terminal, for example, a 2780 or 2770, can be used for entering jobs into and receiving data from JES2. The second type is a *remote workstation* that does have a CPU. A processor, for example, System/3 or System/370, executes a JES2 generated program that allows the processor to send jobs to and receive data from JES2. Also part of the workstation are printers, punches, card readers, and a console. A remote workstation is established by a JES2 program, RMTGEN, during system generation or later. See "Remote Generation" in this section for a description of the procedure and parameters used. A *remote station* is a composite term for a remote terminal and a remote workstation.

Reading, printing, and punching between the CPU and the remote terminal take place one action at a time. For example, it is either transmitting print data or transmitting punch data or reading an input stream. The remote operator may influence the order of these events. A discussion of how this is done is presented later in this section under, "Altering the Sequence of Operations from a Remote Terminal."

Communication between the local CPU and remote workstations uses a JES2 facility called MULTI-LEAVING that allows multiple print and punch streams to be transmitted at the same time and multiple console messages and input streams to be received by JES2. With MULTI-LEAVING, you can have several operations going simultaneously. Operators at remote terminals and at workstations that have no console can enter commands into the input stream in the normal manner. Command replies will be scheduled back to the remote station for printing on a remote printer.

JES2 provides remote station support for the following programmable MULTI-LEAVING workstations:

IBM System/360 Model 20, Submodels 2, 4, 5, and 6 with the following selectable options:

2560 Multi-Function Card Machine

2520 Card Reader/Punch

2501 Card Reader

1442 Card Punch

2203 Printer

1403 Printer

2152 Console

IBM System/360 Models 22 and up and System/370 Models 115, 125, 135, 145, 155, 158, 165, 168, and 195 with the following selectable options:

2540 Card Reader/Punch

2520 Card Reader/Punch

1442 Card Reader/Punch

3525 Card Punch (suppotted as a 1442)

2501 Card Reader

3504 Card Reader (supported as a 2501)

3505 Card Reader (supported as a 2501)

1403 Printer

1443 Printer

3211 Printer

3203 Printer

5203 Printer

1052 Console

3210/3215 Console (supported as a 1052)

5313 Console for the Model 125 (requires 1052 compatibility feature)

(Note: System/370 RMS support is not provided)

IBM 1130 System with the following selectable options:

1403 Card Printer

1132 Printer

1442 Card Reader/Punch

1442 Card Punch

2501 Card Reader

Standard printer-keyboard

IBM System/3 Model 10 with the following selectable options:

5203 Printer

5424 Card Reader/Punch

1442 Card Reader/Punch

5471 Printer-Keyboard

5475 Data Entry Keyboard

Remote lines can be configured as dedicated or non-dedicated. This configuration is established during initialization when the remote stations are specified. If the station parameter, RMTnnn, designates a line number, the line is dedicated to that station. Lines that are not pointed to by a station parameter at initialization are non-dedicated lines and are eligible to be dynamically connected to any non-dedicated station.

Remote stations that are not physically connected to the CPU, that is, stations that must be connected via dial facilities, normally do not specify a dedicated line so that the station may be connected to any available non-dedicated line. There are other reasons for specifying a line as non-dedicated even if the line is physically connected to a remote station.

- A sign-on card is not required for connecting stations to dedicated lines, and is ignored, since the station is considered active when the line is started. Therefore, line and station password authorization is only enforced for non-dedicated lines and stations.
- One physically connected station can be initialized as multiple non-dedicated stations for use by different groups or at different times. The period of use of each such logical station would be defined by sign-on and sign-off. Data routed to the logical station will only be transmitted while that logical station is signed on.
- If remote stations are initialized as non-dedicated, one remote station can be used as backup for an inoperable station by being signed on with the inoperable station's id.
- A station attached to a dedicated line is considered active whenever the line is active. Line activation is under control of the central operator. The central operator is not aware of station usage in this case. (He is aware of station usage when non-dedicated stations are signed on and off via the console). Also, JES2 allocates resources for remote lines while they are active, which is only between sign-on and sign-off for non-dedicated lines.

One advantage in specifying lines as dedicated is that the station can be used without signing on the station, a manual process at all remote terminals.

It is possible to configure lines and stations that must be connected by dial facilities as dedicated. However, there can be only one station id and set of station characteristics associated with the dedicated line.

RMT Generation

RMT generation is the JES2 procedure for generating MULTI-LEAVING remote terminal processor (RTP) programs for remote job entry from programmable remote workstations. RMT generation requires other procedures for its processing; for example, procedures for allocating space and cataloging. It also requires certain spool data sets for job processing after generation. These procedures and data sets, also required by JES2 generation, are described in the chapter "Installing JES2."

The following sections describe the RMT parameters used and the processing involved in an RMT generation.

Specifying RMT Parameters

If RTP programs are to be generated, parameters that define those programs must also be specified. Additionally, if changes are to be made to the RTP program source modules, these changes must be specified in control statements.

For an RMT generation, the input deck contains one or more RTP program descriptions. Each terminal program to be generated is described by card entries in the following order:

1. JES2 remote terminal processor program identification
2. RMT generation parameter cards
3. $.UPDATE control card (optional)
4. Update cards if $.UPDATE is specified
5. $.RMTEND end of RMT generation description

RMT Parameters

Each parameter is coded, beginning in column 1, in the format

parameter=value

parameter represents a valid option specified in the appropriate RTP program options section (see "RMT Parameter Descriptions"), and

value represents a character string of up to seventeen characters.

The format cannot have embedded blanks. Comments can be included in a parameter statement but they must be separated from the value by one or more blanks.

RMT generation parameters may appear in any order after the RTP program identification card. If the same parameter occurs more than once in the input deck, the last occurrence determines the parameter value.

RMT Control Cards

The general format for RMT control cards is:

Columns	Field	Description
1-2	$.	Control card identification
3-71	*operands*	Variable length, separated by a comma and containing no embedded blanks (the last operand must be followed by a blank).
73-80	*ignored*	

The first card in the RMT generation input deck is a JES2 remote terminal processor program identification card. It serves two functions:

- Selects the appropriate standard options group and source member from SYS1.HASPSRC, and
- Sets the remote terminal identification number.

The card format is:

$.*name,n*

name is the name of the RTP program to be generated (see Figure 5-1), and

n is a one to three digit terminal number that specifies the remote sign-on number (the first number cannot be 0). This number must be followed by a blank.

There are two additional RMT control cards:

- $.UPDATE which sets the update mode and causes the cards following this card to be used to modify the RTP program source modules for the current generation description, and
- $.RMTEND which is required to signal the end of the RMT generation description.

JES2 Remote Terminal Processor Program for	**Terminal Program Identification Card (First Card of Each Remote Description)**
System/360 Model 20, 2922	**$.RMTM20,*n***
System/360 (other than Model 20) or System/370	**$.RMT360,*n***
1130 Loader	**$.RTPLOAD,*n***
1130	**$.RTP1130,*n***
System/3	**$.RMTSYS3,*n***
	where *n*=remote sign-on number

Figure 5-1. RTP Program Identification Cards

RMT Update Control Cards

The update control cards may be used only during an update run, after a $.UPDATE CARD. The format of an update control card is shown in Figure 5-2.

The DELETE card is used to delete one or more source card images from the source code of the described RTP program (see "RMT Control Cards") as the source code is being prepared for the assembler. The DELETE card may be mixed with insertion and replacement update cards containing new source statements for the assembler. When a DELETE control card is specified, the source card images for the RPT program, starting with the serial number specified in SEQ1 through and including the serial number specified in SEQ2, are omitted from the assembler input source. ENDUP terminates the remote terminal program description. It may be replaced by $.RMTEND, which also serves this function.

RMT Update Cards

Update cards are assembly language source cards in the format described in *OS/VS–DOS/VS VM/370 Assembler Language.* Each card may be serialized in columns 73 through 80 or may have blanks in columns 73 through 80. Cards with blank serial numbers will be inserted in the source deck after the last serialized input card or, if following a DELETE control card, in place of the deleted source cards. All serialized input, including DELETE control cards, must have the serial numbers in columns 73-80 in ascending order.

Columns	Field	Description
1-2	./	Control identification
3-*n*	*Blank*	0 to 44 blanks
(*n+1*) - (*n+6*)	DELETE	Verb for delete source cards indicated
(*n+7*) - *m*	*Blank*	1 to 45-number of previous blanks
(*m+1*) - (*m+13*)	SEQ1=*serial1*	Starting card serial number
(*m+14*)	,	Control separator
(*m+15*) - (*m+27*)	SEQ1=*serial2*	Ending card serial number

Figure 5-2. The Format of an RMT Update Control Card

RMT Parameter Descriptions

The following subsections describe the parameters for each of five different types of RMT generations: System/360 (models other than Model 20) and System/370 binary synchronous communication (BSC) remote terminal processor program, System/360 Model 20 binary synchronous communication (BSC) remote terminal processor program (including the 2922), 1130 remote terminal processor program, 1130 loader program, and System/3 remote terminal processor·program.

Refer to the overview at the beginning of this chapter for the devices that are supported in each type of remote workstation.

The following conventions are used in this manual to describe the RMT parameters:

- The RMT parameters for each RTP program are discussed alphamerically; the first character is ignored if it is & or $.
- Letters and numbers in bold type must be coded as shown.
- Lowercase letters in italics represent variables for which you must substitute specific information or specific values.
- If an alternative item is underlined, it is the default value. This value will be used automatically if the parameter is not specified.

RMT Parameters for the System/360 Model 20 BSC RTP Program

This section describes the parameters used to specify the machine configuration and programming options required in the assembly of the System/360 Model 20 BSC remote terminal processor program for JES2 MULTI-LEAVING remote job entry.

Parameter	Value	Explanation
&CCT=	*number* 4	is an integer from 3 to 31 that specifies, for all text compression (except trailing blank compression), the minimum number of characters to be compressed. A duplicate character string of fewer than the number specified is treated as a string of nonduplicate characters for compression purposes. If a small value is specified, efficiency of communication line usage is increased at the expense of the compute time that is required for compression. If the &CMPTYPE parameter is specified as 1, this parameter is ignored.
&CMPTYPE=	**1** **2** **3**	specifies the type of compression to be applied to all data transmitted from the Model 20 to JES2. 1 specifies trailing blank compression. 2 specifies compression of leading, embedded, and trailing blanks. 3 specifies compression of all duplicate character strings. If this parameter is specifies as 1, the &CCT parameter is ignored.
&CORESIZ=	*number* 8	is an integer from 8 to 32 that specifies the size of Model 20 main storage in 1K bytes (1K bytes equals 1024 bytes).

Parameter	Value	Explanation
&CORESIZ= (continued)		This parameter must never be greater than the actual storage size of the object machine. Warning: It is possible to specify combinations of parameters such that the resulting workstation program is too large for the available storage (&CORESIZ=255). Such a program will fail to load into the object machine.
&ERRMSGN=	*number* 10	is a value greater than or equal to 8 that specifies the number of 4-byte entries to be assembled in the Model 20 RTP program as an error message log table.
&LINESPD=	*number* 2000	is an integer that specifies the speed, in baud, of the communication line to be used between the Model 20 and JES2.
&NUMBUFS=	*number* 8	specifies the number of teleprocessing buffers to be constructed by the Model 20 RTP program. The specification must be an integer no less than: **2X+1** where: X = 1, if either a 2520 or a 2560 is to be used as both a reader and a punch X = 0, if a 2520 or a 2560 is not to be used as both a reader and a punch The length of each buffer is the value specified in the &MLBFSIZ parameter plus 5 bytes (rounded upward to the next fullword). (The value of the JES2 parameter &MLBFSIZ is automatically propagated to the RMT generation.) If this parameter specifies more buffers than can be built in available storage, the RTP program will build as many buffers as it can. It is recommended that at least two buffers be provided for each output device and for the communication adapter.
&NUMTANK=	*number* 8	is an integer greater than or equal to 2 that specifies the number of decompression buffers to be assembled in the Model 20 RTP program. It is recommended that at least two buffers be provided for each printer and punch. The length of each decompression buffer is the value specified in the &PRTSIZE parameter plus 6. For an 8K Model 20, specifying this parameter greater than 8 may cause the RTP program to assemble more than X'1F00' bytes (8K – 256). If this occurs, the resultant program will fail to load.

Parameter	Value	Explanation
&PDEV(1)=	*devtype* 2203	specifies the device type for the Model 20 printer. The specification must be either 1403 or 2203.
&PRTCONS=	0 1 2	specifies the usage of the printer as an output console. This parameter is dependent upon the specifications given during JES2 initialization that pertain to the handling of messages for the remote. If JES2 is informed, by means of the RMT*nnn* parameter at JES2 initialization, that the remote has a console, &PRTCONS should be·specified as one of the following: 0 specifies that error logging and display will be suppressed and operator messages that are created while the remote is online to JES2 will be discarded. 1 specifies that the printer will be used as an output console when sufficient operator messages from JES2 have been queued for output at the remote. If the printer is busy with job stream output, that output will be interrupted for the printing of operator messages from JES2 and from the remote error log. When the console queue is empty, job stream output will continue. 2 specifies that the printer will be used as an output console but will not interrupt the printing of jobs. Operator messages received from JES2 while jobs are being printed will be discarded. If JES2 is informed, by means of the RMT*nnn* parameter at JES2 initialization, that the remote does not have a console, and if JES2 does not have message spooling capability (as determined by the JES2 parameter &SPOLMSG), &PRTCONS should be specified as follows: 0 specifies that error logging and display will be suppressed (JES2 will not return operator messages to the workstation). 1 or 2 specifies that error log messages will be displayed when the printer is free to print them and no job's printed output will be interrupted. If JES2 is informed, by means of the RMT*nnn* parameter at JES2 initialization, that the remote does not have a console and if JES2 has message spooling capability (as determined by the JES2 parameter &SPOLMSG), &PRTCONS should be specified as if the remote did not have a console and JES2 did not have message spooling capability (the second specification, above). The definitions are the same but with an additional capability in that operator messages queued for the remote by JES2, transmitted

Parameter | **Value** | **Explanation**

&PRTCONS= (continued)

to the remote when the printer is free, and set to receive messages (via the $TRMr.PR1 command) are printed.

If &WDEV(1) is not specified as 0, this parameter should be set to 0.

Regardless of the settings of &WDEV(1) and this parameter, error messages resulting from loggable errors detected by the remote will be discarded when the errors occur at a rate that is faster than the output device can display them.

Refer to the &SPOLMSG and &WDEV(1) parameters for additional information.

&PRTSIZE= *number* 120

specifies the length, in bytes, of the text portion of each decompression buffer. Each buffer must be long enough to hold a maximum-length output record for either the printer, the punch, or the operator console. The specification must be an integer that is the largest of 80 (if &UDEV(1) is not 0), 120 (if &WDEV(1) is not 0), or the line width of the printer.

&RADR(1) *unitadr* 1

specifies the unit address of the Model 20 card reader. The specification must correspond to the specification for the &RDEV(1) parameter as follows:

&RDEV(1)	**&RADR(1)**
2501	1
2520	2
2560	2

This parameter should not be altered when generating a 2922 work station program.

&RDEV(1)= *devtype* 2501

specifies the device type for the Model 20 card reader. The specification must be either 2501, 2520, or 2560.

This parameter should not be altered when generating a 2922 work station program.

Refer to the &RADR(1) parameter for additional information.

&SUBMOD= *submodel* 2

specifies the submodel number of the Model 20 for the specified remote terminal. The specification must be a valid System/360 Model 20 submodel number.

This parameter should not be altered when generating a 2922 work station program.

Parameter	Value	Explanation
&UADR(1)=	*unitadr* 3	specifies the unit address of the Model 20 card punch. The specification must correspond to the specification of &UDEV(1) as follows: **&UDEV(1)** **&UADR(1)** 1442 3 2520 2 2560 2 0 not present
&UDEV(1)=	*devtype* 1442	specifies the device type for the Model 20 card punch. The specification must be either 1442, 2520, 2560, or 0. Specify 0 when the Model 20 does not include a card punch. Specify &UDEV(1)=0 for the 2922, unless the RPQ punch is included (in which case &UDEV(1) should not be altered). Refer to the &UADR(1) parameter for additional information.
&WDEV(1)=	*devtype* 0	specifies the device type for Model 20 console. The specification must be either 2152, if a console is present, or 0, if a console is not present. If a console is present, console support should be indicated for this remote terminal at JES2 initialization.
&WTOSIZE=	*number* 120	is an integer less than or equal to 120 that specifies the maximum length, in bytes, of a JES2 operator command to be transmitted from the Model 20 to the central computer. If &WDEV(1) is specified as 0, this parameter is ignored.
&XPARENT=	NO YES	specifies the inclusion or exclusion of support for the text transparency feature. If the binary synchronous communication adapters at both the Model 20 and the central computer have the text transparency feature, the default should be used; otherwise, NO should be specified.

RMT Parameters for the 2922 Remote Workstation RTP Program

This section describes the parameters used to specify the machine configuration and program options required in the assembly of the 2922 remote terminal processor program for JES2 MULTI-LEAVING remote job entry.

To install a 2922 RTP program, the parameters and procedures for the System/360 Model 20 BSC should be used, subject to the following specific parameter setting:

&LIÑESPD=*xxx* where *xxx* is the actual line speed used.
&PDEV(1)=1403
&PRTSIZE=132
&UDEV(1)=0
&WDEV(1)=2152, if the optional typewriter console is installed.
&XPARENT=NO, if the optional text transparency feature is not installed.

The default values should be used for the following parameters:

&CORESIZ=
&RADR(1)=
&RDEV(1)=
&SUBMOD=
&UADR(1)=

The remaining Model 20 BSC parameters may be allowed to default or may be changed.

RMT Parameters for the System/360 (Except Model 20) and System/370 BSC RTP Program

This section describes the parameters used to specify the machine configuration and program options required in the assembly of the System/360 and System/370 BSC remote terminal processor program for JES2 MULTI-LEAVING remote job entry.

Parameter	Value	Explanation
&ADAPT=	*unitadr* 020	specifies the unit address of the binary synchronous communication adapter used by the System/360 or System/370 remote terminal to communicate with JES2 at the central computer. The specification must be a valid unit address.
&CCT=	*number* 4	is an integer from 3 to 31 that specifies, for all text compression (except trailing blank compression), the minimum number of characters to be compressed. A duplicate character string of fewer than the number specified is treated as a string of nonduplicate characters for compression purposes. If a small value is specified, efficiency of communication line usage is increased at the expense of the compute time that is required for compression. If the &CMPTYPE parameter is specified as 1, this parameter is ignored.
&CMPTYPE=	1 2 3	specifies the type of compression to be applied to all data transmitted from the System/360 or System/370 remote terminal to JES2. 1 specifies trailing blank compression. 2 specifies compression of leading, embedded, and trailing blanks.

Parameter	Value	Explanation
&CMPTYPE= (continued)		3 specifies compression of all duplicate character strings. If this parameter is specified as 1, the &CCT parameter is ignored.
&CORESIZ=	*number* 8	is an integer from 8 to 32 that specifies the size of main storage for the System/360 or System/370 remote terminal in 1K bytes (1K byte equals 1024 bytes). If the System/360 or System/370 remote terminal is larger than 32K bytes, this parameter must be specified as 32.
&ERRMSGN=	*number* 10	is a value greater than or equal to 8 that specifies the number of 4-byte entries to be assembled as an error message log table in the System/360 or System/370 remote terminal.
&LINESPD=	*number* 2000	is an integer that specifies the speed, in baud, of the communication line to be used between the System/360 or System/370 remote terminal and JES2.
&MACHINE=	*model* 30	specifies the model number of the System/360 or System/370 to be used as a JES2 remote terminal. The specification must be a valid number for a System/360 or System/370 that includes the standard instruction set and the decimal instruction set.
&NUMBUFS	*number* 8	specifies the number of teleprocessing buffers to be constructed by the System/360 or System/370 remote terminal program. The specification must be an integer no less than: **2X+1** where: X = *n*, the number of 2520 or 1442 units to be used as both readers and punches X = 0, if neither a 2520 nor a 1442 is to be used as both a reader and a punch The length of each buffer is the value specified in the &MLBFSIZ parameter plus 5 bytes (rounded upward to the next fullword). (The value of the JES2 parameter &MLBFSIZ is automatically propagated to the RMT generation.) If this parameter specifies more buffers than can be built in available storage, the RTP program will build as many buffers as it can. It is recommended that at least two buffers be provided for each output device and for the communication adapter.

Parameter	Value	Explanation
&NUMTANK=	*number* 5	specifies the number of decompression buffers to be assembled in the System/360 or System/370 RTP program. The specification must be an integer greater than zero and not less than 2 (number of 2540 punches attached). The length of each decompression buffer is the value specified in the &PRTSIZE plus 6. It is recommended that at least two decompression buffers be provided for each printer and each punch (three buffers for a 2540 punch).
&PADR(*n*)=	*unitadr*	specifies the unit address of each remote terminal printer defined by the &PDEV(*n*) parameter. The *n* is a sequential number (1-7) that you code to identify each device being specified. For each &PDEV(*n*) parameter that is not specified as 0, the corresponding parameter &PADR(*n*) must specify the device's three-character hexadecimal unit address. All devices at the remote terminal workstation must be on separate non-shared subchannels (that is, all I/O devices must be capable of running simultaneously). If this parameter is not specified, the following values are used as defaults: **&PADR(1)=00E** **&PADR(2)=00F** **&PADR(3)=FFF** **&PADR(4)=FFF** **&PADR(5)=FFF** **&PADR(6)=FFF** **&PADR(7)=FFF**
&PDEV(*n*)=	*devtype*	specifies the existence and device type of each remote terminal printer. The specification must be either 1403, 1443, 3211, 3203, 5203, or 0. A specification of 0 indicates that the associated printer does not exist. The *n* is a sequential number (1-7) that you code to identify each device being specified. If this parameter is not specified, the following values are used as defaults: **&PDEV(1)=1403** **&PDEV(2)=0** **&PDEV(3)=0** **&PDEV(4)=0** **&PDEV(5)=0** **&PDEV(6)=0** **&PDEV(7)=0** &PDEV(1) must not be specified as 0.

Parameter	**Value**	**Explanation**
&PDEV(*n*)= (continued)		If &PDEV(*n*+1) is specified as a device type, &PDEV(*n*) must be specified as a device type. If &PDEV(*n*) is specified as a device type, &UDEV(8-*n*) must be specified as 0. If more than one printer is specified, more than one printer should also be specified in the RMT*nnn* parameter at JES2 initialization.
&PRTSIZE=	*number* 132	specifies the length, in bytes, of the text portion of each decompression buffer. Each buffer must be long enough to hold a maximum-length output record for either a printer, a punch, or the operator console. The specification must be an integer that is the larger of 120 or the line width of the widest printer.
&RADR(*n*)=	*unitadr*	specifies the unit address of each remote terminal card reader defined by the &RDEV(*n*) parameter. The *n* is a sequential number (1-7) that you code to identify each device being specified. For each &RDEV(*n*) parameter that is not specified as 0, a corresponding parameter &RADR(*n*) must specify the device's three-character hexadecimal unit address. All devices at the remote terminal workstation must be on separate nonshared subchannels (that is, all I/O devices must be capable of running simultaneously). If this parameter is not specified, the following values are used as defaults: **&RADR(1)=00C** **&RADR(2)=FFF** **&RADR(3)=FFF** **&RADR(4)=FFF** **&RADR(5)=FFF** **&RADR(6)=FFF** **&RADR(7)=FFF**
&RDEV(*n*)=	*devtype*	specifies the existence and device type of each remote terminal card reader. Each specification must be either 1442, 2501, 2520, 2540, or 0. A specification of 0 indicates that the associated remote terminal card reader does not exist. The *n* is a sequential number (1-7) that you code to identify each device being specified.

Parameter	Value	Explanation
&RDEV(*n*)= (continued)		If this parameter is not specified, the following values are used as defaults: **&RDEV(1)=2540** **&RDEV(2)=0** **&RDEV(3)=0** **&RDEV(4)=0** **&RDEV(5)=0** **&RDEV(6)=0** **&RDEV(7)=0** &RDEV(1) must not be specified as 0. If &RDEV(*n*+1) is specified as a device type, &RDEV(*n*) must be specified as a device type. If more than one reader is specified, more than one reader should also be specified in the RMT*nnn* parameter at JES2 initialization.
&UADR(*n*)=	*unitadr*	specifies the unit address of each remote terminal punch defined by the &UDEV(*n*) parameter. The *n* is a sequential number (1-7) that you code to identify each device being specified. For each &UDEV(*n*) parameter that is not specified as 0, the corresponding parameter &UADR(*n*) must specify the device's three-character hexadecimal unit address. All devices at the remote terminal workstation must be on separate nonshared subchannels (that is, all I/O devices must be capable of running simultaneously). If this parameter is not specified, the following values are used as defaults: **&UADR(1)=00D** **&UADR(2)=FFF** **&UADR(3)=FFF** **&UADR(4)=FFF** **&UADR(5)=FFF** **&UADR(6)=FFF** **&UADR(7)=FFF**
&UDEV(*n*)=	*devtype*	specifies the existence and device type of each remote terminal punch. The specification must be either 1442, 2520, 2540, or 0. A specification of 0 indicates that the associated punch does not exist. The *n* is a sequential number (1-7) that you code to identify each device being specified.

Parameter	Value	Explanation
&UDEV(*n*)= (continued)		If this parameter is not specified, the following are used as defaults: **&UDEV(1)=2540** **&UDEV(2)=0** **&UDEV(3)=0** **&UDEV(4)=0** **&UDEV(5)=0** **&UDEV(6)=0** **&UDEV(7)=0** If &UDEV(*n*+1) is specified as a device type, &UDEV(*n*) must be specified as a device type. If &UDEV(*n*) is specified as a device type, &PDEV(8-*n*) must be specified as 0. If more than one punch is specified, more than one punch should also be specified in the RMT*nnn* parameter at JES2 initialization.
&WADR(1)=	*unitadr* **01F**	specifies the unit address of the 1052 or 1052-compatible operator console on the System/360 or System/370 remote terminal. The specification must be a three-character hexadecimal unit address.
&WTOSIZE=	*number* **120**	is an integer less than or equal to 120 that specifies the maximum length, in bytes, of a JES2 operator command to be transmitted from the System/360 or System/370 remote terminal to JES2.
&XPARENT=	NO **YES**	specifies the inclusion or exclusion of support for the text transparency feature. If the binary synchronous communication adapters at both the System/360 or System/370 remote terminal and the central computer have the text transparency feature, the default value should be used; otherwise, NO should be specified.

RMT Parameters for the 1130 RTP Program

This section describes the parameters used to specify the machine configuration and program options required in the assembly of the 1130 remote terminal processor program for JES2 MULTI-LEAVING remote job entry.

Parameter **Value** **Explanation**

&CLOCK=
0
1

specifies the type of communication adapter clocking available on the 1130.

0 specifies that data set clocking is being used.
1 specifies internal 1130 clocking.

The rate of insertion of the synchronous idle sequence in the transmitted data is determined by the &CLOCK, &LINESPD, and &TRANPRN parameters. The relationship of these parameters to the insertion rate is:

&CLOCK	&TRANPRN	Insertion Rate
0	0	Every &LINESPD/8 characters
0	1	Every &LINESPD/8 characters
1	0	Every 70 characters
1	1	Every &LINESPD/8 characters

The equation used for the insertion rate is:

(&LINESPD/8)T

where T is 1.00 second, which is the nominal 2701 timer value.

&CMPTYPE=
0
1
2

specifies the type of compression to be applied to all data transmitted to JES2.

0 specifies no compression of duplicate characters and no truncation of trailing blanks.
1 specifies trailing blank truncation.
2 specifies full compression, trailing blank truncation and encoding of duplicate characters.

The process of compressing input data offers optimum performance with respect to efficient line utilization. However, factors such as line speed, CPU availability, buffer size, line turnaround time, and nature of the data to be compressed contribute to the overall operation of the RTP program. Since compression and truncation require considerable CPU time, you may decide, on the basis of the other variables, to respecify the compression technique.

&DELAY=
number
3

specifies the number of time intervals that the RTP program will delay in transmitting a "handshaking" sequence (DLE-ACK0) to the central computer. The machine program timer clock is used to measure the delay and is assumed to be set to a minimal value of .35 seconds.

Parameter	Value	Explanation
&DELAY= (continued)		The purpose of the delay when "handshaking" is to minimize CPU processing at the central computer when no data is being transmitted. Using the default value results in a delay of 1.05 seconds, assuming a timer interval of .35 seconds. The value of this parameter must not be set to such a large increment that the delay will be greater than the time-out period of the central computer's 2701/2703.
&FULLIST=	**0** **1**	specifies the type of assembly listing produced by the OS/VS assembler during RMT generation. 0 specifies that the assembly listing will be produced according to the PRINT NOGEN stipulation of the assembler. 1 specifies that the listing will be produced according to the PRINT GEN stipulation. Because most of the code in the RTP1130 and RTPLOAD programs is created by macro instructions, the specification of 0 will, essentially, produce a source listing (cross-referenced) without the 1130 assembled instructions.
&LINESPD=	*number* **2000**	is an integer that specifies the speed, in baud, of the communication line to be used between the 1130 and central computer. The value should correspond to the selected setting of the baud rate switch on the 1130 SCA control panel: 1200, 2000, ... The rate of insertion of the synchronous idle sequence (DLE-SYN or SYN-SYN) in the transmitted data is determined by the &CLOCK, &LINESPD, and &TRANPRN parameters. Refer to the &CLOCK parameter for the relationship of these parameters.
&MACHSIZ=	*number* 8192	specifies the amount of 1130 storage to be used by the RTP program. The value is specified in 1130 words. The value specifies the number of words, starting at location 0, that are available to the RTP programs (RTPBOOT, RTPLOAD, and RTP1130). The value specified may be less than the actual available storage but must not be greater. The same parameter must be defined for the assembly of the 1130 loader program and should have the same value.

Parameter	Value	Explanation
&PN1442=	0 1	specifies the inclusion or exclusion of support for the 1442 punch in the RTP program. 0 specifies that support is not to be included. 1 specifies that support for punched card output produced by jobs at the central computer is to be included. Refer to the &RD1442 parameter for information about the reader function of the 1442.
&PRFOTLW=	120 132	specifies the line width of the 1403 printer. The specification of the line width for all printers on a remote terminal is a JES2 installation requirement.
&PR1132=	0 1	specifies the inclusion or exclusion of the support for the 1132 printer in the RTP program. 0 specifies that support is not to be included. 1 specifies that support for printing job output using the 1132 is to be included.
&PR1403	0 1	specifies the inclusion or exclusion of the support for the 1403 printer in the RTP program, where: 0 specifies that support is not to be included, and 1 specifies that support is to be included.
&RD1442=	0 1	specifies the inclusion or exclusion of the support for a 1442 card reader in the RPT program. 0 specifies that support is not to be included, and 1 specifies that support is to be included.
&RD2501=	0 1	specifies the inclusion or exclusion of the support for the 2501 card reader in the RTP program. 0 specifies that support is not to be included. 1 specifies that support is to be included.
&RTPLORG=	*number*	defines the location in 1130 storage of the RTPLOAD program, which is used to load the 1130 RTP program. If this parameter is not specified, the default value is: **2(&MACHSIZ–1024)** The RTPLOAD program must reside in an area of storage that is available between the beginning of the buffer pool and the end of main storage, as defined in the &MACHSIZ parameter, minus the length of the RTPLOAD program. The default value of this parameter allows 1024 words for the RTPLOAD program. Assuming &MACHSIZ=8192, the default value is 14336. This value is twice the actual 1130 storage address because the value is used in an ORG operation and must be in terms of bytes, not 1130 words.

Parameter	Value	Explanation
&TRANPRN=	0 1	specifies the simulation of the binary synchronous transparency feature. 0 specifies that no simulation will occur. In this case, data containing transparent characters cannot be properly processed by the RTP program. 1 specifies that simulation will occur in the same manner as the 2701 SDA-II adapter that is equipped with the transparency feature. If 0 is specified, the conversion of card code data is monitored and all BSC control characters are converted to hexadecimal 0. This prevents mispunched data from causing an infinite error retry if the central computer does not have transparency. If 1 is specified, the RTP program will communicate only with a 2703 or with a 2701 adapter that has the text transparency feature.

RMT Parameters for the 1130 Loader Program

This section describes the parameters used to specify the machine size, loader origin, and assembly list option that are used in the assembly of RTPLOAD, the 1130 loader program, that loads the 1130 remote terminal processor program.

RMT generation produces the object decks for the RTPLOAD and RTP1130 programs. The bootstrap loader program (RTPBOOT) cannot be produced on System/360 or System/370 and must be keypunched as indicated in the RTP section of *OS/VS2 JES2 Logic*.

Parameter	Value	Explanation
&FULLIST=	0 1	specifies the type of assembly listing produced by the OS/VS assembler during the RMT generation. 0 specifies that the assembly listing will be produced according to the PRINT NOGEN stipulation of the assembler. 1 specifies that the listing will be produced according to the PRINT GEN stipulation. Since most of the code in the RTP1130 and RTPLOAD programs is created by macro instructions, the specification of 0 will, essentially, produce a source listing (cross-referenced) without the 1130 assembled instructions.
&MACHSIZ=	*number* 8192	specifies the amount of 1130 main storage to be used by the RTP program. The value is specified in 1130 words. The value specifies the number of words, starting at location 0, that are available to the RTP programs (RTPBOOT, RTPLOAD, and RTP1130). The value specified may be less than the actual available storage but must not be greater. The same parameter must be defined for the assembly of the 1130 RTP program and should have the same value.

Parameter	Value	Explanation
&RTPLORG=	*number*	defines the location in 1130 main storage of the RTPLOAD program, which is used to load the 1130 RTP program. If this parameter is not specified, the default value is: **2(&MACHSIZ – 1024)** The RTPLOAD program must reside in an area of storage that is available between the beginning of the buffer pool and the end of main storage, as defined in the &MACHSIZ parameter, minus the length of the RTPLOAD program. The default value of this parameter allows 1024 words for the RTPLOAD program. Assuming &MACHSIZ=8192, the default value is 14336. This value is twice the actual 1130 storage address because the value is used in an ORG operation and must be in terms of bytes, not 1130 words.

RMT Parameters for the System/3 RTP Program

This section describes the parameters used to specify the machine configuration and programming options in the assembly of the System/3 remote terminal processor program for JES2 MULTI-LEAVING remote job entry.

Parameter	Value	Explanation
&COMP=	**0** **1** **2**	specifies the type of compression to be applied to all data transmitted to the central computer. 0 specifies that no compression of duplicate characters and no truncation of trailing blanks is performed. 1 specifies that trailing blanks are truncated. 2 specifies that compression takes place after truncation. Strings of from two to thirty-one blanks are compressed to a single byte; strings of from three to thirty-one duplicate characters are compressed to two bytes.
&DEBUG=	**0** **1**	specifies the inclusion or exclusion of certain validity tests and a main storage dump program in the RTP program. 0 specifies that support is not to be included. 1 specifies that support is to be included.
&DIAL= **&DIAL1=**	*number* null string *number* null string	specifies the telephone number to be used during JES2 initialization. The values specified will be included on the /*SIGNON card that is assembled into the RTP program and will be preceded by the keyword DIAL (unless the default values are used). Each specification is a string of up to eight decimal digits. If the telephone number is eight or fewer digits, it should be specified in the &DIAL parameter. If the telephone number is more than eight digits, the leftmost eight digits are specified in the &DIAL parameter and the remaining digits are specified in the &DIAL1 parameter.

Parameter **Value** **Explanation**

&MACHSIZ= *number* 8192

specifies the size of System/3 main storage. The value specified must be the appropriate specification for the System/3 main storage size, specified as follows:

Value	Main Storage Size
8192	8K
12288	12K
16384	16K
24576	24K
32768	32K

&PASSWD= *character-string* null string

specifies the password that is to be used during the sign-on process. The value specified will be included in the /*SIGNON card that is assembled into the System/3 RTP program. The specification must be a character string of from one to eight characters. If you want blanks, let the parameter default.

&PC(*n*)= *number*

specifies skip information for the 5203 or 1403 printer. The *n* is a number from 1 to 12 that specifies the channel. The value specified in this parameter determines the print line number to which paper will be skipped when the RTP program simulates the 1403 command "Skip to Channel *n*." A specification of 0 causes no forms movement.

If this parameter is not specified, the following values are used as defaults:

&PC(1)=1
&PC(2)=0
&PC(3)=0
&PC(4)=0
&PC(5)=0
&PC(6)=0
&PC(7)=0
&PC(8)=0
&PC(9)=0
&PC(10)=0
&PC(11)=0
&PC(12)=&S3FORML–5

&PRTCONS= **0** **1** **2**

specifies utilization of the 5203 or 1403 printer as an operator's console.

0 specifies that the printer will not be used as an operator's output console.

1 specifies that the RTP program will attempt to hold operator messages from JES2 until a job has completed printing. However, if two or more MULTI-LEAVING buffers contain JES2 operator messages, the printer will eject a page (skip to channel 1), print the JES2 operator messages, eject another page, and resume printing the job.

Parameter	Value	Explanation
&PRTCONS= (continued)		2 specifies that the RTP program will throw away all operator messages while the printer is printing a job. While the printer is dormant, it will print any received messages. Regardless of the setting of this parameter, messages temporarily saved on a direct-access volume for a remote terminal will be printed to the terminal as a job. Thus, they will always appear on the printer, even if another console exists. If this parameter is specified greater than 0, MULTI-LEAVING console support should be specified in the RMT*nnn* parameter at JES2 initialization. If &S35471=1, the value of &PRTCONS is ignored and assumed to be 0.
&S3BSCA=	1 2	specifies the number of the System/3 BSC adapter to be used for RJE communication. 1 specifies the first BSCA. 2 specifies the second BSCA. The assembled System/3 RTP program uses the adapter specifiedin this parameter, only.
&S3CMDS=	0 1	specifies the inclusion or exclusion of a command facility and of commands to assist the System/3 operator. 0 specifies support is not to be included. 1 specifies that support is to be included. The commands that are available with this facility are detailed in *Operator's Library: OS/VS2 Remote Terminals (JES2)*.
&S3FORML=	*number* 66	is an integer greater than or equal to 6 that specifies the number of print lines on a page for the continuous forms used on the 5203 or 1403 printer.
&S3NPUNS=	1 2 3	specifies the maximum number of jobs that can be punched simultaneously at the System/3 remote terminal. A value of 3 allows simultaneous operation of both 5424 hoppers and the 1442 hopper as punches. If this parameter is specified greater than 1, additional punches should also be specified in the RMT*nnn* parameter at JES2 initialization.
&S3NRDRS=	1 2 3	specifies the maximum number of jobs that can read simultaneously from the System/3 remote terminal. A value of 3 allows simultaneous operation of both 5424 and 1442 hoppers as card readers.

Parameter	Value	Explanation
&S3NRDRS= (continued)		If this parameter is specified greater than 1, additional card readers should also be specified in the RMT*nnn* parameter at JES2 initialization.
&S3OBJDK=	0 1	specifies the inclusion or exclusion of the facility to punch OS/VS2 object decks. 0 specifies support is not to be included. 1 specifies support is to be included. If this facility is to be included, the text transparency feature should be present. If 1 is specified, each card of an OS/VS2 object deck will be expanded and punched into two 96-column cards. These cards will be recognized when read by the System/3 RTP program. For each two 96-column cards read, one OS/VS2 object deck card image will be transmitted.
&S3SIP=	0 1	specifies usage of those bytes of System/3 main storage between X'100' and X'1FF'. 0 specifies that the RTP program will use the bytes. 1 specifies that the bytes will be used by the System/3 card system initialization program.
&S3TRACE=	*number* 10	is an integer greater than 1 that specifies the number of 4-byte entries in the RTP program's internal error message table.
&S3XPAR=	0 1	specifies the presence or absence of the EBCDIC text transparency feature. 0 specifies that the EBCDIC text transparency feature is not present. 1 specifies that both the central computer's communications adapter and the System/3 BSCA have the EBCDIC text transparency feature.
&S31442=	0 1	specifies inclusion or exclusion of support for the 1442 card reader/punch. 0 specifies that support is not to be included. 1 specifies that support is to be included.
&S35424=	0 1	specifies inclusion or exclusion of support for the 5424 multi-function card unit. 0 specifies that support is not to be included. 1 specifies that support is to be included. If this parameter is specified as 0, &S31442 must be specified as 1.
&S35471=	0 1	specifies inclusion or exclusion of support for the 5471 printer-keyboard for use as an operator's input/output console.

Parameter	Value	Explanation
&S35471= (continued)		0 specifies that support is not to be included. 1 specifies that support is to be included. If console support is desired, MULTI-LEAVING console support must be specified in the RMT*nnn* parameter at JES2 initialization. Regardless of the specification of this parameter, messages from JES2 can print on the printer. Refer to the &PRTCONS parameter in this section and to the &SPOLMSG parameter in the chapter "Installing JES2" for additional information.
&S35475=	0 1	specifies inclusion or exclusion of support for the 5475 data entry keyboard on the System/3 for use as an operator's console. 0 specifies that support is not to be included. 1 specifies that support is to be included. If console support is desired, MULTI-LEAVING console support must be specified in the RMT*nnn* parameter at JES2 initialization. If &S35471=1, this parameter is ignored.
&S396COL=	0 1	specifies inclusion or exclusion of support for the load-mode punch option. 0 specifies that support is not to be included. 1 specifies that support is to be included. If this parameter is specified, the resultant System/3 RTP program will be capable of receiving the punched output of a System/3 RMT generation.

RMT Generation Under a Production Batch System

An RMT generation may be executed as part of a batch stream job. Figure 5-3 shows a sample job stream for a batch RMT generation. The RMT parameters and control statements are the same as for an RMT generation that is done with a JES2 generation.

RTP program modules usually write messages to the SYSPRINT data set using record format FBM with a record length of 121. The data set may be changed to SYSLIST by including a SYSLIST DD statement in the RMTGEN step. This will cause the listings from the REMOTGEN utility and assembler to be placed on separate data sets. For example:

```
//GENJOB   JOB  ...
//STEP     EXEC  RMTGEN
//RMTGEN.SYSLIST  DD    SYSOUT=A
//RMTGEN.OPTIONS  DD    *
        .
        .
        .
/*
```

Procedures for Generating RTP Programs

Input to an RMT generation is read in from the card reader after the JES2 parameters have been processed. If the JES2 parameter &BSCCPU is specified to include programmable RJE support, the following WTOR message is issued:

> *nn*PLACE RMTGEN OPTIONS IN UNIT *xxx* AND REPLY 'GO' OR REPLY 'CANCEL'
>
> where *xxx* is the unit address of the card reader.

You should make sure that the specified card reader is not being used for any other function (a JES2 card reader, for example). You should clear any cards remaining in the card reader, load the card reader with the RMT parameters, and reply "go." If no RMT generations are being performed, reply "cancel."

```
//RMTGENJB   JOB   (0000,0000),'GEN REMOTE PROGRAMS',
//                 MSGLEVEL=1
//RMTGEN     EXEC  RMTGEN
//RMTGEN.OPTIONS DD   *
$.RMTM20,2
&RDEV(1)=2560
&RADR(1)=2
&UDEV(1)=2560
&UADR(1)=2
&WDEV(1)=2152
&NUMTANK=5
$.RMTEND
$.RMT360,3
&CMPTYPE=3
&PDEV(2)=1403
&ADAPT=030
&WADR(1)=009
&NUMTANK=7
&CORESIZ=16
$.RMTEND
```

Figure 5-3. Input Deck for a Batch RMT Generation

Processing

An RMT generation, a part of the JES2 generation, begins by executing the REMOTGEN utility program. This utility acts as a monitor and links to the various RMT utility programs that generate the remote terminal load decks as follows:

1. The GENRMT utility program is invoked to read the card input stream for the remote terminal program identification, to select the appropriate standard options list for the desired RTP program, and to print the default values on the SYSOUT=A device.
2. The GENRMT utility program reads the overriding options from the card reader and changes the current values.
3. When $.UPDATE or $.RMTEND is encountered in the input deck, the remote terminal program source module is copied to a temporary data set by the GENRMT utility program. During this transfer, the options (as specified in the RMT parameters) are used to update the source module. If $.UPDATE is specified, the update cards are used to modify the source module.
4. After the source module is updated, the assembler is invoked by the REMOTGEN utility program to assemble the RTP programs in the temporary data set and, except for 1130 and System/3 programs, punch the self-loading object decks on the SYSOUT=B data set. The 1130 and System/3 assemblies write the object decks to a scratch data set.
5. On return from the assembler, if the program is for the 1130 or System/3, the REMOTGEN utility program invokes a post-processor (LETRRIP or SYS3CNVT, respectively) that creates a load-deck image on the SYSPUNCH data set. The output of this is:
   - The RTPLOAD or RTP1130 load deck for the 1130.
   - A complete load deck for the System/3 without the 5424 multi-function card unit.
   - A deck to be further processed for the System/3 with the 5424 multi-function card unit (see "Output" in this chapter).

This procedure is repeated for each RTP program to be generated.

Completion Codes

During both the JES2 and RMT generations, the success of the generation process is determined and a completion code is returned. Refer to *OS/VS Message Library: VS2 System Codes* and *OS/VS Message Library: VS2 System Messages* for a discussion of the completion codes that are returned by the system.

Output

The output from an RMT generation is a card deck for each RMT program generated. In addition, the GENRMT utility prints an information listing, the RMT parameter default values, and the parameter values you sepcified. Also, a listing of each assembly is produced.

All listings produced by the GENRMT utility and the assembler have the remote terminal sign-on identification number at the top of each page. With the exception of loader bootstrap cards, all object deck cards have the identification number punched in columns 74 through 76.

System/3 96-Column Card Output

As described in "Processing," the REMOTGEN utility invokes the post-processor SYS3CVNT to produce the System/3 object-deck image on the SYSPUNCH data set. The cards created are 80-column cards which, if routed (by use of a /*ROUTE card or the $R operator command) to a System/3 remote terminal utilizing the System/3 starter system, are punched as 96-column System/3 load mode cards. They may also be punched locally or remotely as 80-column cards (with the punched output of other RMT generations) and later be separated and routed to a System/3 starter system, as the punched output of an 80/80 card-to-punch job. The utilities IEBPTPCH or IEBGENER may be used for this. (Refer to *Operator's Library: OS/VS2 Remote Terminals (JES2)* for a description of the System/3 starter program and to *OS/VS Utilities* for a description of the utility programs.)

System/3 96-column load mode cards must be punched in order to use the output of an RMT generation on a System/3 if the System/3 configuration includes a 5424 multi-function card unit and the RMT parameter $35424 was specified as 1. The 80-column cards are loadable on a System/3 only if a 1442 card reader is attached and the RMT parameter &S35424 was specified as 0.

Instead of the System/3 starter system, any JES2 System/3 remote terminal processor program generated with &S396COL=1 specified may be used to punch RMT generation output that is routed to a System/3.

Input Deck for an RMT Generation

Figure 5-4 illustrates the generation of RTP programs for a System/360 Model 20 workstation and for a System/360 (other than a Model 20) or System/370 workstation.

Starting Remote Job Entry

Since teleprocessing lines are never considered active at JES2 initialization, each line must be activated using a JES2 start command ($S) either by the operator, through a command stream entered, for example, through the JES2 initialization deck, into a job stream, or through the automatic command processor.

```
Column
1

$.RMTM20,2
&RDEV(1)=2560
&RADR(1)=2
&UDEV(1)=2560
&UADR(1)=2
&WDEV(1)=2152
&NUMTANK=5
$.RMTEND
$.RMT360,3
&CMPTYPE=3
&PDEV(2)=1403
&ADAPT=030
&WADR(1)=009
&NUMTANK=7
&CORESIZ=16
$.RMTEND
```

Figure 5-4. Input Deck for an RMT Generation

The first action taken at the non-dedicated remote station is the submission of a sign-on statement. (Sign-on is ignored for dedicated lines.) The format of this statement must be:

Column	Description
1	/*SIGNON
25	REMOTEnnn
25	password1
73	password2

- REMOTEnnn defines the remote station requesting sign-on. *The numbers must be left justified with no leading zeros.*
- Password1 defines the password established at initialization or changed by the operator for that line. If the line has a password, then password1 is required. To establish password1, set the LINEnnn JES2 initialization parameter. This password can be changed or invoked by the operator with the $T command.
- Password2 defines the password established at initialization that is assigned to each terminal. If the terminal has a password, then password2 is required. To establish password2, set the RMTnnn JES2 initialization parameter. The password ensures that the station signing on is a valid station.

A line is dynamically allocated when activated. A line can be deactivated and deallocated using the operator's JES2 stop command ($P).

A remote device is considered active when its remote station becomes active provided that the device is specified for automatic start (by the START subparameter in the Rnnn.RDm, Rnnn.PRm, or Rnnn.PUm initialization parameters). Otherwise, the device is considered inactive and must be started either by the remote or local operator command.

Altering the Sequence of Operations from a Remote Terminal

Two JES2 options are provided to allow the remote terminal operator control of the sequence of operations at the remote terminal.

During JES2 generation, the system programmer can specify a delay time, using the $WAITIME parameter, that will take effect between printing and punching the output of each job. This delay gives the operator the opportunity to ready the card reader and change the terminal status to transmit data. JES2 will snese this condition and read the input stream before resuming printing or punching.

When each printer or punch device is defined at JES2 initialization, using the Rnnn.PRm or Rnnn.PUm parameters, the suspend mode of operation can be specified or negated. If the suspend mode is in effect, the remote operator can alter the sequence of operations by stopping the output device. When the device is again readied by the operator, JES2 will simulate an interrupt situation by flushing its current I/O buffers and printing the remote separator page, if any. JES2 will then determine if the remote card reader is ready. If so, the input will be read in. If not, the highest priority output will be selected. This can be resumption of the current operation or another data set. The delay must be sufficiently long for the terminal to notify JES2 of the stopped device state. The time depends on the terminal type. If suspend mode is not in effect, the current operation is resumed after the device is readed again.

Options for Disconnecting Remote Lines

At JES2 initialization, the system programmer can use the LINEnn statement to choose whether each line is to have the abortive disconnect feature. If the feature is selected, a line is automatically disconnected by simulating a $E command sequence when the transmission control unit detects a not-ready data set. If the feature is not selected, the line will remain active and wait for the data set to be made ready or for operator action. The conditions under which a transmission control unit may detect a not-ready data set are dependent on line configurations.

The system programmer can also cause JES2 to automatically disconnect an inactive station by coding a non-zero value into the DISCINTV parameter of RMTnnn at JES2 initialization. When this amount of time has elapsed with no data sent or received on the line, JES2 will disconnect the line by simulating a $E command sequence.

SMF Accounting Record

SMF accounting records, types 47, 48, and 49, contain information useful for tracking the use of remote stations.

- Type 47 indicates whenever a line is started or a station signs on.
- Type 48 indicates whenever a line is stopped or a station signs off. It also contains statistical information.
- Type 49 indicates whenever a station uses the wrong password when trying to sign on.

CHAPTER 6. MISCELLANEOUS JES2 FACILITIES

The JES2 patching facility, automatic command processing, the flow for time sharing and started tasks, and the multi-access spool are described in this chapter.

Automatic Command Processing

The operator may specify from the console or through a local reader that certain commands or strings of commands take effect automatically at specific times or at regular intervals. The procedures for using the following commands to do this are in *OS/VS2 Operator's Library: Reference (JES2).*

- Start Automatic ($SA): starts automatic command processing.
- Set Automatic ($TA): displays, specifies, or modifies the strings of commands (the "automatic command elements"). This command can also selectively cancel selected commands.
- Cancel Automatic ($CA): cancels all previously entered automatic commands.
- Halt Automatic ($ZA): stops all automatic command processing until it is restarted.

Typical reasons for using automatic command processing are to provide periodic status displays and to cause the operator to do no more work than necessary for common, preset routines or schedules. For example, if it is normal at the installation to do one specific kind of processing at 8 AM, and another 9 AM every morning, it is possible to preset automatic command processing to issue the operator commands that would ordinarily be necessary at those times.

Writing a Day's Work Scheduler

Establish the use of automatic command processing with the &NUMACE parameter at initialization. Enter the commands with the $T operator command or write a program to put cards through an internal reader.

The following statements represent sample cards placed in the initialization deck:

(1) $TA,T=10.30,'$SLNE1,LNE2,LNE3'

(2) $TA,T=12.30,'$I1,ABC;TI2,XBC;L=A'

(3) $TA,T=16.15,'$PLNE1,LNE3;DM1-9, "PLEASE SIGN OFF ASAP"'

(4) $TA,T=16.45,'$ELNE1,LNE2,LNE3'

These four statements mean the following:

(1) Start the three remote job entry lines defined.

(2) Modify these initiators.

(3) Prepare to stop the remote job entry lines and give a warning to users who are currently using the system.

(4) Halt the remote job entry lines.

The sample cards show various times of the day set aside for routine processing that are part of the standard day's work.

A common source of these commands may be a user-written program for scheduling the day's work. This program can use the internal reader to get the commands into the subsystem. To write this program, observe the following considerations:

- When more than one command is to execute at approximately the same time, you should combine them into one command text entry of multiple commands.

- The responses to the command within the text will normally be directed to the in-line messages area and to any consoles receiving MCS route code 1 unless you use the L=caa operand as described in the operator's reference manual cited above.
- Multiple commands with responses to out-of-line area on graphic consoles will normally be automatically overlaid too rapidly for the operator to view. Avoid this kind of command sequencing.
- The authority of the internal reader determines which of your command entered through it will be valid. See the operator's reference manual for discussion of command authority relative to automatic command processing.
- The day's work scheduler program should limit the number of automatic command entries to a value that does not overload the system consoles or leave the operator insufficient resources for his interval status displays.
- An entire automatic command processing entry must fit on an 80-column card image.

Limiting Considerations

When automatic command processing is active, the command entered at system speed (rather than at operator speed) may tend to congest the system. In turn, the system response to the commands may tend to flood the console with the response messages.

If the installation is experiencing difficulty either with congesting the system or flooding the console with messages, re-evaluate the mix of commands submitted with automatic command processing and try some changes.

Automatic command processing may also terminate itself prematurely under the following conditions:

- The operator enters the "$ZA" command to halt automatic command processing and then lets 24 hours or more elapse without restarting it.
- The operator specifies a start time for automatic command processing that is either before midnight of the current day or more than 24 hours later than the current moment.
- The system becomes so congested that the automatic commands are delayed approximately five minutes.

Make sure the operator fully understands the procedures for using automatic command processing. They are fully outlined in the operator's reference manual cited at the beginning of this section.

The JES2 Patching Facility

The JES2 patching facility makes temporary patches to any module in JES2 or to any absolute storage address in the address space into which JES2 is loaded. Because these patches are valid only until a module is reloaded, they must be applied every time that JES2 is started. These patches are applied at the time JES2 is initialized. The patching facility statements are submitted to the JES2 initialization data set.

Modules which are marked refreshable should not be patched since a system refresh will nullify the effect of the patch. Since pages in the Pageable Link Pack Area (PLPA) are not paged out, any patches applied to modules residing in this area will not be effective once the page in which the patch resides has been paged in. For this reason, modules in SYS1.LPALIB data set (for example, HASPSSSM) must be "fixed" via an entry in the SYS1.PARMLIB fixed list before patches are applied via the JES2 patching facility.

The JES2 patching facility in the JES2 initialization data set can be specified in either the JES2 patching format or in the SPZAP format. All patches in the JES2 patching format should appear before any SPZAP format patches. These two methods for patching are explained in the following sections.

Rules for Coding Patching Statements

The following conventions are used in the parameter descriptions:

- Uppercase letters must be coded exactly as shown.
- Lowercase letters represent variables for which you must substitute specific information or specific values.

The following syntax rules apply to the coding of the parameters.

- The size of a patching facility statement is 71 bytes.
- The statements cannot be continued on successive cards.
- The statements may begin in any column, but the operation name must precede the parameters.
- A statement beginning with an asterisk is a comment statement.
- There must be at least one blank between the specified operation name and the first parameter.
- All parameters must be separated by at least one blank space.

Format of the JES2 Patching Facility Statements

The format of the JES2 patch statement is as follows:

```
operation  csect   address  data  comments
```

operation

defines the operation to be performed as follows:

REPLACE
REP
R
The data on the statement will replace the data at the location specified by the "csect" and "address" fields.

VERIFY
VER
V
The data on the statement will be compared with the data at the location specified by the "csect" and "address" fields. If the data does not compare, an error message is displayed in the Parameter Library List data set; subsequent REP operations are still performed. (Note: a verify request will not prevent subsequent REP operations from being performed.)

BASE
B
The base used to adjust address values that are to be specified in any subsequent VER and REP statements is to be modified. This offset is initialized to a value which is based upon the distributed CSECT and assembly module relationships of JES2, and the BASE statement need only be used if this relationship is modified locally. The "data" field on the BASE statement is ignored and may be omitted.

csect

specifies the control section (or control block) in which the data to be verified and/or modified is resident. If an asterisk (*) is coded, the CSECT in effect on the previous JES2 patch statement is used. Figure 6-1 contains a list of the possible names which can be coded and CSECTs to which these names refer. Note that the patch statement name is simply the CSECT name with the first four character (always "HASP") omitted.

address
specifies the hexadecimal address of the data to be verified and/or modified. This address does not have to be aligned in any way and can consist of one to six digits (with or without leading zeros). The address should be taken directly from a JES2 assembly listing containing the referenced CSECT. If an asterisk (*) is coded, the address will be interpreted as one greater than the last address reference on the previous JES2 patch statement.

data
specifies the bytes of data that are to be verified and/or modified at the specified location. The number of bytes of data defined must be specified as a multiple of two hexadecimal digits. If desired, the data within the parameter may be separated by commas (never blanks). If all the data will not fit into one patch statement (71 bytes), then another patch statement must be used.

If the data specified contains the address of a location within a JES2 CSECT, the JES2 patch processing routine will reloacte this data by the base location of the CSECT if indicated. This relocation is indicated by following the data to be relocated with the name of the CSECT (abbreviated as in "csect" above) enclosed in parentheses. The address specified in the "data" should be taken directly from a JES2 assembly listing containing the referenced CSECT. The data to be relocated should contain at least six hexadecimal digits (three bytes), and, if more than six digits are specified only the last eight digits (four bytes) will be considered in the relocation process. If an asterisk (*) is coded instead of a CSECT name, the CSECT in effect for the location of the current patch statement is used.

comments
following the last required parameter and its blank delimiter, the rest of the control statement space can be used for comments.

JES2 Patch Name	AMASPZAP Patch Name	CSECT Referenced
ABS	HASPABS	Absolute Storage Location
ACCT	HASPACCT	HASPACCT
BLKS	HASPBLKS	HASPBLKS
COMA	HASPCOMA	HASPCOMA
COMM	HASPCOMM	HASPCOMM
CON	HASPCON	HASPCON
INIT	HASPINIT	HASPINIT
MISC	HASPMISC	HASPMISC
NUC	HASPNUC	HASPNUC
PRPU	HASPPRPU	HASPPRPU
RDR	HASPRDR	HASPRDR
RDRO	HASPRDRO	HASPRDRO
RSCN	HASPRSCN	HASPRSCN
RTAM	HASPRTAM	HASPRTAM
SSSM	HASPSSSM	HASPSSSM
SSVT	HASPSSVT	Subsystem Vector Table
XEQ	HASPXEQ	HASPXEQ

Figure 6-1. Patch Name to CSECT Reference

Examples of JES2 Patching facility statements:

```
*
*                      CORRECT PROGRAMMING ERROR IN HASPRDR
*
VER RDR       1E2    41E00001                          VERIFY INSTRUCTION
REP   *       1E2    4590B258                          BAL TO PATCH SPACE
VER NUC       258    B258,B25A,B25C,B25E,B260          VERIFY PATCH SPACE
REP   *       258    41202000                          ADD INSTRUCTION
REP   *         *    41E00001                          REPLACE INSTRUCTION
REP   *         *    07F9                              RETURN
 *
 *                     CORRECT BAD ADDRESS CONSTANT IN HASPPRPU
 *
VER PRPU      32E    58F0C65C                          VERIFY INSTRUCTION
REP   *       330    B264                              MODIFY DISPLACEMENT
VER NUC       264    B264,B266                         VERIFY PATCH SPACE
REP   *       264    00000520(PRPU)                    ADDRESS CONSTANT
```

SPZAP Patch Statement Formats

Two formats are required for defining a SPZAP patch. They are the same formats of the control statements for the OS/VS2 AMASPZAP service aid. The first format type defines what module you want to change; the second format type defines what change you want made to the module.

The first format type is used to indicate the control section that is to be the object of subsequent operations. The format of this section is as follows:

NAME member csect comments

NAME
specifies a keyword that must be coded.

member
specifies the member name on the AMASPZAP control statement. This field is ignored on a SPZAP patch statement, but must be provided for AMASPZAP compatibility.

csect
specifies the control section (or control block) in which the data to be verified and/or modified is resident. While this field is optional on the AMASPZAP control statement, it is required on the SPZAP patch statement. Figure 6-1 contains a list of the possible CSECTs which can be coded.

comments
following the last required parameter and its blank delimiter, the rest of the control statement space can be used for comments.

The second format type is used to indicate what operation is to be performed. The format of this section is as follows:

operation offset data comments

operation
specifies the operation to be performed as follows:

REP
The data on the statement will replace the data at the offset into the CSECT specified on the previous NAME statement.

VERIFY
VER
The data on the statement will be compared with the data at the offset into the CSECT specified on the previous NAME statement. If the data does not compare, an error message is displayed in the Parameter Library List data set.

BASE
The base used to adjust offset values that are to be specified in any subsequent VERIFY and REP statements is to be modified. This statement should be used when the offsets given in the VERIFY and REP statements for a CSECT are to be obtained from an assembly listing in which the starting address of the CSECT is not location zero. The "data" field on the BASE statement is ignored and may be omitted.

offset
specifies the hexadecimal displacement of the data to be verified and/or modified in the specified CSECT. This displacement does not have to be aligned in any way and can consist of two, four, or six digits.

data
specifies the bytes of data that are to be verified and/or modified at the specified location. As with the offset parameter, the number of bytes of data defined must be specified as a multiple of two hexadecimal digits. If desired, but again, the number of digits between commas must also be a multiple of two. If all the data will not fit into one SPZAP statement (71 bytes), then another SPZAP statement must be used.

comments
following the last required parameter and its blank delimiter, the rest of the control statement space can be used for comments.

Time Sharing Logon and Starting Task Flow

Time sharing logon and started system tasks appear to JES2 as special form of jobs that are received from designated internal readers. These jobs are enqueued in special job classes (TSU and STC) and are assigned a MSGCLASS that is set during JES2 initialization (TSUMCLAS and STCMCLAS). They are presented to the converter with parameters (&RDROPSU or RDROPST) established first during JES2 generation and later during JES2 initialization.

The time sharing message class (TSUMCLAS) becomes the output class for all dynamically allocated sysout data sets for which a class it not specified, and becomes the MSGCLASS for all submitted jobs with no MSGCLASS parameter in the JOB statement.

Time sharing users can dynamically allocate sysout data sets, dynamically unallocate them (spinoff), and print them at the time sharing terminal (OUTPUT command).

Multi-Access Spool

Previous sections have described JES2 functions on a single system (uniprocessor, MP158, or MP168) operating under a single copy of the MVS control program, as shown in Figure 1-1. It is also possible to operate from two to seven such systems (each a uniprocessor or MP) as members of a multi-access spool configuration, as shown in Figure 6-2.

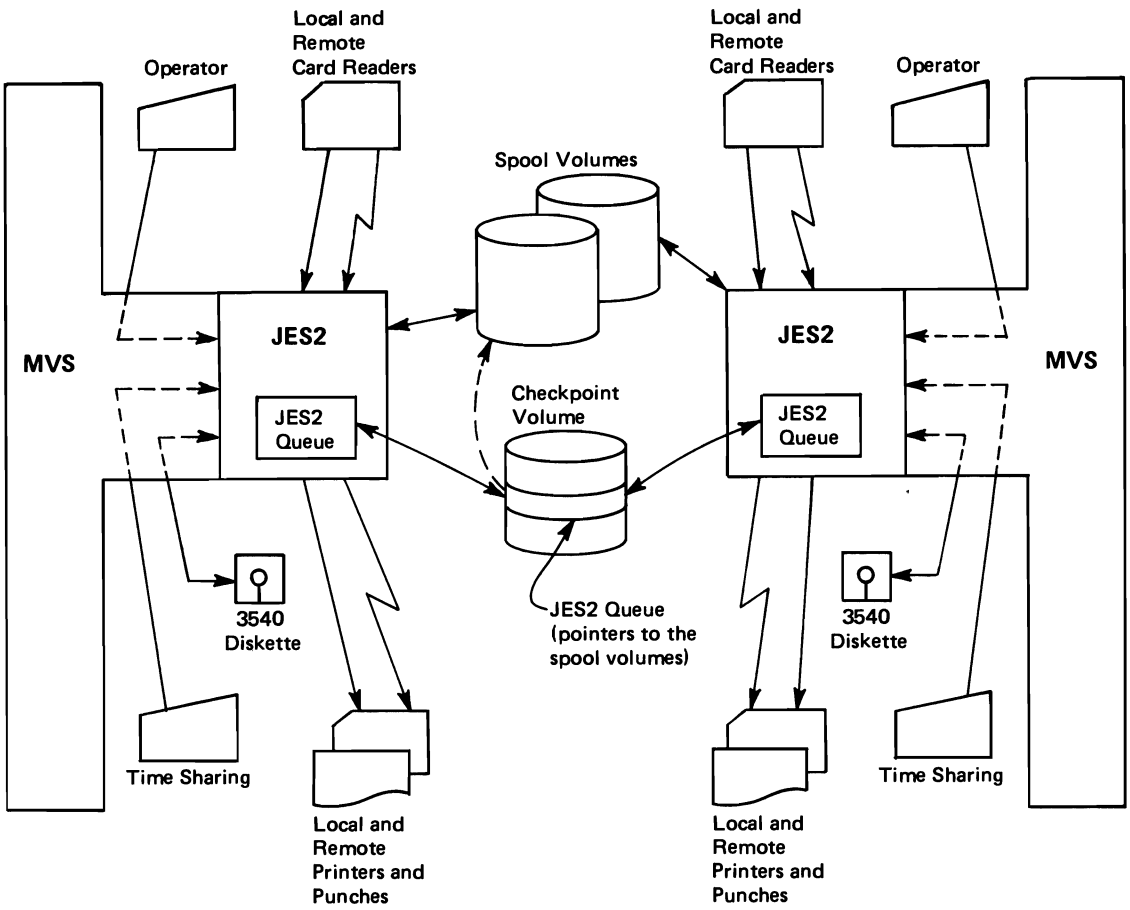
Operator
Local and Remote Card Readers
Local and Remote Card Readers
Operator
Spool Volumes
MVS
JES2
JES2 Queue
Checkpoint Volume
JES2
JES2 Queue
MVS
3540 Diskette
JES2 Queue (pointers to the spool volumes)
3540 Diskette
Time Sharing
Time Sharing
Local and Remote Printers and Punches
Local and Remote Printers and Punches

Figure 6-2. Two-System Shared JES2 Configuration

The operation of each system in the configuration is independent and includes all functions previously described for single JES2 systems. That is, each JES2 system can read jobs from local and remote card readers, schedule jobs for conversion and execution under MVS initiators, print and punch results at local and remote output devices, and communicate with operators and time sharing users. However, all spool volumes and the volume containing the SYS1.HASPCKPT data set are used by all system in the configuration.

The systems logically share a common JES2 queue. The workload may be balanced among systems by allowing jobs to execute on whatever system has an idle initiator with the correct class and print or punch, on whatever system has an idle device with the correct class, routing, setup, etc.

Since all systems are functionally the same, if one system in the complex fails, the others may continue processing from the common queue. Only work in process on the failed system is interrupted; this work may be recovered by a warmstart of the failed system while other systems continue processing, or, as explained later, by operator command on one of the other systems.

Shared DASD hardware features (two channel switch, two channel switch additional, and string switching) are used to access data on all spool and checkpoint volumes. A copy of the JES2 queue and other status information (for example, spool space allocation maps) is written to the SYS1.HASPCKPT data set for possible warmstart, as with a single JES2 system. This information is available to all systems, one at a time, as needed.

RESERVE/RELEASE channel commands are used to prevent simultaneous referencing and updating of information kept in the SYS1.HASPCKPT data set.

Each system in the complex must have at least one channel path to each spool and checkpoint volume, and these devices must be specified as SHARED during MVS system generation. It is recommended that each CPU of an MP158 system in the complex have a channel path to each shared volume.

Configuration

To use the multi-access spool feature, the initialization or generation &SPOOL parameter and the &CHKPT initialization parameter must specify the same volumes for all systems in the complex. To make the common spool and checkpoint data compatible, all systems must specify the same values for the &BUFSIZE, &NUMDA, &NUMTGV, &MAXJOBS, &NUMJOES, &NUMRJE, and &SPOLMSG generation parameters.

For operational consistency, it is recommended that the &MINJOES, &TGWARN, &XBATCH, and &XBATCHN generation parameters be specified the same in all systems of the configuration.

It is also recommended that local unit record devices and RJE lines be given unique JES2 device names over the whole complex. The &NUMLNES, &NUMPRTS, &NUMPUNS, and &NUMRDRS generation parameters of each JES2 system should be specified as the total number of each type of device in the configuration. This allows all devices to be attached to one system (with appropriate manual switching) if other systems are not operational.

Similarly, the LINEnn, PRINTERnn, PUNCHn, and READERn initialization parameters should be set so that a device has the same name no matter which system it is attached to. For example, if a 3211 printer is one of four local printers on a two-system, configuration it could be initialized as:

```
PRINTER4 UNIT=102
```

for one system and:

```
PRINTER4 UNIT=302
```

for the other, if it were attached to different channels on the two systems.

A local unit record device or RJE line can only be attached to one system at any instant. JES2 initialization will detect devices which are not online and place them in a DRAINED state. Later, the device may be activated by entering the $P device and VARY OFFLINE commands on the system to which it is attached, performing hardware switching, then entering the VARY ONLINE and $S device commands on the new system. The $S command will fail if no hardware path exists.

The &NUMRJE generation parameter must be the same in all systems of the configuration, as previously described. This parameter represents the total number of RJE lines known to the entire configuration. Each RJE line has a unique name, no matter which system it is attached to. Therefore, the RMTnnn, Rnnn.RDm, Rnnn.PRm, and Rnnn.PUm initialization parameters should be specified the same in all JES2 systems of a multi-access spool configuration.

Starting the Multi-Access Spool Configuration

Before starting the configuration, the TOD clocks on each system should be carefully synchronized with a single time source. Since this synchronization is externally performed and subject to error, the generation parameter $SYNCTOL is provided to specify the maximum error (in seconds) which JES2 should assume. If the synchronization error is actually greater than $SYNCTOL, then JES2 will not be able to detect certain illegal operator actions (for example, performing cold start with other systems active). On the other hand, certain legal operator actions (for example, warm start after system failure) will be disallowed if attempted before $SYNCTOL seconds have elapsed since system failure.

The members of the configuration are specified by the Sn initialization parameters. For example:

```
S1 SID=K158
S2 SID=L168
```

defines a two-system configuration where K158 and L168 are the SMF system ids set during IPL of the systems. One system must initially do a cold JES2 start with no other systems active and must define all members of the configuration. Other members join operation by warmstart and must also specify identical Sn parameters. A cold start is required to change or add members of the configuration. If only one or no Sn parameter is specified, JES2 operates as a single system.

There are three types of warm starts:

- If a warm start is specified by the operator and JES2 detects that no other members of the configuration are active, after operator confirmation a total configuration warm start is performed. New spool volumes may be added, all in-process work will be recovered, and all unused spool space will be accounted for, as in single system operation.
- A warm start is performed when warm start is specified and other members of the configuration are active. The warmstarting system joins the active configuration and recovers only work in-process on the system at a previous failure, if any. No spool volumes may be added.
- Restart for another system is performed when a system has failed and cannot be immediately warm started. The operator enters the $ESYS command on any active member of the configuration. In-process work on the specified system is recovered and made available for selection by other members of the configuration, subject to system affinity for execution restart as discussed later under "Job Submission and Queueing."

The algorithm for using the common JES2 queues and other information in the SYS1.HASPCKPT data set is determined by the HOLD=, MINDORM=, and MAXDORM= keywords of the QCONTROL initialization parameter. These need not be the same for all systems in the configuration and should be set according to characteristics such as the number of members in the configuration relative CPU speeds, and response requirements. See the chapter "JES2 Initialization" for details.

Job Submission and Queueing

In a multi-access spool configuration, jobs enter the common queue from any input source (local or remote) attached to any system in the configuration. Unless special actions are taken, jobs will normally be eligible to execute in any system in the configuration, selected by priority and the classes of idle initiators as in single system operation.

Started tasks and TSO users are an exception which execute only in the system in which they are entered. However, job queue entries also contain a system affinity for up to seven systems on the maximum configuration and may contain an independent mode affinity.

Individual jobs may be given affinity to one or more systems (less than the total configuration) and may be given affinity for independent mode by the SYSAFF=keyword on the /*JOBPARM card. Any input device (local or remote) may be set by the $T command to give system and/or independent mode affinities to all jobs read from that device. The /*JOBPARM card overrides the input device default.

If a job's affinity is to specific systems in the configuration or to independent mode, the job can be selected only by the system(s) specified and only if the mode of the system (independent or not) matches that of the job.

System affinity may be useful for special processing requirements (for example, emulation) not available on all system of the configuration. Independent mode may be useful for testing new components with selected jobs while in a shared configuration.

The display commands ($DA, $DN, $DQ, $DJ) indicate (by SMF system id) the system in which a job is active or the system(s) eligible to process a queued job. The $TJ and $TALL commands permit affinities of jobs or all jobs with given affinity to be changed. The $TSYS command allows a system to be placed in independent mode. The $LSYS command displays the states of all systems in the configuration.

If a system fails and jobs in execution are recovered and requeued for automatic restart either by a warmstart or the $ESYS command, those jobs are given affinity only to the failed system. If the failed system is unavailable, the operator may change affinity with the $TJ or $TALL commands to attempt restart on another system.

Priority aging is done only by the lowest numbered active system in the configuration.

Duplicate job name protection extends to all systems; i.e., if a job name matches another active in execution anywhere in the configuration, the job is temporarily delayed. See the TSO section that follows.

Output

Printed and punched output processing is very little different from single system operation. System affinity does not apply to selection of work from the JOEs.

Output work is selected by eligible devices, no matter to which system in the configuration devices are attached. Selection criteria are output class, routing (local or remote number), and set up just as in single systems. The automatic setup algorithm which prevents the same special forms from being requested for more than one local printer operates for all local printers in the configuration.

The $CJ command entered from any system in the configuration will cancel a job active on an input or output device attached to another system.

RJE

Configuration considerations for RJE lines were discussed previously.

JES2 enforces that the same remote number cannot sign on more than one line anywhere in the configuration at any given time. For dedicated lines, the user must insure this uniqueness by proper setting of line and remote initialization parameters as previously described.

The remote operator message queue operates across the entire configuration. That is, any remote operator can send messages to any other remote (even if attached to different systems) and any central operator can send a message to any remote.

TSO

TSO userids are job names to JES2 and, in a multi-access spool configuration, the duplicate job name protection extends across the configuration. A TSO logon will be rejected if a user of the same id is logged on elsewhere in the configuration.

Jobs submitted by TSO users may execute anywhere in the configuration, subject to affinities as previously discussed. However, held output data sets are accessible by the TSO OUTPUT command by the submitting user regardless of where logged on or where the job executed. Messages produced by NOTIFY= are also returned to whereever the TSO user is logged on or to where the job was submitted from, if the user is no longer logged on.

SMF

The SMF type 26 record contains system ids indicating which systems in the configuration performed each major function of processing for a job: input, convert, execute, post-execute break into output elements, and purge.

The SMF type 6 records contain the system id which processed each element of output work.

CHAPTER 7. JES2 PERFORMANCE

This chapter describes four aspects of JES2 performance:

- Factors affecting JES2 performance
- How the operator can control the batch job workload
- Comparison between HASP II Version 4.0 and JES2 performance factors
- Changing JES2 from nonswappable to privileged

For related information, such as discussion of job output elements (JOEs) and the the use of job classes under JES2, refer to the chapter "JES2 Processing". Additional information on JES2 output facilities, the external writer and JES2/HASP differences is discussed in "Chapter 4 JES2" of the *OS/VS2 Conversion Notebook.*

JES2 Performance Factors

JES2 performance depends on the careful specification of at least four generation parameters (&BUFSIZE, &NOPRCCW, &NUMJOES, and &NUMWTOQ*), page alignment of JES2 csects, the allocation of sufficient spool space, selection of the spool device(s), and using unblocked records for SYSIN and SYSOUT data sets.

&BUFSIZE Parameter

This parameter specifies the size in bytes of each JES2 buffer. The two recommended values of &BUFSIZE are a compromise between best performance and optimum device utilization:

- Value of 1960 for spool volumes on 2314. This value becomes a half page. (A full page wastes spool space and will likely increase seek time significantly.)
- Value of 4008 for spool volumes on 3330. This value becomes a full page, which allows three records per 3330 track.

Note: *The 2305 fixed head device is not recommended for holding buffers.*

Avoid reducing the buffer size below 1960, since the CPU time to print the buffers will increase. Larger &BUFSIZE values increase performance by requiring increased blocksize. However, avoid increasing buffer size excessively, since buffers are not allowed to cross page boundaries. A value of 4008 is the maximum.

&NOPRCCW Parameter

This parameter specifies the maximum number of channel command words per channel program for local printers. The value should be so chosen that all print lines in a spool buffer can on average be printed with a single channel command. You can compute this value from the formula:

&NOPRCCW=&BUFSIZE/average line length

Estimate the average line length, allowing for truncation of trailing blanks by JES2.

If the value is too small, the CPU time for printing will increase. If, on the other hand, the value is grossly overspecified, the size of the address space will increase, requiring more page space and potentially more page faults.

&NUMJOES Parameter

This parameter specifies the number of job output elements (JOEs) to be generated for the queueing of work for printers and punches. Each JOE takes 26 bytes of virtual storage. If the value is set too small, jobs will wait excessively to be eligible for printing or page space and potentially more page faults.

*&NUMWTOQ is both a generation parameter and an initialization parameter.

punching. If the value is set too large, the size of the address spaces will increase (since job output elements are in virtual storage), and the CPU time and the number of page faults needed to search the elements will increase. The default value is ten times the maximum number of printers and punches, both local and remote, plus ten times the number of user-written writers. This value should keep printers and punches busy without tying up too much virtual storage. As a rough approximation, you can determine the starting value as 2N JOEs per job, where N is the number of output classes per job. (For further discussion of the factors affecting the number of job output elements, see the chapters "Installing JES2" and "JES2 Processing."

&NUMWTOQ Parameter

This parameter specifies the number of message buffers for JES2. A value of 24 can serve as a starting value. If it proves too small, it can be increased in multiples of 24. If the value is too small, the system is slowed because console messages must wait for buffers. If the value is too large, excessive common service area (CSA) virtual storage becomes allocated and is thus unavailable for other uses. As a rough approximation, estimate the value for &NUMWTOQ as:

≅ 2 times &MAXPART + the number of readers (local, remote, and internal)

+ the number of printers and punches (local and remote).

Primary JES2 Spool Volume

If the primary JES2 spool volume (&SPOOL JES2 initialization parameter) is not mounted and ready when JES2 is started, a message is issued to request mounting the volume, and JES2 is terminated. Because the requested MOUNT command processing requires that JES2 be already initialized, the mounting operation is not possible unless the operator makes the spool volume ready and performs a re-IPL. To avoid this situation, you should include an entry (that specifies the required spool without suppressing the mount option) in the VATLSTnn member of SYS1.PARMLIB. VATLST processing will then request volume mounting as part of the IPL process.

Page Alignment of JES2 Control Sections

Alignment of the HASPINIT csect on a page boundary is mandatory. The purpose is to place buffers on a page boundary. The alignment is handled through the Linkage Editor, via the standard process. Further improvement is possible by aligning the other JES2 modules so that csects do not cross page boundaries. By this process unnecessary page faults can be minimized during JES2 execution.

Allocation of Sufficient Spool Space

JES2 allocates spool space for a job by dividing each spool volume into *track groups.* A track group is a group of DASD tracks whose size is indirectly specified by the &NUMTGV parameter at JES2 generation. (The &NUMTGV parameter specifies the number of track groups per volume, from which JES2 calculates the track group size). JES2 allocates to a job one track group at a time. When a given track group is exhausted, the next track group that is allocated is the closest one to the last one used. Seek time is therefore minimized.

As a rough approximation, you can compute spool space by noting that a 2314 pack can hold approximately 200,000 lines of output, plus normal input. A 3330 pack can contain approximately 700,000 lines of output, plus normal input. (These figures assume that an output line contains 120 non-blank characters.) If sufficient spool space is not allocated, performance will degrade, since jobs will periodically have to wait for spool space. (For detailed information on how to compute spool space, see *OS/VS2 System Programming Library: Storage Estimates.)*

Selection of Spool Devices

The three types of JES2 spool volumes are the primary spool volume, secondary spool volume, and checkpoint spool volume. The primary spool volume contains:

- JES2 control blocks
- Job input and output data
- Spool message queue records (for remote terminals)

The secondary spool volume contains:

- JES2 control blocks
- Job input and output data

The checkpoint spool volume contains the checkpoint records (which were formerly located on the primary spool volume). These records are in the data set SYS1.HASPCKPT.

When selecting devices for spools, consider that the volume(s) which contain job input and output data should preferably go on one or more 3330s. This device type has reasonable speed, rotational position sensing and good capacity. For best performance it is desirable to dedicate spool volumes (that is, don't share a volume with paging data sets or other non-spool data sets), so that JES2 can do ordered seeks. If there is more than one spool device, they should be put on separate channels. The channel need not be dedicated, however, since JES2 channel utilization is low.

Because the checkpoint records are in a separate data set, the space allocation for spooling can be evenly spread across the primary and secondary spool volumes. This space is is defined as the first extent of SYS1.HASPACE on each of the spool volumes. Note that this space on the primary spool volume must be sufficient to provide for the spool message queue records. The &BUFSIZE JES2 generation parameter defines the length of each record. The spool message queue record area is calculated as &SPOLMSG x &NUMRJE.

The checkpoint data set, SYS1.HASPCKPT, should be located on a high speed direct access device with low activity to improve performance. The JES2 initialization parameter, &CHKPT, indicates the volume serial number of the checkpoint volume. The checkpoint data set should be allocated as a single extent within one cylinder. JES2 uses only the first extent. For further information about space allocation refer to the description of SYS1.HASPCKPT in the chapter "Installing JES2."

Use of Unblocked Records for SYSIN and SYSOUT Data Sets

If the installation is using BSAM, it may not be desirable to specify that SYSIN and SYSOUT data sets be blocked. Otherwise, the SAM Compatibility Interface wll increase overhead by unnecessarily deblocking and blocking sets.

Held Internal Readers in JES2

All internal readers are treated as a single facility. Thus, if one is held, all internal readers are held. This can be particularly troublesome if TSO users are submitting jobs and the central operator has held the internal readers. This can be overcome by several operating techniques.

1. All jobs submitted via an internal reader can be assigned a class and that class can be held via a JES2 Parameter Library entry or the $HQn operator command.
2. Jobs submitted via an internal can use the TYPRUN=HOLD parameter on the JOB card.
3. Job submitted via an internal reader can be individually held with the $HJ operator command.

How the Operator can Control the Batch Job Workload

It may be a good idea to always have more batch jobs initiated and more time sharing users logged on than the number of address spaces that will fit together in real storage. The purpose of such over-initiation is to allow the System Resources Manager (SRM) a varied mixture of swapped-out jobs to choose from whenever any resouce becomes under-utilized. On the other hand, when a resource becomes a bottleneck, the SRM can remedy the problem by swapping out the heavy resource-using job(s).

From the resource management point of view, swapping an address space out of real storage is an ideal control mechanism, since the associated job immediately stops using the three main system resources—CPU, real storage, and I/O paths. The only resource a swapped-out address space continues to hold is allocated auxiliary storage. Thus, it may be more practical to cause some jobs to be selected, initiated, and swapped out, rather than to have them remain unselected on the job queue.

You may feel that over-initiation may ccompete with and slow up the progress of the more important jobs. However, this concern is addressed by the SRM's Workload Manager, which tries to ensure that individual jobs are processed according to the jobs' PERFORM parameter and related parameters in the IPS.

The operator can ensure that the System Resource Manager has a sufficient number and variety of jobs to keep the system busy. He can do this by varying the number of logical initiators and the classes from which they dequeue jobs. The mechanisms for such control are these JES2 commands: Start Initiator, Stop Initiator, Set Initiator, and Halt Initiator ($s Inn, $t Inn, and $z Inn).

Each logical initiator controls the selection of one job at a time. The maximum number of logical initiators is specified at JES2 generation by the &MAXPART parameter. During JES2 initialization, the Innn-sublist parameter assigns classes to logical initiators. Each initiator is given a status of started or "drained" that is, stopped. (For information on JES2 generation parameters, see the chapter "Installing JES2.")

JES2 indirectly causes an address space to be created for each "active" logical initiator. The operator may activate logical initiators and cause additional batch job address spaces to be created, by means of the JES2 Start Initiator command, $s Inn. Such increase, is of course, subject to the maximum number of logical initiators specified by the &MAXPART parameter. In a similar manner, the operator may "drain" logical initiators and cause termination of their address spaces by issuing the JES2 Stop Initiator command, $p Inn.

The operator can inactivate a logical initiator, but not terminate the address space (which will be swapped out), by issuing the Halt Initiator command, $z Inn. Some time is saved, since the address space need not later be recreated through the Start Initiator command, $s Inn.

The four commands useful in controlling batch job workload are summarized functionally in Figure 7-1. For syntax information regarding the $t, $s, $p, and $z commands, refer to *Operator's Library: OS/VS2 Reference (JES2).*

Comparison of HASP II Version 4.0 and JES2 Performance Factors

JES2 performance factors in MVS are compared with similar factors in HASP II Version 4.0 under VS2 Release 1.

Desired Function	JES2 Operator Command
Control the number of batch jobs that are executable at the same time, by assigning classes from which these jobs can be selected.	Set Initiator: $t Inn
Activate logical initiators and cause creation of additional batch job address spaces.	Start Initiator: $s Inn
Stop ("drain") logical initiators and cause termination of batch job address spaces.	Stop Initiator: $p Inn
Inactivate a logical initiator, but not terminate the address space.	Halt Initiator: $z Inn

Note: The initiator identifier nn represents a one-character or two-character ID. Some examples are 1, 2, cd.

Figure 7-1. JES2 Commands Useful in Controlling the Batch Job Workload

Fixed Storage

HASP II Version 4.0 fixes a minimum of three pages to satisfy EXCP requirements of VS2 Release 1. It also fixes space for commonly used control blocks.

JES2 does not long-fix any pages. Pages are fixed only for I/O.

Region Size

A large part of JES2 is common with HASP II Version 4.0 Changes in the size from the HASP region are not expected to introduce noticeable performance changes.

Checkpoint Records

Checkpoint records on the spool volumes are not compatible between HASP II Version 4.0 and JES2. Therefore neither HASP II Version 4.0 nor JES2 can be warmstarted from the other's spool pack.

Spool I/O

HASP II Version 4.0 uses pseudo devices to process SYSIN and SYSOUT data sets. A cross-region POST is issued for I/O when a buffer is full. Spool I/O is organized to minimize seek time.

In JES2, spool I/O is from the job's address space. Pseudo devices are not neded. No cross-region POST is needed. Task switch time is reduced. Spool I/O also minimizes seek time, if the secondary spool volume is not shared with non-spool data sets.

Changing JES2 from Nonswappable to Privileged

In the generated system, JES2 is automatically marked *nonswappable* in the program properties table. Under the following conditions you may improve performance by changing the JES2 attribute from *nonswappable* to *privileged.*

You should consider this modification only if your installation's job stream has *both* these traits:

1. Batch jobs only, *and*
2. Extended periods during which jobs are neither read in, scheduled, nor written out.

Do no make JES2 *privileged* if you plan to do any remote processing, either conversational or remote batch.

By making JES2 privileged, you can cause the system to swap JES2 out, if no JES2 activity has occurred in a 10-second interval. Such swapout would permit swapin of a job step awaiting real storage availability.

The disadvantage of this modification is a short delay to swap JES2 in when it is needed; that is, during Job Select, or when the operator issues a JES2 command, a START command, or a MOUNT command, or when an internal reader is allocated.

The method by which you can modify entries in the program properties table is described in *OS/VS2 System Programming Library: Job Management.*

Questions and Answers

The following JES2 questions have been asked by users.

Does JES2 support FCB loading for System/360 or System/370 Remote Workstations?

If FCB loading capability is indicated for a remote printer via the JES2 Rnnn.PRm initialization parameter, the same FCB image that would have been loaded locally is transmitted to the remote workingstation. If the related printer is 3211, the image will be loaded as received by the remote workstation. If the related printer is a 3203 or 5203, the FCB image will be stripped of its indexing byte before loading. UCS loading, via stand-alone utility, remains the responsibility of the remote operator.

Can user-written output writers be invoked in MVS?

In MVS JES2 spools all SYSOUT data. The OS/MVT writer has been modified to operate MVS, and is called an External Writer. IBM supplies a procedure, XWTR, which the operator can invoke to start the external writer. All of the documented interfaces for user-written output writers for OS/MVT and OS/VS2 Release 1 still apply to MVS. The External Writer control program requests spooled data sets from JES2, based upon various selection criteria, via a formal subsystem interface. User-written separator routine interfaces for the External Writer also remain the same for MVS as with previous OS releases. (Note tha JES2 will not process data sets that specified a user-written writer name except by the external writer facility.)

Will the current HASP MULTI-LEAVING work station programs operate correctly with JES2?

Yes, but with some limitations. The HASP Versions 3.1 and 4.0 work station programs can be used with JES2 as a transition aid. The Version 3 System/3 program will operate unpredictably if punch jobs with 4-digit job numbers are sent to it. The Version 3 and 4 System/360 and Sytem/370 programs will operate unpredictably if an FCB image is sent to them. All Version 3 and 4 work station programs will operate unpredictably when the new disconnect control record is sent to them at the end of a session. It is strongly recommended that all work station programs be regenerated from JES2 libraries to realize the benefits of new feature support and to insure that the latest maintenance level is obtained.

Can JES2-supplied MULTI-LEAVING workstation programs be used with prior versions of HASP?

Yes. But certain new functions (for example, signoff control record and FBC support) are partly implemented in JES2 are are not supported in prior versions of HASP.

Does JES2 support the 3781 punch?

Yes, the JES2 RMTnnn initialization parameter can be used to define a 3780 terminal. Then, if a 3781 punch is attached to the terminal, the NUMPU subparameter should be set to 1.

Should the user specify blocked sysout data?

No, JES2 automatically blocks sysout data. The user should specify DCB parameters on the DD card if the executing program requires them. BLKSIZE, RECFM, and LRECL should indicate unblocked records where possible.

What remote terminal support is currently available for the System/3?

The System/3 MULTI-LEAVING Remote Job Entry Work Station (MRJE/WS) program feature operates under the Model 6 System or Model 10 Disk System (with or without the Dual Programming Feature). MRJE/WS communicates with HASP (Version 3.1 or 4.0) or JES2 over point-to-point (switched or non-switched) communication lines via the BSC Adapter. MRJE/WS supports full MULTI-LEAVING of reader, console input, console output, printer, and punch data streams. A minimum program partition of 8.25K is required.

A variety of physical devices can be assigned to the logical processors, including disk and tape input and output. Individual input and output files need not be defined at program load time, but can be dynamically allocated as required during a session. Individual printer and punch data sets can be directed to tape or disk; or an entire stream can be directed to disk, which facilitate DPF operation and may reduce line connect time.

Hardware configuration and system operation are described in the *IBM System/3 MRJE/WS Support Reference Manual*, GC21-7621.

How can you pool remote output devices with JES2?

JES2 supports logical pooling for remote output devices so that the output devices can be used more efficiently. For example, an installation has two or more remote workstations physically located in the same vicinity and the user wants to be able to submit jobs through any of them and not be concerned with which one receives the output. Remote workstations 45 and 71 can be pooled by specifying the following JES2 initialization parameter:

```
RMT71 ROUTECDE=45
```

The output routine code of 45 now applies to both workstations. Output is returned to either, and the operator at either workstation can control devices and job output at both workstations.

Responses to operator commands are made to the remote workstation that enters the command unless the response is spooled (message spooling). If one of the workstations does not have a console, all nonspooled responses can be directed to the workstation with a console by also specifying:

RMT71 CONDEST=45 if workstation 45 has the console, or:

RMT45 CONDEST=71 if workstation 71 has the console

Spooled responses, like other print data sets, are routed to either workstation.

GLOSSARY

This glossary defines JES2 terms and other data processing and communication terms used in this publication. For definitions of terms not included in this glossary, see *IBM Data Processing Glossary,* GC20-1699.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the *American National Standard Vocabulary for Information Processing* (copyright © 1970 by American National Standards Institute, Inc.), which was prepared by Subcommittee X3K5 on Terminology and Glossary of American National Standards Committee X3. A complete commentary taken from ANSI is identified by an asterisk (*) that appears between the term and the beginning of the commentary; a single definition taken from ANSI is identified by an asterisk after the item number for that definition.

A

address space. The complete range of addresses that is available to a programmer.

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

automatic mode. The mode of operation in which the setup and selection of jobs on a printer is controlled by JES2 rather than by the operator through the use of operator commands.

B

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

C

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

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

checkpoint. (1) * A place in a routine where a check, or a recording of data for restart purposes, is performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later.

cold start. (1) The initialization procedure that causes an operating system to commence operation. (2) Synonymous with initial program load.

D

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

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

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

E

external writer. In OS/VS2, a program that supports the ability to write SYSOUT data in ways and to devices not supported by the job entry subsystem.

H

HASP. Houston automatic spooling program. A computer program that provides supplementary job management, data management, and task management functions such as control of job flow, ordering of tasks, and spooling. See also JES2.

I

initial program load (IPL). Synonym for cold start.

J

JES2. A functional extension of the HASP II program that receives jobs into the system and processes all output data produced by the job.

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

job entry subsystem (JES). A system facility for spooling, job queuing, and managing the scheduler work area. See also JES2.

job output element (JOE). Information that describes a unit of work for the HASP or JES2 output processor and represents that unit of work for queuing purposes.

JOE. Job output element.

M

multi-access spool configuration. Two to seven systems sharing the JES2 input, job, and output queues through the use of shared DASD.

P

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

R

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

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

remote terminal. (1) A terminal attached to a system through a data link. (2) In telephony, a terminal attached through a trunk or tieline.

remote terminal access method (RTAM). A facility that controls operations between the job entry subsystem (JES2 or JES3) and remote terminals.

RJE. Remote job entry.

RMT generation. Generation of remote work stations for remote job entry.

routing. The assignment of the communications path by which a message or telephone call will reach its destination.

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

RTAM. Remote terminal access method.

RTP. Remote terminal processor.

S

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

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

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

system restart. A restart that allows reuse of previously initialized input and output work queues. Synonymous with warm start.

W

warm start. Synonym for system restart.

INDEX

OS/VS2 MVS System Programming Library: JES2 (File No. S370-36) Printed in U.S.A. GC23-0001-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)