GC26-3792-1
File No. S370-34

Systems

OS/VS2 System Generation Reference

VS2 Release 1

IBM

Second Edition (March 1973)

This edition replaces the previous edition (numbered GC26-3792-0) and its technical newsletter (numbered GN26-0760) and makes them both obsolete.

This edition applies to Release 1.6 of OS/VS2 and to all subsequent releases until otherwise indicated in new editions or technical newsletters. Changes to the information in this book may be made at any time; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/360 and System/370 Bibliography,* GA22-6822, the latest *IBM System/370 Advanced Function Bibliography,* GC20-1763, and the technical newsletters that amend those bibliographies, to learn which editions and technical newsletters are applicable and current.

Significant system changes are summarized under "Summary of Amendments" following the list of figures. Each technical change is marked by a vertical line to the left of the change.

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

Forms for reader's comments are provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Center – Publishing, Department D58, Monterey and Cottle Roads, San Jose, California 95193.

PREFACE

This publication is intended for system programmers who are to generate an OS/VS2 System Control Program. This publication does not explain the concepts underlying VS2 operations or the options available in VS2. Information of this type is contained in the following publications:

OS/VS2 Planning and Use Guide, GC28-0600

OS/VS System Generation Introduction, GC26-3790

IBM System/370 System Summary, GA22-7001

This publication provides you with the information necessary to:

- Prepare for system generation
- Execute system generation
- Restart system generation if errors occur during processing
- Test the newly created System Control Program

The information in this publication is divided into nine chapters and six appendixes.

"Introduction," briefly defines system generation and discusses the three types of system generation: a complete system generation, a nucleus generation, and an I/O device generation. It also provides a list and explanation of the procedures for generating a VS2 System Control Program.

"Requirements for Generating a New System Control Program," is divided into three sections. The first section gives the real-storage requirements for system generation. The second section discusses the machine requirements for system generation using the starter system. The third section tells about the programming requirements for system generation.

"Specifying the New System Control Program," is divided into two sections. The first section explains the conventions used in this publication to present the system generation macro instructions. The second section discusses the system generation macro instructions and what must be considered in specifying the new System Control Program. Included are:

- The required and optional macro instructions for the three types of system generation
- A list of the system generation macro instructions by type and in the order they should be considered
- An explanation of each system generation macro instruction, given alphabetically by the macro instruction name. The format, parameters, and defaults of each macro are explained. Whether each macro instruction is required or optional is also shown
- A coding example of each macro instruction

"Selecting the New System Data Sets," describes the procedures for allocating space to the system data sets and cataloging them in the system catalog. Also included is a description of each of the required and optional system data sets and what must be considered in selecting and specifying them, and a summary description of the system data sets.

"Preparation for System Generation," tells what to do before system generation. Included are discussions and examples of direct-access volume initialization, preparatory steps for using both the starter system and an existing VS2 system as the generating system, and an explanation of how to add your own routines to the new system during system generation.

"System Control Program Installation," describes Stage I input, execution, and output, and Stage II input, execution, and output.

"Restart Procedures," gives the procedures for restarting system generation if processing ends unsuccessfully during either Stage I or Stage II.

"Testing the System Control Program," gives the procedures used to test the newly installed System Control Program using the installation verification procedure (IVP).

"Examples of System Generation," presents examples that illustrate the generation of a particular type of System Control Program and the various types of system generation.

Appendix A, "Device Types," lists and describes the device types that can be specified in the UNIT parameter of the IODEVICE macro instruction.

Appendix B, "Description of the Starter System and Distribution Library Tapes," discusses the contents of the starter system tape and distribution library tape, and lists the contents of the distribution library tape.

Appendix C, "System Generation Messages," lists and explains the messages that are produced by the assembler program during Stage I.

Appendix D, "Functionally Equivalent Devices," lists I/O device types that are functionally equivalent and discusses how they are specified in an IODEVICE macro instruction.

Appendix E, "Diagnostic Override," gives the macro instructions that can be used to override errors that may occur during Stage I processing.

Appendix F, "Console Sheet of the Procedures for Starting the Starter System," shows the commands and responses that are printed on the console during the procedure for making the starter system operational.

The following items are described in this publication for planning purposes only, and will not be available with the initial release of VS2:

- System/370 Model 155II
- System/370 Model 165II

Availability dates for support of the above items may be obtained from the local IBM branch office.

The following publications provide detailed explanation of job control language, utility programs, and coding macro instructions:

OS/VS and DOS/VS Assembler Language, GC33-4010

OS/VS JCL Reference, GC28-0618

OS/VS Utilities, GC35-0005

The following publications are referred to in this manual. They provide detailed information about topics associated with system generation:

OS/VS Assembler Programmer's Guide, GC33-4021

OS/VS BTAM, GC27-6980

OS/VS Checkpoint/Restart, GC26-3784

OS/VS Data Management Macro Instructions, GC26-3793

OS/VS Data Management for System Programmers, GC28-0631

OS/VS Data Management Services Guide, GC26-3783

OS/VS2 Debugging Guide, GC28-0632

OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit, GC27-6971

OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment), GC27-6972

OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I, GC27-6973

OS/VS JCL Services, GC28-0617

OS/VS Linkage Editor and Loader, GC26-3813

OS/VS SYS1.LOGREC Error Recording, GC28-0638

OS/VS Message Library: Service Aids and OLTEP Messages, GC38-1006

OS/VS Message Library: Routing and Descriptor Codes, GC38-1004

OS/VS Message Library: VS2 System Messages, GC38-1002

OS/VS Message Library: VS2 System Codes, GC38-1008

OS/VS OLTEP, GC28-0636

Operator's Library: OS/VS Console Configurations, GC38-0120

Operator's Library: OS/VS2 Display Consoles, GC38-0260

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

Operator's Library: OS/VS TCAM, GC38-0305

Operator's Library: OS/VS2 TSO, GC38-0220

OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP, GC27-6974

OS/VS Recovery Management Support Logic, SY27-7239

OS/VS Service Aids, GC28-0633

OS/VS2 Storage Estimates, GC28-0604

OS/VS2 Supervisor Logic, SY27-7244

OS/VS Supervisor Services and Macro Instructions, GC27-6979

OS/VS2 System Data Areas, SY28-0606

OS/VS System Management Facilities (SMF), GC35-0004

OS/VS Tape Labels, GC26-3795

OS TCAM Concepts and Facilities, GC30-2022

OS/VS TCAM Programmer's Guide, GC30-2034

OS/VS2 TSO Command Language Reference, GC28-0646

OS/VS2 TSO Guide, GC28-0644

OS/VS2 TSO Terminal User's Guide, GC28-0645

An introduction to the IBM 3270 Information Display System, GA27-2739

CONTENTS

FIGURES

SUMMARY OF AMENDMENTS

Release 1.6

- VS2 Release 1.6 now fully supports DSS (dynamic support system) and HASP.
- VS2 Release 1.6 now fully supports the System/370 central processing units, Model 158 and Model 168.
- The SIZE parameter of the EDITOR macro has been changed to default to sizel=118 and size2=24.
- The DCB subparameters for SYS1.SAMPLIB have been changed to RECFM=F, LRECL=80, and BLKSIZE=80.
- During Stage II execution, an additional job step has been added to allow the IQADCM00 utility program to format the DSS paging system data set.
- The PFK (programmed-function-key) parameter of the SECONSLE macro instruction is also valid for the 3158.
- The SYS1.DCMLIB data set is required if the programmed-function-key (PFK) command entry is specified in the SCHEDULR or SECONSLE macro instructions.
- There is a new component for VSAM (SYS1.AOSA0) that has been added to the distribution library. This component is for planning purposes only and will not be useable until VSAM is released.

INTRODUCTION

This chapter presents an overview of the system generation process and a discussion of the three types of system generation. Also in this chapter is a list and brief discussion of what to do after you have planned for your new System Control Program and received the starter system and/or distribution library tapes, but before the actual system generation.

The System Generation Process

System generation is a process that creates an OS/VS2 System Control Program tailored to both the data processing requirements and machine configuration of an installation.

The System Control Program to be generated is specified with system generation macro instructions that you code. During system generation, IBM-supplied components and user-written components are combined in a variety of ways to build a new System Control Program according to the specifications in the macro instructions (see Figure 1). The new VS2 System Control Program is composed of the standard programs incorporated into every System Control Program and optional programs selected from the distrubution libraries.

A VS2 System Control Program is generated in two stages (see Figure 1). During Stage I, the macro instructions that you coded to describe the installation's machine configuration and the programming options you selected are assembled and analyzed for errors. If no errors are found, a job stream consisting of job control language and control statements is produced. During Stage II, these statements are used to select and process components from the distribution libraries and from user-written routines from user-libraries to form the new VS2 System Control Program.

Types of System Generation

There are three types of system generation: a complete VS2 System Control Program (system) generation, a nucleus generation, and an I/O device generation. The type of generation is specified in the GENERATE macro instruction.

For a complete system generation, the job stream produced during Stage I is processed during Stage II to create an entirely new VS2 System Control Program. For a nucleus or I/O device generation, the job stream is used to modify an existing VS2 system.

Complete System Generation

This is the generation of a complete VS2 System Control Program. It is done when you are installing a VS2 system for the first time or when you must create a new System Control Program.

Nucleus Generation

This is the generation of a new nucleus. It is done when changes need to be made to the nucleus of the System Control Program only. The program options and system configuration that are specified must be compatible with the unchanged part of the System Control Program. For a nucleus generation, the generating system can be the system that is being modified. In a nucleus generation, SYS1.LPALIB, SYS1.NUCLEUS, and SYS1.PARMLIB are the system data sets that are reconstructed.

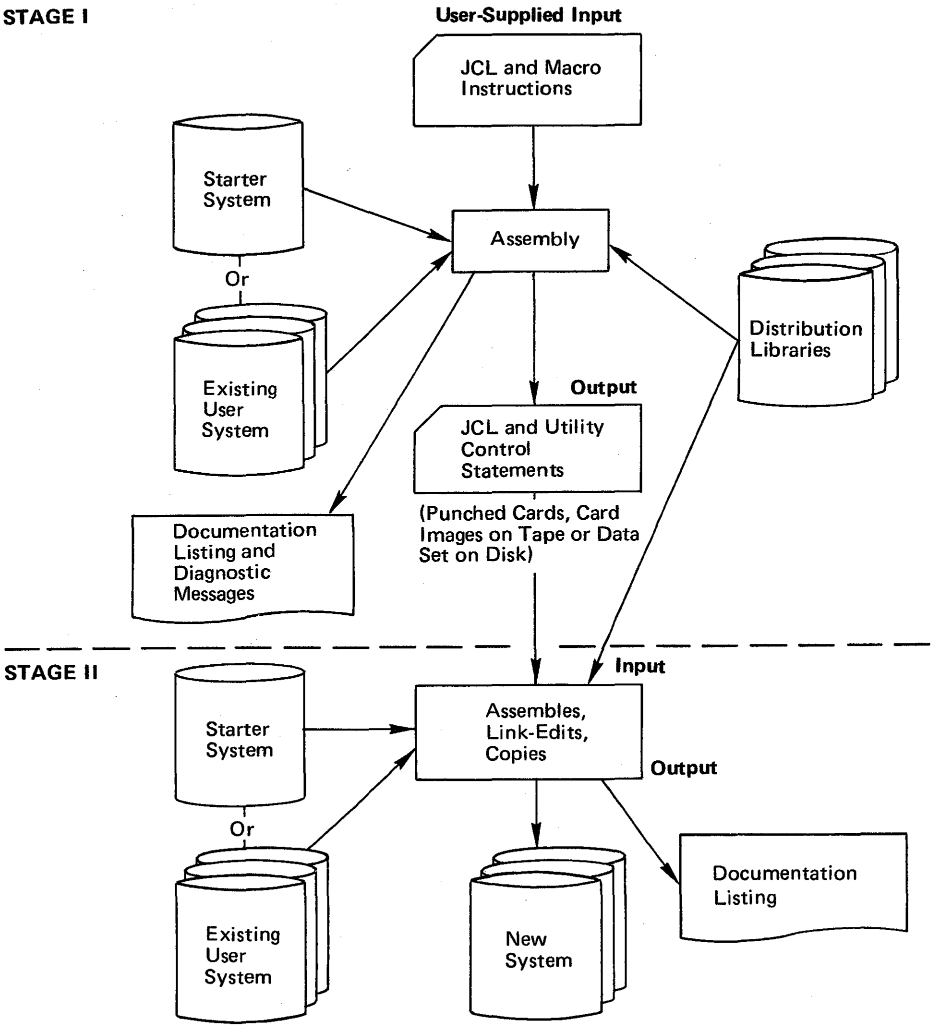

STAGE I
User-Supplied Input
JCL and Macro Instructions
Starter System
Or
Existing User System
Assembly
Distribution Libraries
Output
JCL and Utility Control Statements
(Punched Cards, Card Images on Tape or Data Set on Disk)
Documentation Listing and Diagnostic Messages
STAGE II
Input
Starter System
Assembles, Link-Edits, Copies
Output
Or
Existing User System
New System
Documentation Listing

Figure 1. The system generation process. System generation macro instructions are assembled and expanded to form a job stream (Stage I). This job stream assembles, link-edits, and copies selected modules from the distribution libraries and user-supplied modules to the new system volumes to form the new system (Stage II).

I/O Device Generation

An I/O device generation is done when I/O devices or channels are to be added to or deleted from the existing system. In an I/O device generation, for example, you can add universal character set (UCS) support for the 1403 printer, change I/O device group names, change console specifications, or add or delete paging devices. You cannot add the programming support for telecommunications or graphics, or add additional access methods. For an I/O device generation, the system being modified cannot be the generating system. In an I/O device generation, SYS1.DCMLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.LOGREC, SYS1.LPALIB, SYS1.NUCLEUS, SYS1.PARMLIB, and SYS1.SVCLIB are the system data sets that are reconstructed.

Overview of System Generation Procedures

This section briefly describes the sequence of procedures to be done before a system is generated. The procedures are discussed in the order in which they should be done. For example, coding system generation macro instructions is presented before specifying the system data sets because, what is specified in the macro instructions helps to determine the specifications for the system data sets. All of the required coding should be done before execution of system generation begins.

Coding System Generation Macro Instructions

After you have planned your new System Control Program, code the system generation macro instructions that specify the options you have selected. Some of the macro instructions are always required, some are required or optional, depending on the program options selected and the type of generation.

Figure 6 lists all the macro instructions, shows whether they are required or optional, and gives the order in which they should be coded. The macro instructions are discussed alphabetically in "Specifying the New System Control Program."

Allocating Space to the New System Data Sets and Cataloging Them in the System Catalog

The system data sets should be considered next. If you are using the DATASET macro instruction to allocate space to the system data sets and catalog them, you should refer to "Selecting the New System Data Sets" when you code the DATASET macros. This chapter describes each of the system data sets and tells whether they are required or optional. Also included is data definition (DD) information and cataloging information. The information presented includes the default values that are used by the DATASET macro, but if you are using the IEHPROGM utility program, you must code this information. How to code the control statements and catalog statements for the IEHPROGM utility program is discussed with an example that may be helpful to refer to.

If you are using IEHPROGM to initialize the system data sets, you would probably want to allocate space to the utility data set (OBJPDS), that is required for Stage II processing, at the same time. This information is found in the chapter "System Control Program Installation" in the section "Stage II Input." If you are using DATASET macros, you can code the IEHPROGM control statements for allocating space to OBJPDS when you code the job control language that is required for Stage I processing.

Preparing to Execute System Generation

The procedures discussed in the chapter "Preparing for System Generation" are done before system generation is executed. This chapter lists the procedures that need to be performed to make the generating system (the starter system or an existing VS2 system) and the distribution libraries ready for execution.

Executing System Generation

The chapter "System Control Program Installation" discusses the actual execution of the Stage I and Stage II parts of system generation. Also given is the job control language required to execute Stage I and Stage II and the space requirements for the utility data sets required during Stage I and Stage II.

Restarting System Generation

Coding errors or machine malfunctions may cause system generation to end unsuccessfully. The chapter "Restart Procedures" presents guidelines and coding examples for restarting system generation.

Testing the New System

After installing a VS2 System Control Program, you can test it by using the installation verification procedure (IVP). IVP is discussed in the chapter "Testing the System Control Program."

REQUIREMENTS FOR GENERATING A NEW SYSTEM CONTROL PROGRAM

You can generate a VS2 system using either an existing VS2 system or a starter operating system that is provided by IBM. This chapter discusses the real-storage requirements for a system generation, the machine requirements for system generation using the starter system, and the programming requirements for system generation using either an existing VS2 system or a starter system.

Real-Storage Requirements

To generate a VS2 System Control Program using a starter system, 384K bytes of real storage is required.

Machine Requirements

VS2 system generation is performed using one of the following System/370 central processing units:

Model 145
Model 155II
Model 158
Model 165II
Model 168

Figure 2 shows the maximum machine configuration allowed for a VS2 system generation using the starter system. Any subset of this configuration can be used if it meets the minimum requirements shown in Figure 3.

The I/O devices that are allowed for system generation when the starter system is used are listed in Figure 4. Like devices, such as direct-access devices, can be referred to by certain group names that support the IBM-supplied cataloged procedures. The group names that can be used are listed in Figure 5. (For additional information on group names, refer to the explanation of the UNITNAME macro instruction in "Specifying the New System Control Program.")

Programming Requirements

System generation is performed under the control of an existing system control program (the generating system) and is executed as any other job. The system that is used as the generating system is an existing VS2 System Control Program. If this is your first VS2 system generation, you must use a starter system that is provided by IBM as the generating system. The starter system contains all of the programs necessary to perform system generation.

Using an Existing VS2 System as the Generating System

For an existing VS2 system to be used as the generating system, it must contain a VS2 Release 1:

- OS/VS Assembler
- Linkage editor
- IEBCOPY utility program

- IEBEDIT utility program
- IEBPTPCH utility program
- IEBUPDTE utility program
- IEHDASDR utility program
- IEHLIST utility program
- IEHPROGM utility program
- IFCDIP00 program
- IFCEREP0 service aid program
- IQADCM00 utility program

These programs are defined in the glossary of this publication.

Using the Starter System as the Generating System

The starter system consists of:

- A control program that supports the central processing unit(s) and I/O devices needed to perform the system generation
- An assembler and a linkage editor
- The utility programs used for data set space allocation, direct-access volume initialization, and for Stage II processing

Before a starter system can be used as a generating system, certain procedures need to be performed to make the starter system operational. These procedures are described in the chapter "Preparing for System Generation."

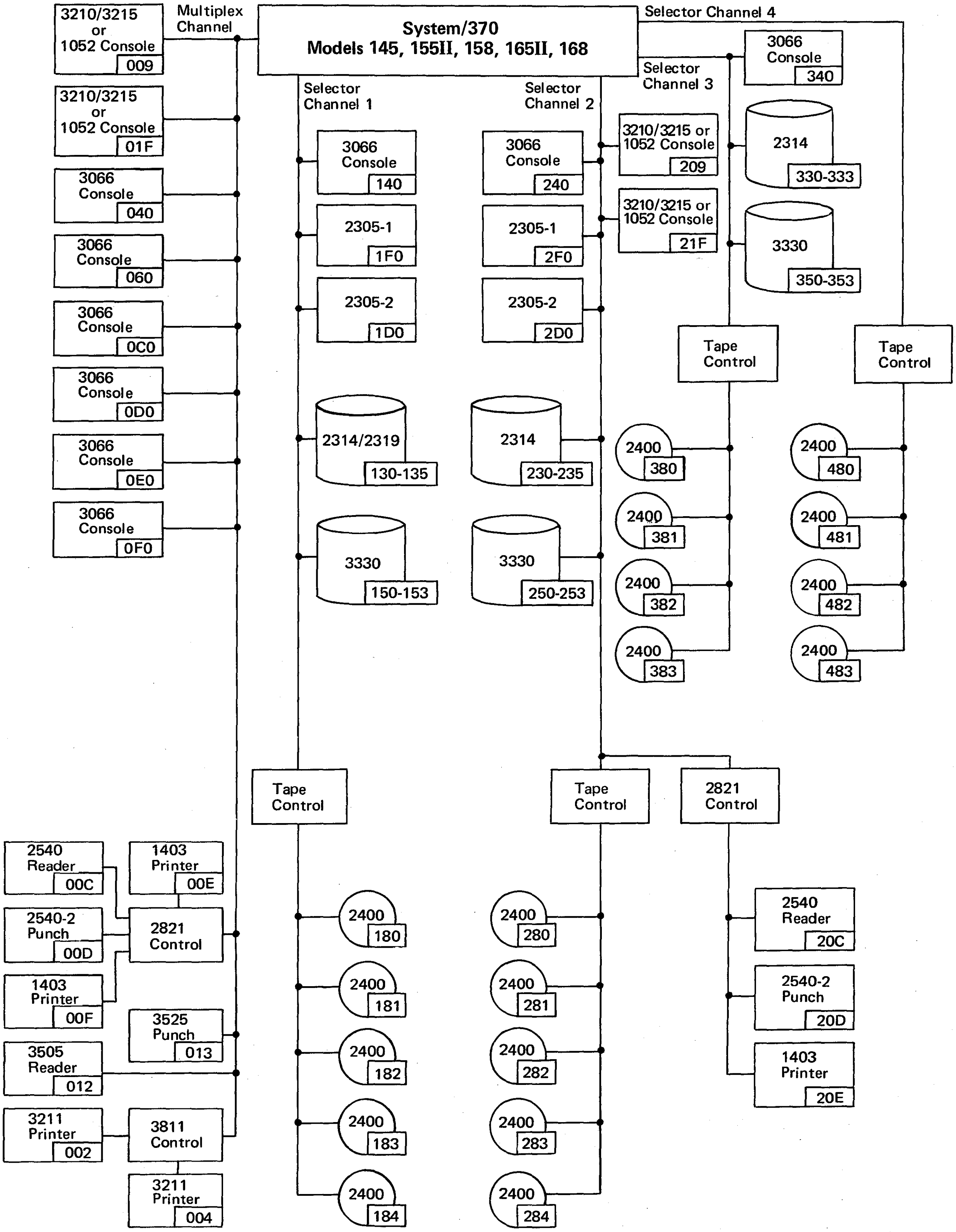


Figure 2. Maximum machine configuration for using the starter system

Use	Minimum Requirement[1]	Device (choose one)	Unit Address (choose one)				
			Multiplexor Channel	Selector Channel 1	Selector Channel 2	Selector Channel 3	Selector Channel 4
System Console	1	3210/3215 or 1052	009, 01F		209, 21F		
		3066	040, 060, 0C0, 0D0, 0E0, 0F0	140	240	340	
System Input	1	2540 Reader	00C		20C		
		3505 Reader	012				
		2400 (7-track with data conversion)		180, 181	280, 281	380, 381	480, 481
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383	482, 483
Punched Output	1	2540-2 Punch	00D		20D		
		3525 Punch	013				
		2400 (7-track with data conversion)		180, 181	280, 281	380, 381	480, 481
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383	482, 483
Printed Output	1	3211	002, 004				
		1403	00E, 00F		20E		
		2400 (7-track with data conversion)		180, 181	280, 281	380, 381	480, 481
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383	482, 483
New System	1[2]	2305-1		1F0	2F0		
		2305-2		1D0	2D0		
		2314 (or 2319 on channel 1)		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333	
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353	

[1] Select the minimum requirement from the list of devices given in each section. For example, the *one* device needed to contain the new system can be either a 2305-1, 2305-2, 2314/2319, or 3330.

[2] The three utility data sets required for Stage I and the object-module data set required for Stage II do not require an additional direct-access device if you are using 3330's. If you are using 2314s/2319s, an additional direct-access device is required.

Figure 3 (Part 1 of 2). Minimum I/O device requirements for performing a system generation using a starter system

			Unit Address (choose one)				
Use	Minimum Requirement[1]	Device (choose one)	Multiplexor Channel	Selector Channel 1	Selector Channel 2	Selector Channel 3	Selector Channel 4
Starter System and Distribution Libraries	2	2314 (or 2319 on channel 1)		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333	
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353	
Four Utility Data Sets[2]	1	2305-1		1F0	2F0		
		2305-2		1D0	2D0		
		2314 (or 2319 on channel 1)		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333	
		3330		150, 151 152, 153	250, 251, 252, 253	350, 351, 352, 353	

[1] Select the minimum requirement from the list of devices given in each section. For example, the *one* device needed to contain the new system can be either a 2305-1, 2305-2, 2314/2319, or 3330.

[2] The three utility data sets required for Stage I and the object-module data set required for Stage II do not require an additional direct-access device if you are using 3330's. If you are using 2314s/2319s, an additional direct-access device is required.

Figure 3 (Part 2 of 2). Minimum I/O device requirements for performing a system generation using a starter system

Magnetic Tape Drives[1]	
2400	2400 series 9-track magnetic tape drive
2400-2	2400 series 7-track magnetic tape drive with data conversion
Direct-Access Devices	
2305-1	2305 fixed head disk storage model 1
2305-2	2305 fixed head disk storage model 2
2314	2314 disk storage facility
2319	2319 disk storage facility
3330	3330 disk storage drive
Unit Record Devices	
1052	1052 console keyboard
1403	1403 printer
2540	2540 reader punch (read feed)
2540-2	2540 reader punch (punch feed)
3066	3066 system console
3210	3210 console printer keyboard
3211	3211 printer
3215	3215 console printer keyboard
3505	3505 card reader
3525	3525 card punch

[1] The 3400 magnetic tape series is functionally equivalent to the 2400 series. The 3400 series may be used for system generation, but you must specify a 2400 series device type.

Figure 4. I/O devices that can be used for system generation using the starter system

Group Name	Devices identified by group name											
SYSSQ	Sequential-access devices at any of the following addresses:											
	2400 series				*2305-1*	*2305-2*	*2314/2319*			*3330*		
	180	280	380	480	1F0	1D0	130	230	330	150	250	350
	181	281	381	481	2F0	2D0	131	231	331	151	251	351
	182	282	382	482			132	232	332	152	252	352
	183	283	383	483			133	233	333	153	253	353
	184	284					134	234				
							135	235				
SYSDA	Direct-access devices at any of the following addresses:											
					2305-1	*2305-2*	*2314/2319*			*3330*		
					1F0	1D0	130	230	330	150	250	350
					2F0	2D0	131	231	331	151	251	351
							132	232	332	152	252	352
							133	233	333	153	253	353
							134	234				
							135	235				

Figure 5. Group names that can be used to identify I/O devices during system generation

SPECIFYING THE NEW SYSTEM CONTROL PROGRAM

This chapter contains the detailed information you need to select and code the system generation macro instructions. Included in this chapter are:

- An explanation of the conventions used in this book to show the macro instructions.
- A summary of the macro instructions by group (machine configuration, control program, etc.) and type of generation.
- An alphabetic listing of the macro instructions. In this section, the function of each macro instruction, the format, a description of the parameters and their defaults, and a coding example is given.

Coding System Generation Macro Instructions

This section describes the rules used to code system generation macro instructions and the notation used in this book to describe the system generation macro instructions.

Rules for Coding System Generation Macro Instructions

The rules for coding system generation macro instructions are those of the assembler language. The following paragraphs are a summary of these rules as stated in *OS/VS and DOS/VS Assembler Language*.

System generation macro instructions have the following standard format:

Name	Operation	Operand
Symbolic name	Macro instruction name	Optional and required parameters

Name symbolically identifies the macro instruction. If included, it can contain from one through eight alphameric characters, the first of which must be alphabetic. The name must begin in the first position of the macro instruction and must be followed by one or more blanks. Except where otherwise indicated in the description of the individual macro instructions, the name field of a system generation macro instruction is ignored during system generation.

Operation identifies the macro instruction. It must be preceded and followed by one or more blanks.

Operand contains parameters coded in any order and separated by commas. The operand field ends with one or more blanks placed after the last parameter. In most system generation macro instructions, keyword parameters are used in the operand field. A parameter consists of a keyword followed by an equal sign (=) and the keyword value. The keyword value may be a single value or a list of values. If it is a list of values, the values must be separated by commas and the list must be enclosed in parentheses.

Comments can be written in a system generation macro instruction, but they must be separated from the last parameter of the operand field by one or more blanks. You can use an entire card for a comment by placing an asterisk in the first column of each card. A macro instruction that has no parameters cannot have comments.

System generation macro instructions are coded in columns 1 through 71 of a card. You can continue a macro instruction that exceeds 71 columns onto one or more additional cards by placing a nonblank character in column 72 to indicate continuation. The system generation macro instruction can be interrupted either at column 71 or after any comma that separates parameters. The continued portion must begin in column 16 of the following card. Comments may appear on every card of a continued statement. Columns 73 through 80 can be used to code identification and/or statement sequence characters if you choose.

Describing System Generation Macro Instructions

This section lists the conventions used in this publication to illustrate the format and coding of system generation macro instructions:

- Letters in boldtype, numbers, and punctuation marks must be coded exactly as shown. Exceptions to this convention are brackets, [ ] ; braces, { } ; ellipses, . . . ; and subscripts, which are never coded.
- Lowercase letters in italics represent variables for which you must substitute specific information or specific values.
- Items or groups of items within brackets are optional. They may be omitted at your discretion. Conversely, the lack of brackets indicates that an item or group of items must be coded.
- Stacked items enclosed in braces represent alternative items. Exactly one of the stacked items should be coded.
- If an alternative item is underlined, it is the default value – the operating system will automatically assume it is your choice if none of the items is coded.
- An ellipsis indicates that the preceding item or group of items can be coded two or more times in succession.

A typical system generation macro instruction might appear as:

```
CHAN1      CHANNEL      ADDRESS=1,TYPE=SELECTOR
```

In this example, CHAN1, in the name field, symbolically identifies the macro instruction. CHANNEL, in the operation field, identifies the macro instruction to the system. ADDRESS is a required parameter and is coded in the operand field. It is followed by an equal sign and the selected value for the parameter. This value is followed by a comma separating the ADDRESS parameter and value from the next parameter. The TYPE parameter is also required for this macro instruction and is coded in the same way as the ADDRESS parameter. Since no parameters follow the TYPE parameter, it is not followed by a comma but by a blank indicating that it is the last parameter in the operand field for this macro instruction.

Macro Instruction Summary

Not all system generation macro instructions are required for every system generation. Figure 6 lists the system generation macro instructions for each type of system generation, indicating whether they are required or optional. If neither required nor optional is indicated, that macro instruction does not apply to that type of system generation. If it is included in the input deck it is ignored, unless it contains a coding error.

Figure 6 also shows which macro instructions can be issued more than once during a system generation. All UNITNAME macro instructions having the same value in the NAME parameter must appear together in the input deck. The GENERATE macro instruction must be the last macro instruction in the input deck for system generation. All other system generation macro instructions can be issued in any order.

Dependencies among the parameters of a macro instruction are illustrated by the macro instruction format and, in some cases, by tables within the macro instruction description. Dependencies between macro instructions are stated in the descriptions of each macro instruction.

Group	Macro Instruction	Complete	Nucleus	I/O Device
Machine Configuration	CENPROCS	required	required	required
	CHANNEL[1]	required	required	required
	IODEVICE[1]	required	required	required
	UCS	optional	--	optional
	UNITNAME[1]	required	--	required
Control Program	CKPTREST	optional	--	--
	CTRLPROG	optional	optional	optional
	EDIT	optional	--	--
	EDITOR	optional	--	--
	LOADER	optional	--	--
	MACLIB	optional	--	--
	PAGE[1]	required	required	required
	SCHEDULR	required	required	required
	SECONSLE[1]	optional	optional[2]	optional[2]
	TSO	optional	optional[2]	optional[2]
Data Management	DATAMGT	optional	optional[2]	optional[2]
	GRAPHICS	optional	optional[2]	optional[2]
User-written Routines	LINKLIB	optional	optional[2]	optional[2]
	LPALIB	optional	optional[2]	--
	RESMODS	optional	optional[2]	--
	SVCTABLE[1]	optional	optional[2]	optional[2]
Generation	DATASET[1,3]	optional	optional	optional
	GENERATE	required	required	required

[1] This macro instruction can be used more than once in the input deck.

[2] This macro instruction is required if it was originally specified in the last complete system generation.

[3] This macro instruction is required under certain conditions. Refer to the DATASET macro instruction text in this chapter.

Figure 6. The required and optional system generation macro instructions and the order in which they should be considered

CENPROCS

Required for:
complete
nucleus
I/O device

The CENPROCS macro instruction describes the central processing unit (CPU) and secondary CPU support for the new system. Recovery management support (RMS) is included for all CPU models specified. (For information on RMS, refer to *OS/VS Recovery Management Support Logic.*)

For a *nucleus* or *I/O device generation,* the CENPROCS macro instruction must be coded with the same parameters and subparameters that were specified in the last complete system generation.

```
CENPROCS        MODEL= ⎧ 145R ⎫
                       ⎪ 155R ⎪
                       ⎨ 158R ⎬
                       ⎪ 165R ⎪
                       ⎩ 168R ⎭
               [SECMODS= ⎧ ALL                  ⎫ ]
                         ⎩ (model [,model] . . .) ⎭
```

Parameter	Subparameter	Explanation
MODEL=	model	specifies the model number of the central processing unit.
SECMODS=		specifies the additional System/370 models for which environment record editing and printing (EREP) support will be included in the new system. If this parameter is omitted, no secondary models will be supported.
	ALL	specifies that all supported CPU models are to be included.
	model	specifies the additional CPU models to be supported.

Example: This example specifies a model 145 central processing unit and additional support to model 155II.

```
CPU     CENPROCS     MODEL=145R,SECMODS=(155R)
```

CHANNEL

Required for:
complete
nucleus
I/O device

The CHANNEL macro instruction describes the channel characteristics. There must be a CHANNEL macro instruction for each channel type in your computing system.

For a *nucleus generation*, the CHANNEL macro instructions must be coded with the same parameters and subparameters that were used in the last complete system generation.

For an *I/O device generation*, each channel must be respecified. Channels can be deleted or added, but each channel in the new system must be specified.

If you do not specify a name in the name field of this macro instruction, a sequential identification number is supplied by the macro. The order in which the numbers are assigned is determined by the order in which they appear in the input stream. For example, if the name is omitted from the third CHANNEL macro instruction, the name CHAN #3 is supplied in each diagnostic message resulting from an error detected during processing of that macro instruction.

```
CHANNEL     ADDRESS= (address [,address] . . .)
            TYPE= ⎧ BLKMPXR     ⎫
                  ⎨ HISPEEDMPXR ⎬
                  ⎨ MULTIPLEXOR ⎬
                  ⎩ SELECTOR    ⎭
```

Parameter	**Subparameter**	**Explanation**
ADDRESS=	address	specifies the address assigned to the channel. The value must be alphameric, 0 through 9 or A through F. Multiple addresses may be specified for the channels having the same value for the TYPE parameter.
TYPE=		specifies the type of channel defined by this macro instruction.
	BLKMPXR	The channel is a block multiplexor channel that is integrated either with a central processing unit or with a 2880 channel frame. An address must be specified for each block multiplexor channel.
	HISPEEDMPXR	The channel is a 2870 channel frame and there are one or more subselector channels on the 2870. You cannot use separate CHANNEL macro instructions or specify different addresses for the subselector channels.
	MULTIPLEXOR	The channel is a byte multiplexor channel integrated with either the central processing unit or a 2870 channel frame (if there are no sub-selector channels on the 2870).

Parameter	Subparameter	Explanation
TYPE= (Continued)	SELECTOR	The channel is a selector channel integrated with either the central processing unit or a 2860 channel frame. An address must be specified for each selector channel.

Example: This macro specifies that channels 1, 2, and 3 are selector channels.

```
CHAN1-3     CHANNEL      ADDRESS=(1,2,3),TYPE=SELECTOR
```

CKPTREST

Optional for:
complete
Not applicable for:
nucleus
I/O device

The CKPTREST macro instruction is used to specify the standard ABEND codes that you do not want eligible for automatic restart, and those codes written by you that you do want eligible for automatic restart. If this macro instruction is omitted, the standard set of ABEND codes will be included in the new system.

For information on checkpoint/restart, refer to *OS/VS Checkpoint/Restart.* For information on ABEND codes, refer to *OS/VS Message Library: VS2 System Codes.*

```
CKPTREST      [ELIGBLE=(code [,code] ...)]
              [NOTELIG=(code [,code] ...)]
```

Parameter	**Subparameter**	**Explanation**
ELIGBLE=	code	specifies ABEND codes written by you which you want eligible for automatic restart. The code specified is a decimal integer from 0 to 4095. A maximum of ten values may be specified
NOTELIG=	code	specifies those standard ABEND codes which you do not want to be eligible for automatic restart. The standard ABEND codes are listed in *OS/VS Checkpoint/Restart.*

Example: This example specifies the standard ABEND codes that are not to be recognized for automatic restart.

```
CKPTRST      CKPTREST      NOTELIG=(001,100,031,113)
```

CTRLPROG

Optional for:
complete
nucleus
I/O device

The CTRLPROG macro instruction is used to specify control program options. If it is not specified, the default values are used.

For a *nucleus generation*, the values that were specified in the DEBTINC, DEBTSZE, REAL, TRACE, and TZ parameters can be changed. The value specified for the QSPACE parameter can be changed, but this parameter cannot be included if it was not specified in the last complete system generation. A fixed BLDL table can be included in or deleted from the system during a nucleus generation.

For an *I/O device generation*, the value specified in the MAXIO parameter can be changed. All other parameters must be specified as they were in the last complete system generation.

For additional information about control program options, refer to *OS/VS2 Planning and Use Guide.* To estimate storage for control program options, refer to *OS/VS2 Storage Estimates.*

```
CTRLPROG      [ASCII= {EXCLUDE} ]
                      {INCLUDE}

              [APG=   (priority, lower bound, upper bound, delta, statistics interval,
                      ratio)] 1

              [DEBTINC= {size} ]
                        {80  }

              [DEBTSZE= {size} ]
                        {80  }

              [LSQACEL=  ((cellsize-number)
                         [,(cellsize-number)] ...)] 1

              [MAXIO= number]

              [OPTIONS= (value [,value] ...) ]1

              [PGCLEAR= {YES} ]
                        {NO }

              [QSPACE= {size} ] 1
                       {1   }

              [REAL= {size} ] 1
                     {0   }

              [SQACEL= ( (cellsize-number)
                       [,(cellsize-number)] ... )] 1

              [TMSLICE= (priority,SLC-time [,priority,SLC-time] ...)]1

              [TRACE= {number} ]1
                      {  0   }

              [TZ= ( {W} ,hh [,mm] )]
                     {E}
```

[1] The values specified in these parameters will become members of the IEASYS00 parameter list in SYS1.PARMLIB.

Parameter	Subparameter	Explanation
ASCII=		specifies the inclusion or exclusion of the ASCII translation routine. If you plan to use ASCII utilities, this parameter must be specified. If this parameter is not specified, EXCLUDE will be assumed.
	EXCLUDE	specifies that the routine will not be included.
	INCLUDE	specifies that the routine will be included as part of the pageable supervisor.
APG=		specifies a single priority level to a group of tasks in an attempt to provide optimum use of CPU and I/O resources by these tasks. If APG (automatic priority group) is specified, all subparameters must be coded. For information on APG, refer to *OS/VS2 Planning and Use Guide.*
	priority	is a decimal number from 0 to 13 that represents the scheduling priority that is to be identified as the automatic priority group.
	lower bound	is a decimal number from 1 to 999 that represents, in milliseconds, the lower bound of a time-slice that may be given to an APG task.
	upper bound	is a decimal number from 2 to 999 that represents, in milliseconds, the upper bound of a time-slice that may be given to an APG task. This value must be greater than the value specified for the lower bound.
	delta	is a decimal number from 1 to 99 that represents, in milliseconds, the delta to be added to or subtracted from the APG time-slice value at the end of each statistics interval.
	statistics interval	is a decimal number from 1 to 9999 that represents, in milliseconds, the length of the statistics interval.
	ratio	is a decimal number from 1 to 9 that represents, in tenths, the ratio of time-slice ends to the sum of voluntary waits plus time-slice ends in the APG.
DEBTINC=	size 80	is a decimal number from 8 to 32752 that specifies by how much the data extent block (DEB) table for DEB validity checking will be expanded if the initial size is insufficient for a jobstep. If this parameter is not specified, the default value is assumed.
DEBTSZE=	size 80	is a decimal number from 8 to 32760 that specifies the initial size, in bytes, of the DEB table for DEB validity checking. If this parameter is not specified, the default value is assumed.

Parameter	Subparameter	Explanation
LSQACEL=		specifies the size and number of cells in the local system queue area. If this parameter is not specified, no local SQA cells will be available.
	cell size	is a decimal number from 8 to 256 (bytes) that specifies the size of the cell in the local system queue area. The value must be specified in even multiples of 8 bytes. If it is not, the value specified will be rounded upward to an even multiple of 8 bytes.
	number	is a decimal number from 1 to 8 that specifies the number of cells. The total size of the LSQA area is limited to 4K (4096 bytes). If 4K is exceeded, the cells will be truncated starting with the largest cell. Storage requirements for the local system queue area can be found in *OS/VS2 Storage Estimates.*
MAXIO=	number	specifies the maximum number of I/O operations that can be simultaneously processed by the new VS2 system. This number is the sum of those I/O operations that are being executed simultaneously and those that are currently queued but are not being executed. The number specified determines the number of request queue elements (RQEs) in the nucleus. If this parameter is not specified, the default value is one RQE for each I/O device when the total number of devices is 50 or less. Beyond 50, one RQE will be assigned for every device other than direct-access. One RQE will be assigned for every four direct-access devices in the new system.
OPTIONS=		specifies which of the following control program options are to be included in, or excluded from, the new VS2 system. If this parameter is not specified, none of these options will be included in the system.
	BLDL	specifies a fixed BLDL table. If this option is not specified, the BLDL table will be pageable.
	DEVSTAT	specifies inclusion of the nucleus initialization (NIP) support that will regard all direct-access devices and magnetic tape drives that are not in a ready condition or that are not currently attached to the system as offline.
	RER	specifies that the reduced error recovery procedure for magnetic tape will be used if requested in the OPTCD parameter of a DD statement or a DCB macro instruction. If RER is not specified, all requests for reduced error recovery will be ignored.

Parameter	Subparameter	Explanation
OPTIONS= (Continued)	RDE	specifies inclusion of the reliability data extractor feature. For information about RDE, refer to *OS/VS SYS1.LOGREC Error Recording.*
PGCLEAR=		specifies whether or not a page frame in real storage will be cleared to zeros before the first page is paged into that area.
	YES	specifies that the area will be cleared to zeros.
	NO	specifies that the area will not be cleared to zeros.
QSPACE=	size 1	is a decimal number that specifies the number of 64K-byte segments reserved for system queue space. If this parameter is not specified, a value of 1 is used. *OS/VS2 Storage Estimates* tells how to estimate the storage required by the system queue space.
REAL=	size 0	is a value from 0 to 999 that specifies the amount of real storage available, in addition to the amount of real storage that is provided by the system. The value specified is the number of 4K (4096 bytes) blocks. The value is expressed as a decimal number without the K. A minimum value of 16 is assumed by the system. Refer to *OS/VS2 Storage Estimates* for information on determining this value.
SQACEL=		specifies the size and number of cells in the system queue area. If this parameter is not specified, no SQA cells will be available.
	cell size	is a decimal number from 8 to 256 (bytes) that specifies the size of a cell in the system queue area. The value must be specified in even multiples of 8 bytes. If it is not, the value specified will be rounded upward to an even multiple of 8 bytes.
	number	is a decimal number from 1 to 8 that specifies the number of cells. The total size of the system queue area (SQA) cells is limited to 4K (4096 bytes). If 4K is exceeded, the area will be truncated to this size. To estimate the size and number of SQA cells, refer to *OS/VS2 Storage Estimates.*
TMSLICE=		specifies time-slicing for the jobs scheduled to be executed that have a given priority number. Uppercase letters and hyphens are coded as shown.
	priority	is a decimal number from 0 to 13 that specifies the priority number of the jobs to be time-sliced.

Parameter	Subparameter	Explanation
TMSLICE= (Continued)	SLC-time (Continued)	is a decimal number from 20 to 9999 that specifies the maximum number of milliseconds for which each ready job of the priority specified is to have control of the CPU during one cycle through the group. If this parameter is not specified, no jobs will be time-sliced. Time-slicing can be added to or deleted from the system during a nucleus generation. For information on time-slicing, refer to *OS/VS2 Planning and Use Guide* and *OS/VS2 Storage Estimates.*
TRACE=	number $\underline{0}$	specifies the number of system trace table entries to be maintained. If this parameter is not specified, a value of 0 is assumed. A tracing routine aids in debugging and maintenance by storing in the trace table information pertaining to start I/O (SIO) instructions, supervisor (SVC) interruptions, dispatcher interruptions, and I/O interruptions. When the table is full, the succeeding entries overlay the existing ones. During system generation, only the size of the table is specified. However, when this parameter is specified, the trace program routines are also included as part of the control program. Additional information is found in *OS/VS2 Planning and Use Guide* and *OS/VS2 Storage Estimates.*
TZ=		specifies that local standard time will be reflected in the output instead of Greenwich Mean Time.
	$\begin{Bmatrix} \mathrm{W} \\ \mathrm{E} \end{Bmatrix}$	specifies whether the time zone is west or east of the Greenwich Meridian.
	hh	specifies the number of hours difference between local time and Greenwich Mean Time. The hh is specified as either one or two decimal digits from 0 to 12.
	mm	specifies the number of minutes to be added to the hh subparameter to obtain a time zone offset from Greenwich Mean Time in hours and minutes. The mm is specified as one or two decimal digits between 0 and 60. If mm is omitted, 0 is used. If the TZ parameter is not specified, Greenwich Mean Time will be produced.

Example: The example specifies the options for the control program for the new system. A fixed BLDL table has been specified, inclusion of nucleus initialization support (NIP) to treat direct-access devices and magnetic tape devices that are in a not-ready condition as off-line, and inclusion of the reliability data extractor feature (RDE) has also been specified. Six 64K-byte segments have been reserved for system queue space, 8 SQA quickcells of 256 bytes have been reserved, and 8 LSQA quickcells of 256 bytes have been reserved. Twenty additional 4K blocks of real storage have been specified for jobs that run in a VIRTUAL=REAL region. The maximum length of time that a job in priority-class 13 will have control is 500 milliseconds. Up to 100 simultaneous I/O requests will be processed. Local standard time will be used with an 8-hour deviation from Greenwich Mean Time. The automatic priority group (APG) for the dispatcher is specified as follows: scheduling priority – 12, time-slice lower bound – 20, time-slice upper bound – 500, time-slice delta – 1, statistics interval – 15, time-slice and wait ratio – 5. Default values are assumed for the ASCII translation routine, the system trace table, and the data extent block (DEB) table size and increment.

```
CTRLPROG CTRLPROG OPTIONS=(BLDL,DEVSTAT,RDE),      X
                QSPACE=6,SQACEL=(256-8),           X
                LSQACEL=(256-8),REAL=20,           X
                TMSLICE=(13,SLC-500),MAXIO=100,    X
                TZ=(W,8),APG=(12,20,500,1,15,5)
```

DATAMGT

Optional for:
complete
nucleus
I/O device

The DATAMGT macro instruction is used to specify the optional access methods to be included in the new system. This macro instruction is optional.

A telecommunications option can be included in the system so that tasks can use the basic telecommunications access method (BTAM) or the telecommunications access method (TCAM). Also, the indexed sequential access method (ISAM) can be included in the new system so that tasks can use the basic indexed sequential access method (BISAM) or the queued indexed sequential access method (QISAM). Additional information can be found in:

OS/VS Data Management Services Guide

OS/VS2 Storage Estimates

OS/VS BTAM

OS TCAM Concepts and Facilities

The following standard access methods are always included in the system:

BDAM – basic direct access method

BPAM – basic partitioned access method

BSAM – basic sequential access method

QSAM – queued sequential access method

For a *nucleus generation*, if this macro instruction was used during the last complete system generation, it must be respecified. The same parameter and subparameters must be coded.

For an *I/O device generation*, if this macro instruction was used during the last complete system generation, it must be respecified. The same parameters and subparameters must be coded. If this macro instruction was not specified during the last complete system generation, it cannot be specified for this type of generation.

```
DATAMGT        ACSMETH= (method[,method] ...)
```

Parameter	Subparameter	Explanation
ACSMETH=	method	specifies one or more of the following access methods: **BTAM** – basic telecommunication access method **TCAM** – telecommunications access method **ISAM** – basic and queued indexed sequential access methods If the TSO macro is specified, TCAM is assumed.

Example: This example specifies that ISAM is to be included in the new system.

```
DATAMGT      DATAMGT      ACSMETH=(ISAM)
```

DATASET

Optional for:
complete
nucleus
I/O device

The DATASET macro instruction is used to allocate space to the new system data sets and catalog them in the system catalog. (See "Allocating Space to the New System Data Sets and Cataloging Them in the System Catalog" in the chapter "Selecting the New System Data Sets.") All of the system data sets that are specified using the DATASET macro are cataloged except SYSCTLG, which contains the system catalog.

If you use the DATASET macro, it must be specified for each system data set that is to be included in the system. If you do not use the DATASET macro, you must use the IEHPROGM utility program. Space allocation and cataloging of system data sets must take place before Stage II. (For information on IEHPROGM, see "Preparation for System Generation.")

A DATASET macro must be specified for any system data set used during system generation that is not on the system residence volume, even if the data set has already had space allocated to it and has been cataloged. Omission of the SPACE parameter in the DATASET macro will prevent the allocation of space for a system data set.

For a *complete generation*, the SPACE parameter must be specified.

For a *nucleus generation*, if SYS1.LINKLIB, and/or SYS1.LPALIB, or SYS1.PARMLIB are not on the system residence volume, you must respecify the DATASET macro(s). The SPACE parameter need not be respecified. If it is respecified, it will be ignored.

For an *I/O device generation,* if SYS1.DCMLIB, and/or SYS1.IMAGELIB, SYS1.LINKLIB, or SYS1.PARMLIB are not on the system residence volume, you must respecify the DATASET macro(s). The SPACE parameter need not be respecified. If it is respecified, it will be ignored.

In both *nucleus* and *I/O device generations,* all SPACE parameters are ignored. Therefore, you need not remove them from the input deck or change them.

The DATASET macro instructions use the default values for the DCB parameters that are required by the system data sets. You cannot specify DCB information for the system data sets if you are using the DATASET macro instructions. For information on the default values for the DCB parameters, refer to "Selecting the New System Data Sets."

```
DATASET      system data set
                      ⎧CYL       ⎫
             [SPACE= (⎨TRK       ⎬,quantity)]
                      ⎩blocksize ⎭
             [VOL= ( ⎧serial,type  ⎫)]
                     ⎩SYSRES, 2314 ⎭
```

Parameter	**Subparameter**	**Explanation**
system data set		specifies the system data set for which space is to be allocated. The system data set to be allocated may be: BRODCAST CMDLIB DCMLIB DSSVM DUMP HELP IMAGELIB LINKLIB LPALIB MACLIB MANX MANY NUCLEUS PARMLIB PROCLIB SAMPLIB SVCLIB SYSCTLG SYSJOBQE SYSVLOGX SYSVLOGY TELCMLIB UADS The following system data sets must reside on the system residence volume: LOGREC[1] NUCLEUS SVCLIB SYSCTLG The following system data sets must exist during system generation and if they are not specified will be assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro instruction (system residence volume): DSSVM LINKLIB LPALIB MACLIB NUCLEUS PARMLIB PROCLIB SAMPLIB SVCLIB SYSCTLG

[1] Space for LOGREC is allocated by the generating system. More explanation is given in the chapter "Selecting the New System Data Sets" under "SYS1.LOGREC."

Parameter	Subparameter	Explanation
system data set (Continued)		IMAGELIB must exist in the system to be generated if the UCS macro instruction is specified. TELCMLIB must exist in the system to be generated if BTAM or TCAM is specified in the DATAMGT macro instruction. If you are generating a system with TSO, CMDLIB must have space allocated to it and it must be cataloged. If HELP and/or UADS is to be included in the system, you must specify a DATASET macro for them even if the data sets are preallocated.
SPACE=		specifies a request that space allocation be based solely on the values given in this subparameter.
	{ CYL TRK blocksize }	specifies the unit of space to be allocated. Space can be allocated in cylinders, tracks, or average block length.
	quantity	specifies how many units of space (cylinders, tracks, or blocks) are to be allocated. Depending on the system data set, the quantity may have one of several forms. Generally the form is: (primary quantity [,secondary quantity] [,directory blocks]) primary quantity specifies how many units of space (tracks, cylinders, or blocks) are to be allocated. secondary quantity specifies how many more tracks or cylinders are to be allocated if additional space is required, or how many more blocks of data may be included if additional space is required for the data set. Secondary quantity space allocation is valid for the following system data sets: CMDLIB HELP LINKLIB LPALIB MACLIB PROCLIB SAMPLIB SVCLIB SYSCTLG TELCMLIB UADS directory blocks specifies the number of 256-byte blocks to be allocated for the directory of a partitioned data set. Directory block information must be specified for the following system data sets if space is being

Parameter	**Subparameter**	**Explanation**
SPACE= (Continued)	quantity (Continued)	allocated for them by means of DATASET macro instructions: CMDLIB DCMLIB HELP IMAGELIB LINKLIB LPALIB MACLIB NUCLEUS PARMLIB PROCLIB SAMPLIB SVCLIB TELCMLIB UADS
		The amount of space for the system data set SYS1.LOGREC is always calculated and allocated by the generating system on the system residence volume of the system being generated. This applies to complete and I/O device generations.
		If the DATASET macro is used to allocate space to SYS1.DSSVM, the following must be specified in the SPACE parameter: SPACE=(2048,(600))
		Estimates for space allocation for the system data sets is in *OS/VS2 Storage Estimates.*
VOL=		specifies the volume serial number and the device type of the volume for the system data set.
	serial number SYSRES	specifies the volume serial number. SYSRES is the default value unless the RESVOL parameter in the GENERATE macro specifies another value.
	device type 2314	specifies the device type of the volume. The device types that may be coded are: 2305-1, 2305-2, 2314, and 3330.
		The VOL parameter works in conjunction with the RESVOL parameter in the GENERATE macro. The default value for RESVOL is the same as that for VOL. If RESVOL is changed, the default value for VOL becomes the changed value.
		If either serial number or device type is specified, both must be specified. If this parameter is not specified, SYSRES, 2314 is used.

Example: This example catalogs and allocates space to the SYS1.NUCLEUS system data set. Two cylinders are required with no secondary space allocation and two 256-byte blocks are requested for the directory.

```
NUCLEUS      DATASET      NUCLEUS,SPACE=(CYL,(2,,2))
```

EDIT

Optional for:
complete

Not applicable for:
nucleus
I/O device

The EDIT macro instruction specifies the physical characteristics and processing attributes of the data sets to be processed by the TSO EDIT command. Three programs may be specified for each data set type: an exit routine for processing user-defined information during the EDIT command initialization, a syntax checker, and a routine that processes the EDIT data set when the RUN subcommand of EDIT is issued. This macro instruction is optional for a complete system generation of a system with TSO.

```
EDIT     BLOCK= ([n],[n] ,...)
         CHECKER= ([name],[name] ,...)
                   ⎧CAPS      ⎫
         CONVERT= (⎨ASIS      ⎬,...)
                   ⎩CAPSONLY  ⎭
         DSTYPE= (name[,name] ...)
         FIXED= ([d-m],[d-m] ,...)
                  ⎧FIXED    ⎫
         FORMAT= (⎨VAR      ⎬,...)
                  ⎩FXDONLY  ⎭
         PRMPTR= ([name],[name] ,...)
         USEREXT= ([name],[name] ,...)
         USERSRC= (⎧DATASET⎫,...)
                   ⎩INCORE ⎭
         VAR= ([d-m],[d-m] ,...)
```

Commas are required to denote missing operands in all parameters except DSTYPE. Defaults for omitted operands are listed in Figure 7.

Parameter	Subparameter	Explanation
BLOCK=	n	specifies, in bytes, the default length of blocks for the data sets created by the TSO EDIT command. The n is an integer from 1 to 32760 that specifies the default blocksize for any data set of the type specified in the corresponding *name* field of the DSTYPE parameter.
CHECKER=	name	specifies the installation-supplied processor that is to be used by the TSO EDIT command to check lines in the data set for proper syntax.

Parameter	Subparameter	Explanation
CHECKER= (Continued)	name (Continued)	The name is the 1 to 8 character module name that the TSO EDIT command will use to invoke the CHECKER for data sets of the type specified by the corresponding *name* field in the DSTYPE parameters.
CONVERT=		specifies whether input data to the TSO EDIT command should be converted to upper case.
	CAPS	specifies that the CAPS operand on the TSO EDIT command, which requests upper case conversion of data, is to be the default for data sets of the type specified by the corresponding *name* field of the DSTYPE parameter.
	ASIS	specifies that the ASIS operand on the TSO EDIT command, which requests that the data remain as entered (for example, upper and lower case), is to be the default for data sets of the type specified by the corresponding *name* field of the DSTYPE parameter.
	CAPSONLY	specifies that the CAPS operand of the TSO EDIT command is the only permitted operand. The ASIS operand may not be specified on the TSO EDIT command.
DSTYPE=	name	specifies IBM-standard data set types or new data set types defined by you that are to have their attributes modified or defined for use by the TSO EDIT command. (See Figure 7). The name specified is the 1 to 8 character data set type as it will appear in the TSO EDIT command. Any data set type permitted by the TSO EDIT command or defined by the installation can be specified. Up to ten new data set types defined and up to fifteen standard data set types can be specified in this parameter.
FIXED=	d-m	specifies, in bytes, the default and maximum allowable length of logical fixed-length records for data sets created by the TSO EDIT command. *d* is an integer from 1 to 255 that specifies the default logical record length for any fixed format data set of the type specified in the corresponding *name* field of the DSTYPE parameter. The corresponding *n* value, if specified in a BLOCK parameter, must be a multiple of this value. If the variable record format is the default for the data set type (VAR=), a value of 0 should be coded for *d* when this subparameter is used to define the maximum size of a fixed-length record. *m* is an integer from 1 to 255 that specifies the maximum logical record length of any fixed-format data set of the type specified in the corresponding *name* field of the DSTYPE parameter. The corresponding

Parameter	Subparameter	Explanation
FIXED= (Continued)	d-m (Continued)	*n* value, if specified in a BLOCK parameter, must be a multiple of this value, and the *m* value must be greater than or equal to the *d* value. The hyphen must be coded as shown.
FORMAT=		specifies the allowable record format for new data sets created by the TSO EDIT command.
	FIXED	specifies that fixed-record format is the default value for data sets of the type specified in the corresponding *name* field of the DSTYPE parameter.
	VAR	specifies that variable record format is the default value for data sets of the type specified by the corresponding *name* field of the DSTYPE parameter.
	FXDONLY	specifies that only the fixed-record format is valid for data sets of the type specified in the corresponding *name* field of the DSTYPE parameter. Specifying the VAR parameter for this data set type is invalid.
PRMPTR=	name	specifies the installation-supplied processor that is to be used by the RUN subcommand of the TSO EDIT command. RUN invokes an installation-supplied processor to execute programs under the TSO EDIT command. The name is the 1 to 8 character prompter name for the data sets of the type specified in the corresponding *name* field of the DSTYPE parameter.
USEREXT=	name	specifies the installation-supplied user exit routine that is to be invoked by the TSO EDIT command to interpret the parameters specified in the subfield of the DSTYPE operand of the TSO EDIT command. The interpreted parameters will be passed as part of the parameter list to the processor that is specified in the CHECKER parameter. This parameter cannot be used with any of the standard data set types. The name, which must be 1 to 8 characters long, is associated with data sets of the type specified in the corresponding *name* field of the DSTYPE parameter.
USERSRC=		specifies the type of input acceptable to the processor specified in the PRMPTR parameter in the TSO EDIT command.
	DATASET	specifies that a sequential data set is the only allowable input to the specified processor.
	INCORE	specifies that the TSO EDIT command may pass the input source to the specified processor in the form of

Parameter	Subparameter	Explanation
USERSRC= (Continued)	INCORE (Continued)	an in-storage data set, if the source does not exceed 4096 bytes, or in the form of a sequential data set.
VAR=	d-m	specifies, in bytes, the default and maximum allowable length of logical, variable-length records with variable-record format for new data sets created by the TSO EDIT command. *d* is an integer from 5 to 255 that specifies the default logical record length for any variable-format data set of the type specified in the corresponding *name* field of the DSTYPE parameter. The value specified must be less than or equal to *n*-4 of the corresponding *n* value specified in the BLOCK parameter. *m* is an integer from 5 to 255 that specifies the maximum logical record length for any variable-format data set of the type specified in the corresponding *name* field of the DSTYPE parameter. The value specified must be less than or equal to *n*-4 of the corresponding *n* value specified in the BLOCK parameter. The hyphen must be coded as shown.

Example: This example specifies that a new data set type, USER01, is defined for the TSO EDIT command. Its default record format is fixed and all other system defaults are used (see Figure 7). The standard data set type, DATA, will have a default blocksize of 2400 instead of 1680 (see Figure 7). All other standard data set types will use the default values that are specified in Figure 7.

```
EDIT     EDIT     DSTYPE=(USER01,DATA),BLOCK=(2400),    X
                  FORMAT=(FIXED)
```

DATA SET TYPE	BLOCK-SIZE	RECORD FORMAT	Default LRECL F	Default LRECL V	Maximum LRECL F	Maximum LRECL V	Data Conversion	Checker Name	Prompter Name	Prompter Input
Standard Data Set Types										
PLIF	400	FXDONLY	80	0	100	0	CAPSONLY	PLIFSCAN	--------	DATASET
FORTE	400	FXDONLY	80	0	80	0	CAPSONLY	IPDSNEXC	--------	DATASET
FORTG	400	FXDONLY	80	0	80	0	CAPSONLY	IPDSNEXC	--------	DATASET
FORTH	400	FXDONLY	80	0	80	0	CAPSONLY	IPDSNEXC	--------	DATASET
ASM	1680	FXDONLY	80	0	80	0	CAPSONLY	--------	ASM*	DATASET
TEXT	1680	VAR	0	255	255	255	ASIS	--------	--------	DATASET
DATA	1680	FIXED	80	0	255	255	CAPS	--------	--------	DATASET
CLIST	1680	VAR	0	255	255	255	CAPSONLY	--------	--------	DATASET
CNTL	1680	FXDONLY	80	0	80	0	CAPSONLY	--------	--------	DATASET
COBOL	400	FXDONLY	80	0	80	0	CAPSONLY	--------	COBOL*	DATASET
PLI	400	VAR	0	104	100	104	CAPS	PLISCAN	PLI	DATASET
IPLI	1680	VAR	0	120	120	120	CAPSONLY	IKJNC211*	IKJNC211*	DATASET
BASIC	1680	VAR	0	120	120	120	CAPSONLY	IKJNC211*	IKJNC211*	DATASET
GOFORT	1680	VAR	0	255	255	255	CAPS	IPDSNEXC	GOFORT*	INCORE
FORTGI	400	FXDONLY	80	0	80	---	CAPSONLY	IPDSNEXC	FORT*	DATASET
Any User Defined Data Set Types										
----	1680	FIXED	80	0	255	255	CAPS	--------	--------	--------

*IBM Program Product.

--Null Value

Figure 7. Attribute defaults for parameters omitted from the EDIT macro

EDITOR

Optional for:
complete

Not applicable for:
nucleus
I/O device

The EDITOR macro instruction specifies the linkage editor options. This macro is optional. If it is not specified, the default values will be assumed. For additional information on the linkage editor, refer to *OS/VS Linkage Editor and Loader.*

```
EDITOR         [SIZE= ({size1,size2 } )]
                       {118,24      }
```

Parameter	Subparameter	Explanation
SIZE=		specifies the default values for the maximum number of bytes of virtual storage available to the linkage editor and to its corresponding text buffer. The values specified are multiples of 1024 (1K) bytes.
	size1 <u>118</u>	is a value from 64K to 9999K (specified without the K). It specifies the amount of virtual storage available to the linkage editor. If this subparameter value is specified it must be coded first. If this subparameter is omitted, a value of 118 is assumed, resulting in 118K bytes.
	size2 <u>24</u>	is a value from 6K to 100K (specified without the K). It specifies the amount of virtual storage available to the text buffer. If both subparameter values for the SIZE parameter are specified, this value must be coded last. If only this one is specified, it must be preceded by a comma. If this subparameter is omitted, a value of 24 is assumed, resulting in 24K bytes.

Example: This macro specifies that 256K bytes of virtual storage will be available to the linkage editor and 24K bytes of virtual storage will be available to the corresponding text buffer.

```
EDITOR      EDITOR      SIZE=(256)
```

GENERATE

Required for:
complete
nucleus
I/O device

The GENERATE macro instruction is used to specify the object-module data set, the volume serial number and device type of the system residence volume of the system to be generated, the output class and jobclass used during system generation, and the type of generation being done. This macro instruction must be specified last.

The GENERATE macro instruction produces the Stage II job stream that consists of several jobs so multijobbing may be used. If multijobbing is not used, the jobs will execute sequentially.

For a *complete generation* the serial number specified in the RESVOL parameter cannot be the serial number of the system residence volume of the generating system.

For a *nucleus generation*, the serial number and device type for the RESVOL parameter must be the same as the system residence volume of the system being modified, which could be the generating system. If you specify a value for the INDEX parameter other than SYS1., you must rename the system data sets that are to be updated to the value you specified.

For an *I/O device generation*, the system being modified cannot be the generating system. The serial number specified in the RESVOL parameter cannot be the volume serial number of the system residence volume of the generating system. If you specify a value for the INDEX parameter other than SYS1., you must rename the system data sets that are to be updated to the value you specified.

```
GENERATE     [GENTYPE= { ALL          } ]
                       { (NUCLEUS,n)  }
                       { (IO,n)       }
             [INDEX= { name }]
                     { SYS1 }
             [JCLASS= { class }]
                      { A     }
             [OBJPDS= { SYS1.name    }]
                      { SYS1.OBJPDS  }
             [OCLASS= { class }]
                      { A     }
             [RESVOL= ( [ { serial } ] [, { type } ] )]
                          { SYSRES }       { 2314 }
```

Parameter	Subparameter	Explanation
GENTYPE=		specifies the type of system generation.
	ALL	specifies a complete system generation.
	NUCLEUS,n	specifies that only a nucleus will be generated. The *n* is a number from 1 through 9 that specifies the number of the new member of SYS1.NUCLEUS being generated. The member name will be IEANUC0n.

Parameter	Subparameter	Explanation
GENTYPE= (Continued)	NUCLEUS,n (Continued)	The value 1 is always assigned to the nucleus to be generated when GENTYPE=ALL is specified. Therefore, if 1 is specified in this subparameter, the new nucleus to be generated will replace the nucleus generated during the last complete system generation.
	IO,n	specifies an I/O device generation. The n is a number from 1 through 9 that specifies the number of the nucleus that is being modified. The member name will be IEANUC0n.
INDEX=	name SYS1	specifies the qualifier for the new system data sets that will be created during system generation. The qualifier may be from 1 to 6 alphameric characters, the first character alphabetic. If an index value other than the default value, SYS1, is specified, the qualifier of the system data set affected is changed to SYS1 at the end of the system generation process.
JCLASS=	class A	specifies the jobclass (A-O) to be used for output from Stage II of system generation. If this parameter is not specified, a value of A is used.
OBJPDS=	SYS1.name SYS1.OBJPDS	specifies the partitioned data set to be used for the storage of the object modules that are assembled during Stage II of system generation. This data set must have been cataloged as SYS1.name in the generating system before Stage II execution. If this parameter is omitted, SYS1.OBJPDS is used.
OCLASS=	class A	specifies the output class (A through Z or 0 through 9) to be used for output from Stage II of system generation. If this parameter is not specified, a value of A is used.
RESVOL=		specifies the volume serial number and device type of the new system residence volume to be generated.
	serial number SYSRES	specifies the volume serial number of the new system residence volume to be generated. If this parameter is omitted, SYSRES is used.
	type 2314	specifies the unit address, device type, or group name for the new system residence volume to be generated. Valid device types are 2305-1, 2305-2, 2314, or 3330. If this parameter is omitted, 2314 is used. If either subparameter is omitted, the default values are SYSRES,2314. If the serial number subparameter is omitted, the type subparameter must be preceded by a comma.

Example: This example specifies a complete system generation. The partitioned data set that is to contain the object modules that are assembled during Stage II is SYS1.TEMP. The system residence volume will reside on a 3330 and its volume serial number will be SYSRES. The qualifier for the new system data sets will be SYS1. The job class for Stage II output is A and the output class is A.

```
GEN  GENERATE  OBJPDS=SYS1.TEMP,                        X
               RESVOL=(SYSRES,3330),INDEX=SYS1,         X
               JCLASS=A,OCLASS=A
```

GRAPHICS

Optional for:
complete
nucleus
I/O device

The GRAPHICS macro instruction specifies the inclusion of graphic programming services (GPS). This macro instruction is optional. If it is not specified, GPS will not be included in the new system. For information about GPS, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2250* and *OS/VS Graphic Programming Services (GPS) for IBM 2260.*

For a *nucleus generation*, if the new nucleus is to support the same set of graphic programming services that was included in the system during the last complete generation, then the GRAPHICS macro instruction must be coded exactly the way it was specified in the last complete system generation. GPS, however, cannot be added during a nucleus generation.

GPS cannot be added during an *I/O device generation*. If the GRAPHICS macro instruction was specified during the last complete system generation, it must be respecified with exactly the same parameters and subparameters.

```
GRAPHICS      [GSP= { INCLUDE  } ]
                    { EXCLUDE  }
              [PORRTNS= { EXCLUDE } ]
                        { INCLUDE }
```

Parameter	Subparameter	Explanation
GSP=		specifies the inclusion of the graphic subroutine package (GSP) in SYS1.LINKLIB.
	INCLUDE	specifies that GSP is to be included.
	EXCLUDE	specifies that GSP is not to be included. If this parameter is not specified, EXCLUDE is used.
PORRTNS=		specifies the inclusion of problem-oriented routines (PORs) in SYS1.LINKLIB.
	EXCLUDE	specifies that PORs are not to be included.
	INCLUDE	specifies that PORs are to be included. If this parameter is not specified, INCLUDE is used.

Example: This macro specifies that problem-oriented routines (PORs) are to be included in SYS1.LINKLIB and that the graphic subroutine package is not to be included in SYS1.LINKLIB.

```
GRAPHICS      GRAPHICS      PORRTNS=INCLUDE
```

IODEVICE

Required for:
complete
nucleus
I/O device

The IODEVICE macro instruction describes the characteristics of an input/output (I/O) device and its system requirements. Each uniquely addressable I/O device in the operating system must be specified in an IODEVICE macro instruction. This macro instruction is required.

For telecommunication devices there must be one IODEVICE macro instruction for each telecommunications line. The IODEVICE macro instruction applies to a telecommunications line, not to a terminal device.

For telecommunication terminals, all terminals on a line must be of the same type with the same features. The type of terminal is used to identify the line in the UNIT parameter.

The value specified in the ADDRESS parameter becomes the unit address of the device. Unit addresses are automatically assigned to the devices during system generation.

During system generation, device types are also provided in the operating system for each type of device specified by the UNIT parameter of an IODEVICE macro instruction. Device types are described in Appendix A.

A device or a collection of devices can be assigned a group name by use of the UNITNAME macro instruction.

A maximum of 768 I/O devices can be specified during system generation.

Figure 8 shows the valid combinations of values for the UNIT, MODEL, and FEATURE parameters.

Figure 9 shows the values that can be specified for I/O devices in the FEATURE parameter.

Figure 10 shows the valid combinations of values for telecommunications devices for the UNIT, ADAPTER, and TCU parameters.

Burst devices cannot be specified for multiplexor channels. Burst devices are: 2250, 2314, 2401, 2420, 3330, 3410, and 3420.

An IODEVICE macro instruction must be specified if a 2955 remote analysis unit is to be attached to the system.

For a *nucleus generation*, the same IODEVICE macro instructions used during the last complete system generation must be respecified without changes.

For an *I/O device generation*, all I/O devices that are to be in the new system must be specified. Except for the system residence device type, there can be additions, deletions, changes, or the same specifications as those in the last complete generation.

```
IODEVICE     ADDRESS= { address                    }
                      { (address, number of units) }

             [DEVTYPE=type]

             [ERRTAB=nnn]

             [IOREQUE= { FIFO     } ]
                       { PRIORITY }
                       { ORDERED  }

             UNIT= { device type      }
                   { DUMMY            }
                   { HASP-device type }
```

The validity of the following parameters depends on the type of unit specified. See Figure 8 for valid parameters and subparameters.

```
             ADAPTER=adapter
             [EXPBFR= { number } ]                  (2250-3 only)
                      { 4096   }
             [FEATURE=(feature [,feature] . . . ) ]
             GCU= { 2848-1  }                       (2260-2 only)
                  { 2848-2  }
                  { 2848-21 }
                  { 2848-22 }
             [NUMSECT={ number } ]                  (2250-3 only)
                      { 16     }
             [MODEL=model]
             OBRCNT=number                          (BSC1, BSC2, BSC3 only for
                                                    2715)
             [OPTCHAN= (address [,address] . . . ) ] (2305, 2314, 3330, 2401, 2420,
                                                    3420, or DUMMY only)
             PCU=n                                  (2250-3 only)
             SETADDR=value                          (TCU=2702 only)
             TCU= { 2701 }                          (Telecommunications devices)
                  { 2702 }
                  { 2703 }
```

Parameter	Subparameter	Explanation
ADAPTER=	adapter	specifies the terminal control or transmission adapter used to connect a telecommunications line to a transmission control unit. This parameter is required for all telecommuncation lines. Figure 10 associates terminal control adapters with transmission control units.

Parameter	Subparameter	Explanation
ADDRESS=		specifies the unit address of devices or telecommunication lines. The value specified in the ADDRESS parameter becomes the unit address of the device. For each device address that is assembled, a unit control block (UCB) is created.
	address	specifies the unit address, consisting of three hexadecimal digits from 000 to FFF. The high-order digit is the address of the channel (specified in the CHANNEL macro instruction) to which the device is attached.
	number of units	specifies the number of units to be used and the total number of sequential addresses to be assembled. The value can be a number from 1 to FFF. For example, if ADDRESS=(130,5) is specified, the unit addresses 130, 131, 132, 133, and 134 would be assembled. If this subparameter is omitted, a value of 1 is assumed for all devices except the 2314, which has a default of 8 and the 3330 which has a default of 2. The maximum value that can be specified for a 3330 in one IODEVICE macro is 8. This parameter value is ignored for the 2305.
DEVTYPE=	type	specifies any additional characteristics of the device. The value specified must be 8 hexadecimal characters. This parameter need not be specified for any IBM-supported device. This parameter should be specified if UNIT=DUMMY is specified. For further information about this parameter value, refer to the description of the UCB in *OS/VS2 System Data Areas.*
ERRTAB=	nnn	specifies that an error routine other than a standard error routine is to be used for the device. Either an IBM-supplied routine or your own routine may be specified. IBM error routines have the values 000 through 219 and 230 through 254. Your own routines can have values 220 through 229. This value is the suffix of the name IGE00, under which the error routine is stored in SYS1.LPALIB. The ERRTAB parameter should be specified if UNIT=DUMMY is specified.
EXPBFR=	number 4096	specifies the amount of buffer space, in bytes, required by programs written for a 2250 model 1 that use EXPRESS attention handling routines and that can be used with a 2250 model 3 attached to a 2840. The value may be an integer from 1 to 8192. If this parameter is not specified, 4096 (bytes) is used.

Parameter	Subparameter	Explanation
EXPBFR= (Continued)	number 4096 (Continued)	For information on the 2250, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2250.*
FEATURE=	feature	specifies the optional features that the device has. These values can be written in any order. Refer to Figure 8 for the valid features that may be specified for the devices and to Figure 9 for a summary of features that may be specified.
GCU=	$\left\{\begin{matrix}\text{2848-1}\\\text{2848-2}\\\text{2848-21}\\\text{2848-22}\end{matrix}\right\}$	specifies the type of graphic control unit (GCU) to which a 2260 model 2 is attached. One of these graphic control units must be specified for each 2260 model 2. For information on the 2260, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2260.*
IOREQUE=		specifies the type of I/O request queuing to be provided by the supervisor for the device. If this parameter is not specified, FIFO is used.
	FIFO	specifies first-in-first-out queuing.
	PRIORITY	specifies queuing according to task priority.
	ORDERED	specifies queuing according to cylinder address. IOREQUE=ORDERED may be specified only for the 2314 or 3330. If it is specified for any other device, FIFO is substituted.
MODEL=	model	specifies the model number, if any, for the device. This parameter must be specified if the device has a model number (see Figure 8).
NUMSECT=	number 16	specifies the number of 256-byte buffer sections in a 2840 display control unit to be assigned to a 2250 model 3. These buffer sections can be used only by the device being specified. The value of the number specified is from 1 to a maximum value that is determined by the following formula: $\frac{A}{256} - B + 1$ where A is the size of the 2840 buffer B is the number of devices attached to the 2840. The total number of buffer sections guaranteed to the devices attached to a 2840 must not exceed the number of sections in the buffer of that 2840.

Parameter	Subparameter	Explanation
NUMSECT= (Continued)	number 16 (Continued)	If this parameter is not specified, the device uses the 2840 control unit buffer sections that are not assigned to other devices attached to the 2840. However, there must always be at least one section available for assignment to each device attached to the 2840. If a 2250 model 3 is specified as an operator console and this parameter is not specified or its value is less than 16, a value of 16 is used. For information on the 2250, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2250.*
OBRCNT=	address 800	is a number from 0 to 800 that specifies the number of area stations connected to the 2715 transmission control in a 2790 data communications system. The total of all values specified in this parameter cannot exceed 800 during system generation. If 800 is exceeded, 800 is assumed.
OPTCHAN=	address	specifies the alternate channels through which any of the following devices may be addressed: 2305, 2314, 2401, 2420, 3330, 3420, or DUMMY (a device not supported by IBM). The value of the address specified must be greater than the high-order digit of the value in the ADDRESS parameter of this macro. The address of a high-speed multiplexor subchannel must be specified as 2 characters; the first, the channel address of the selector channel, the second, D, E, or F. There is a maximum of 768 optional channel paths per configuration. Each value specified is the address of an alternate channel that was specified in the CHANNEL macro instruction. The alternate path retry feature of the I/O supervisor becomes part of the system when the OPTCHAN parameter is specified. There must be no more than one IODEVICE macro instruction for each I/O device, regardless of the number of alternate addresses given to the device. For example, if the primary address of a device is 181, and if it can also be addressed through channels 2, 3, and 4, there must not be separate IODEVICE macro instructions defining the address of the device as either 281, 381, or 481. The primary address of the device, that is, the one with the lowest channel address, must be specified in the ADDRESS parameter. The other channel addresses must be specified in the OPTCHAN parameter. In this example, the macro instruction for the device must contain the parameters ADDRESS=181 and OPTCHAN=(2,3,4).

Parameter	Subparameter	Explanation
PCU=	n	specifies the number assigned to the physical control unit (2840) to which this 2250 model 3 is attached. The n is a number from 1 to 99. Each physical 2840 must be uniquely identified by this parameter. The same 2840 should be considered a different physical control for each channel attached to it. For example, for two 2250s with addresses 3D2, 3D3, 4D2, and 4D3 attached to the same 2840(D) but attached to two different channels (3 and 4), the PCU parameter might be PCU=1 for 3D2 and 3D3 and PCU=2 for 4D2 and 4D3. A 2250 on any 2840 cannot have a unit address that is within the range of addresses of any 2250 on another 2840 or any other unit address. For example, 122, 124 cannot be assigned to any 2250s if 121, 125, and 126 are addresses of 2250s on another 2840 or if 123 is the address of another device.
SETADDR=	value	specifies which of the four set address (SAD) commands is to be issued to the transmission control unit (TCU) for operations on the line specified by the ADDRESS operand (2702 only). The SAD command selects the appropriate line speed for the terminal connected to the line. The association between the specific command and the corresponding line speed is established by internal connections within the 2702. The *value* for the SAD command is one of the following: 0 2 1 3 This parameter is required for the 2702. If the TCU is a 2701 or 2703, the SAD commands are ignored.
TCU=	2701 2702 2703	specifies the teleprocessing control unit for a telecommunications line. This parameter is required for all telecommunications lines. Figure 10 associates terminal control adapters with transmission control units.
UNIT=	device type	specifies the device type. In the case of telecommunication lines, this parameter specifies the device type that is connected to a telecommunication line, or the type of binary synchronous line configuration: nonswitched point-to-point (BSC1), switched point-to-point (BSC2), or nonswitched multipoint (BSC3). Binary synchronous may also be specified by UNIT=2780, 1130, 2020, or S360. Figure 8 lists and defines the devices that may be specified in the UNIT parameter and the valid combinations of parameters that may be specified. Appendix A describes the device types. The 2319 drives are functionally equivalent to the 2314 drives. The default value is eight drives. If you use less than eight drives, you must specify the number of

Parameter	Subparameter	Explanation
UNIT= (Continued)	device type (Continued)	drives (see ADDRESS= in this section). To use a 2319, specify UNIT=2314. Appendix D contains a description of the 2319. The IBM 3333 Disk Storage and Control is functionally equivalent to the IBM 3330 Disk Storage Drive. To use a 3333, specify UNIT=3330. Appendix D contains a description of the 3333.
	DUMMY	specifies a device that is not supported by IBM. When DUMMY is specified, a 32-byte UCB with all its standard fields is generated. It is assumed that you provide your own I/O support routines for the device. If you want to refer to the device using job control language statements, you must generate a unit address with the UNITNAME macro instruction. Unit addresses for DUMMY devices are not automatically generated.
	HASP-device type	specifies a HASP device. Valid device types that may be specified are: 1403, 1442, 1443, 2520, 2540R, and 2540P.

Example: This macro defines a 3210 console with a unit address of 009. The IOREQUE parameter was not specified so the default option for the I/O request queuing of FIFO (first-in-first-out) is assumed.

```
C009       IODEVICE       UNIT=3210,ADDRESS=009
```

Example: This macro defines a 2540 Model 1 card punch with the CARDIMAGE feature. The unit address for the device is 00D. The default value, FIFO, was taken for the I/O request queuing parameter.

```
P-00D      IODEVICE     UNIT=2540P,ADDRESS=(00D),      X
                FEATURE=(CARDIMAGE),MODEL=1
```

Example: This macro defines a 2314 with 6 drives. UCBs will be generated for addresses 130 through 135. Channel 2 is defined as an alternate channel. (No other device may be specified with addresses 230 through 235.) The default value, FIFO is assumed for the I/O request queuing parameter (IOREQUE).

```
D2314      IODEVICE     UNIT=2314,ADDRESS=(130,6),     X
                OPTCHAN=(2)
```

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Direct-Access Devices				
2314		SHARED	IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\\ \text{ORDERED}\end{array}\right\}$ OPTCHAN= $\left\{\begin{array}{l}\text{n}\\ \text{(n,n [,n])}\end{array}\right\}$	Direct Access Storage Facility The 2314 must have the 2-channel switch feature or the 2844 to allow FEATURE= SHARED and/or OPTCHAN. The 2844 Auxiliary Storage Control allows the 2314 to be addressed through up to four channels: the primary and three alternate, or the primary, one or more shared with another processing unit, and one or more shared in the same processing unit.
3330		SHARED	IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\\ \text{ORDERED}\end{array}\right\}$ OPTCHAN= $\left\{\begin{array}{l}\text{n}\\ \text{(n,n [,n])}\end{array}\right\}$	Disk Storage Drive The device's control unit must have the 4-channel switch feature for FEATURE= SHARED or OPTCHAN to be supported. A maximum of three alternate channels may be specified. The addresses of alternate channels may be written.
Drum				
2305	1 or 2	SHARED	IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{array}\right\}$ OPTCHAN= n	Fixed Head Disk Storage The 2305 Model 1 can only be specified in a CPU Model 165II system configuration. FEATURE=SHARED and the OPTCHAN parameter are mutually exclusive for the device.
Display devices				
1053	4		IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{array}\right\}$	Printer
2250	1	ABSLTVEC ALKYB2250 $\left\{\begin{array}{l}\text{BUFFER4K}\\ \text{BUFFER8K}\end{array}\right\}$ CHARGNTR DESIGNFEAT LIGHTPEN PRGMKYBD	IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{array}\right\}$ EXPBFR=n	Display unit Refer to the "Unit Record" part of this figure for specifying this device as an operator's console.

Figure 8 (Part 1 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Display Devices (Continued)				

2250 (Cont)	3	ALKYB2250
		PRGMKYBD
			IOREQUE=
			FIFO
			PRIORITY
			NUMSECT=n
			EXPBFR=n
			PCU=n
				Display unit
				The parameter PCU is required for the 2250 Model 3.
				Refer to the "Unit Record" part of this figure for specifying this device as an operator's console.

2260	1 or 2	ALKYB2260 DEKYB2260 LINEADDR NODESCUR NMKYB2260	IOREQUE= FIFO PRIORITY GCU=control unit	Display station The GCU parameter is required for the 2260 Model 2. Refer to the "Unit Record" part of this figure for specifying this device as an operator's console.
2260 or 2265			IOREQUE= FIFO PRIORITY ADAPTER= IBM3 TCU=2701	Display station When the unit is specified as a remote device, the MODEL and FEATURE parameters are not specified.
3158		EBKY3277 ASKY3277 DEKY3277 OCKY3277 SELPEN KB78KEY		Display unit Refer to the "Unit Record" part of this figure for specifying this device as an operator's console.
3277	1 or 2	ASCACHAR ASCBCHAR KACHAR DOCHAR FRCHAR GRCHAR UKCHAR AUDALRM MAGCDRD NUMLOCK SELPEN		Display unit Refer to the "Unit Record" part of this figure for specifying this device as an operator's console. Only one type of character generator may be specified. If none are specified, DOCHAR is used as the default. Only one type of keyboard may be specified. If none are specified, a 66-key keyboard is assumed. The AUDALRM feature can only be specified if a keyboard has been specified.

Figure 8 (Part 2 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Display Devices (Continued)				
3277 (Cont)	1 or 2 (Cont)	EBKY3277 ASKY3277 DEKY3277 OCKY3277 KB70KEY KB78KEY KB81KEY		
3284	1 or 2	DOCHAR FRCHAR GRCHAR KACHAR UKCHAR		Printer
3286	1 or 2	DOCHAR FRCHAR GRCHAR KACHAR UKCHAR		Printer
Magnetic Tape Units				
2401	1, 2, 3	READWRITE MDECOMPAT 7-TRACK 9-TRACK DATACONV	IOREQUE= FIFO PRIORITY OPTCHAN=n	Magnetic tape unit FEATURE=9-TRACK is assumed if the FEATURE parameter is omitted.
	4, 5, 6	READWRITE 9-TRACK DUALDENS		
2401	8	7-TRACK READWRITE	IOREQUE= FIFO PRIORITY OPTCHAN=n	Magnetic tape unit The DATACONV feature is standard for the 2401-8. READWRITE is required and valid only if the 2401-8 is attached to a 2804-3.
2420			IOREQUE= FIFO PRIORITY OPTCHAN=n	Magnetic tape unit This is a 9-track 1600 BPI drive only so the FEATURE parameter is not required.
2495			IOREQUE= FIFO PRIORITY	Magnetic tape cartridge reader

Figure 8 (Part 3 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Magnetic Tape Units (Continued)				
3410	1, 2, 3	7-TRACK 9-TRACK DUALDENS	IOREQUE= FIFO PRIORITY	Magnetic tape drive
3420	3, 5, 7	7-TRACK 9-TRACK DUALDENS SHARABLE	IOREQUE= FIFO PRIORITY OPTCHAN= n	Magnetic tape drive
Optical Character Readers				
1275 or 1287 or 1288			IOREQUE= FIFO PRIORITY	The address specified must be the address of the primary control unit and must be an even number. There must be only one IODEVICE macro for each 1275. Two addresses are generated by the one IODEVICE macro. The 1275 is available through IBM World Trade Corporation branch offices.
Magnetic Ink Character Readers				
1419			IOREQUE= FIFO PRIORITY	The address specified must be the address of the primary control unit and must be an even number. There must be only one IODEVICE macro for each 1419. Two addresses are generated by the one IODEVICE macro instruction.
Unit Record Devices[2]				
1052	7		IOREQUE= FIFO PRIORITY	Printer-keyboard Installed in combination with the 2150 Control Unit.
2250	1	ALKYB2250 BUFFER4K BUFFER8K CHARGNTR LIGHTPEN PRGMKYBD	IOREQUE= FIFO PRIORITY	Display unit LIGHTPEN and PRGMKYBD are optional; other features are required if the 2250 Model 1 is used as a console device.

Figure 8 (Part 4 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Unit Record Devices[2] (Continued)				

2250	3	ALKYB2250
		PRGMKYBD
			IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$
			NUMSECT= n
			EXPBFR= n
			PCU=n
				Display unit
				If the 2250 Model 3 is being used as a console device:
				• ALKYB2250 is required.
				• If NUMSECT is omitted or less than 16, 16 is used.
				• The PCU parameter is required.

2260	1	LINEADDR ALKYB2260 NODESCUR	IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$	Display Station If the 2260 Model 1 is being used as a console device, LINEADDR and ALKYB2260 are required.
3066			IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$	The system console for the System/370 Models 165II and 168.
3210 or 3215			IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$	Console-printer keyboard
3277	2	$\left\{\begin{matrix}\text{ASKY3277}\\ \text{DEKY3277}\\ \text{EBKY3277}\\ \text{OCKY3277}\end{matrix}\right\}$ KB78KEY SELPEN		If the 3277 is being used as a console device with input capability, a keyboard must be specified. The 3277 Model 1 can be used only as an output-only console to display operator messages; optional features or parameters cannot be specified.
3158		$\left\{\begin{matrix}\text{EBKY3277}\\ \text{ASKY3277}\\ \text{DEKY3277}\\ \text{OCKY3277}\end{matrix}\right\}$ SELPEN KB78KEY		Display console with keyboard
1403	N1 or 2	UNVCHSET	IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$	Printer
	7			UNVCHSET is invalid for the 1403-7.
1443	N1	SELCHSET 24ADDPOS	IOREQUE= $\left\{\begin{matrix}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{matrix}\right\}$	Printer

Figure 8 (Part 5 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Unit Record Devices[2] (Continued)				
3213			IOREQUE= {FIFO PRIORITY}	Console-printer with no keyboard
3211			IOREQUE= {FIFO PRIORITY}	High-speed printer The universal character set is a standard feature.
2501	B1 or B2	CARDIMAGE	IOREQUE= {FIFO PRIORITY}	Card reader
2520	B1	CARDIMAGE	IOREQUE= {FIFO PRIORITY}	Card reader punch
	B2 or B3			Card punch only
2540R or 2540P	1	CARDIMAGE	IOREQUE= {FIFO PRIORITY}	Card reader punch The 2540R and 2540P are specified for the same 2540 Card Reader Punch. Two IODEVICE macro instructions must be specified.
3505		CARDIMAGE	IOREQUE= {FIFO PRIORITY}	Card reader and control unit
3525		CARDIMAGE {TWOLINE MULTILINE}	IOREQUE= {FIFO PRIORITY}	Card punch Every 3525 specified must be attached to a 3505.
2671	1		IOREQUE= {FIFO PRIORITY}	Paper tape reader
Telecommunications[3]				
1030		AUTOPOLL	IOREQUE= {FIFO PRIORITY} SETADDR= value	Data collection system

Figure 8 (Part 6 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURES[1] (optional)	Optional Parameters	Notes
Telecommunications[3] (Continued)				
1050		AUTOANSR AUTOCALL AUTOPOLL	IOREQUE= $\left\{\begin{array}{l}\underline{\text{FIFO}}\\ \text{PRIORITY}\end{array}\right\}$ SETADDR= value	Data communication system AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified.
1050X		AUTOCALL AUTOANSR		Refers to the 1050 with the time-out suppression feature.
1060		AUTOPOLL		Data communication system
2740		AUTOANSR AUTOCALL AUTOPOLL CHECKING SCONTROL XCONTROL OIU INTERRUPT		Communication terminal AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified. If the 2740 is specified as a console, then it must have the record checking feature. The communications line must be non-switchable and only one 2740 per communications line can be specified as a console device. CHECKING must be specified if the OIU feature is specified. SCONTROL and XCONTROL cannot be specified if OIU is specified. If RPQ #S330031 is installed, FEATURE= INTERRUPT may be specified.
2740C		CHECKING AUTOANSR		Communications terminal with correspondence code. CHECKING is required.
2740X		CHECKING AUTOCALL AUTOANSR		Communications terminal with PTTC code. CHECKING is required.
2741C or 2741P		AUTOANSR		Communications terminal with correspondence code or communications terminal with PTTC code.
115A				Western Union* Terminal

*Trademark of Western Union Telegraph Company

Figure 8 (Part 7 of 9). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Telecommunications[3] (Continued)				
83B3			IOREQUE= {FIFO PRIORITY} SETADDR= value	AT&T* Selective Calling Terminal
TWX		AUTOANSR AUTOCALL		Teletype** Models 33 or 35
WTTA				IBM World Trade Corporation Telegraph Terminal

BSC1		DUALCODE[5]
		DUALCOMM
			IOREQUE=
			{FIFO
			PRIORITY}
			OBRCNT=n
				BSC[4] station nonswitched point-to-point line.

BSC2		DUALCODE[5] DUALCOMM AUTOANSR AUTOCALL		BSC[4] station switched point-to-point line.
BSC3		DUALCODE[5] DUALCOMM AUTOPOLL		BSC[4] station nonswitched multipoint line. AUTOPOLL is assumed.
7770	3		IOREQUE= {FIFO PRIORITY}	Audio response unit
2955			IOREQUE= {FIFO PRIORITY}	Remote analysis unit If this device is to be attached, an IODEVICE macro instruction must be specified.
3705			ADAPTER= {CA1 CA2}	Communications control unit An adapter must be specified for the 3705.

*Trademark of American Telephone and Telegraph Company.
**Trademark of the Teletype Corporation.

Figure 8 (Part 8 of 9). Parameter values that may be specified in an IODEVICE macro instruction

[1] Figure 9 lists and defines all of the options that can be specified through the FEATURE parameter.

[2] Figure 11 lists all of the devices that can be used as consoles.

[3] Figure 10 lists the terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device (UNIT) to a transmission control unit (TCU).

[4] BSC (Binary Synchronous Communications) stations can be any of the following:
System/3 Processor Station
System/360 Processor Station
System/370 Processor Station
System/360 Model 20 Processor Station
1130 Processor Station
1800 Processor Station
2770 Data Communications System
2780 Data Transmission Terminal
2790 Data Communications System
2972 Model 8 and 11 Bank Terminal
3275 Display Station (BSC3 only)
3277 Display Station (BSC3 only)
3284 Printer Controller (BSC3 only)
3286 Printer Controller (BSC3 only)
3670 Brokerage Terminal
3735 Programmable Buffer Terminal (BSC2 and BSC3 only)

[5] If DUALCODE and/or DUALCOMM is specified, the telecommunications device can only be connected to a 2701. If AUTOANSR and/or AUTOCALL or AUTOPOLL is specified, the telecommunications device can be connected to either a 2701 or 2703.

Figure 8 (Part 9 of 9). Parameter values that may be specified in an IODEVICE macro instruction

Feature	Unit	Description
ABSLTVEC	2250-1	Absolute vectors and control; enables the device to trace continuous straight lines at any angular position within the display area.
ALKYB2250 or ALKYB2260	2250 2260	Alphameric keyboard; permits you to enter messages consisting of letters, numbers, and other symbols.
ASCACHAR	3277	ASCII A character generator.
ASCBCHAR	3277	ASCII B character generator.
ASKY3277	3158 3277	ASCII typewriter keyboard.
AUDALRM	3277	Audible alarm feature.
AUTOANSR	BSC2 TWX 1050 1050X 2740 2740C 2740X 2741C 2741P	The modem connecting the telecommunications line specified by the address operand to the TCU is a switched line over which calls are to be answered.
AUTOCALL	BSC2 TWX 1050 1050X 2740 2740X	The TCU to which the remote station is connected is equipped with the auto call feature and the line is connected to the TCU terminal adapter by means of an automatic calling unit and an appropriate modem.
AUTOPOLL	BSC3 1030 1050 1060 2740	The automatic polling feature of the TCU is to be used. This feature is standard for the 2703 and optional for the 2702. This feature, when the TCU is a 2701, is valid only for lines connected through a synchronous data adapter type II. When this feature is on the 2740, SCONTROL must also be specified.
BUFFER4K or BUFFER8K	2250-1	This feature provides the display unit with either 4096 bytes or 8192 bytes of virtual storage for display regeneration.

Figure 9 (Part 1 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
CARDIMAGE	2501 2520 2540P-1 2540R-1 3505 3525	This feature provides reading and/or punching in card-image mode. For the 2540R and 2540P, the 2821 Control Unit must have the column-binary feature installed.
CHARGNTR	2250-1	Character generator; allows alphameric characters to be displayed on a cathode ray tube. The feature is standard on the 2250-3 and 2260.
CHECKING	2740 2740C 2740X	The 2740 Communication Terminal is equipped with the record checking feature.
DATACONV	2401	The data conversion feature allows the writing and reading of binary data on 7-track 2400 tape units.
DEKYB2260	2260	This feature specifies an alphameric keyboard with numeric inset for the 2260. The numeric keys are inset in the keyboard in a block arrangement for rapid numeric data entry.
DEKY3277	3158 3277	EBCDIC data entry keyboard.
DESIGNFEAT	2250-1	Graphic design feature for the 2250 Model 1 provides incremental vectors and point plotting, a special fiber optics light pen, and light pen control orders. This feature is standard on the 2250-3. ABSLTVEC must also be specified.
DOCHAR	3277	Domestic character generator.
DUALCODE	BSC1 BSC2 BSC3	The TCU (2701 only) is equipped with the dual code feature. The feature allows processing program selection of the transmission code to be used on the communications line.
DUALCOMM	BSC1 BSC2 BSC3	The TCU (2701 only) is equipped with the dual communication interface feature. This feature allows program selection of either of two modems over which transmission is to occur.
DUALDENS	2401 3410 3420	The dual density feature allows a program to utilize the tape unit as either an 800 BPI or a 1600 BPI machine.
EBKY3277	3158 3277	EBCDIC typewriter keyboard.
FRCHAR	3277 3284 3286	French character generator.

Figure 9 (Part 2 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
GRCHAR	3277 3284 3286	German character generator.
INTERRUPT	2740	This feature indicates that the 2740 is a Model 1 with RPQ #30031 added to allow improved performance when the 2740 is being used as a console.
KACHAR	3277 3284 3286	Katakana character generator.
KB70KEY	3277	70-key keyboard on the 3277.
KB78KEY	3158 3277	78-key keyboard. The feature can only be specified in ASKY3277, EBKY3277, or OCKY3277 is specified.
KB81KEY	3277	81-key keyboard.
LIGHTPEN	2250-1	A light pen is a pen-like device that enables the operator to identify to the program a particular point, line, or character in the displayed image.
LINEADDR	2260	Line addressing is a special feature on the 2848 control unit for the 2260. The feature permits selection of display starting location on incoming data under program control.
MAGCDRD	3277	This feature specifies a magnetic card reader adapter.
MDECOMPAT	2401	The mode compatibility feature enables NRZI tape units (Models 1, 2, and 3) to operate with phase-encoding (PE) tape controllers (2803-2 or 2804-2).
MULTILINE	3525	This feature allows the 3525 Card Punch with the print feature to print up to 25 lines on a card.
NMKYB2260	2260	The numeric keyboard feature specifies that the keyboard is organized like a 10-key adding machine.
NODESCUR	2260	The nondestructive cursor is a feature on the 2848 control unit for the 2260. The feature allows the operator to move the cursor anywhere on the display without changing displayed information.
NUMLOCK	3277	Numeric lock feature.
OCKY3277	3158 3277	This feature specifies a 78-key operator console keyboard on the 3277 or 3158.

Figure 9 (Part 3 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
OIU	2740	The presence of a 2760 Optical Image Unit at a 2740 Communications Terminal is specified by this feature. CHECKING must also be specified. AUTOANSR and AUTOCALL are the only additional optional features that are valid with this feature.
PRGMDYBD	2250	Programmed-function keyboard is a 32-key general-purpose keyboard. The keys of the keyboard are basically unidentified, with their functions defined by application programs.
READWRITE	2401	The feature is specified when the tape device is attached to a simultaneous read-write control unit (2804-1). When this feature is used, OPTCHAN must specify an alternate channel.
SCONTROL	2740	The 2740 Communication Terminal is equipped with the station control feature which allows the terminal to react to a poll or address from a user program. This feature and the AUTOANSR, AUTOCALL, OIU, and XCONTROL features are mutually exclusive.
SELCHSET	1443	The selective character set feature specifies character sets other than the standard 52-character set.
SELPEN	3158 3277	Selector pen.
SHARABLE	3420	This feature allows 3420 magnetic tape drives to be shared between two central processing units when the 3803 two-channel switch is used for partitioning.
SHARED	2305-2 2314 3330	This feature allows the system to share direct-access storage devices with other systems. The device's control unit must be equipped with the 2-channel or 4-channel switch feature.
TWOLINE	3525	This feature allows the 3525 card punch with the print feature to print 1 or 2 lines on a card.
UKCHAR	3277 3284 3286	United Kingdom character generator.
UNVCHSET	1403-2 1403-N1	The universal character set feature provides for printing any set of graphic characters (240 maximum) by the printer specified that is attached to a 2821 Control Unit. The IBM character set images are specified in the UCS macro instruction.
XCONTROL	2740	The 2740 Communication Terminal is equipped with the dial-up feature. The AUTOANSR or AUTOCALL feature, or both, as appropriate, must also be specified. XCONTROL and OIU are mutually exclusive.

Figure 9 (Part 4 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
24ADDPOS	1443	The standard printed line for all character sets on the 1443 is 120 characters long. This feature specifies 24 additional print positions.
7-TRACK or 9-TRACK	2401 3420	The 7-track feature enables the 2400 and 3400 tape units to process tapes that are compatible with other IBM computers that utilize such tape units as the 727, 729, or 7330. These tape units read and write tape in the binary coded decimal (BCD) or binary format. Nine-track is the default. These features are mutually exclusive. The 3410 can only process 9-track tape units.

Figure 9 (Part 5 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Adapter	Unit	TCU	Description
BSCA	BSC1 BSC2 BSC3	2701	Synchronous data adapter, type II, and an appropriate modem.
		2703	Synchronous terminal control and an appropriate modem.
IBM1	1050 1050X 1060 2740	2701	IBM terminal adapter, type I, and an appropriate modem or an IBM line adapter.
	2740C 2740X 2741C 2741P	2702 2703	IBM terminal control, type I, and either a data set line adapter and an appropriate modem or an IBM line adapter.
IBM2	1030	2701	IBM terminal adapter, type II, and either an appropriate modem or an IBM line adapter.
		2702 2703	IBM terminal control, type II, and either a data set line adapter and an appropriate modem or an IBM line adapter.
IBM3	2260 2265	2701	IBM terminal adapter, type III, and an appropriate modem.
IBMT	1050 1050X	2701	IBM telegraph adapter.
		2703	IBM terminal control, type I, and a telegraph line adapter.
TELE1	115A 83B3	2701	Telegraph adapter, type I.
		2702 2703	Telegraph terminal control, type I, and a telegraph line adapter.
TELE2	TWX	2701	Telegraph adapter, type II, and an appropriate modem.
		2702 2703	Telegraph terminal control, type II, and an appropriate modem.
TELEW	WTTA	2701	IBM World Trade Corporation telegraph adapter.
		2702 2703	IBM World Trade Corporation adapter and a telegraph line adapter.

Figure 10. Terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device to a transmission control unit (TCU).

LINKLIB

Optional for:
complete
nucleus
I/O device

The LINKLIB macro instruction is used to add user-written routines, in load-module form, to the link library (SYS1.LINKLIB) in the new system. Before these load modules can be included in the link library, they must be members of a partitioned data set. This data set must have been cataloged as SYS1.name in the generating system. Although the MEMBERS and RESIDNT parameters are optional, at least one must be specified if this macro instruction is included.

You may specify, at your option, those routines written by you that are reentrant and that are to be added to the list of modules that will become a part of the system's link pack area (SYS1.LPALIB).

For a *nucleus generation*, members can be added to or deleted from SYS1.LINKLIB. If you want the same members included that were included in the last complete system generation, you must respecify them.

For an *I/O device generation*, members can be changed or deleted from SYS1.LINKLIB but they cannot be added. If you want the same members included that were included in the last complete system generation, you must respecify them.

```
LINKLIB     [MEMBERS=(name [,name] . . .)]
            PDS=SYS1.name
            [RESIDNT=(name [,name] . . .)]
```

Parameter	Subparameter	Explanation
MEMBERS=	name	specifies the member(s) to be added to the system link library.
PDS=	SYS1.name	specifies the partitioned data set that contains the load-module routines to be added. The name cannot exceed 8 alphameric characters. The first character must be alphabetic.
RESIDNT=	name	specifies the member or members to be added to SYS1.LINKLIB and to the IEAFIX01 list of members that will become part of the resident portion of the link pack area. MEMBERS and RESIDNT are mutually exclusive. A name appearing as a value in either of these parameters must not appear as a value in the other. The total number of names (MEMBERS and RESIDNT) must not exceed 20.

Example: This macro specifies that IOSMAIN, IOSAA, and IOSA1, members of the partitioned data set SYS1.USERLINK, are to be included in SYS1.LINKLIB in the new system.

```
LINK     LINKLIB    PDS=SYS1.USERLINK,                    X
               MEMBERS=(IOSMAIN,IOSAA,IOSA1)
```

LOADER

Optional for:
complete

Not applicable for:
nucleus
I/O device

The LOADER macro instruction specifies the options to be included in the loader processing program. This macro instruction is optional. If it is specified, it can appear only once in the input deck. If it is not specified, the default values are assumed.

Information on the loader program can be found in *OS/VS Linkage Editor and Loader.*

```
LOADER      [LIB=  {lib ddname}]
                   {SYSLIB    }
            [LIN=  {in ddname}]
                   {SYSLIN  }
            [PARM=(option [,option] . . .)]
            [PRINT={print ddname}]
                   {SYSLOUT     }
            [SIZE= {size}]
                   {128 }
```

Parameter	Subparameter	Explanation
LIB=	library ddname SYSLIB	specifies the ddname of the library that will be searched to resolve external references. If this parameter is omitted, SYSLIB is used.
LIN=	input data set ddname SYSLIN	specifies the ddname of the primary input data set that will contain the input to the loader program. If this parameter is omitted, SYSLIN is used.
PARM=		specifies the options that will be used by the loader program. The following options can be listed in any order. If this parameter is not specified, the underlined values will be used.
	PRINT	specifies that the diagnostic messages and the map of external references will be placed on the data set specified by the PRINT parameter of this macro instruction.
	NOPRINT	specifies that neither diagnostic messages nor the map of external references will be produced.
	NOMAP	specifies that the map of external references will not be produced.
	MAP	specifies that the map of external references will be produced on the data set specified by the PRINT parameter of this macro instruction.

Parameter	Subparameter	Explanation
PARM= (Continued)	MAP (Continued)	If you specify this subparameter, the PRINT subparameter of the PARM parameter must also be specified.
	NOLET	specifies that execution of the loaded program will not be attempted when a severity 2 error occurs during loading.
	LET	specifies that execution of the loaded program will be attempted when a severity 2 error occurs during loading.
	CALL	specifies that the partitioned data set specified in the LIB parameter will be searched for any unresolved external references that remain after the input to the loader has been processed.
	NOCALL	specifies that the partitioned data set specified in the LIB parameter of this macro instruction will not be searched.
	NORES	specifies that the link pack area will not be searched.
	RES	specifies that an automatic search of the link pack area is to be made. This search is always made after processing the primary input data set (LIN=ddname) but before searching the SYSLIB data set (LIB=ddname). If you specify this subparameter, CALL must also be specified in the PARM parameter of this macro instruction.
PRINT=	print ddname SYSLOUT	is the ddname of the data set used for the map of external references and diagnostic messages. If this parameter is omitted, SYSLOUT is assumed.
SIZE=	size 128	specifies the amount of virtual storage required by the loader program to hold its own buffers, tables, and the problem program. You specify a decimal number from 2 to 1024, indicating the number of 1K blocks required. If this parameter is omitted, 128 is assumed.

Example: This macro selects the options for the loader. The amount of virtual storage needed is 128K bytes. The input data set is SYSLIN, the output data set is SYSLOUT, and the external reference library is SYSLIB. Diagnostic messages and the map of external references will be placed on the SYSLOUT data set. Default values are assumed for the CALL and NORES subparameters.

```
LOADER   LOADER   LIN=SYSLIN,LIB=SYSLIB,                  X
                PRINT=SYSLOUT,SIZE=128,                  X
                PARM=(PRINT,MAP)
```

LPALIB

Optional for:
complete
nucleus

Not applicable for:
I/O device

The LPALIB macro instruction is used to add user-written routines, such as type 3 or 4 SVCs, in load module form, to the link pack area (SYS1.LPALIB). Additionally, you may specify those user-written routines that are to be added to the resident portion of the link pack area.

Before the load modules can be included in SYS1.LPALIB, they must be members of a partitioned data set. This data set must have been cataloged as SYS1.name in the generating system.

If type 3 or 4 SVCs are specified, an SVCTABLE macro must also be specified.

For a *nucleus generation*, members can be added to or deleted from SYS1.LPALIB. If you want the same members included that were included in the last complete system generation, you must respecify them.

For additional information on user-written routines, refer to *OS/VS2 Planning and Use Guide.*

```
LPALIB      [MEMBERS=(name[,name]...)]
            PDS=SYS1.name
            [RESIDNT=(name[,name]...)]
```

Parameter	Subparameter	Explanation
MEMBERS=	name	specifies the member or members to be added to SYS1.LPALIB.
PDS=	SYS1.name	specifies the partitioned data set that contains the load module routines to be added to SYS1.LPALIB. The name cannot exceed 8 alphameric characters. The first character must be alphabetic.
RESIDNT=	name	specifies the member or members to be added to SYS1.LPALIB and to the IEAFIX01 list of members that will become part of the resident portion of the link pack area. MEMBERS and RESIDNT are mutually exclusive. A name appearing as a value in either of these parameters must not appear as a value in the other. The total number of names (MEMBERS and RESIDNT) must not exceed 20.

Example: This macro specifies that two user-written accounting routines and two user-written SVC routines, members of SYS1.USERLIB, are to be included in SYS1.LPALIB and to the list of members that are to be included in the resident portion of the link pack area.

```
USERLIB  LPALIB    PDS=SYS1.USERLIB,                           X
                 MEMBERS=(ACCT1,ACCT2),                        X
                 RESIDNT=(IGC248,IGC249)
```

MACLIB

Optional for:
complete

Not applicable for:
nucleus
I/O device

The MACLIB macro instruction is used to exclude groups of macro definitions from the macro library of the new VS2 system. This macro instruction is optional. If it is omitted, the macro library is used in its entirety.

```
MACLIB      EXCLUDE=(option [,option] . . .)
```

Parameter	Subparameter	Explanation
EXCLUDE=		specifies groups of macro definitions to be excluded from the new system's macro library (SYS1.MACLIB). These values can be listed in any order.
	BTAM	specifies that the macro definitions used for BTAM are to be excluded.
	GPS	specifies that the macro definitions used for graphics programming services (GPS) are to be excluded.
	OCR	specifies that the macro definitions used for optical character readers are to be excluded.
	TCAM	specifies that the macro definitions used for TCAM are to be excluded. If the TSO macro is specified, TCAM will not be excluded even if it is specified in this macro instruction.
	TSO	specifies that the macro definitions used for TSO are to be excluded.

Example: This macro specifies that the BTAM macro definitions are to be excluded from the macro library.

```
MACLIB    MACLIB    EXCLUDE=(BTAM)
```

PAGE

Required for:
complete
nucleus
I/O device

The PAGE macro instruction is used to provide the nucleus initialization program (NIP) with the information needed to allocate space to the page data sets. Up to sixteen page data sets may be defined. You must specify a PAGE macro for each data set required. Valid devices are: 2305-1, 2305-2, 2314/2319, 3330. (The 2314 and 2319 are considered the same device type.)

The parameter values specified for the PAGE macro are placed in the IEASYS00 list in SYS1.PARMLIB during Stage II. Because of this, you may redefine page data sets at IPL, or after system generation, using the IEBUPDTE utility program.

Specifications for the page data sets may be added to or deleted from the system in a *nucleus or I/O device generation.*

For information about paging, refer to *OS/VS2 Planning and Use Guide.* For information on the size of page data sets, refer to *OS/VS2 Storage Estimates.*

```
PAGE      [LPA= {YES} ]
                {NO } 
                 ---
          [SIZE= { ( {BLK}                 } ]
                 {   {TRK} ,number)        }
                 {   {CYL}                 }
                 { MAX                     }
                   ---
          [TYPE= {S} ]
                 {P}
                  -
          [ {UNIT=address       } ]
            {VOLNO=volume serial}
```

Parameter	Subparameter	Explanation
LPA=		specifies that the device being specified is to contain the pageable link pack area. If LPA is not specified in any PAGE macro instruction, the first PAGE macro will be assumed to contain the pageable link pack area.
	YES	specifies that this device will contain the pageable link pack area. YES may only be specified in one PAGE macro.
	<u>NO</u>	specifies that this device will not contain the pageable link pack area.
SIZE=		specifies the size of the page data set. The space that will be allocated will be contiguous. If this parameter is not specified, MAX is used.

Parameter	Subparameter	Explanation
SIZE= (Continued)	BLK	specifies that space is to be allocated in 4K blocks.
	TRK	specifies that space is to be allocated in tracks.
	CYL	specifies that space is to be allocated in cylinders
	number	specifies the number of blocks, tracks, or cylinders to be allocated.
	MAX	specifies that the largest contiguous extent on the selected volume be used for the page data set. For storage estimates for page data sets, refer to *OS/VS2 Storage Estimates.*
TYPE=		specifies the type of auxiliary storage device that is to be used for paging operations.
	P	specifies a primary device is to be used for paging operations.
	S	specifies a secondary device is to be used for paging operations. Information about primary and secondary auxiliary storage devices is in *OS/VS2 Planning and Use Guide.*
UNIT=	address	specifies the address of the device that is to contain the page data set. This device must also be specified in an IODEVICE macro instruction. IOREQUE= PRIORITY should also be specified in an IODEVICE macro for the device that is to contain the page data set. If PRIORITY is not specified, performance will be degraded.
VOLNO=	volume serial	specifies the volume serial number of the volume that is to contain the page data set. If UNIT or VOLNO is not specified, VOLNO=PAGEnn is used (where 00 ⩽ nn ⩽ 15) to specify the first through sixteenth page data set.

Example: This macro specifies a secondary page data set (TYPE=S). The unit address of this page data set is 430 and the size is to be the largest contiguous extent available on the selected volume.

```
PAGE     SIZE=MAX,TYPE=S,UNIT=430
```

RESMODS

Optional for:
complete
nucleus

Not applicable for:
I/O device

The RESMODS macro instruction is used to add your own routines, in load module form, to the nucleus (member name IEANUC01). Before these routines can be included, they must be members of a partitioned data set. This data set must have been cataloged as SYS1.name in the generating system.

A RESMODS macro instruction must be specified for each type 1 or type 2 user-written SVC routine to be included in the new system. If type 1 or 2 SVCs are specified, an SVCTABLE macro must also be specified.

For a *nucleus generation*, sufficient space must have been allocated to SYS1.NUCLEUS in the previous generation. Members may be added to or deleted from SYS1.NUCLEUS. If you want the same members included, you must respecify this macro with the same parameters and sub-parameters and you must also respecify the SVCTABLE macro instruction.

RESMODS **MEMBERS=(*name* [,*name*] . . .)**
PDS=SYS1.*name*

Parameter	**Subparameter**	**Explanation**
MEMBERS=	name	specifies the name of the member of the partitioned data set to be included in the nucleus. The name must be 1 to 8 alphameric characters, the first of which is alphabetic. A maximum of ten load modules can be included in the nucleus. For further information on writing your own SVC routines, refer to the *OS/VS2 Planning and Use Guide.*
PDS=	name	specifies the partitioned data set that contains the load modules to be included in the nucleus. The name must be 1 to 8 alphameric characters, the first of which is alphabetic.

Example: This macro specifies that two user-written SVC routines that are members of the partitioned data set SYS1.USERLIB are to be included in the nucleus (SYS1.NUCLEUS).

```
RESMODS  RESMODS  PDS=SYS1.USERLIB,                       X
                  MEMBERS=(IGC240,IGC241)
```

SCHEDULR

Required for:
complete
nucleus
I/O device

The SCHEDULR macro instruction specifies the job scheduler and master scheduler options. This macro instruction is required.

For a *nucleus generation*, only WTOBFRS and REPLY can be changed. All other parameters must be respecified exactly as they were during the last complete system generation.

For an *I/O device generation*, the CONSOLE and ALTCONS parameters can be changed. The console and alternate console addresses may be changed. No other parameters can be changed and they must be respecified exactly as they were during the last complete system generation. The addresses specified in the STARTR and STARTW parameters must be the same as those specified in the last complete system generation, but they can be changed at IPL.

```
SCHEDULR        [ALTCONS= {address               }]
                          {(I-address,O-address) }

                [AREA=(nn[,nn]...)]

                [BCLMT= {integer}]
                        {100    }

                CONSOLE= {address               }
                         {(I-address,O-address) }

                [ESV= {SMF}]
                      {CON}

                [EVA=(n1,n2)]

                [HARDCPY=({address}
                          {SYSLOG }

                          {,(routing-code[,routing-code]...)}
                          { ALL                             }

                          [,{CMDS  }])]
                            {INCMDS}
                            {STCMDS}

                 [INITQBF=number]

                IOC=address

               *[JOBQFMT= {number}]
                          {12    }

               *[JOBQLMT= {number}]
                          {60    }

               *[JOBQTMT= {number}]
                          {60    }

               *[JOBQWTP= {number}]
                          {2     }
```

*These parameters may be overridden by the operator during IPL.

```
 [OLDWTOR=(routing-code [,routing-code] . . .)]
 [PFK=nn]
 [REPLY= {number} ]
         {10    }
 [ROUTCDE= (routing-code [,routing-code] . . .)]
*[STARTI= {AUTO  } ]
          {MANUAL}
*[STARTR=(A-address [,V-serial] [,D-dsname] )]
*[STARTW=(A-address [,V-serial] [,D-dsname] )]
 [VLMOUNT=AVR] [,TAVR= {200} ]
                       {556}
                       {800}
 [WTLBFRS= {number} ]
           {4     }
 [WTLCLSS= {classname} ]
           {L        }
 [WTOBFRS=number]
```

*These parameters may be overridden by the operator during IPL.

Parameter	Subparameter	Explanation
ALTCONS=	address	specifies the unit address of a device having input and output capability that is to be used as an alternate console.
	I-address	specifies, for a composite console, the unit address of an input device.
	O-address	specifies, for a composite console, the unit address of an output device.
		Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. The devices that can be used are listed in Figure 10.
		A device specified as a part of a composite console cannot be specified in any other way (as a single device or as a part of a composite console with a different companion device). However, the same combination may be repeated more than once in the SECONSLE macro instruction as an alternate console.
		The device specified as an alternate console in this parameter must also be specified as a secondary console in the SECONSLE macro.
AREA=	nn	specifies the dimensions of the display areas to be set aside for status displays on the display screen of the console specified in the CONSOLE parameter of the SCHEDULR macro instruction.

Parameter	Subparameter	Explanation
AREA= (Continued)	nn (Continued)	The value specified must be a decimal number equal to the number of display screen lines to be in the display area. Each nn defines one display area of the size indicated. The first nn defines the bottommost display area on the screen (the bottom lines of the message area). Subsequent nn's define areas stacked above the bottommost area. The minimum specification is 4 lines. The maximum specification for all areas is: 47 if a 2250 is used 8 if a 2260 is used 30 if a 3066 is used 19 if a 3158 is used 19 if a 3277 Model 2 is used The default is a single area with a length of: 14 if a 2250 is used 8 if a 2260 is used 14 if a 3066 is used 14 if a 3158 is used 14 if a 3277 Model 2 is used This parameter is invalid for any other consoles. For further information about display consoles, refer to *Operator's Library: OS/VS2 Display Consoles.*
BCLMT=	number 100	is an integer from 1 to 1000 that specifies the number of 130-byte records that will be set aside for broadcast messages in the SYS1.BRODCAST system data set. If this parameter is specified, the SYS1.BRODCAST system data set should be specified in a DATASET macro instruction. If this parameter is not specified, a value of 100 is used.
CONSOLE=	address	specifies the unit address of a console device, having input and output capability, to be used as the master console. A master console must always be specified.
	I-address	specifies, for a composite console, the unit address of an input device.
	O-address	specifies, for a composite console, the unit address of an output device. Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. The devices that can be used are listed in Figure 10. A device specified as a part of a composite console cannot be specified in any other way (as a single device or as a part of a composite console with a different companion device).

Parameter	Subparameter	Explanation
CONSOLE= (Continued)	O-address (Continued)	If a graphics device will be active as a console, a device that produces printed output must also be specified. If a device is specified as the master console (in the CONSOLE parameter of the SCHEDULR macro), it cannot be specified as the secondary console (in the CONSOLE parameter of the SECONSLE macro); it can, however, be specified again as an alternate console in the SECONSLE macro.
ESV=		specifies the destination of volume error statistics (ESV) records. If this parameter is not specified, CON is used.
	CON	specifies that an abridged version of the records is to be constructed and put on the console.
	SMF	specifies that the records are to be constructed and written on the SYS1.MANX or SYS1.MANY system data sets.
EVA=		specifies the use of error volume analysis (EVA) and the number of temporary read and write errors that will cause an ESV message to be printed on the console. If this parameter is not specified, EVA will not be included in the new system.
	n_1, n_2	are integers from 1 to 255. n_1 is the number of temporary read errors that will cause an ESV message to be printed on the console. n_2 is the number of temporary write errors that will cause an ESV message to be printed on the console.
HARDCPY=		specifies that a hard-copy log will be used to record operator commands, system commands and responses, and write-to-operator (WTO and WTOR) messages. (Control (K) commands, which control console functions rather than system functions, are not recorded on the hard-copy log but are recorded in the SYSLOG data set.) If this parameter is not specified, SYSLOG, ALL, and NOCMDS are used. For information about operator communication with the system, the hard-copy log, and the system log, refer to *Operator's Library: OS/VS Console Configurations* and *OS/VS Supervisor Services and Macro Instructions.* The subparameters for HARDCPY are positional and must be coded in the sequence shown in the macro instruction format. For any subparameter omitted, a comma must be written to indicate its absence. For example, HARDCPY=(,ALL,CMDS) indicates the absence of the unit address subparameter.

Parameter	Subparameter	Explanation
HARDCPY= (Continued)	SYSLOG	specifies that the data that is supposed to go to the hard-copy log will go to the system log.
	address	specifies the unit address of a device with at least output capability to be used as the hard-copy log device. (See Figure 10 for a list of the devices that can be used.) The unit address specified for the device must also be specified for that device in an IODEVICE macro instruction. This device must also be specified in either the CONSOLE parameter of the SCHEDULR macro or a SECONSLE macro. A graphics device cannot be specified as the hard-copy log device.
	ALL	specifies that all write-to-operator (WTO and WTOR) messages are to be put on the hard-copy log.
	routing code	is a number from 1 to 16 that designates the routing code that the hard-copy log is authorized to receive for each operator's console specified in the CONSOLE parameter of the SCHEDULR and SECONSLE macro instructions. For information on routing and descriptor codes, refer to *OS/VS Message Library: Routing and Descriptor Codes.*
	CMDS	specifies that operator and system commands, responses and status displays (static and time-interval updated) are to be written on the hard-copy log.
	INCMDS	specifies that operator and system commands and responses (but not status displays) are to be written on the hard-copy log.
	STCMDS	specifies that operator and system commands, responses, and status displays (except time-interval updated status displays) are to be written on the hard-copy log. If neither CMDS, INCMDS, nor STCMDS is specified, no operator or system commands or responses will be put on the hard-copy log.
INITQBF=	number	specifies the number of 1024-byte buffers in real storage that will hold logical tracks from SYS1.SYSJOBQE. The number specified must be a decimal integer from 0 to 255. Information about the storage required by SYS1.SYSJOBQE can be found in *OS/VS2 Storage Estimates.*

Parameter	Subparameter	Explanation
IOC=	address	is the unit address of an integrated operator console. The value specified must be the unit address of a 3066, 3158, 3210, or 3215 console. This parameter must be specified.
JOBQFMT=	number 12	specifies the number of 176-byte queue records that can be written on a track allocated to SYS1.SYSJOBQE. The number specified is an integer from 5 to 255. This number represents the total number of 176-byte records on each track. This value can be changed at IPL when the system is loaded. If this parameter is not specified, 12 is used. Refer to *OS/VS2 Storage Estimates* for information on determining this value.
JOBQLMT=	number 60	specifies the number of 176-byte records in SYS1.SYSJOBQE to be reserved for each initiator started. The number specified is an integer from 60 to 9999. This value can be changed at IPL when the system is loaded. If this parameter is not specified, 60 is used. Refer to *OS/VS2 Storage Estimates* for information on determining this value.
JOBQTMT=	number 60	specifies the number of 176-byte records in SYS1.SYSJOBQE to be reserved for the termination of jobs that require more records for initialization than those specified in JOBQLMT. The number specified is an integer from 60 to 9999. This value can be changed at IPL when the system is loaded. If this parameter is not specified, 60 is used. Refer to *OS/VS2 Storage Estimates* for information on determining this value.
JOBQWTP=	number 2	specifies the number of 161-byte records in SYS1.SYSJOBQE that the write-to-programmer routine can use for messages during a job. The number specified is an integer from 2 to 20. This value can be changed at IPL when the system is loaded. If this value is not specified, 2 is used. Refer to *OS/VS2 Storage Estimates* for information on determining this value.
OLDWTOR=	routing code	is a number from 1 to 16 that specifies the routing code to be assigned to all write-to-operator (WTO and WTOR) messages that do not already have routing and descriptor codes. If this parameter is omitted, the master console will receive the WTO and WTOR messages that do not have routing and descriptor codes. For information on routing and descriptor codes, refer to *OS/VS Message Library: Routing and Descriptor Codes.* For information on specifying WTO and WTOR macros, refer to *OS/VS Supervisor Services and Macro Instructions.*

Parameter	Subparameter	Explanation
PFK=	number	specifies that the console is to have programmed-function-keyboard (PFK) command entry and/or light-pen command entry. This parameter is valid only for a 2250, 3158 or 3277 Model 2 that is specified as a console with input-output capability or as the input half of a composite console. The number specified is a decimal number from 1 to 12 that indicates: • the number of PFK keys that the operator can associate with commands after IPL (2250 or 3277 Model 2), or • the number of light-pen-detectable numerical indicators in the PFK line of the screen that the operator can associate with commands after IPL (2250, 3158, or 3277 Model 2). If the specified console has both a PFK and a light pen, both methods of command entry will be made available when the PFK parameter is specified. If this parameter is specified, space must be allocated for SYS1. DCMLIB. If this parameter is specified for a 2250, either the LIGHTPEN or PRGMKYBD features, or both, must be specified in an IODEVICE macro instruction. If this parameter is specified for a 3277 Model 2, either the SELPEN or programmed-function keyboard (KB78KEY) features, or both, must be specified in an IODEVICE macro instruction. The SELPEN feature must be specified for the 3158. Information on the programmed-function keyboard, selector pen, and light pen features can be found in *Operator's Library: OS/VS2 Display Consoles.*
REPLY=	number 10	specifies the number of reply queue elements to be used by the WTOR routines. Each reply queue element is 24 bytes. If this parameter is not specified, 10 is assumed.
ROUTCDE=	routing code	is a number from 3 to 16 that specifies the additional routing codes the master console can receive. Routing codes 1 and 2 are always automatically assigned for use by the master console (see *OS/VS Message Library: Routing and Descriptor Codes).*
STARTI=		specifies whether a START INIT command is to be executed automatically after each IPL. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS2 Reference* for this information.) If this parameter is not specified, MANUAL is used.
	AUTO	specifies that the command is to be executed automatically

Parameter	Subparameter	Explanation
STARTI= (Continued)	MANUAL	specifies that it is not to be executed automatically.
STARTR=		specifies that a START RDR command is to be executed automatically for a device each time the new system is loaded after each IPL. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS2 Reference* for this information.)
	A-address	is the unit address of the input device to be started.
	V-serial	When the device to be started is on tape or disk: specifies the volume serial number of the labeled volume associated with the device.
	D-dsname	specifies the name of the data set associated with the device to be started. The data set name can be from 1 to 8 characters; the first character must be alphabetic.
STARTW=		specifies that a START WTR command is to be executed automatically for a device each time the new system is loaded after IPL. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS2 Reference* for this information.)
	A-address	is the unit address of the output device to be started.
	V-serial	When the device to be started is tape or disk: specifies the volume serial number of the labeled volume associated with the device.
	D-dsname	specifies the name of the data set associated with the device to be started. The data set name must be from 1 to 8 characters, the first character must be alphabetic.
TAVR=	200 556 800	specifies the standard density for 7-track magnetic tape volumes used with automatic volume recognition (AVR). If this parameter is not specified, 800 BPI is used. This parameter is specified only if the VLMOUNT= AVR parameter is coded.
VLMOUNT=	AVR	specifies automatic volume recognition. For information about AVR, refer to *OS/VS JCL Services.*
WTLBFRS=	number 4	specifies the number of buffers that will be used as temporary storage areas for write-to-log (WTL) messages that will be written on the SYS1.SYSVLOGX or SYS1.SYSVLOGY system data sets. The number specified is an integer from 4 to 255. Although each system has different system log requirements, an optimum number for best log performance is from

Parameter	Subparameter	Explanation
WTLBFRS= (Continued)	number 4 (Continued)	10 to 15. If this parameter is not specified, 4 is assumed. The number you specify will be multiplied by 148 (which is the maximum size of a WTL message) to determine the size of the buffer area. No matter what value is specified, the size of the area will not be greater than twice the track size of the device type for which SYS1.SYSVLOGX and SYS1.SYSVLOGY are cataloged. For information on WTL messages, refer to *OS/VS Supervisor Services and Macro Instructions.*
WTLCLSS=	classname L	specifies the classname to be used as a default for SYSOUT write-to-log messages. Classname is a letter from A to Z or a number from 0 to 9. If this parameter is not specified, L is used.
WTOBFRS=	number	specifies the number of buffers to be used by the write-to-operator (WTO) routines. The number specified is an integer greater than or equal to 4. The number specified should at least be equal to four times the number of initiators expected to be active at any one time. Each buffer will be 168 bytes. If the value specified is less than two buffers per console, then the specified value is ignored and two buffers per console are assigned. If this parameter is not specified, two buffers are assigned for each operator's console specified in the SECONSLE macro instruction. For information on communiation between the operator and the system, refer to *OS/VS Supervisor Services and Macro Instructions.*

Device	Model	ALTCONS/ CONSOLE address	I-address	O-address	HARDCPY address	Notes
1052	7	X	X	X	X	Console keyboard-printer. Can only be specified when attached through the 2150 adapter.
1403				X	X	Printer (note 1)
1443	N1			X	X	Printer (note 1)
2250		X	X	X		Display unit (note 2)
2260	1	X	X	X		Display station (notes 1 and 3)
2501			X			Card reader
2520	B1		X			Card reader punch
2540R			X			Card reader
2740	1	X	X	X	X	Communication terminal (note 3)
3066		X	X	X		System console. May only be specified in a S370/165II or 168 system configuration, and can only appear once in the system as a master or secondary console (note 3).
3158		X	X	X		Display unit (notes 1 and 3)
3210		X	X	X	X	Console print keyboard
3211				X	X	High-speed printer. The universal character set is included. (note 1)
3213				X	X	Printer
3215		X	X	X	X	Console print keyboard
3277	1			X		Display unit. May only be specified as an O-address and must only be specified in the SECONSLE macro.

Figure 11 (Part 1 of 2). Console and alternate console support. This figure lists the devices that can be used as consoles and alternate consoles in the SCHEDULR and SECONSLE macro instructions.

Device	Model	ALTCONS/ CONSOLE address	I-address	O-address	HARDCPY address	Notes
3277	2	X	X	X		Display unit (note 3)
3284	1			X	X	Printer (note 1)
3284	2			X	X	Printer (note 1)
3286	1			X	X	Printer (note 1)
3286	2			X	X	Printer (note 1)
3505			X			Card reader
3525			X			Card punch. Must have read feature.

Notes:

1. These devices may be specified as O-address only consoles in the SECONSLE macro instruction.
2. Refer to Figure 7 for IODEVICE specifications and parameters required when this device is used as an operator's console. The GRAPHICS macro instruction is required for a 2250 Model 3.
3. Refer to Figure 7 for IODEVICE specifications and parameters required when this device is used as an operator's console.

Figure 11 (Part 2 of 2). Console and alternate console support. This figure lists the devices that can be used as consoles and alternate consoles in the SCHEDULR and SECONSLE macro instructions.

Example: This macro specifies the options for the scheduler. The following options are specified:

- The system start commands are to be issued manually
- Automatic volume recognition
- The density for 7-track tapes is 200 BPI
- The output class for the system log is SYSOUT=D
- Twelve queue records per logical track
- SYS1.BRODCAST is to have room for 100 notice messages
- The console address is 009
- The alternate console addresses are 00C for input and 00E for output
- Ten reply queue requests will be permitted
- The integrated operator console address is 009
- The system log will be the hard-copy log
- All WTO messages will be written on the hard-copy log
- Status displays will not be written to the hard-copy log
- Operator commands and responses will be written to the hard-copy log.

```
SCHEDULR  SCHEDULR  CONSOLE=009,                    X
               ALTCONS=(I-00C,O-00E),STARTI=MANUAL, X
               REPLY=10,HARDCPY=(SYSLOG,ALL,INCMDS),X
               BCLMT=100,VLMOUNT=AVR,TAVR=200,      X
               WTCLSS=D,ESV=NO,IOC=009
```

SECONSLE

Optional for:
complete
nucleus
I/O device

The SECONSLE macro instruction is used to specify secondary support for the standard multiple console support function. This macro instruction is optional. It must, however, be specified if the ALTCONS parameter of the SCHEDULR macro instruction is specified.

For a *nucleus generation*, this macro instruction must be specified if it was specified in the last complete system generation. The same parameters and subparameters must be coded. If the SECONSLE macro instruction was not specified in the last complete generation, it cannot be specified.

For an *I/O device generation*, this macro instruction must be specified if it was specified in the last complete system generation. This macro instruction can be used to add, delete, or change the secondary console specifications. If this macro instruction was not specified in the last complete generation, it cannot be specified.

The alternate console for the master console that is specified in the ALTCONS parameter of the SCHEDULR macro instruction must be used in the CONSOLE parameter of this macro instruction.

This macro instruction must be used to specify each additional secondary console.

You must specify a SECONSLE macro instruction for each secondary console to be included in the new system. A maximum of 31 secondary consoles can be specified. If more are specified, the system generation process is terminated.

Each device specified in an ALTCONS parameter of a SCHEDULR or SECONSLE macro instruction must also be specified in a CONSOLE parameter of a SCHEDULR or SECONSLE macro instruction.

```
SECONSLE      [ALTCONS= ( address                 ) ]
                        { O-address               }
                        ( (I-address,O-address)   )
              [AREA=(nn [,nn] . . . )]
              CONSOLE= ( address                 )
                       { O-address               }
                       ( (I-address,O-address)   )
              [PFK=nn]
              [ROUTCDE= { ALL                                       } ]
                        { (routing code[,routing code] . . .)      }
              [USE= { SD } ]
                    { MS }
              [VALDCMD=(command code [,command code] . . .) ]
```

Parameter	Subparameter	Explanation
ALTCONS=		specifies the address or addresses of the alternate console.
	address	is the unit address of an alternate console device that can be used for input and output.
	I-address	is the unit address of an input device for a composite console.
	O-address	is either the unit address of an output device for a composite console or the unit address of a console that can be used only for output. If this parameter is omitted, the master console specified in the CONSOLE parameter of the SCHEDULR macro instruction is assigned as the alternate console. The device specified must also be specified in the CONSOLE parameter of either the SCHEDULR or a SECONSLE macro instruction. Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. Figure 11 lists the devices that can be used. A console that can be used only for output cannot be specified as the alternate console for a console that can be used for both input and output. A device specified as a part of a composite console cannot be specified in any other way (as a single device or as a part of a composite console with a different companion device).
AREA=	nn	specifies the dimensions of the display areas to be set aside for status displays on the display screen of the console specified in the CONSOLE parameter of the SECONSLE macro instruction. The value specified must be a decimal number equal to the number of display screen lines to be in the display area. Each nn defines one display area. The first nn defines the bottom-most display area on the screen (the bottom lines of the message area). Subsequent nn's define areas stacked above the bottom-most area. The minimum specification is 4 lines. The maximum specification for all areas is: 47 if a 2250 is used 8 if a 2260 is used 11 if a 2260 is used only for output (USE=SD) 30 if a 3066 is used 19 if a 3277 Model 2 is used 23 if a 3277 Model 2 is used only for output (USE=SD)

Parameter	Subparameter	Explanation
AREA= (Continued)	nn (Continued)	The default values are: 14 if a 2250 is used 8 if a 2260 is used 11 if a 2260 is used only for output (USE=SD) 14 if a 3066 is used 14 if a 3277 Model 2 is used (13, 10) if a 3277 Model 2 is used only for output (USE=SD) This parameter is invalid for any other console and is also invalid for a 2250, 2260, or 3277 Model 2 when USE=MS has been specified in the SECONSLE macro. For further information about display consoles, refer to *Operator's Library: OS/VS2 Display Consoles.*
CONSOLE=		specifies the address or addresses of the secondary console.
	address	specifies the unit address of the secondary console that can be used for input and output.
	I-address	is the unit address of an input device for a composite console.
	O-address	is either the unit address of an output device for a composite console or the unit address of a console that can be used only for output. Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. For a list of the devices that can be used, see Figure 11. A device specified as a part of a composite console cannot be specified in any other way (as a single device or as a part of a composite console with a different companion device). If a device is specified as the master console (in the CONSOLE parameter of the SCHEDULR macro), it cannot be specified as the secondary console (in the CONSOLE parameter of the SECONSLE macro); it can, however, be specified again as an alternate console.
PFK=	number	specifies that the console is to have programmed-function-keyboard (PFK) command entry and/or light pen command entry. This parameter is valid only for a 2250, 3158, or 3277 Model 2 that is specified as a console with input/output capability or as the input half of a composite console. The number specified is a decimal number from 1 to 12 that indicates: • the number of PFK keys that the operator can associate with commands after IPL (2250 or 3277 Model 2), or

Parameter	Subparameter	Explanation
PFK= (Continued)	number (Continued)	• the number of light-pen-detectable numerical indicators in the PFK line of the screen that the operator can associate with commands after IPL (2250, 3158, or 3277 Model 2). If the specified console has both a PFK and a light pen, both methods of command entry will be made available when the PFK parameter is specified. If this parameter is specified, space must be allocated for SYS1.DCMLIB. If this parameter is specified for a 2250, either the LIGHTPEN or PRGMKYBD features or both must be specified in an IODEVICE macro instruction. If this parameter is specified for a 3277 Model 2, either the SELPEN or programmed-function keyboard (KB78KEY) features, or both, must be specified in an IODEVICE macro instruction. The SELPEN feature must be specified for the 3158. Information on the programmed-function keyboard, selector pen, and light pen can be found in *Operator's Library: OS/VS2 Display Consoles.*
ROUTCDE=		specifies which routing codes the console will receive. If this parameter is not specified, no routing codes are assigned to this console.
	ALL	specifies that the console can receive all routing codes.
	routing code	is a number from 1 to 16 that designates which routing code will be recognized and accepted by this console. (Refer to *OS/VS Message Library: Routing and Descriptor Codes* for additional information.)
USE=		specifies the intended use of a 2260, 3277 Model 2, or 3158 display (CRT) console. If this parameter is not specified, full capacity is assumed.
	SD	specifies that the console is to be used as an output-only console for status displays.
	MS	specifies that the console is to be used as an output-only console to display operator messages.
VALDCMD=	command code	is a number from 1 to 3 that specifies which commands can be entered from this console. One or more numbers may be specified to indicate which command groups can be entered from the console. For information on command groups, refer to *Operator's Library: OS/VS2 Reference.*

Example: This example specifies that a 3277 Model 2 at address 018 is to be used as a secondary console. A 2260 Model 1 at address 001 is to be used as the alternate console. (These devices and addresses were also specified in IODEVICE macros.) The status display areas are to have lengths of 7. No routing codes have been assigned to this console.

```
SECON018  SECONSLE  CONSOLE=018,ALTCONS=001,             X
                    AREA=(7,7)
```

SVCTABLE

Optional for:
complete
nucleus
I/O device

The SVCTABLE macro instruction is used to specify the number, type, and function code of the user-written supervisor call (SVC) routines that are to be added to the new system. This macro may be specified more than once.

Type 1 and 2 SVCs: For each type 1 or type 2 SVC, a corresponding member should be specified in the RESMODS macro instruction. One member may contain more than one resident SVC routine.

Type 3 and 4 SVCs: For each type 3 or type 4 SVC, a corresponding member should be specified in the LPALIB macro instruction. Each member may contain only one SVC routine.

Nucleus generation: An SVCTABLE macro instruction must be specified for each type 1 or 2 SVC specified in the RESMODS macro instruction and for each type 3 or 4 SVC specified in an LPALIB macro instruction.

I/O device generation: If this macro instruction was not used in the last complete generation, it cannot be specified for an I/O device generation. If it was specified, it must be respecified with the same operands.

SVCTABLE *operand* [*,operand*] . . .

Parameter	**Subparameter**	**Explanation**
operand		Each operand is written in the format shown below. Uppercase letters and hyphens must be coded as shown.

$$\text{SVC-nnn-}\left\{\begin{matrix}\text{Da}\\ \text{Ea}\end{matrix}\right\}\left[\text{-}\left\{\begin{matrix}\text{FC01}\\ \underline{\text{FC00}}\end{matrix}\right\}\right]$$

where:

nnn specifies the SVC number as a decimal number

D specifies that the SVC is being entered with interruptions disabled.

E specifies that the SVC is being entered with interruptions enabled

a specifies the SVC type as either 1, 2, 3, or 4

E1 (type 1 SVC entered enabled) is an invalid parameter. All type 1 SVCs must be entered disabled.

FC01 indicates a function code of 01 is to be assigned to the SVC. This code will restrict the use of the SVC to authorized job steps. If this value is not specified, a value of FC00 is assumed making the SVC unrestricted.

Parameter	Subparameter	Explanation
operand (Continued)		For additional information on the Authorized Program Facility (APF), refer to *OS/VS2 Planning and Use Guide.* You must assign unique numbers (nnn) to your SVC routines. You should assign them in descending order starting with 255 and ending with 200 to avoid conflict with the numbers assigned to IBM-written routines. (For an example of adding a user-written SVC routine, refer to "Adding User-Written Routines to the System Control Program.") For information about SVCs and APF, refer to *OS/VS2 Planning and Use Guide.*

Example: This macro specifies the number and type of three user-written SVC routines. SVC255 is a type 1 SVC that is being entered with interruptions disabled. SVC254 is a type 4 SVC, SVC253 is a type 3 SVC and both are being entered with interruptions enabled. An authorization code (APF) was not specified so these SVCs are not restricted.

```
SVCTABLE  SVCTABLE  SVC-255-D1,SVC-254-E4,          X
                    SVC-253-E3
```

TSO

Optional for:
complete
nucleus
I/O device

The TSO macro instruction is used to specify a system with TSO.

If the TSO macro is specified, TCAM is included in the system.

For a *nucleus* or *I/O device generation,* this macro must be respecified if it was originally specified. The same parameters and subparameters must be coded.

```
TSO      [LOGLINE={ integer } ]
                  { 10      }
         [LOGTIME={ time } ]
                  { 300  }
         [OUTCLS=( { [classname1] [,classname2] } )]
                   {      X            X        }
         [SUBMITQ={ number } ]
                  { 50     }
         [TSOAUX={ number } ]
                 { 0      }
```

Parameter	Subparameter	Explanation
LOGLINE=	number 10	specifies the number of lines that may be entered before an attempt to log on is automatically canceled. The number specified is an integer from 1 to 16,777,215. If this parameter is not specified, a 10 is used.
LOGTIME=	time	specifies the number of seconds you may wait without a terminal response during a LOGON. The number specified is an integer from 1 to 16,777,215. If this parameter is not specified, 300 seconds is used.
OUTCLS=		specifies the class defaults for the OUTPUT command. The value must be specified as a letter from A to Z or a number from 0 to 9. If this parameter is not specified, X is used for both classname values.
	classname 1	is the background-message class default.
	classname 2	is the printed-output class default.
SUBMITQ=	number 50	specifies the maximum number of logical tracks to be reserved in the system job queue for TSO jobs submitted for background running. The value specified is an integer from 0 to 9999. If this parameter is not specified, 50 is used.

Parameter	Subparameter	Explanation
TSOAUX=	number <u>0</u>	specifies the percentage of the first 4096 pages of external page storage to be made available to TSO. The value specified is an integer from 0 to 100. If this parameter is not specified, 0 is used.

Example: This macro specifies a printed-output class of A, that 600 seconds may elapse before LOGON issues a message, that 5 lines may be entered before a LOGON attempt is canceled, and that 100 logical tracks are to be reserved for TSO foreground-initiated-background (FIB) jobs in the system job queue. The default value of X is used for the background message class. No additional external page storage will be used.

```
TSODF    TSO    LOGTIME=600,LOGLINE=5,SUBMITQ=100,      X
                OUTCLS=(,A)
```

UCS

Optional for:
complete
I/O device

Not applicable for:
nucleus

The UCS macro instruction specifies the IBM standard character-set images for a 1403 printer with the universal character set (UCS) feature or for a 3211 printer. If this macro instruction is omitted, it is expected that the IBM or user character-set images will be included by a process other than system generation. For information on including your own character-set images, refer to *OS/VS Data Management for System Programmers.*

For an *I/O device generation* this macro instruction can be specified to add UCS support for the printer if support was not supplied in the last complete system generation. This macro instruction does not have to be respecified if it has already been specified in a previous generation.

```
UCS      [DEFAULT={ALL                   }]
                  {(image [,image] . . .)}
         IMAGE={ALL 1403              }
               {ALL 3211              }
               {ALL                   }
               {(image [,image] . . .)}
```

Parameter	Subparameter	Explanation
DEFAULT=		specifies the UCS images to be used as defaults when a job does not specify an image through its job control language statements.
	ALL	specifies that all the images specified in the IMAGE parameter are to be considered defaults.
	image	specifies the images in the IMAGE parameter that will be defaults. It is recommended that all UCS images that can produce valid results as defaults be specified.
IMAGE=		specifies the IBM standard character-set images to be included as one or more of the following values. These values may be listed in any order.
	ALL1403	specifies that all the 1403 UCS images are to be included.
	ALL3211	specifies that all the 3211 UCS images are to be included.
	ALL	specifies that both the 1403 and 3211 UCS images are to be included.

Parameter	Subparameter	Explanation
IMAGE= (Continued)	image	specifies particular UCS images to be included; the images are one or more of the values listed in Figure 12. These values may be listed in any order.

Value	IBM Standard Character-Set Image
1403 Printer	
AN	Arrangement A, standard EBCDIC character set, 48 graphics.
HN	Arrangement H, FORTRAN/COBOL EBCDIC character set, 48 graphics.
PCAN	Arrangement A, preferred character set, 48 graphics.
PCHN	Arrangement H, preferred character set, 48 graphics.
PN	PL/I character set, 60 graphics.
QNC	PL/I, commercially preferred character set, 60 graphics.
QN	PL/I, scientifically preferred character set, 60 graphics.
RN	FORTRAN/COBOL commercial, 52 graphics.
SN	Text printing, commercial, 84 graphics.
TN	Text printing, scientific, 120 graphics.
XN	High speed alphameric, 1403 Model 2, 40 graphics.
YN	High speed alphameric, 1403 Model N1, 42 graphics.
3211 Printer	
A11	Standard commercial character set, 48 graphics.
G11	ASCII character set, 63 graphics.
H11	Scientific character set, 48 graphics.
P11	PL/I character set, 60 graphics.
T11	Text printing, 120 graphics.

Figure 12. Standard character-set images.

Example: This macro specifies that the standard character set images AN, HN, PN, and QNC are to be included in the system. The images identified by the PN and AN identifications are to be the default values.

```
UCS    UCS    IMAGE=(AN,HN,PN,QNC),DEFAULT=(PN,AN)
```

UNITNAME

Required for:
complete
I/O device

Not applicable for:
nucleus

The UNITNAME macro instruction is used to name a group of I/O devices. A UNITNAME macro instruction is required for each named group of I/O devices in the new VS2 system, except for unique device types. All UNITNAME macro instructions having the same value in the NAME parameter must appear consecutively in the input stream.

This macro instruction must be used to assign certain names to groups of I/O devices for the IBM-supplied cataloged procedures in SYS1.PROCLIB and the installation verification procedure (IVP) in SYS1.SAMPLIB. The names required are:

SYSSQ	for magnetic tape and/or direct-access devices
SYSDA	for direct-access devices only.

A maximum of 100 uniquely named groups can be specified for a system. A maximum of 255 addresses can be in one group. If more addresses must be listed for a particular name, another UNITNAME macro instruction is coded using the same name. A maximum of 255 characters can be used in the operand of any UNITNAME macro instruction. The maximum number of addresses that may be specified is 1028 minus the number of names. The addresses used must also be specified in the IODEVICE macro instruction. For a list of I/O devices, see Appendix A.

```
UNITNAME        NAME=name
                UNIT=(address[,address] . . .)
```

Parameter	Subparameter	Explanation
NAME=	name	specifies the name to be given to a group of I/O devices. The name consists of from 1 to 8 characters. These characters can be alphameric, national (#, @, or $), or the two special characters slash (/) and hyphen (-).
UNIT=		specifies the addresses of a group of I/O devices that will be recognized by the name assigned.
	address	specifies the unit address of an I/O device to be included by the name assigned.
	(address, n)	specifies the lowest unit address of a group of sequential addresses being specified. The n is the number of sequential addresses being assigned. The two forms of the UNIT parameter may be mixed. If more than one value is expressed, the values must be enclosed in parentheses.

Parameter	Subparameter	Explanation
UNIT= (Continued)	(address, n) (Continued)	If the form (address, n) is used as the only subparameter of the macro instruction, double parentheses must be used. For example, UNIT=((180,4)) would create a group of four devices that have the addresses 180, 181, 182, and 183. The only combination of unlike device types permitted in a group is magnetic tape and direct-access devices.

Example: This macro specifies that TAPE is the symbolic name associated with devices at addresses 180, 181, 280, 281, and 282.

```
UNITNAM1    UNITNAME    NAME=TAPE,                               X
                  UNIT=(180,181,280,281,282)
```

Example: This macro specifies that DISK is the symbolic name associated with eight consecutive addresses, beginning at address 130.

```
UNITNAM2    UNITNAME    NAME=DISK,UNIT=((130,8))
```

Example: This example illustrates the use of the UNITNAME macro instruction to assign a specific unit name to an unsupported I/O device. The unsupported device is located at address 167 (specified as UNIT=DUMMY,ADDRESS=167 in an IODEVICE macro instruction).

```
UNITNAM3    UNITNAME    NAME=167,UNIT=167
```

SELECTING THE NEW SYSTEM DATA SETS

This chapter is divided into two sections. The first section tells how to allocate space to the system data sets and catalog them in the system catalog by using either the DATASET macro or the IEHPROGM utility program.

Figure 15 presents a summary of the required and optional system data sets. This figure may be helpful to refer to when you are specifying the new system data sets. This section also contains examples of coding the parameters for the DATASET macro and the control statements for executing IEHPROGM.

The information in this section about specifying DD statements and catalog statements applies only if you are using the IEHPROGM utility program for allocating space and cataloging. If you are using the DATASET macro, you only specify values for the SPACE and VOL parameters and the default values for the remaining parameters are supplied for you.

The second section contains detailed information about the required and optional system data sets for the new VS2 system. In this section, the system data sets are arranged in alphabetical order by the fully qualified data set name. The following information is supplied for each of the system data sets:

- Whether the data set is required or optional
- Whether the data set is sequential, partitioned, or direct
- What the data set contains
- What you must know when you are using the IEHPROGM utility program for cataloging and allocating space for the system data sets

Allocating Space to the New System Data Sets and Cataloging Them in the System Catalog

Before the components of the distribution libraries can be placed in the system data sets, space must be allocated to them, a system catalog must be built, and the data sets must be cataloged in the system catalog.

SYS1.SVCLIB must be allocated space entirely on the system residence volume. SYS1.SVCLIB may not occupy more than 1023 tracks on the system residence volume. The size of SYS1.LOGREC is determined during system generation and space is allocated during system generation. The maximum space that can be allocated to each of the remaining system data sets is one volume, except for SYS1.SYSJOBQE which may not occupy more than 1215 tracks on a 2314/2319 or more than 745 tracks on a 3330. Alternate track assignment is accepted for the system data sets. To achieve maximum efficiency in the new system, alternate tracks should not be used for SYS1.LPALIB, SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.SYSJOBQE.

Space for the new system data sets can be allocated and the system data sets can be cataloged either during system generation, by specifying the DATASET macro in the Stage I input deck, or before Stage II, by executing the IEHPROGM utility program.

Allocating Space and Cataloging Using the DATASET Macro

The DATASET macro instruction allocates space to the system data sets and catalogs them in the system catalog during the Stage II part of system generation. The parameters in the DATASET macro specify the system data set for which space is to be allocated, the volume on which it will reside, and the amount of space required on the specified volume. All of the system data sets are cataloged for you except SYSCTLG.

If you use the DATASET macro, one DATASET macro must be specified for each system data set that is to make up the new system to be generated. Refer to the description of the DATASET macro in "Specifying the New System Control Program."

Figure 13 illustrates the parameter specifications for the DATASET macro instruction. The default values for the system residence volume, specified in the GENERATE macro, are RESVOL=(SYSRES,3330).

```
SYSCTLG   DATASET  SYSCTLG,SPACE=(CYL,(15,5))
BRODCAST  DATASET  BRODCAST,SPACE=(CYL,(10)),           X
                VOL=(SYSLIB,3330)
CMDLIB    DATASET  CMDLIB,SPACE=(TRK,(20,1,1)),         X
                VOL=(SYSLIB,3330)
DSSVM     DATASET  DSSVM,SPACE=(2048,(600))
HELP      DATASET  HELP,SPACE=(CYL,(10,5,10)),          X
                VOL=(SYSLIB,3330)
IMAGELIB  DATASET  IMAGELIB,SPACE=(TRK,(20,,8))
SAMPLIB   DATASET  SAMPLIB,SPACE=(TRK,(100,20,2)),      X
                VOL=(SYSLIB,3330)
LINKLIB   DATASET  LINKLIB,SPACE=(CYL,(20,10,5))
LPALIB    DATASET  LPALIB,SPACE=(CYL,(30,3,150))
MACLIB    DATASET  MACLIB,SPACE=(CYL,(30,10,5)),        X
                VOL=(SYSLIB,3330)
NUCLEUS   DATASET  NUCLEUS,SPACE=(CYL,(2,,2))
PARMLIB   DATASET  PARMLIB,SPACE=(TRK,(50,,5))
DCMLIB    DATASET  DCMLIB,SPACE=(TRK,(50,,8)),          X
                VOL=(SYSLIB,3330)
PROCLIB   DATASET  PROCLIB,SPACE=(CYL,(20,5,10))
SVCLIB    DATASET  SVCLIB,SPACE=(CYL,(10,2,15))
SYSJOBQE  DATASET  SYSJOBQE,SPACE=(CYL,(100)),          X
                VOL=(111111,2305-2)
TELCMLIB  DATASET  TELCMLIB,SPACE=(CYL,(20,2,100)),     X
                VOL=(SYSLIB,3330)
UADS      DATASET  UADS,SPACE=(TRK,(20,2,5)),           X
                VOL=(SYSLIB,3330)
```

Figure 13. Allocating space for the new system data sets and cataloging them in the system catalog using the DATASET macro

Allocating Space and Cataloging Using IEHPROGM

Instead of using the DATASET macro, the IEHPROGM utility program can be used to allocate space to the system data sets and to catalog them in the system catalog.

If you use IEHPROGM instead of using DATASET macros, space must be allocated and the system data sets must be cataloged before you begin Stage II.

The following text describes the use of IEHPROGM for allocating space for the new system data sets and cataloging them. Detailed descriptions of the control cards and functions of IEHPROGM are in *OS/VS Utilities.*

The input deck for IEHPROGM must contain:

- A JOB statement with any parameters required by your installation.
- An EXEC statement with the PGM=IEHPROGM parameter.
- A DD statement for the message output data set (SYSPRINT).
- A DD statement for each of the new system data sets (except for the SYS1.LOGREC data set). These DD statements have the following format:

```
//ddname  DD  DSNAME=dsname,                        X
//            VOLUME=(,RETAIN,SER=serial),          X
//            UNIT=unit,LABEL=EXPDT=99350,          X
//            SPACE=(allocation),                   X
//            DISP=(,KEEP),DCB=(see Figure 15)
```

- A DD * statement (SYSIN).
- A CATLG statement for each new system data set to be cataloged. Each CATLG statement must have the following format:

```
CATLG  DSNAME=dsname,CVOL=unit=serial,              X
       VOL=unit=serial
```

The DD and CATLG statements for IEHPROGM are discussed in the following sections. For more information on the coding of parameters, refer to *OS/VS JCL Reference*.

Specifying the DD Statements for the IEHPROGM Input Deck

The DD statements in the input deck for IEHPROGM have the following parameters and subparameters:

Parameter	**Subparameter**	**Explanation**
DSNAME=	dsname	specifies the system data set to be initialized.
VOLUME=	(,RETAIN, SER=serial)	specifies the serial number of the direct-access volume on which the system data set is to reside.
UNIT=	unit	specifies the name of the direct-access device on which the volume containing the data set is to reside.
LABEL=	EXPDT=99350	specifies the expiration date for the system data set to prevent accidental deletion.
SPACE=	space	specifies the amount of storage to be allocated to the system data set. Information on space allocation is found in *OS/VS2 Storage Estimates*.

Parameter	Subparameter	Explanation
DISP=	(,KEEP)	specifies the disposition of the system data set. This parameter must be coded as shown.
DCB=	(DCB information)	specifies any DCB information that may be required by the system data sets.

Specifying the CATLG Statements for the IEHPROGM Input Deck

The parameters that are required for the catalog statements are described in the following text. Refer to "Selecting the New System Data Sets" for the system data sets that need to be cataloged.

Parameter	Subparameter	Explanation
DSNAME=	dsname	specifies the name of the system data set to be cataloged.
CVOL=	unit=serial	specifies the unit name and serial number of the new system residence volume. (The values must be the same as specified in the DD statement for SYS1.NUCLEUS.)
VOL=	unit=serial	specifies the unit name and serial number of the volume on which the system data set resides. These values must be the same as specified in the corresponding DD statement for the system data set.

Figure 14 illustrates the control statements for executing IEHPROGM. Space requirements for the system data sets are determined by several factors, especially the type of device used and the characteristics of the system control program to be generated. Exact storage requirements on various types of direct-access devices are in *OS/VS2 Storage Estimates.*

```
//NEWSYS    JOB
//          EXEC PGM=IEHPROGM
//SYSPRINT   DD  SYSOUT=A
//SYSCTLG    DD  DSNAME=SYSCTLG,DISP=(,KEEP),                  X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,            X
//           LABEL=EXPDT=99350,SPACE=(CYL,(15,5))
//CMDLIB     DD  DSNAME=SYS1.CMDLIB,DISP=(,KEEP),              X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,            X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),    X
//           SPACE=(TRK,(20,1,1))
//HELP       DD  DSNAME=SYS1.HELP,DISP=(,KEEP),                X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,            X
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960),            X
//           SPACE=(CYL,(10,5,10))
```

Figure 14 (Part 1 of 3). Allocating space for the new system data sets and cataloging them in the system catalog using IEHPROGM

```
//SAMPLIB    DD  DSNAME=SYS1.SAMPLIB,DISP=(,KEEP),               X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=F,LRECL=80,BLKSIZE=80),X
//           SPACE=(TRK,(100,20,2))
//LINKLIB    DD  DSNAME=SYS1.LINKLIB,DISP=(,KEEP),               X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),      X
//           SPACE=(CYL,(20,10,5))
//LPALIB     DD  DSNAME=SYS1.LPALIB,DISP=(,KEEP),                X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),      X
//           SPACE=(CYL,(30,3,150))
//MACLIB     DD  DSNAME=SYS1.MACLIB,DISP=(,KEEP),                X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=FB,BLKSIZE=12960,      X
//           LRECL=80),SPACE=(CYL,(30,10,5))                     X
//PROCLIB    DD  DSNAME=SYS1.PROCLIB,DISP=(,KEEP),               X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80),         X
//           SPACE=(CYL,(20,5,10))
//SVCLIB     DD  DSNAME=SYS1.SVCLIB,DISP=(,KEEP),                X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,           X
//           BLKSIZE=4096),SPACE=(CYL,(10,2,15))
//TELCMLIB   DD  DSNAME=SYS1.TELCMLIB,DISP=(,KEEP),              X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),      X
//           SPACE=(CYL,(20,2,100))
//UADS       DD  DSNAME=SYS1.UADS,DISP=(,KEEP),                  X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,              X
//           DCB=(RECFM=FB,DSORG=PO,BLKSIZE=800,LRECL=80),       X
//           SPACE=(TRK,(20,2,5))
//IMAGELIB   DD  DSNAME=SYS1.IMAGELIB,DISP=(,KEEP),              X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,SPACE=(TRK,(20,,8)),              X
//           DCB=(RECFM=U,BLKSIZE=13030)
//NUCLEUS    DD  DSNAME=SYS1.NUCLEUS,DISP=(,KEEP),               X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,SPACE=(CYL,(2,,2),,CONTIG)
//PARMLIB    DD  DSNAME=SYS1.PARMLIB,DISP=(,KEEP),               X
//           VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,              X
//           LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80),         X
//           SPACE=(TRK,(50,,5),,CONTIG)
//BRODCAST   DD  DSNAME=SYS1.BRODCAST,DISP=(,KEEP),              X
//           VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,              X
//           SPACE=(CYL,(10),,CONTIG)
```

Figure 14 (Part 2 of 3). Allocating space for the new system data sets and cataloging them in the system catalog using IEHPROGM

```
//SYSJOBQE    DD  DSNAME=SYS1.SYSJOBQE,DISP=(,KEEP),                     X
//            VOLUME=(,RETAIN,SER=111111),UNIT=2305-2,                   X
//            SPACE=(CYL,(100),,CONTIG)
//DCMLIB      DD  DSNAME=SYS1.DCMLIB,DISP=(,KEEP),                       X
//            VOLUME=(,RETAIN,SER=SYSLIB),UNIT=3330,                     X
//            SPACE=(TRK,(50,,8))
//DSSVM       DD  DSNAME=SYS1.DSSVM,DISP=(,KEEP),                        X
//            VOLUME=(,RETAIN,SER=SYSRES),UNIT=3330,                     X
//            SPACE=(2048,(600),,CONTIG)
//SYSIN  DD   *
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.CMDLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.HELP
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.SAMPLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.LINKLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.LPALIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.MACLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.PROCLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.SVCLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.TELCMLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.UADS
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.IMAGELIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.NUCLEUS
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.PARMLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.BRODCAST
    CATLG    CVOL=3330=SYSRES,VOL=2305-2=111111,DSNAME=SYS1.SYSJOBQE
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSLIB,DSNAME=SYS1.DCMLIB
    CATLG    CVOL=3330=SYSRES,VOL=3330=SYSRES,DSNAME=SYS1.DSSVM
/*
```

Figure 14 (Part 3 of 3). Allocating space for the new system data sets and cataloging them in the system catalog using IEHPROGM

System Data Set Summary

Figure 15 lists the required and optional system data sets. This figure may be helpful to refer to when you are allocating space to the new system data sets and cataloging them. The values given for the DCB subparameters for the data sets must be specified if you are using the IEHPROGM utility program. If you use the DATASET macro, these values are the default values that are used. These values must not be specified if you are using the DATASET macro.

System Data Sets	Type	System Residence	Secondary Volume Allocation	DCB Subparameters	Cataloged
Required System Data Sets					
SYSCTLG	sequential	required	yes	none	no
SYS1.DSSVM	sequential	optional	no	none[6]	yes
SYS1.LINKLIB[1]	PDS	optional	yes	RECFM=U, BLKSIZE=7294[2]	yes
SYS1.LPALIB	PDS	optional	yes	RECFM=U, BLKSIZE=7294[2]	yes
SYS1.LOGREC[4]	sequential	required	no	none	no
SYS1.MACLIB	PDS	optional	yes	RECFM=FB, LRECL=80, BLKSIZE=7280[3]	yes
SYS1.NUCLEUS	PDS	required	no	none	yes
SYS1.PARMLIB	PDS	optional	no	RECFM=F,BLKSIZE=80	yes
SYS1.PROCLIB	PDS	optional	yes	RECFM=FB,LRECL=80, BLKSIZE=7280[3]	yes
SYS1.SAMPLIB	PDS	optional	yes	RECFM=F, LRECL=80,BLKSIZE=80	optional
SYS1.SVCLIB[1]	PDS	required	yes	DSORG=POU,RECFM=U, BLKSIZE=4096	yes
SYS1.SYSJOBQE	sequential	optional	no	none	yes
Optional System Data Sets					
SYS1.BRODCAST[5]	direct	optional	no	none	yes
SYS1.CMDLIB[5]	PDS	optional	yes	RECFM=U, BLKSIZE=7294[2]	yes
SYS1.DCMLIB	PDS	optional	no	none	yes
SYS1.DUMP	sequential	optional	no	none	optional
SYS1.HELP[5]	PDS	optional	yes	RECFM=FB, LRECL=80, BLKSIZE=7280[3]	yes
SYS1.IMAGELIB	PDS	optional	no	RECFM=U, BLKSIZE=1024	yes
SYS1.MANX	sequential	optional	no	none	optional
SYS1.MANY	sequential	optional	no	none	optional
SYS1.SYSVLOGX	sequential	optional	no	RECFM=VB,BLKSIZE=7294[2]	yes
SYS1.SYSVLOGY	sequential	optional	no	RECFM=VB,BLKSIZE=7294[2]	yes
SYS1.TELCMLIB	PDS	optional	yes	RECFM=U, BLKSIZE=7294[2]	yes
SYS1.UADS[5]	PDS	optional	yes	DSORG=P0, LRECL=80, RECFM=FB, BLKSIZE=800	yes

[1] Space should be allocated in cylinders.

[2] Use BLKSIZE=7294 if the system data set resides on a 2314 or 2319. Use BLKSIZE=14136 for a 2305-1, BLKSIZE=14660 for a 2305-2, and BLKSIZE=13030 for a 3330.

[3] The value of BLKSIZE must be a multiple of 80 that is less than or equal to 7280 for a 2314 or 2319, 14080 for a 2305-1, 14640 for a 2305-2, or 12960 for a 3330.

[4] Space must not be allocated by the user.

[5] Required for a system with TSO.

[6] If this data set is specified using IEHPROGM, the following must be specified in the SPACE parameter: SPACE=(2048, (600),,CONTIG)

Note: The blocksize found in the DSCB for system data sets whose record format is undefined (RECFM=U) will be the maximum blocksize for the device being used. This is not necessarily the size of the current record.

Figure 15. Summary of the required and optional system data sets

SYSCTLG

Required
Sequential

Contents

This system data set contains pointers to all the cataloged data sets in the new VS2 system.

Creation

The initial entries are created by the system generation process.

Requirements for Specification

Location:

This data set must be on the system residence volume.

DD Statement:

The standard format is used.

The serial number of the new system residence volume must be specified for this data set.

Secondary space allocation can be specified.

CATLG Statement:

This statement must not be coded for this data set.

For further information on this data set, refer to *OS/VS2 Planning and Use Guide.*

SYS1.BRODCAST

Required for TSO systems
Direct

Contents

This system data set stores two types of TSO messages:

- Notices – messages available for all users of the system
- Mail – messages available for specific users of the system

To facilitate the access of each type of message, the system data set also contains a Notice Directory and a Mail Directory.

For additional information about the data set, refer to *OS/VS2 TSO Command Language Reference.*

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Space must be allocated.

Secondary space allocation cannot be specified.

DCB parameters cannot be specified.

CATLG Statement:

This data set should be cataloged in the system catalog (SYSCTLG).

Notes

Initialization:

After system generation, the first use of the data set causes it to be formatted and initialized. The data set will not be reformatted until another system generation.

This data set does not contain an expiration date.

SYS1.CMDLIB

Required for TSO systems
Partitioned

Contents

This system data set is a load library for command processor programs.

Requirements for specification

Location:

This system data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Secondary space allocations can be specified.

One of the following DCB subparameters must be specified:

RECFM=U, BLKSIZE=	14136	for a 2305-1
	14660	for a 2305-2
	7294	for a 2314
	13030	for a 3330

If you use the DATASET macro, these are the values that are used for the specified devices.

CATLG Statement:

This data set should be cataloged in the system catalog (SYSCTLG).

SYS1.DCMLIB

Required if programmed-function-key (PFK) command is specified in the SCHEDULR or SECONSLE macro instructions.
Partitioned

Contents

This system data set contains the definitions of programmed-function-keys assigned for operator command entry.

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Secondary space cannot be allocated.

CATLG Statement:

This data set should be cataloged in the system catalog (SYSCTLG).

SYS1.DSSVM

Required
Sequential

Contents

This system data set contains DSS language processing routines, a nucleus map area, a work space area, a nucleus swap area, and DSS change pages.

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

If space is allocated with the DATASET macro, the following values must be specified in the SPACE parameter:

```
SPACE=(2048,(600))
```

If IEHPROGM is used instead, the following values must be specified in the DD statement that identifies SYS1.DSSVM:

```
SPACE=(2048, (600), , CONTIG)
```

Secondary space cannot be allocated.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.DUMP

Optional
Sequential

Contents

This system data set contains system dumps that are used to record areas of virtual storage in case of system task failures.

For further information on this data set, refer to *OS/VS Debugging Guide.* For information on the AMDPRDMP service aid that is used to process the dumps, refer to *OS/VS Service Aids.*

Requirements for specification

Location:

This data set can be on either a direct-access volume or a magnetic tape volume. It can be on the system residence volume.

The volume that contains this data set can be on a:

2305-1	fixed-head storage facility
2305-2	fixed-head storage facility
2314	direct-access storage facility
2319	direct-access storage facility
2400	series magnetic tape unit
3400	series magnetic tape unit
3330	disk storage drive

DD Statement:

This statement is used only if this system data set is on a direct-access volume and you choose to allocate space. In this case, you must also write an end-of-file (EOF) record as the first record in the data set. If you do not allocate space for this data set and it is cataloged, the nucleus initialization program (NIP) allocates it for you and writes the EOF record when the system is loaded.

Secondary space cannot be allocated.

CATLG Statement:

Code this statement only if this data set is going to be on a direct-access volume.

Notes

If this data set is going to be on a magnetic tape volume, it must be specified at IPL. It cannot be cataloged during the preparation for system generation.

SYS1.HELP

Required if the TSO HELP command is to be used in a system with TSO
Partitioned

Contents

Each member of this system data set contains TSO HELP information regarding the syntax, operands, and functions for each TSO command.

Requirements for specification

Location:

This data set must be on a direct-access volume. It can be on the system residence volume.

DD Statement:

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

RECFM=FB, LRECL=80,
BLKSIZE= a multiple of 80, which is less than or equal to:

80	if unblocked
14080	for a 2305-1
14640	for a 2305-2
7280	for a 2314
12960	for a 3330

If you use the DATASET macro, these are the values that are used for the specified devices.

CATLG Statement:

This data set should be cataloged in the system catalog (SYSCTLG).

Notes

For further information on this data set, see *OS/VS2 TSO Command Language Reference.*

SYS1.IMAGELIB

Required if a 1403 printer with the universal character-set features (UNVCHSET) or a 3211 printer is in the system.
Partitioned

Contents

This data set contains the universal character-set (UCS) and forms control buffer (FCB) image modules.

For information about this system data set, refer to *OS/VS Data Management for System Programmers.*

Requirements for specification

Location:

This data set must be permanently mounted on a direct-access volume, which can be the system residence volume.

DD Statement:

Space must be allocated.

No secondary space allocation is allowed.

The following DCB subparameters must be specified:

RECFM=U, BLKSIZE=1024

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.LINKLIB

Required
Partitioned

Contents

This system data set contains programs and routines that are referred to by the XCTL, ATTACH, LINK, and LOAD macro instructions and nonresident VS2 system routines.

This system data set also contains an assembler-language processor, a linkage editor, the utility programs, and service aids.

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Space should be allocated in cylinders. For maximum efficiency, alternate tracks should not be used.

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

$$\text{RECFM=U, BLKSIZE=} \left\{\begin{array}{r} 14136 \\ 14660 \\ 7294 \\ 13030 \end{array}\right\} \begin{array}{l} \text{for a 2305-1} \\ \text{for a 2305-2} \\ \text{for a 2314} \\ \text{for a 3330} \end{array}$$

If you use the DATASET macro, these are the values that are used for the specified devices.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.LOGREC

Required
Sequential

Contents

This system data set is used to contain statistical data about machine failures (CPU failures, I/O device errors, and channel errors).

Requirements for specification

Location:

The system generation process allocates space for this data set on the system residence volume.

DD Statement:

This statement must not be coded for this data set.

CATLG Statement:

This statement must not be coded for this data set.

Notes

The size of this data set can be increased or decreased after system generation by use of the IFCDIPOO program. Information on this program is in *OS/VS SYS1.LOGREC Error Recording.* For information on space allocation for reinitializing this data set, refer to *OS/VS2 Storage Estimates.*

SYS1.LPALIB

Required
Partitioned

Contents

This system data set contains all of the modules that are loaded into the link pack area (LPA). This includes system routines, SVC routines, data management access methods, nonresident machine-check handler modules, and some TSO modules.

Information about this data set can be found in *OS/VS2 Planning and Use Guide.*

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

The standard format is used.

Secondary space allocation can be specified.

The following DCB subparameters must be specified:

RECFM=U, BLKSIZE=	14136	for a 2305-1
	14660	for a 2305-2
	7294	for a 2314
	13030	for a 3330

If you use the DATASET macro, these are the blocksizes that are used.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.MACLIB

Required
Partitioned

Contents

This system data set contains the macro definitions for supervisor and data management macro instructions.

Requirements for specification

Location:

This data set must be on a direct-access volume, which may be the system residence volume.

DD Statement:

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

RECFM=FB, LRECL=80,
BLKSIZE= a multiple of 80 which is less than or equal to:

80	if unblocked
14080	for a 2305-1
14640	for a 2305-2
7280	for a 2314
12960	for a 3330

If you use the DATASET macro, these are the values that are used for the specified devices.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.MANX, SYS1.MANY

Required if the recording function of the system management facility (SMF) feature is to be used.
Sequential

Contents

Both of these system data sets contain the data collected by the SMF routines. (For information on SMF, refer to *OS/VS System Management Facilities (SMF).*)

Requirements for specification

Location:

Both data sets should be defined on the same direct-access device type. They must reside on a permanently mounted volume, which can be the system residence volume.

DD Statement:

Secondary space allocations, if they are coded, are ignored.

Space allocation for both data sets should be the same.

CATLG Statement:

These data sets should be cataloged in the system catalog (SYSCTLG).

SYS1.NUCLEUS

Required
Partitioned

Contents

This system data set contains the resident portion (nucleus) of the control program. The member name is IEANUC01.

Requirements for specification

Location:

This data set must be on the system residence volume.

DD Statement:

The serial number of the new system residence volume must be specified for this data set.

Secondary space cannot be allocated.

CATLG Statement:

You must catalog this data set in the system catalog (SYSCTLG).

SYS1.PARMLIB

Required
Partitioned

Contents

This system data set contains the IEASYS00 and SMFDEFLT system parameter lists that are used by the nucleus initialization program (NIP), the PRESRES list that is used by the master scheduler, the system parameters that are defined during system generation, the LNKLST00 list that is used to concatenate data sets to SYS1.LINKLIB and the IEAFIXnn lists that are used to indicate the routines from SYS1.LINKLIB and SYS1.LPALIB that are to become part of the resident portion of the link pack area.

For additional information about this system data set, refer to *OS/VS2 Planning and Use Guide.*

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Space must be allocated for this data set.

Secondary space cannot be allocated.

The following DCB subparameters must be specified:

RECFM=F,BLKSIZE=80

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG) if it does not reside on the system residence volume. You should, however, always catalog this data set.

SYS1.PROCLIB

Required
Partitioned

Contents

This system data set contains the cataloged procedures used to perform certain system functions, such as reader, writer, and initiator procedures. The cataloged procedures can be for system tasks or processing program tasks invoked by the operator or the programmer.

For information on this system data set, refer to *OS/VS2 Planning and Use Guide.*

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Space must be allocated for this data set.

Secondary space allocations can be specified.

It is recommended that the following DCB subparameters be used:

```
RECFM=FB, LRECL=80,
  BLKSIZE=  a multiple of 80, which is less than, or equal to:
                 80  if unblocked
              14080  for a 2305-1
              14640  for a 2305-2
               7280  for a 2314
              12960  for a 3330
```

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG) if it does not reside on the system residence volume.

You should always catalog this data set.

SYS1.SAMPLIB

Required
Partitioned

Contents

This system data set contains the installation verification procedure (see "Testing the System Control Program"), the independent utilities, and the IPL text. It also contains SMF sample exit routines. (For information about punching the independent utilities, IPL text, and SMF exit routines, see "Preparing for System Generation.")

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=F,LRECL=80,BLKSIZE=80
```

If you use the DATASET macro, these are the values that are used for the specified devices.

CATLG Statement:

This data set need not be cataloged.

SYS1.SVCLIB

Required
Partitioned

Contents

This system data set contains system routines that are used by the nucleus initialization program (NIP) when the system is loaded.

Requirements for specification

Location:

This data set must be on the system residence volume.

DD Statement:

The serial number of the new system residence volume must be specified for this data set.

This data set cannot occupy more than 1023 tracks on the system residence volume.

Secondary space allocations can be specified.

Space should be allocated in cylinders. For maximum efficiency, alternate tracks should not be used.

The following DCB subparameters must be specified:

```
DSORG=POU, RECFM=U, BLKSIZE=4096
```

Catalog Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.SYSJOBQE

Required
Sequential

Contents

This system data set contains the input queue and output queue. It is used by the job scheduler as a storage and work area for information about the input and output streams. For information about this system data set, refer to *OS/VS2 Planning and Use Guide.*

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

For maximum efficiency, alternate tracks should not be used.

Contiguous space must be allocated for this data set. No more than 1215 tracks can be on a 2314 or 2319, and no more than 745 on a 3330. A full volume can be allocated on a 2305-1 or 2305-2. Secondary space allocations cannot be specified.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG) if it does not reside on the system residence volume.

You should always catalog this data set.

Notes

If this data set is date-protected, the operator will have to reply to an ENTER message if he formats the job queue.

This data set must be formatted when the new VS2 system is loaded for the first time. (For detailed information, see the *Operator's Library: OS/VS2 Reference.*)

This data set is not used during system generation. Therefore, it does not have to be cataloged and have space allocated for it until just before the IPL procedure. For convenience, it is recommended that you catalog it and allocate space for it with the other system data sets.

SYS1.SYSVLOGX, SYS1.SYSVLOGY

Required if a system log is to be used in the system
Sequential

Contents

These system data sets contain system log data that consists of:

- Write-to-log (WTL) messages
- Data entered by the operator
- The time it takes for a job and step to-run, and data from the JOB and EXEC statements of a job that has ended
- Write-to-operator (WTO) and write-to-operator reply (WTOR) messages
- Accepted replies to WTOR messages
- Operator commands

Requirements for specification

Location:

These data sets must be on direct-access volumes, which can be the system residence volume.

DD Statement:

Secondary space allocations cannot be specified.

One of the following DCB subparameters must be specified:

RECFM=VB, BLKSIZE=	14136	for a 2305-1
	14660	for a 2305-2
	7294	for a 2314
	13030	for a 3330

CATLG Statement:

This statement must be specified.

Notes

These data sets are not used during system generation. Therefore, they do not have to be cataloged and have space allocated for them until just before the new system is loaded. For convenience, it is recommended that you allocate space for these data sets and catalog them at the same time you do it for the other system data sets.

SYS1.TELCMLIB

Required if BTAM or TCAM will be in the system.
Partitioned

Contents

This system data set contains telecommunications subroutines in load module form.

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U, BLKSIZE=  ⎧14136⎫ for a 2305-1
                   ⎨14660⎬ for a 2305-2
                   ⎪ 7294⎪ for a 2314
                   ⎩13030⎭ for a 3330
```

If you use the DATASET macro instruction, these are the values that are used for the specified devices.

CATLG Statement:

This data set must be cataloged in the system catalog (SYSCTLG).

SYS1.UADS

Required for
systems with TSO
Partitioned

Contents

This system data set contains a list of authorized terminal users, and information about each of them.

Requirements for specification

Location:

This data set must be on a direct-access volume, which can be the system residence volume.

DD Statement:

The standard format is used.

Secondary space allocations can be specified.

The following DCB subparameters must be specified:

DSORG=PO, LRECL=80, BLKSIZE=800, RECFM=FB

CATLG Statement:

This data set should be cataloged.

Notes

This data set does not contain an expiration date.

For further information about this data set, refer to *OS/VS2 TSO Command Language Reference.*

PREPARING FOR SYSTEM GENERATION

This chapter describes the preparation for a system generation. The following topics are discussed:

- Initializing direct-access volumes
- Using the starter system as the generating system
- Using an existing VS2 system as the generating system
- Adding your own routines to the new VS2 system to be generated

Initializing Direct-Access Volumes

Before a VS2 System Control Program can be generated, the volumes that are to contain the starter system, the distribution libraries, and the new system must be initialized.

Volume initialization is the process of writing home addresses, a volume label, and a volume table of contents (VTOC) on a direct-access volume. In addition, the initial program load (IPL) program must be written on the direct-access volume that is to become the new system residence volume.

Volumes are initialized by either the IBCDASDI or the IEHDASDR utility programs. The IBCDASDI utility program is self-loading and its operation is independent of an operating system. The IEHDASDR utility program operates under the control of an operating system. The IEHDASDR utility program is on the starter system tape in SYS1.LINKLIB or in your existing system in SYS1.LINKLIB.

IBCDASDI, which is in the first file of the starter system tape, is used to initialize the volume that will contain the starter system. After the starter system has been restored to the initialized volume and the IPL procedure performed, the IEHDASDR utility program, under the control of the starter system, can be used to initialize the volumes that are to contain the distribution libraries and the VS2 System Control Program.

The IBCDASDI or IEHDASDR utility programs can also be used to include the IPL program on the volume that is to be the new system residence volume. The section "System Generation Using the Starter System" in this chapter describes the procedure used to obtain the IPL program cards for inclusion on the new system residence volume.

These utility programs and the control statements they require are described in *OS/VS Utilities,* their use in preparing for system generation is discussed in the "System Generation Using the Starter System" section of this chapter.

System Generation Using the Starter System

You must use the starter system provided by IBM for your first system generation (see "Requirements for Generating a New System Control Program"). After your first system generation, you can use your existing VS2 system for later system generations (see "System Generation Using an Existing VS2 System" in this chapter).

This section discusses what you must consider before using the starter system and the processing procedures that must be performed before the starter system and distribution libraries can be used for system generation.

Starter System Special Considerations

Special considerations that apply only to the starter system are described here.

I/O Devices

All devices to be used should be ready before IPL; any device not ready will automatically be taken off line. However, all teleprocessing control units such as the 2701, 2702, or 2703 should be disabled before IPL. If a device that was not ready at IPL is required during a job step, the operator should enter a VARY ONLINE command for that device.

The Dual-Density Feature and 9-Track Magnetic-Tape Drives

For 9-track magnetic-tape drives with or without the dual-density feature, the starter system will assume a default value for tape of 800 BPI, regardless of whether the tape volumes mounted on the drives have standard labels or no labels.

If you want to use 1600 BPI for 9-track tape drives with or without the dual-density feature, then you must specify the density in the DCB parameter of the DD statements for all the data sets that will reside on these drives. This applies if the tape volumes mounted on the drives have standard labels or no labels. The density specification for 1600 BPI will not be passed on from one step of a job to the next, and 1600 BPI cannot be used for SYSOUT.

System/370 Model 158

When using the System/370 Model 158, the 3215 mode must be used.

Processing the Starter System and Distribution Library Tapes

Certain procedures need to be followed before the starter system and distribution libraries can be used for system generation. These procedures consist of:

- Preparing for volume initialization
- Initializing the volumes that are to contain the starter system and distribution libraries
- Dumping the starter system from tape to a direct-access volume (STARTR)
- Starting the starter system
- Copying the distribution libraries from the tape volume (DLIBT1) to another direct-access volume (DLIB01)
- Punching the independent utilities and IPL text on DLIB01
- Listing the volume table of contents (VTOC) on STARTR and DLIB01
- Initializing the volume that is to contain the new VS2 system

This section describes a procedure that can be followed to prepare for system generation. In this example, the direct-access devices that are used for the volumes that are to contain the starter system, distribution libraries, and the new system are 3330s. This procedure can also be used with a 2314/2319. If a 2314/2319 is being used, the parameters that specify the device must be changed.

For illustrative purposes, the procedures assume the following set of devices and unit addresses:

I/O Device	Device Function	Address
3210 console	console keyboard	01F
3330 disk storage drive	starter system volume -- STARTR	150
3330 disk storage drive	distribution library volume -- DLIB01	151
3330 disk storage drive	new system volume -- SYSRES	152
2540 reader	system input	00C
2540 punch	punched output	00D
1403 printer	printed output	00E
2400 tape drive	tape drive for starter system tape and distribution library tape (DLIBT1)	182

Prepare to Initialize the Volume That Will Contain the Starter System

1. Mount the disk volumes onto which the contents of the starter system tape are to be dumped and the contents of the distribution library tape are to be copied.
2. Mount the starter system tape. The first file of the tape contains the IBCDASDI independent utility program. The second file contains the IBCDMPRS independent utility program.
3. Load the IBCDASDI independent utility program from the starter system tape by setting the load selector switches and pressing the console LOAD key. After the program has been loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.

Initialize the Volume That Will Contain the Starter System

4. Place the following IBCDASDI control statements in the input device:

```
INITVOL    JOB
           MSG     TODEV=1403,TOADDR=00E
           DADEF   TODEV=3330,TOADDR=150,VOLID=SCRATCH,FLAGTEST=NO
           VLD     NEWVOLID=STARTR,OWNERID=DEPT38
           VTOCD   STRTADR=05,EXTENT=10
           END
```

Note: The underlining in this and subsequent examples indicates that the device, unit address, or parameter value was arbitrarily chosen.

If the volume is being initialized for the first time, the parameter FLAGTEST=NO must be included in the DADEF statement.

5. Define the control statement input device by pressing the REQUEST key of the console keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the reply INPUT=2540, 00C where 2540 is the device type and 00C is the unit address. If this is not the device type and/or unit address of your input device, enter the correct information.
6. When the volume initialization is complete, the message END OF JOB is printed on the message output device, and the system enters the wait state.

Dump the Contents of the Starter System Tape to a Direct-Access Volume

7. Load the IBCDMPRS independent utility program from the second file of the starter system tape by pressing the console LOAD key again. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.

Place the following IBCDMPRS control statements in the input device to dump the contents of the starter system tape to disk:

```
DUMPTAPE JOB
         MSG      TODEV=1403,TOADDR=00E
         RESTORE FROMDEV=2400,FROMADDR=182,TODEV=3330,              X
               TOADDR=150,VOLID=STARTR
         END
```

8. Define the control statement input device by pressing the REQUEST key of the console keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the reply INPUT=2540,00C where 2540 is the device type and 00C is the unit address. If this is not the device type and/or unit address of your input device, enter the correct information.

9. When the contents of the starter system tape have been dumped to disk, the message END OF JOB is printed on the message output device, and the system enters the wait state.

Starting the Starter System

Making the starter system operative includes initial program loading (IPL), initializing the nucleus (NIP), readying the scheduler, and starting a reader, a writer, and an initiator. The following procedures can be used to make the starter system operative. Appendix F shows the entire process as it would appear on the console.

10. Set the load selector switches on the control panel to the unit address of the volume that contains the starter system. Then press the LOAD key.

11. Signal EOB to the SPECIFY SYSTEM PARAMETERS message.

12. Signal EOB to the request for a DUMP data set.

13. When the starter system is ready to begin working, it sends you the message: READY.

14. The date and time of day are printed along with the message REPLY WITH SET PARAMETERS OR U. Reply with the parameter for formatting the job queue data set SYS1.SYSJOBQE; R 00,‘Q=(,F)’ and signal EOB. If the date is incorrect, type R 00, ‘Q=(,F),DATE=yy.ddd,CLOCK=hh.mm.ss’, where yy is the year (00-99), ddd is the day (001-366), hh is the hours (00-59), mm is the minutes (00-59), and ss is the seconds (00-59), and set the TOD switch on the console.

15. Reply U to the message SPECIFY JOBQUE PARAMETERS and signal EOB.

16. Reply U to the message SPECIFY WITH SMF VALUES and signal EOB.

17. You must enter the commands to start a reader, a writer, and an initiator. Type the following commands:

```
MN JOBNAMES,T
START RDR,00C
START WTR,00E
START INIT,,,classname
```

If 00C and 00E are not the addresses of the reader and writer, enter the correct addresses.

Initialize the Volume That is to Contain the Distribution Libraries

18. The volume to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device and enter a START RDR,00C command. The volume will be initialized. After IEHDASDR has been executed, the initialized volume must be varied online.

```
// VARY 151,OFFLINE
//INIT         JOB      ACCT123,PROGRAMMER,MSGLEVEL=1
//DLIB01      EXEC      PGM=IEHDASDR,PARM='N=1'
//SYSPRINT      DD      SYSOUT=A
//SYSIN         DD      *
   ANALYZE      TODD=(151),VTOC=2,EXTENT=10,                X
                NEWVOLID=DLIB01,OWNERID=ID
/*
```

Copy the Distribution Libraries to a Direct-Access Volume

19. Mount the distribution library tape.

20. The IEBCOPY utility program is used to load the distribution libraries from tape to a direct-access volume. The IEBCOPY control cards are in the first file of the distribution library tape in the LOAD010 data set. When IEBCOPY is executed, it will allocate space to each of the distribution libraries and catalog them in the catalog of the generating system (in this case, the starter system).

    To access the IEBCOPY control cards for copying to either a 2314, 2319, or 3330, type the following:

    ```
    START RDR,182,LABEL=(,SL),DSNAME=LOAD010,VOL=SER=DLIBT1
    ```

    (If 182 is not the unit address of your tape drive, enter the correct address.)

    The control cards will be read in from tape and the reader will end and close.

    The IEBCOPY utility program will be executed. Completion of the preceding steps provides operable direct-access volumes with backup tapes.

Punch the Utility Programs and IPL Text

21. *This step is not necessary but may be performed at this time if you want the independent utilities and IPL text in card form.* The independent utilities operate outside the control of the operating system and are loaded as card decks or card images on tape. The independent utilities and IPL text, however, must be of the same release level as the system you are generating. These programs are distributed in the SYS1.ASAMPLIB distribution library. Once these programs are punched you need only include them and they are ready for use.

    Use the IEBPTPCH utility program to punch the IBCDMPRS, IBCDASDI, and ICAPRTBL independent utility programs and the IPL text (IEAIPL00). Place the following IEBPTPCH control deck in the input device and enter a START RDR,00C command.

```
//JOB1        JOB    ACCT123,PROGRAMMER,MSGLEVEL=1
//            EXEC   PGM=IEBPTPCH
//SYSUT1        DD   DSNAME=SYS1.ASAMPLIB,                X
//                   DISP=(OLD,KEEP)
//SYSUT2        DD   UNIT=00D
//SYSPRINT      DD   SYSOUT=A
//SYSIN         DD   *
      PUNCH          TYPORG=PO,MAXNAME=4
      MEMBER         NAME=IBCDMPRS
      MEMBER         NAME=IBCDASDI
      MEMBER         NAME=ICAPRTBL
      MEMBER         NAME=IEAIPL00
/*
```

A MEMBER card can be added to the above control statements for the sample exit routines that are contained in SYS1.ASAMPLIB for use if the System Management Facility (SMF) is to be used in the new VS2 system. By listing the cards that have been punched out from SYS1.ASAMPLIB, sample coding is provided for you to use as a reference while you are writing your own accounting routine. (For information on SMF, refer to *OS/VS System Management Facilities (SMF).* The SMF routines that are contained in SYS1.ASAMPLIB are:

SMFE15, SMFE35, SMFSORT, SMFEXITS, SMFFRMT, and TESTEXIT.

If you are including these sample programs to be punched out, be sure to adjust the MAXNAME field in the PUNCH control statement to show the revised number of member cards in the deck.

List the Volume Table of Contents (VTOC)

22. Before Stage I, you must allocate space to the three utility data sets that are used during Stage I and the OBJPDS utility data set that is used during Stage II by the assembler. If you choose to allocate this space on either the 3330 starter system volume or the 3330 distribution library volume, you may want to list the VTOCs of the two volumes at this point to determine if enough space is available. (If you are using a 2314/2319, you will need to allocate this space on a utility volume.)

Place the following IEHLIST control deck in the input device and enter a START RDR, 00C command.

If JOB2 is run immediately after JOB1 (in step 21), eliminate the JOB2 card.

```
//JOB2            JOB   ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1          EXEC   PGM=IEHLIST
//SYSPRINT         DD   SYSOUT=A
//SYSRES           DD   UNIT=3330,DISP=OLD,               X
//                      VOLUME=SER=STARTR
//DLIBTAPE         DD   UNIT=3330,DISP=OLD,               X
//                      VOLUME=SER=DLIB01
//SYSIN            DD   *
      LISTVTOC          VOL=3330=STARTR,FORMAT
      LISTVTOC          VOL=3330=DLIB01,FORMAT
/*
```

Initialize the Volume That Will Contain the New VS2 System

23. Mount the disk volume that is to contain the VS2 system that is to be generated.
24. The volume to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device and enter a START RDR,00C command. The volume will be initialized and the IPL text will be written on the volume that is to be the new system residence volume. After IEHDASDR has been executed, the initialized volume must be varied online.

```
// VARY 152,OFFLINE
//INIT          JOB    ACCT123,PROGRAMMER,MSGLEVEL=1
//SYSRES       EXEC    PGM=IEHDASDR
//SYSPRINT       DD    SYSOUT=A
//ASAMPLIB       DD    DISP=OLD,                             X
//                     DSNAME=SYS1.ASAMPLIB(IEAIPL00)
//SYSIN          DD    *
     ANALYZE     TODD=(152),VTOC=2,EXTENT=10,                X
                 NEWVOLID=SYSRES,IPLDD=ASAMPLIB,             X
                 OWNERID=ID
/*
```

After you have selected and specified the system generation macro instructions, selected and specified the system data sets, and specified the control statements for initializing the required volumes, dumping the contents of the starter system tape to a direct-access volume, and copying the distribution library tape to a direct-access volume, you are ready to specify the job control language required to execute Stage I. Refer to the "Stage I Input" section of "System Control Program Installation" for this information. After you have specified the Stage I job control language, you are ready to begin the actual processing. Refer to the beginning of this section ("System Generation Using the Starter System") and follow the procedures for executing the utility programs using the control statements you coded earlier. Then, refer to "System Control Program Installation" for the information for Stage I execution. If you plan to add your own routines to the system to be generated, refer to "Adding User-Written Routines to the System Control Program," in this chapter before executing Stage I.

System Generation Using an Existing VS2 System

After your first system generation using the starter system, your existing VS2 system can be used as the generating system for system generation. When you use an existing VS2 system as the generating system you can either:

- Perform the system generation as the only job, or
- Perform the system generation as one of several jobs in the job stream

When you use the first method, you can modify the nucleus of the generating system (nucleus generation). When you use the second method, you cannot modify the generating system.

Preparation for System Generation Using an Existing VS2 System as the Generating System

This section lists the procedures for using an existing VS2 system as the generating system in performing a system generation.

Initialize the Volume That Will Contain the Distribution Libraries

1. Mount the volume onto which the contents of the distribution library tape will be copied.
2. The volume to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device and enter a START RDR,00C command. After IEHDASDR has been executed, the initialized volume must be varied online.

```
// VARY 151,OFFLINE
//DLIB01     JOB    ACCT123,PROGRAMMER,MSGLEVEL=1
//INIT01     EXEC   PGM=IEHDASDR,PARM='N=1'
//SYSPRINT    DD    SYSOUT=A
//SYSIN       DD    *
    ANALYZE   TODD=(151),VTOC=00004,EXTENT=00010,    X
              FLAGTEST=NO,NEWVOLID=DLIB01
/*
```

Initialize the Volume That Will Contain the New VS2 System

3. Mount the volume that is to contain the new VS2 system.
4. The volume to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device and enter a START RDR,00C command. After IEHDASDR has been executed, the initialized volume must be varied online.

```
// VARY 152,OFFLINE
//INIT         JOB    ACCT123,PROGRAMMER,MSGLEVEL=1
//SYSRES      EXEC    PGM=IEHDASDR
//SYSPRINT     DD     SYSOUT=A
//ASAMPLIB     DD     DISP=OLD,                          X
//                    DSNAME=SYS1.ASAMPLIB(IEAIPL00)
//SYSIN        DD     *
    ANALYZE    TODD=(152),VTOC=2,EXTENT=10,              X
               NEWVOLID=SYSRES,IPLDD=ASAMPLIB,           X
               OWNERID=ID
/*
```

Copy the Distribution Libraries to a Direct-Access Volume

5. The IEBCOPY utility program is used to load the distribution libraries from tape to a direct-access volume. The IEBCOPY control statements are in the first file of the distribution library tape in the LOAD010 data set. When IEBCOPY is executed, it will allocate space to each of the distribution libraries and catalog them in the catalog of the generating system (in this case, an existing VS2 system).

To access the IEBCOPY control statements for copying to either a 2314/2319 or 3330, type the following:

```
START RDR,182,LABEL=(,SL),DSNAME=LOAD010,VOL=SER=DLIBT1
```

(If 182 is not the unit address of your tape drive, enter the correct address.)

The control statements will be read in from tape and the reader will end and close.

The IEBCOPY utility program will be executed. Completion of the preceding steps provides operable direct-access volumes with a backup tape of the distribution libraries.

After you have selected and specified the system generation macro instructions, selected and specified the system data sets, and specified the control statements for initializing the required volumes, dumping the contents of the starter system tape to a direct-access volume and copying the distribution library tape to a direct-access volume, you are ready to specify the job control language required to execute Stage I. Refer to the "Stage I Input" section of "System Control Program Installation" for this information. After you have specified the Stage I job control language, you are ready to begin the actual processing. Refer to the beginning of this section ("System Generation Using an Existing VS2 System") and follow the procedures for executing the utility programs using the control statements you coded earlier. Then, refer to "System Control Program Installation" for the information for Stage I execution. If you plan to add your own routines to the system to be generated, refer to "Adding User-Written Routines to the System Control Program" in this chapter before you begin Stage I.

Preparation for System Generation on the Generating System

A nucleus generation can be performed against the generating system. The following procedures can be used to add a new nucleus to the generating system.

- Modify the original Stage I macro instruction deck to include the new macro specifications. The remaining macros need not be removed as they will be ignored (see "Examples of System Generation").
- Initialize the volume onto which the distribution library tape is to be copied. (See "Processing the Starter System and Distribution Library Tapes" in this chapter.)
- Copy the distribution library tape to a direct-access volume. (See "Processing the Starter System and Distribution Library Tapes" in this chapter.)
- Specify the job control language to run Stage I and Stage II. (See "System Control Program Installation".)
- Run Stage I and Stage II. (Refer to "System Control Program Installation" for the procedures to execute Stage I and Stage II.)

Adding User-written Routines to the System Control Program

During system generation, you can add your own routines to the new nucleus, SYS1.LINKIB, and SYS1.LPALIB.

Routines that are to be added to SYS1.NUCLEUS, SYS1.LINKLIB, or SYS1.LPALIB must be load modules residing in a cataloged partitioned data set in the generating system. That is, each routine must be compiled, link-edited, and placed in a cataloged partitioned data set. (Each load module must be a member of the data set.) The name of the partitioned data set must be of the form SYS1.name. The name must consist of from 1 to 8 alphameric characters, the first character being alphabetic. The name of the partitioned data set and of the members that contain your routines are specified with system generation macro instructions.

The RESMODS macro instruction specifies load modules to be added to the nucleus during the nucleus link-edit step in Stage II (see "System Control Program Installation").

The LINKLIB macro instruction specifies load modules, such as accounting routines, to be added to SYS1.LINKLIB during an IEBCOPY step in Stage II (see "System Control Program Installation"). These load modules become members of SYS1.LINKLIB.

The LPALIB macro instruction specifies load modules, such as type 3 or type 4 SVC routines and nonstandard label routines, to be added to SYS1.LPALIB during an IEBCOPY step in Stage II (see "System Control Program Installation").

If SVCs are added to the nucleus or SYS1.LPALIB, the SVCTABLE macro instruction must also be used. This macro instruction adds to the SVC table an entry that specifies the characteristics of each SVC added.

Figure 16 is an example of adding a user-written routine to the new system control program. In the example, IGC255, a type 1 user-written SVC routine is to be included in SYS1.NUCLEUS. In the first step, the CSECT is assembled and placed in a temporary data set, &&LOADSET. In the second step the CSECT is link-edited and becomes member IGC255 of SYS1.USERLIB, a partitioned data set that is cataloged in the generating system. During system generation the SVC will be included in the new nucleus by the following system generation macro instructions:

```
RESMODS     PDS=SYS1.USERLIB,MEMBERS=(IGC255)
SVCTABLE    SVC-255-D1
```

The RESMODS, LINKLIB, LPALIB, and SVCTABLE macro instructions are described in the section "Specifying the New System Control Program." For a description of the control statements required by the assembler and the linkage editor, refer to *OS/VS Assembler Programmer's Guide.*

```
//USERSVC      JOB    MSGLEVEL=1
//STEP1       EXEC    PGM=ASMBLR,PARM='NODECK,OBJECT'
//SYSPRINT      DD    SYSOUT=A
//SYSUT1        DD    UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSUT2        DD    UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSUT3        DD    UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSLIB        DD    DSN=SYS1.MACLIB,DISP=SHR
//SYSGO         DD    DSN=&&LOADSET,UNIT=SYSDA,DISP=(,PASS),        X
//                    SPACE=(CYL,(5,1,1)),DCB=BLKSIZE=400
//SYSIN         DD    *
IGC255          CSECT
                ENTRY   IGC255
            SOURCE STATEMENTS
                  .
                  .
                  .
                END
/*
```

Figure 16 (Part 1 of 2). Adding a user-written routine to the system

```
//STEP2       EXEC    PGM=LINKS,PARM='LIST,NCAL,XREF,RENT'
//SYSPRINT     DD     SYSOUT=A
//SYSUT1       DD     SPACE=(1024,(200,20)),UNIT=SYSDA
//SYSLMOD      DD     DSN=SYS1.USERLIB(IGC255),DISP=(,CATLG),             X
//                    VOL=SER=PACK01,SPACE=(CYL,(2,1,10)),                X
//                    DCB=(RECFM=U,BLKSIZE=7294),UNIT=SYSDA
//SYSLIN       DD     DSN=&&LOADSET,DISP=(OLD,DELETE)
/*
```

Figure 16 (Part 2 of 2). Adding a user-written routine to the system

Figure 17 is another example of adding a user-written routine to the system control program. In this example, a CSECT is to be added to SYS1.NUCLEUS (member name IEANUC01). The CSECT consists of a series of constants describing the nucleus to be generated. This information will appear in SYSABEND, virtual-image, and stand-alone dumps as an additional means of identifying the nucleus of the System Control Program.

During the first step, the CSECT is assembled and placed in a temporary data set (DSNAME=&LOADSET). The CSECT is link-edited in the second step and the resulting module becomes member IEAXYZ1 of the SYS1.USER data set. SYS1.USER is a partitioned data set residing on volume PACK01 and cataloged in the generating system. During system generation, the CSECT will be included in the nucleus by the following command:

```
RESMODS   PDS=SYS1.USER,MEMBERS=(IEAXYZ1)
```

```
//USER      JOB MSGLEVEL=1
//STEP1    EXEC PGM=ASMBLR,PARM='NODECK,OBJECT'
//SYSLIB    DD  DSNAME=SYS1.MACLIB,DISP=OLD
//SYSUT1    DD  UNIT=SYSSQ,SPACE=(1700,(400,50))
//SYSUT2    DD  UNIT=SYSSQ,SPACE=(1700,(400,50))
//SYSUT3    DD  UNIT=(SYSSQ,SEP=(SYSUT1,SYSUT2,SYSLIB)),                  X
//              SPACE=(1700,(400,50))
//SYSPRINT DD   SYSOUT=A
//SYSGO     DD  DSNAME=&LOADSET,UNIT=SYSSQ,SPACE=(80,(200,50)),           X
//              DISP=(MOD,PASS)
//SYSIN     DD  *
IEAXYZ1    CSECT
           DC   C'XXXXXXXXXX-NUCLEUS ID CSECT-XXXXXXXXXX'
           DC   C'SYSTEM CONTROL PROGRAM GENERATED 7/1/73'
           DC   C'OWNER--DEPT. D58'
           DC   C'NUCLEUS--01'
           DC   C'XXXXXXXXXX-END ID CSECT-XXXXXXXXXX'
           END
/*
```

Figure 17 (Part 1 of 2). Preparing a user-written nucleus-identification load module

```
//STEP2    EXEC PGM=LINKS,PARM=(XREF,LIST,NCAL)
//SYSLIN    DD  DSNAME=&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD   DD  DSNAME=SYS1.USER(IEAXYZ1),UNIT=2314,DISP=(,CATLG), X
//              VOLUME=(,RETAIN,SER=PACK01),SPACE=(1024,(50,20,5))
//SYSUT1    DD  UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),                 X
//              SPACE=(1024,(50,20,5))
//SYSPRINT DD   SYSOUT=A
/*
```

Figure 17 (Part 2 of 2). Preparing a user-written nucleus-identification load module

SYSTEM CONTROL PROGRAM INSTALLATION

The first section of this chapter presents an explanation of Stage I and discusses Stage I input, Stage I processing, and the job stream that is produced. The second section presents an explanation of Stage II and discusses Stage II input, job stream processing, and Stage II output. Also included are procedures that you may need to perform after system generation.

Stage I: Producing the Job Stream

In Stage I, the macro instructions are assembled and analyzed for errors. If errors are found, error messages are written and the job stream is not produced. If no errors are found, a job stream consisting of job control language and control statements is produced. During Stage II these statements are used to select and process components from the distribution libraries and user-written components to form the new VS2 System Control Program.

It is possible to produce a job stream even though there are errors in the Stage I macros. Refer to Appendix E for the macro that can be used to override any errors that may be found during execution of Stage I.

Stage I Input

The input deck required for Stage I consists of job control language statements and system generation macro instructions. The sequence of the cards in the deck and the job control language statements are shown in Figure 18.

The input deck for Stage I must contain the following if the system generation program is executing as an independent job:

- A JOB statement with any parameters required by your installation. If the system generation process immediately follows the space allocation and cataloging of system data sets using IEHPROGM, this card is not to be specified.
- An EXEC statement with PGM=ASMBLR.
- A SYSLIB DD statement that allocates the SYS1.AGENLIB macro library to this job step.
- Three DD statements names SYSUT1, SYSUT2, and SYSUT3 that are used to allocate space to the three utility data sets required for Stage I. (Refer to Figure 18 for space allocation.)
- A SYSPUNCH DD statement defining the data set that is to contain the job stream produced during Stage I.
- A SYSPRINT DD statement defining the output classname.
- A SYSIN DD * statement
- The system generation macro instructions
- An END statement
- A /* card

```
//USER       JOB   MSGLEVEL=1
//          EXEC   PGM=ASMBLR
//SYSLIB      DD   DSNAME=SYS1.AGENLIB,DISP=SHR
//SYSUT1      DD   UNIT=unit address,SPACE=(space)
//SYSUT2      DD   UNIT=unit address,SPACE=(space)
//SYSUT3      DD   UNIT=unit address,SPACE=(space)
//SYSPUNCH    DD   UNIT=unit address,LABEL=(,NL)
//SYSPRINT    DD   SYSOUT=A
//SYSIN       DD   *
       system generation macro instructions
             END
/*
```

Figure 18. Input deck for Stage I

Figure 19 shows the values to be given to the SPACE parameter of each of the DD statements that define the utility data sets according to the type of direct-access volume on which they may reside. To determine if there is enough space available on the direct-access volume, list its volume table of contents (VTOC) before Stage I using the IEHLIST utility program. (See "Preparing for System Generation.")

	DD Statement			
Device Type	**SYSUT1**	**SYSUT2**	**SYSUT3**	**OBJPDS[1]**
2305 or 3330	3,2	5,2	8,2	5,1,12
2314 or 2319	15,2	10,2	16,2	7,1,12

[1] The OBJPDS utility data set is used during Stage II.

Figure 19. Space allocation (in cylinders) for the utility data sets

Stage I Execution

The system generation macro instructions are executed as one assembler job. During execution of the Stage I input, the macro instructions are assembled and analyzed for valid parameters and dependencies upon other macro instructions.

For the macro instructions to be executed, the volumes containing the starter system and distribution libraries must be mounted.

Stage I Output

If invalid macro instructions are found during Stage I execution, error messages are printed (see Appendix C) and the job stream is not produced. (For information on restarting Stage I, refer to "Restart Procedures.") If there are no errors, the job stream is produced as punched cards, as card images on tape or in a data set on disk according to what you specified in your Stage I SYSPUNCH DD statements. In addition to the job stream, Stage I also produces a documentation listing. This is a printout of the expansion of all the macro instructions that you specified, including the PUNCH statements that comprise the input to Stage II.

When you have completed State I and produced a job stream, you are ready to begin Stage II.

Stage II: Processing the Job Stream

In Stage II, the job stream produced during Stage I is processed as follows:

- Selected modules are assembled.
- The linkage-editor combines the modules that are to be included in the resident portion of the control program (the nucleus).
- The linkage-editor processes selected modules to construct members of the new VS2 system data sets.
- Utility programs complete the construction and initialization of the system data sets that make up the new VS2 system.

The completion of Stage II results in a System Control Program that reflects what was specified in the system generation macro instructions.

Stage II Input

The input for Stage II consists of the control statements required to catalog and allocate space for the utility data set, OBJPDS, and the job stream that is the output from Stage I.

Allocating Space for and Cataloging the OBJPDS Utility Data Set

You can catalog and allocate space for the OBJPDS utility data set at the same time you catalog and allocate space for the system data sets if you are using the IEHPROGM utility program (see "Selecting the New System Data Sets"). If you are using the DATASET macro instruction to catalog and allocate space for the system data sets, you will need to catalog and allocate space for OBJPDS before Stage II is executed. Figure 20 shows the job control statements used to catalog and allocate space for the OBJPDS utility data set. Figure 19 gives the space requirements for the utility data set. If Stage II immediately follows Stage I, the JOB card is not specified.

```
//OBJPDS      JOB  MSGLEVEL=1
//           EXEC  PGM=IEHPROGM
//OBJPDS       DD  DSNAME=SYS1.OBJPDS,SPACE=(space), X
//                 DISP=(,CATLG),UNIT=(unit address),X
//                 VOLUME=(,RETAIN,SER=serial number)
//SYSPRINT     DD  DUMMY
//SYSIN        DD  DUMMY
/*
```

Figure 20. Job control statements for allocating space to and cataloging the OBJPDS utility data set

The Job Stream

If no error messages are printed during Stage I, the job stream is produced on the SYSPUNCH data set. The job stream is the input to Stage II.

The job stream contains several JOB statements, each followed by one or more EXEC statements. Each EXEC statement is followed by its associated DD statements and other data required to execute the assembler, linkage editor, and utility programs during Stage II.

The Job Statement. The format of the Stage II JOB statements that are produced by the system generation process is:

```
//SYSGENnn JOB1,"SYSTEM GENERATION",CLASS=&SGCTRLC(42),          X
         MSGCLASS=&SGCTRLC(41),MSGLEVEL=1
```

The nn represents sequential identification numbers supplied by the system generation process. If a value other than 'SYS1' is specified in the INDEX parameter of the GENERATE macro instruction, that value will be substituted for 'SYSGEN' in the jobname field. &SGCTRLC(42) will be substituted with the appropriate job class and &SGCTRLC(41) will be substituted with the appropriate output class.

You may choose to use your own JOB statements by changing the job name or the accounting information on the JOB statement before Stage I. You can do this by using the IEBUPDTE utility program to modify the JOB statement information that is contained in the SYS1.AGENLIB distribution library. (Refer to *OS/VS Utilities* for information on using the IEBUPDTE utility program.)

When modifying the JOB statement, the name parameter must be *jobname*&SGCTRLA(3). Jobname consists of 1 through 6 alphameric or national (#,@, or $) characters, the first character being alphabetic or national. &SGCTRLA(3) is a counter that will be incremented by the system and which produces unique jobnames. (For information on coding the JOB statement, refer to *OS/VS JCL Reference*.)

Figure 21 is an example of creating a unique JOB statement. In this example, the jobname will be 'SYS#10nn.' The input class for the job stream will be 'C', and the output class for the system messages will be 'C'.

```
Col
1 3     9
//JOBA     JOB  (78700,J22),NAME,MSGLEVEL=(1,1)
//UPDTE   EXEC  PGM=IEBUPDTE,PARM=MOD
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSN=SYS1.AGENLIB,DISP=OLD
//SYSUT2    DD  DSN=SYS1.AGENLIB,DISP=OLD
//SYSIN     DD  *
./        REPL  NAME=JOBCARD,LIST=ALL
./      NUMBER  NEW1=10000,INCR=10000
         MACRO
       JOBCARD
          COPY  SGGBLPAK
  PUNCH '//SYS#10&SGCTRLA(3) JOB (78700,J22),              X
                NAME,MSGLEVEL=1,CLASS=C,MSGCLASS=C
          MEND
./       ENDUP
/*
```

Figure 21. Creating unique job statements for the Stage II input deck

The EXEC Statement. The format of the Stage II EXEC statements that are produced by the system generation process is:

//SGxx EXEC PGM=program[,COND=condition] [,PARM=value(s)]

or

//SGxx EXEC procname[,PARM=value(s)]

where

SGxx is the step name. The xx represents a sequential identification number supplied by the system generation process.

PGM indicates the name of the program being executed. The names are IFCDIP00, IEHPROGM, IEHLIST, IEBUPDTE, IFCERCP0 and IQADCM00. The programs are executed in the order in which they are discussed in "Stage II Execution." Some of the programs are executed several times.

procname is the name of the procedure being executed. The names are ASMS and LINKS. The procedures are executed in the order in which they are discussed in "Stage II Execution." Some of the procedures are executed several times.

COND specifies the conditions for executing or bypassing the job step according to the success or failure of the previous step in the job. If the previous step was unsuccessful, the remaining steps are bypassed and the next job is initiated.

PARM specifies any parameter information that may be required for that job step.

Stage II Execution

This section discusses the programs that are executed and their sequence of execution in creating a VS2 System Control Program. Also in this section is a list of the components of the distribution libraries, the names of the distribution libraries, and the system data sets where the components are placed during Stage II execution.

During Stage II execution, the job stream (job control language and utility control statements) assembles, link-edits, and copies specific modules from the distribution libraries and user libraries into system data sets that were allocated on the new system volume(s) to form the new VS2 system.

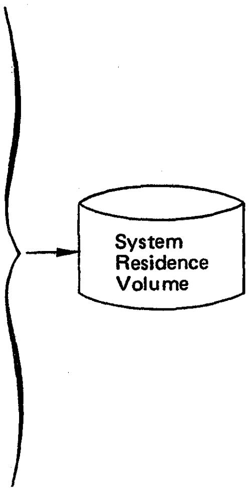
Execution of the job stream occurs in thirteen jobs for a complete system generation. For a nucleus or I/O device generation, job stream execution occurs in less than thirteen jobs. For those jobs that are executed, the sequence of execution is the same as in a complete system generation. Also, the number of job steps in a job may vary according to what is specified ın the macro instructions.

The following example illustrates the execution of the job stream for a complete system generation.

Job 1. If DATASET macros were specified, the IEHPROGM utility program catalogs and allocates space for the system data sets. If the system data sets were cataloged and space was allocated before Stage II this job step does not occur.

```
//SYSGEN1  JOB
//SG1      EXEC     IEHPROGM
```

```
SYSCTLG
SYS1.DCMLIB
SYS1.DSSVM
SYS1.IMAGELIB
SYS1.SAMPLIB
SYS1.LINKLIB
SYS1.MACLIB
SYS1.NUCLEUS
SYS1.PARMLIB
SYS1.PROCLIB
SYS1.SVCLIB
SYS1.LPALIB
SYS1.SYSJOBQE
SYS1.TELCMLIB
```

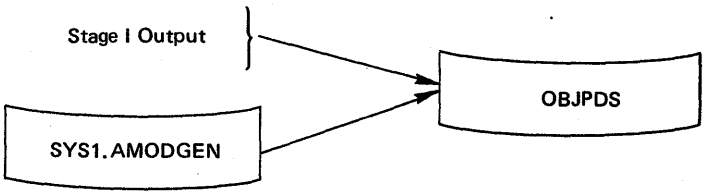


If this was generation of a TSO system, the following data sets would also be cataloged and have space allocated for them:

```
SYS1.CMDLIB
SYS1.BRODCAST
SYS1.HELP
SYS1.UADS
```

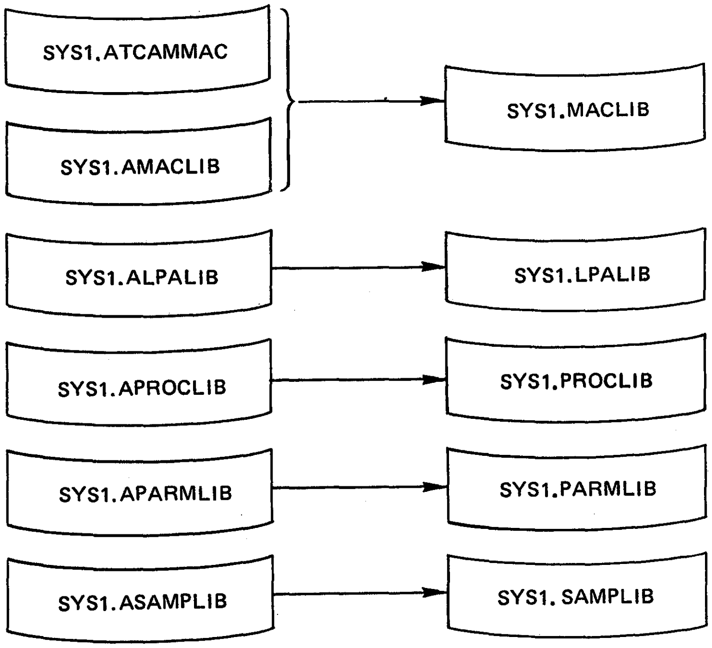
Job 2. Modules are assembled, based on parameter values specified in the macro instructions and from SYS1.AMODGEN, and placed in the object-module utility data set (OBJPDS).

```
//SYSGEN2  JOB
//SG1      EXEC  ASMS
             .
             .
             .
//SG50     EXEC  ASMS
```

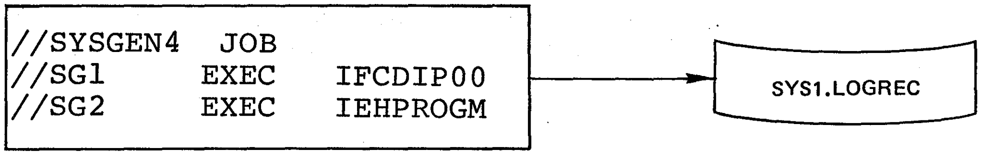


Job 3. The IEBCOPY utility program copies modules from SYS1.AMACLIB and SYS1.ATCAMMAC (if TCAM was specified) to SYS1.MACLIB in the new system. It also copies SYS1.ALPALIB, SYS1.APARMLIB, SYS1.APROCLIB, and SYS1.ASAMPLIB to SYS1.LPALIB, SYS1.PARMLIB, SYS1.PROCLIB, and SYS1.SAMPLIB, respectively, in the new system. If this was the generation of a TSO system, SYS1.TSOMAC, SYS1.AHELP, and SYS1.AUADS would be copied to SYS1.MACLIB, SYS1.HELP, and SYS1.UADS, respectively.

```
//SYSGEN3    JOB
//SG1       EXEC   IEBCOPY
```

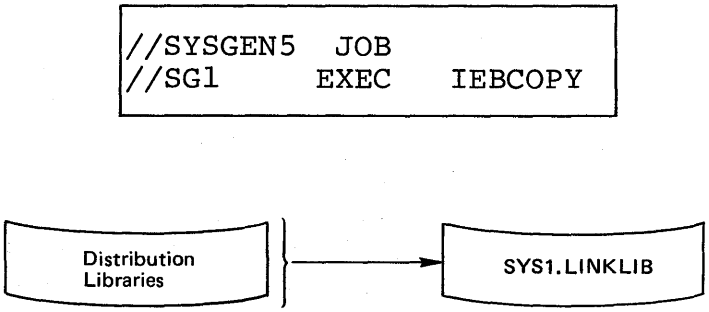


Job 4. This job catalogs and allocates space for SYS1.LOGREC on the new system residence volume (TSO and non-TSO generation).



```
//SYSGEN4  JOB
//SG1      EXEC   IFCDIP00
//SG2      EXEC   IEHPROGM
```

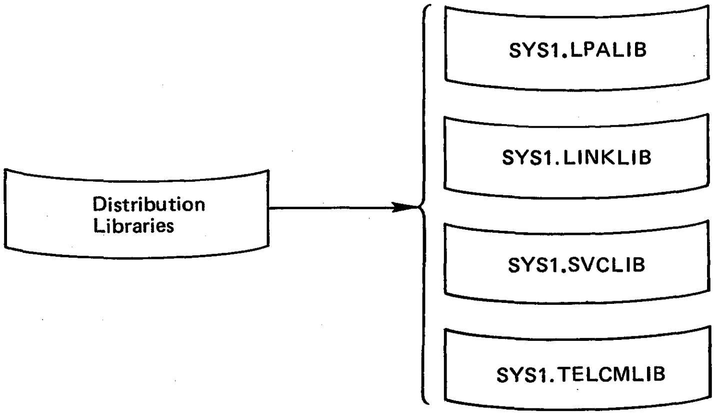
Job 5. The IEBCOPY utility program copies the service aid programs and the loader program from the distribution libraries to SYS1.LINKLIB in the new system (TSO and non-TSO generation).



```
//SYSGEN5  JOB
//SG1      EXEC   IEBCOPY
```

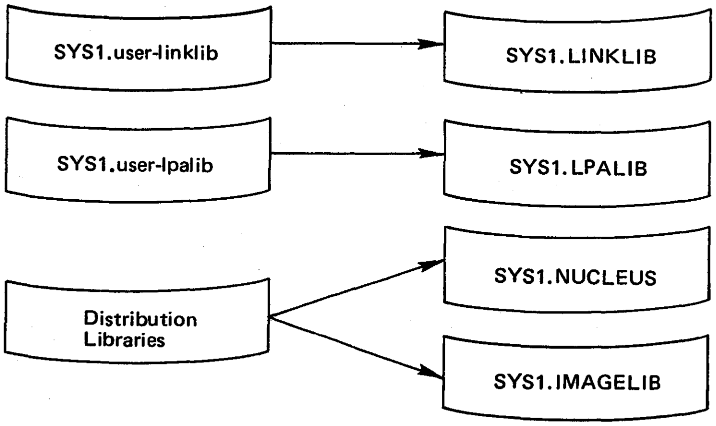
Job 6. The IEBCOPY utility program copies modules from the distribution libraries to SYS1.LPALIB, SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.TELCMLIB.

```
//SYSGEN6  JOB
//SG1      EXEC   IEBCOPY
```



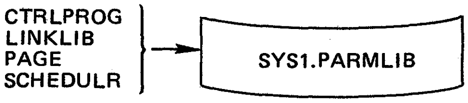
Job 7. The IEBCOPY utility program copies modules from the distribution libraries and user-defined libraries to SYS1.LINKLIB, SYS1.LPALIB, SYS1.NUCLEUS, and SYS1.IMAGELIB.

```
//SYSGEN7  JOB
//SG1      EXEC   IEBCOPY
```



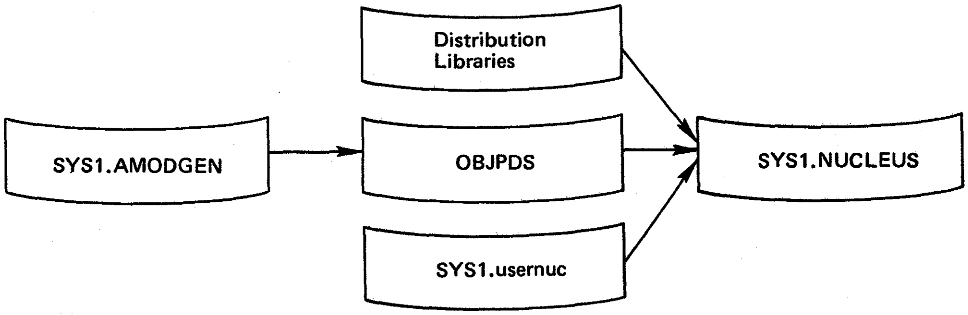
Job 8. The IEBUPDTE utility program adds members to the IEASYS00 and IEAFIX01 lists in SYS1.PARMLIB. The members that are added are specified in the CTRLPROG, LINKLIB, PAGE and SCHEDULR macro instructions (TSO and non-TSO generation).

```
//SYSGEN8  JOB
//SG1      EXEC   IEBUPDTE
             •
             •
             •
//SG2      EXEC   IEBUPDTE
```

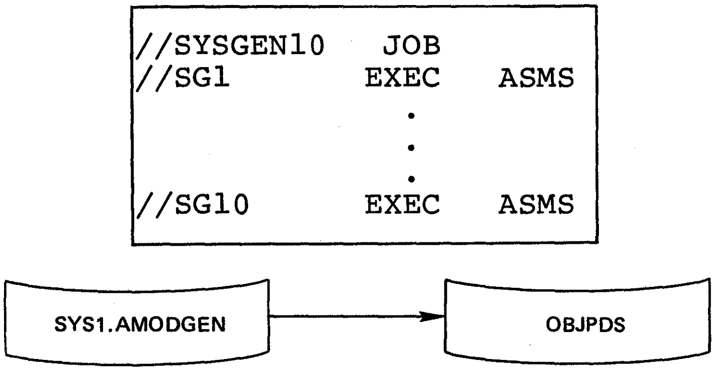


Job 9. Modules from SYS1.AMODGEN are assembled and placed in the object-module utility data set (OBJPDS). Modules from the distribution libraries, the OBJPDS utility data set, and user-libraries are then link-edited and placed in SYS1.NUCLEUS (TSO and non-TSO generation).

```
//SYSGEN9   JOB
//SG1       EXEC   ASMS
              •
              •
              •
//SG3       EXEC   ASMS
//SG4       EXEC   LINKS
```

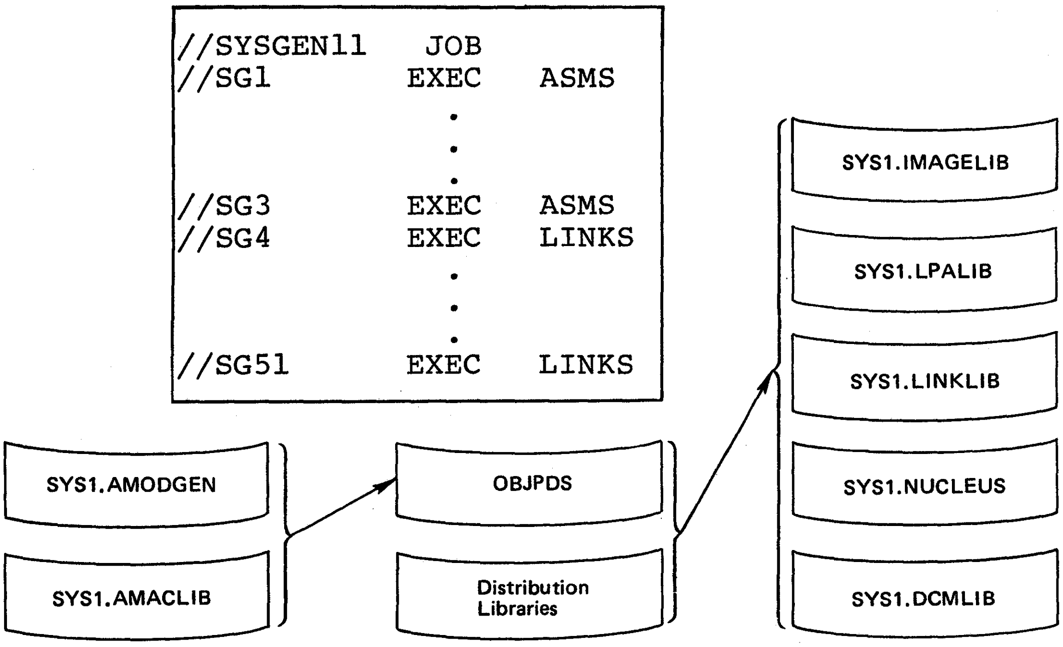


Job 10. Modules from SYS1.AMODGEN are assembled and placed in the object-module utility data set (OBJPDS). In a TSO-generation, modules would also be assembled from the SYS1.AMACLIB, SYS1.AMODGEN, and SYS1.ATSOMAC distribution libraries.



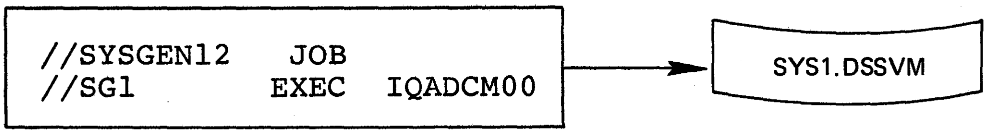
```
//SYSGEN10   JOB
//SG1        EXEC   ASMS
              .
              .
              .
//SG10       EXEC   ASMS
```

Job 11. Modules from SYS1.AMODGEN and SYS1.AMACLIB are assembled and placed in the object-module utility data set (OBJPDS). These modules and modules from the distribution libraries are then link-edited and placed in SYS1.IMAGELIB, SYS1.LPALIB, SYS1.LINKLIB, SYS1.NUCLEUS, and SYS1.DCMLIB. In a TSO-generation, this job step would also include link-edits into SYS1.CMDLIB.



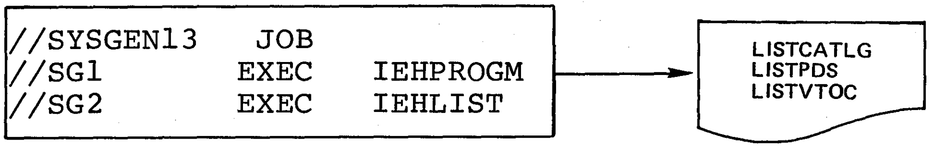
```
//SYSGEN11   JOB
//SG1        EXEC   ASMS
              .
              .
              .
//SG3        EXEC   ASMS
//SG4        EXEC   LINKS
              .
              .
              .
//SG51       EXEC   LINKS
```

Job 12. The IQADCM00 utility program formats the DSS paging system data set.



```
//SYSGEN12   JOB
//SG1        EXEC   IQADCM00
```

Job 13. The IEHPROGM utility program renames the index of the system data sets to SYS1. if you specified other than SYS1. in the INDEX parameter of the GENERATE macro instruction (complete, nucleus, and I/O device generation). The IEHLIST utility program lists the system catalog, the directories of the partitioned data sets in the new VS2 system, and the VTOC (complete and I/O device generation).



```
//SYSGEN13   JOB
//SG1        EXEC   IEHPROGM
//SG2        EXEC   IEHLIST
```

Figure 22 lists the components of the distribution libraries, the names of the distribution libraries from where the components are obtained, and where the components of the distribution libraries are placed during Stage II.

Component	Obtained from	Placed in
Access Methods	SYS1.AOSD0	SYS1.LPALIB
BDAM	SYS1.AOSD7	SYS1.LPALIB
BTAM	SYS1.AOS06 SYS1.AOS20 SYS1.AOS21	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB SYS1.TELCMLIB
Data Management Routines	SYS1.AOSD0	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB
DIDOCS	SYS1.AMODGEN SYS1.AOSC4	SYS1.DCMLIB SYS1.LPALIB
Direct-Access Device Space Management (DADSM)	SYS1.AOSD0	SYS1.LPALIB
EREP	SYS1.AOSCD	SYS1.LINKLIB
ERP	SYS1.AOSCD	SYS1.LOGREC
Generalized Trace Facility (GTF)	SYS1.AOS11 SYS1.AOSC5 SYS1.AMACLIB	SYS1.LINKLIB SYS1.NUCLEUS SYS1.MACLIB
Graphic Programming Services	SYS1.AOS07 SYS1.AOSG0	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS

Figure 22 (Part 1 of 4). The components of the distribution libraries and where they are placed in the system

Component	Obtained from	Placed in
Independent Utilities IPL Text SMF Sample Exit Routines	SYS1.ASAMPLIB	SYS1.SAMPLIB
Installation Verification Procedure	SYS1.ASAMPLIB	SYS1.SAMPLIB
IOS	SYS1.AOSCA SYS1.AOSG0 SYS1.AOSC5 SYS1.AOSD0 SYS1.AOSD7	SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB
ISAM	SYS1.AOSD8	SYS1.LPALIB SYS1.NUCLEUS
Linkage Editor	SYS1.AOS04 SYS1.AMODGEN	SYS1.LINKLIB
Loader	SYS1.AOS05 SYS1.AMODGEN	SYS1.LINKLIB
Macro Library	SYS1.AMACLIB SYS1.ATCAMMAC	SYS1.MACLIB
On-Line Test Executive Program (OLTEP)	SYS1.AOS06 SYS1.AOSC5	SYS1.LINKLIB SYS1.LPALIB
Open/Close/End-of-Volume	SYS1.AOSD0 SYS1.AOSD8	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS
Overlay Supervisor	SYS1.AOSC2	SYS1.LPALIB
Parameter Library	SYS1.APARMLIB	SYS1.PARMLIB
Procedure Library	SYS1.APROCLIB	SYS1.PROCLIB
Recovery Management	SYS1.AOSCD SYS1.AOSCE	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB
Service Aids	SYS1.AOS11 SYS1.AOS12 SYS1.AOSCD SYS1.AMACLIB	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB

Figure 22 (Part 2 of 4). The components of the distribution libraries and where they are placed in the system

Component	Obtained from	Placed in
Scheduler	SYS1.AOSC6 SYS1.AOSB3 SYS1.AOS00 SYS1.AOSC5 SYS1.AOS21 SYS1.AOSCE SYS1.AOST4 SYS1.AMODGEN	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS
Stage I System Generation Macro Definitions	SYS1.AGENLIB	Not applicable
Stage II System Generation Macro Definitions	SYS1.AMODGEN	Not applicable
Supervisor	SYS1.AOSC5 SYS1.AOS10 SYS1.AOSD0	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB
System Management Facility	SYS1.AOS00 SYS1.APARMLIB	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
OS/VS Assembler	SYS1.AOS03	SYS1.LINKLIB
TCAM	SYS1.AOS21	SYS1.LINKLIB SYS1.LPALIB SYS1.NUCLEUS SYS1.SVCLIB SYS1.TELCMLIB
TCAM Macro Definitions	SYS1.ATCAMMAC	SYS1.TELCMLIB
TSO Data Management Routines	SYS1.AOST3 SYS1.AOS21	SYS1.CMDLIB SYS1.DCMLIB SYS1.LINKLIB SYS1.LPALIB
TSO Linkage Editor Prompter	SYS1.ACMDLIB	SYS1.CMDLIB
TSO Editor	SYS1.ACMDLIB SYS1.ATSOMAC SYS1.AMACLIB	SYS1.CMDLIB SYS1.DCMLIB SYS1.LINKLIB SYS1.LPALIB
TSO Scheduler	SYS1.AOST4 SYS1.ACMDLIB SYS1.AMACLIB SYS1.ATSOMAC	SYS1.CMDLIB SYS1.LINKLIB SYS1.LPALIB

Figure 22 (Part 3 of 4). The components of the distribution libraries and where they are placed in the system

Component	Obtained from	Placed in
TSO Supervisor	SYS1.AOST4	SYS1.LINKLIB SYS1.LPALIB SYS1.SVCLIB
TSO Test	SYS1.ACMDLIB	SYS1.CMDLIB
TSO Type 1 Utilities	SYS1.ACMDLIB SYS1.AOSB3 SYS1.AOST4 SYS1.ATSOMAC SYS1.AMACLIB	SYS1.CMDLIB SYS1.LINKLIB SYS1.LPALIB
UCS Images	SYS1.AOSD0	SYS1.IMAGELIB
Utility Programs	SYS1.AOSU0	SYS1.LINKLIB SYS1.LPALIB
VSAM	SYS1.AOSA0	SYS1.LPALIB

Figure 22 (Part 4 of 4). The components of the distribution libraries and where they are placed in the system

Stage II Output

The output from Stage II is your new VS2 System. The new VS2 system that is generated is a System Control Program. Program Products are neither distributed nor generated with the System Control Program. Manuals that describe specific Program Products contain the procedures and requirements for installing them.

Stage II also produces a documentation listing. This is a printout of all the steps executed by the assembler, the linkage editor, and the utility programs.

Your new VS2 system should be tested to eliminate any possible malfunctions. Your VS2 system can be tested by the using the installation verification procedure (IVP). For the information necessary to use IVP, see "Testing the New System Control Program."

Removing the Names of the Distribution Libraries from the Catalog of the Generating System

After you have successfully installed your new VS2 system using an existing VS2 system or after you have performed a nucleus generation against the generating system, you will need to remove the names of the distribution libraries from the catalog of the generating system. This can be done by using the IEHPROGM utility program. Place the following IEHPROGM control statements in the input device and enter a START RDR,00C command. (If 00C is not the address of your input device, enter the correct address.) An UNCATLG statement must be specified for each entry to be removed from the catalog. The CVOL parameter should point to the catalog from which the entries are to be removed.

```
//UNCATLG       JOB
//            EXEC    PGM=IEHPROGM
//SYSPRINT      DD    SYSOUT=A
//CATLG         DD    UNIT=3330,DISP=OLD,                 X
//                    VOLUME=SER=volumeserial
//SYSIN         DD    *
        UNCATLG       DSNAME=dsname,                      X
                      CVOL=device=volumeserial
/*
```

Renaming the New Nucleus

If you performed a nucleus generation against the generating system, you may want to make the newly created secondary nucleus the primary nucleus (IEANUC01). This can be done in two ways:

- Rename the old primary nucleus and rename your new primary nucleus to IEANUC01, or
- Specify the new nucleus at IPL

RESTART PROCEDURES

The system generation process may come to an unsatisfactory completion because of errors that occurred during Stage I or Stage II. This chapter contains the information necessary to restart system generation.

The first section of this chapter discusses the most common causes of error during Stage I and the restart procedures for Stage I.

The second section of this chapter discusses the most common causes of error during Stage II, the restart techniques, and the reallocation of data sets.

Restarting Stage I

The most common causes of error during Stage I are keypunching errors in the input deck and contradictory or invalid specifications in the macro instructions.

Keypunching errors are indicated by system generation error messages or assembler error indications. Invalid specifications are indicated with the system generation error messages (see Appendix C) printed in the SYSPRINT data set. If any errors are found during Stage I, the job stream is not produced.

Stage I consists of a single assembly of the system generation macro instructions. It can be restarted only from the beginning. To restart Stage I, correct the errors in the input deck and resubmit the job.

Restarting Stage II

The most common causes of error during Stage II are:

- Machine interruptions and noncontinuous machine time.
- Faulty space allocation of the system data sets during the preparation for system generation.
- Errors in the input deck that cannot be detected during Stage I. For example, if SYS1.NUCLEUS was allocated space on volume 111111 during the preparation for system generation, and if RESVOL=A11111 was specified in the GENERATE macro instruction, an error would occur.
- Procedural errors, such as improper volume mounting.

Restart Techniques

Stage II can be restarted at the beginning of any job or job step. If any statements in the job stream are to be changed, the job stream must be on cards. If no statements are to be changed, the IEBEDIT utility program can be used to restart a job stream. Procedures discussed in the section "Guidelines for Restarting Stage II" may need to be performed before restarting Stage II. This section discusses the techniques used for restarting the job stream after any other necessary operations have been performed. The topics include restarting from cards, punching the job stream, and restarting from tape or disk.

Restarting From Cards

If the job stream is on cards, you can restart a job step by placing a JOB card ahead of the job step's EXEC card and entering a START RDR command for the card reader.

Punching the Job Stream

If the unit (SYSPUNCH) specified for the job stream during Stage I was not a card punch, the IEBPTPCH utility program can be used to punch the job stream. Figure 23 shows the statements required to punch the job stream using IEBPTPCH. In this example, the underlined fields may require modification for different installations.

```
//PUNCH       JOB
//           EXEC  PGM=IEBPTPCH
//SYSUT1       DD  UNIT=182,LABEL=(,NL),                    X
//                 VOLUME=SER=EXLABL,DISP=OLD
//SYSUT2       DD  UNIT=2540-2
//SYSPRINT     DD  SYSOUT=A
//SYSIN        DD  *
      PUNCH        TYPORG=PS,MAXFLDS=1
      RECORD       FIELD=(80)
/*
```

Figure 23. Punching the job stream on any 2540 card read punch

When you use the IEBPTPCH utility program to punch the job stream, consider the following:

- The value of the UNIT parameter of the SYSUT1 DD statement is the specific unit address of the magnetic tape drive or direct-access storage device on which the job stream resides. Unless the job stream tape or disk has been demounted, the value of this UNIT parameter is the same as the value of the UNIT parameter of the SYSPUNCH DD statement in the input deck for Stage I. If the job stream is on disk, the LABEL parameter must specify a standard label and a DSNAME parameter must be specified.
- The value of the VOLUME parameter of the SYSUT1 DD statement is either any external serial number you have assigned to the job stream tape reel or the volume serial number of the tape or direct-access storage device. The system will issue a MOUNT command for the specified volume on the magnetic tape or direct-access storage device indicated by the UNIT parameter.
- You can specify sequence numbers for the punched cards by specifying the CDSEQ or CDINCR parameters in the PUNCH control cards of the IEBPTPCH input deck. (Refer to *OS/VS Utilities* for information about the CDSEQ and CDINCR parameters).

Restarting From Tape or Disk

The IEBEDIT utility program can be used to restart Stage II from any job step after the first when the job stream is on tape or disk. To restart from the first job step, issue a START RDR command for the tape drive or direct-access storage device that contains the job stream.

IEBEDIT creates a new job stream by editing and selectively copying the job stream provided as input. The IEBEDIT utility program can copy an entire set of jobs, including JOB statements and associated job step statements, or selected job steps in a job. Figure 24 shows the control statements required by IEBEDIT.

```
//RESTART    JOB
//          EXEC  PGM=IEBEDIT
//SYSPRINT   DD   SYSOUT=A
//SYSUT1     DD   UNIT=xxxx,LABEL=(,NL),                 X
//                VOLUME=SER=serial,DISP=(OLD,KEEP),X
//                DSN=dsname,                            X
//                DCB=(DCB information)
//SYSUT2     DD   UNIT=xxxx,LABEL=(,NL),                 X
//                VOLUME=SER=serial,DISP=(,KEEP),        X
//                DSN=dsname,                            X
//                DCB=(DCB information)
//SYSIN      DD   *
    EDIT       START=SYSGENnn,STEPNAME=SGxx[,NOPRINT]
or  EDIT       START=SYSGENnn,TYPE=INCLUDE,              X
               STEPNAME=(SGxx[,SGxx]...)[,NOPRINT]
or  EDIT       START=SYSGENnn,TYPE=EXCLUDE,              X
               STEPNAME=(SGxx[,SGxx]...)[,NOPRINT]
/*
```

Figure 24. Control Statements for IEBEDIT when the job stream is on tape

When you use the IEBEDIT utility program to restart Stage II consider the following:

- The value of the UNIT parameter of the SYSUT1 DD statement is the unit address of the magnetic tape drive or direct-access storage device on which the job stream resides.

  Unless the job stream tape or disk had been demounted, the value of the UNIT parameter is the same as the value of the UNIT parameter of the SYSPUNCH DD statement in the Stage I input deck. If the job stream is on disk, the LABEL parameter must specify a standard label.

- The value of the VOLUME parameter of the SYSUT1 DD statement is either any serial number you have assigned to the job stream tape reel or the volume serial number of the tape or direct-access storage device. The system will issue a MOUNT command for the specified volume on the magnetic tape drive or direct-access storage device indicated with the UNIT parameter.

- The value of the UNIT parameter of the SYSUT2 DD statement is the unit address of a magnetic tape drive or direct-access storage device. If the job stream is on disk, the LABEL parameter must specify a standard label.

- You can specify one or more EDIT statements when executing IEBEDIT. If the TYPE parameter is omitted, STEPNAME specifies the first job step in the job specified by the START parameter to be placed in the new job stream.

- If TYPE=INCLUDE or TYPE=EXCLUDE is specified, STEPNAME specifies the job steps to be included or excluded, respectively, from the new job stream. Individual job steps and sequences of job steps can be specified for inclusion or exclusion. For example:

  START=SYSGEN9,TYPE=INCLUDE,STEPNAME=(SG3,SG6-SG9)

  indicates that job steps 3, 6, 7, 8, and 9 of job 9 are to be included in the restart of system generation.

- NOPRINT must be included if you do not want a listing of the new job stream. After the new job stream is created, a START RDR command must be issued for the magnetic tape drive or direct-access storage device designated by the SYSUT2 DD statement.

For additional information on the IEBEDIT utility program, refer to *OS/VS Utilities*.

Figure 25 shows an IEBEDIT input deck for restarting Stage II, jobs 11, 12, and 13 from tape. In this example, space allocation for SYS1.LINKLIB was not sufficient, causing the subsequent job steps to fail. Since job 11 included link-edits with output to SYS1.LPALIB and SYS1.LINKLIB that did not execute, they need to be restarted. Also, although job 12 executed, it needs to be restarted to obtain accurate results. The restart job stream consists of job 11, steps 4 through 24, and job 12 (IEHLIST).

```
//RESTART    JOB
//          EXEC    PGM=IEBEDIT
//SYSPRINT   DD     SYSOUT=A
//SYSUT1     DD     UNIT=2400,LABEL=(,NL),DSN=STAGE1, X
//                  VOL=SER=JOBSTM,                   X
//                  DCB=(RECFM=F,BLKSIZE=80,DEN=2),   X
//                  DISP=(OLD,KEEP)
//SYSUT2     DD     UNIT=2400,DISP=(,KEEP),           X
//                  VOL=SER=001234,DSN=OUTTAPE,       X
//                  LABEL=(,NL),                      X
//                  DCB=(RECFM=F,BLKSIZE=80,DEN=2)
//SYSIN      DD     *
       EDIT         START=SYSGEN11,TYPE=EXCLUDE,      X
                    STEPNAME=(SG1,SG2,SG3)
       EDIT         START=SYSGEN12
/*
```

Figure 25. Restarting Stage II from tape

Guidelines for Restarting Stage II

The following section gives guidelines for restarting Stage II. Restarting may require the scratching and reallocation of space for the system data sets. When this is necessary, refer to the section, "Reallocating Space for Data Sets" for the procedure to be followed. After the necessary corrections have been made, the actual restarting of Stage II can be accomplished by one of the methods described in "Restart Techniques."

If the problem encountered is other than space allocation, such as component failures or machine malfunctions, you should follow the instructions printed out in the error messages or error codes and then refer to *OS/VS Message Library: VS2 System Messages* or *OS/VS Message Library: System Codes.*

Restarting Job 1: IEHPROGM. Restart the entire job.

Restarting Job 2: Assemblies. To restart any assembly step, place the JOB card in front of the EXEC card for that assembly, or use the IEBEDIT program if the job stream is on tape or disk.

Restarting Job 3: IEBCOPY. To restart, space for SYS1.MACLIB, SYS1.PROCLIB, SYS1.PARMLIB, and SYS1.SAMPLIB must be reallocated if space allocation was the problem. If this was a TSO-generation, space for SYS1.HELP and SYS1.UADS must also be reallocated. Scratch the space for the data sets on their original volume using the IEHPROGM utility program. Reallocate the space, increasing the amount, for the data sets on their original volume. Then restart Stage II at the beginning of the job.

Restarting Job 4. IEHPROGM, IFCDIP00. To restart, place the JOB card in front of the job step that failed, or use the IEBEDIT utility program if the job stream is on tape or disk.

Restarting Jobs 5, 6, and 7: IEBCOPY. If the problem was not enough space, you can carry out one of these procedures:

- Copy the system data set to a scratch volume using the IEBCOPY utility program. Scratch the space for the system data set on its original volume using IEHPROGM. Reallocate the space, increasing the amount of space, for the system data set on its original volume. Use IEBCOPY to copy the system data set from the scratch volume to its original volume. Restart Stage II and the beginning of the job, or
- Reallocate the system data sets affected and restart with the first job that performs copies to those system data sets.

If the problem is not insufficient space, restart the job, as any members already copied will not be copied again. (See "Reallocating Space for Data Sets.")

Restarting Job 8: IEBUPDTE. To restart, place the JOB card in front of the first EXEC statement to be reexecuted or use the IEBEDIT program if the job stream is on tape or disk.

Restarting Job 9: Assemblies and Nucleus Link-Edit. To restart any assembly step, place the JOB card in front of the EXEC card for that assembly or use the IEBEDIT program if the job stream is on tape or disk. All assemblies must execute successfully before the link-edit step will start. If space allocation was the problem:

- Scratch SYS1.NUCLEUS and reallocate space for it
- Execute Job 7 (IEBCOPY) to copy modules from the distribution library to SYS1.NUCLEUS
- Execute the nucleus link-edit job step

Restarting Job 10: Assemblies. To restart, place the JOB card in front of the EXEC card of the step that failed or use the IEBEDIT utility program if the job stream is on tape or disk.

Restarting Job 11: Assemblies and Link-Edits. To restart any assembly step, place the JOB card in front of the EXEC card for that assembly or use the IEBEDIT program if the job stream is on tape or disk. All assemblies must execute successfully before the link-edit steps will start. To restart a link-edit step:

- If the problem is not enough space for the system data set, scratch any members that were added by the failing step. Copy the system data set to a scratch volume using the IEBCOPY utility program. Scratch the system data set from its original volume. Reallocate space for the data set and copy the data set from the scratch volume. Place the JOB card in front of the EXEC card of the job step that failed or use the IEBEDIT program if the job stream is on tape or disk.
- If the problem is not one of space, place the JOB card in front of the EXEC card of the job step that failed or use the IEBEDIT program if the job stream is on tape or disk.

Restarting Job 12: IQADCM00. Restart the entire job.

Restarting Job 13: IEHPROGM, IEHLIST. If this job step executed before all previous job steps have executed successfully and you specified an index value in the GENERATE macro instruction, you must rename the index of the system data sets to the value specified in the INDEX parameter of the GENERATE macro instruction. Then reexecute the jobs that failed and rerun job 13.

Reallocating Space for Data Sets

The following sections discuss the reallocation of space for the utility data set, OBJPDS, and the reallocation of space for the system data sets.

Reallocating space for data sets includes:

- Scratching the space that was originally allocated to the data set
- Allocating new space to the data set
- Possible recataloging if you are switching volumes

The reallocations described in this section may alter the sequence of the new system data sets on a volume. If their sequence is important, the IEBCOPY utility program can be used after system generation to rearrange the new system data sets.

Reallocating Space for the OBJPDS Utility Data Set

Figure 26 shows the statements used to reallocate space for the utility data set OBJPDS – space that was originally allocated for Stage II of system generation. (See "Stage II Input" in the chapter "System Control Program Installation".) When scratching the data set, the DSNAME that is specified in the first DD statement must be the DSNAME that was specified for that data set in the original allocation.

```
//OBJPDS   JOB
//STEP1    EXEC   PGM=IEHPROGM          SCRATCH OBJPDS
//OBJPDS   DD     DSNAME=SYS1.name,DISP=(OLD,DELETE)
//SYSPRINT  DD    SYSOUT=A
//SYSIN    DD     DUMMY
//STEP2    EXEC   PGM=IEHPROGM          REALLOCATE OBJPDS
//OBJPDS   DD     DSNAME=dsname,SPACE=(space),          X
//                DISP=(,KEEP),UNIT=unit address,       X
//                VOLUME=(,RETAIN,SER=volume serial)
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     DUMMY
/*
//
```

Figure 26. Reallocating space for the OBJPDS utility data set

Reallocating Space for System Data Sets

The method for reallocating space for a system data set depends on whether the data set contains data that must be saved. If the data set does not contain data, the IEHPROGM utility program can be used to scratch and reallocate the system data set. If the system data set contains data that must be saved, the data will have to be copied into a temporary data set, the original data set will have to be reallocated, and the contents of the data set copied from the temporary data set into the reallocated data set.

The input deck for scratching and reallocating system data sets must contain the following statements in the order shown:

- A JOB statement with any parameters required by your installation.
- An EXEC statement with the PGM=IEHPROGM parameter.
- A SYSPRINT DD statement defining the system output unit.
- A DD statement defining the unit address and serial number of the generating system's system residence volume:

```
//SYSRES        DD  UNIT=unit,VOLUME=SER=serial,DISP=OLD
```

- A DD statement defining any other permanent volume on which the system data sets to be reallocated reside:

```
//OTHERVOL      DD  UNIT=unit,VOLUME=SER=serial,DISP=OLD
```

- A DD statement for each type of removable volume on which the system data sets to be reallocated reside:

```
//DDNAME        DD  UNIT=(unit,,DEFER),VOLUME=PRIVATE,DISP=OLD
```

- A DD * statement (SYSIN).
- A SCRATCH statement for each new system data set to be reallocated. The SCRATCH statement must have the following format:

```
SCRATCH         DSNAME=dsname,VOL=device=serial,PURGE
```

- A /* statement.
- An EXEC statement with the PGM=IEHPROGM parameter.
- A DD statement defining the unit address and serial number of the generating system's system residence volume (example shown above).
- A DD statement for each permanent volume on which the system data sets to be reallocated reside (example shown above).
- A DD statement for each type of removable volume on which the system data sets to be reallocated reside (example shown above).
- A SYSPRINT DD statement defining the system output unit.
- A DD statement for each of the system data sets to be reallocated:

```
//ddname  DD  DSNAME=dsname,VOLUME=(,RETAIN,SER=serial),          X
//            UNIT=unit,LABEL=EXPDT=99350,SPACE=(allocation),     X
//            DISP=(,KEEP),DCB=(parameters)
```

- A DD * statement (SYSIN).
- A /* statement.

As an example of the reallocation of system data sets that do not contain meaningful data, suppose SYS1.LPALIB and SYS1.LINKLIB must be reallocated. The original allocation for these two data sets was:

```
//LPALIB    DD  DSNAME=SYS1.LPALIB,VOLUME=(,RETAIN,SER=AAA111),     X
//              UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(15,1,187)),      X
//              LABEL=EXPDT=99350,                                  X
//              DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//LINKLIB   DD  DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=AAA112),    X
//              UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(15,5,100)),      X
//              LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030)
```

The generating system residence volume is a 3330 volume. Figure 27 shows the input deck for this reallocation:

```
//SAME       JOB
//STEP1     EXEC PGM=IEHPROGM              SCRATCH
//NEWRES     DD  UNIT=3330,VOLUME=SER=AAA111,DISP=OLD
//LINVOL     DD  UNIT=3330,VOLUME=(,RETAIN,SER=AAA112),DISP=OLD
//SYSPRINT   DD  SYSOUT=A
//SYSIN      DD  *
      SCRATCH    DSNAME=SYS1.LPALIB,VOL=3330=AAA111,PURGE
      SCRATCH    DSNAME=SYS1.LINKLIB,VOL=3330=AAA112,PURGE
/*
//STEP2     EXEC PGM=IEHPROGM              ALLOCATE
//NEWRES     DD  UNIT=3330,VOLUME=SER=AAA111,DISP=OLD
//LINVOL     DD  UNIT=(3330,,DEFER),VOLUME=PRIVATE,DISP=OLD
//SYSPRINT   DD  SYSOUT=A
//LPALIB     DD  DSNAME=SYS1.LPALIB,VOLUME=(,RETAIN,SER=AAA111),     X
//               UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(20,5,75)),       X
//               LABEL=EXPDT=99350,                                  X
//               DCB=(RECFM=U,BLKSIZE=1024,DSORG=POU)
//LINKLIB    DD  DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=AAA112),    X
//               UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(50,10, 150)),    X
//               LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030)
//SYSIN      DD  DUMMY
/*
//
```

Figure 27. Reallocating SYS1.LPALIB and SYS1.LINKLIB

If the system data set to be reallocated contains data, one of two procedures can be followed if there is enough space on the volume for a new allocation:

- Rename the system data set
- Allocate space for the system data set (with its correct name) on the same volume using the IEHPROGM utility program

- Copy the data in the renamed data set onto the newly allocated system data set using the IEBCOPY utility program
- Scratch the renamed data set using the IEHPROGM utility program

Figure 28 illustrates reallocation of a data set on the same volume. The system data set to be reallocated is SYS1.PARMLIB. It was allocated space during the preparation for system generation with the following IEHPROGM DD statement:

```
//PARMLIB   DD  DSNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=SYSTEM),  X
//              UNIT=2314,DISP=(,KEEP),                            X
//              SPACE=(TRK,(7,1,3),,CONTIG),LABEL=EXPDT=99350,     X
//              DCB=(RECFM=F,BLKSIZE=80)
```

The new system residence volume is a 2314 volume whose serial number is SYSTEM. The renamed SYS1.PARMLIB will be called SYS1.TEMPPARM.

```
//MOVE     JOB
//STEP1   EXEC PGM=IEHPROGM                RENAME
//SYSPRINT DD  SYSOUT=A
//NEWRES   DD  UNIT=2314,VOLUME=SER=SYSTEM,DISP=OLD
//SYSIN    DD  *
      RENAME   DSNAME=SYS1.PARMLIB,VOL=2314=SYSTEM, X
               NEWNAME=SYS1.TEMPPARM
/*
//STEP2   EXEC PGM=IEHPROGM                REALLOCATE
//SYSPRINT DD  SYSOUT=A
//PARMLIB  DD  DSNAME=SYS1.PARMLIB,                  X
//             VOLUME=(,RETAIN,SER=SYSTEM),          X
//             UNIT=2314,DISP=(,KEEP),               X
//             SPACE=(TRK,(8,,3),,CONTIG),           X
//             LABEL=EXPDT=99350,                    X
//             DCB=(RECFM=F,BLKSIZE=80)
//SYSIN    DD  DUMMY
/*
//STEP3   EXEC PGM=IEBCOPY                 COPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=SYS1.TEMPPARM,DISP=OLD,        X
//             UNIT=2314,VOL=SER=SYSTEM
//SYSUT2   DD  DSNAME=SYS1.PARMLIB,DISP=OLD
//SYSIN    DD  *
     COPY      INDD=SYSUT1,OUTDD=SYSUT2
/*
//STEP4   EXEC PGM=IEHPROGM                SCRATCH
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
     SCRATCH   DSNAME=SYS1.TEMPPARM,VOL=2314=SYSTEM
/*
```

Figure 28. Reallocating SYS1.PARMLIB

If there is not enough room on the volume for a temporary data set:

- Use the IEBCOPY utility program to copy the data set onto another volume.
- Scratch the data set from the original volume and reallocate the space for the data set on the original volume.
- Use IEBCOPY to copy the contents of the data set from the other volume to the original volume.

TESTING THE SYSTEM CONTROL PROGRAM

This chapter discusses the installation verification procedure. Included in this chapter are:

- A definition of the installation verification procedure
- A discussion of the installation verification procedure
- The system configuration required to use the installation verification procedure
- The procedures for using the installation verification procedure.

The Installation Verification Procedure (IVP)

The installation verification procedure (IVP) is a program that tests whether the newly installed VS2 System Control Program is operational. It also tests whether the newly created System Control Program supports your machine configuration.

The jobs in IVP test only the System Control Program. IVP does not test any Program Products, programs with service classification "C," or similar programs added after the new system control program has been generated. Any program of this type for which verification or demonstration is required should be tested by its own testing procedure or sample program.

IVP Minimum Configuration

The IVP job stream will execute with all machine configurations and requires only the minimum VS2 configuration.

The IVP Job Stream

The IVP job stream is contained in your new VS2 system in the partitioned data set SYS1.SAMPLIB. Its member name is IVPJOBS. The IVP job stream consists of jobs that:

- Provide device information, such as unit address, device type, device status, volume serial number, volume state, ordered output, and device bypass.
- Execute an assemble/link-edit/go to test the basic functions of the newly installed System Control Program.
- Provide a listing of SYS1.PARMLIB.
- Force an 806 ABEND and the resulting dump (see *OS/VS Message Library: VS2 System Codes).*

Procedures for Using IVP

To verify that your new System Control Program is correctly installed, carry out the following steps:

- Upon successful completion of Stage II (no unaccounted for errors are indicated in the Stage II listings), carry out the IPL procedure.
- Issue a START RDR command to the IVP job stream (IVPJOBS)

- To issue a START RDR command to the IVP job stream, type the following command:

```
START RDR.R,132,VOL=SER=SYSRES,LABEL=(,SL),
DSNAME=SYS1.SAMPLIB(IVPJOBS)
```

- If 132 and SYSRES are not the unit address and volume serial number of the system residence volume, enter the correct information.
- Review the I/O device information produced by the job stream to verify the machine configuration.
- Review the listing of SYS1.PARMLIB for completeness.
- Review the results of the assemble/link-edit/go for errors.
- Verify that an 806 ABEND occurred and that a dump was provided.
- Save the IVP output for future reference.

The completion of the above steps and normal end-of-job (EOJ) of the IVP job stream (excluding the 806 ABEND) constitutes successful verification of your new VS2 system.

EXAMPLES OF SYSTEM GENERATION

This chapter contains examples of system generation. Included in this chapter are:

- A diagram of the system to be generated
- The system generation macro instructions for a complete system generation
- The system generation macro instructions for a nucleus generation
- The system generation macro instructions for an I/O device generation

Machine Configuration

Figure 29 shows the machine configuration for the system to be generated using the macro instructions shown in Figure 30. The CPU is a System/370 Model 145.

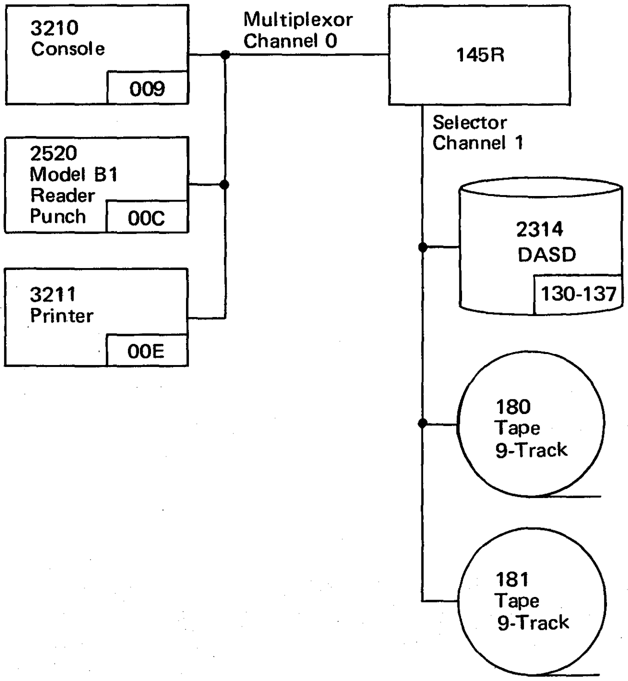


Figure 29. Machine Configuration

System Generation Macro Instructions for a Complete System Generation

Figure 30 shows the system generation macro instructions for a complete system generation.

The new system will support the devices shown in Figure 29. The following options have been specified:

- User-assigned checkpoint/restart codes
- The size of the area of virtual storage that is to contain the linkage editor
- The optiohs for the loader
- Exclusion of BTAM
- Exclusion of GPS,TCAM, OCR, and TSO macro definitions from SYS1.MACLIB
- The size of the page data sets
- Inclusion of 1403 UCS images
- The unitname SYSSQ for magnetic tape
- The unitname SYSDA for direct-access devices
- A user-library SYS1.USER that contains modules to be placed in the nucleus and SYS1.LPALIB during system generation
- A user-library SYS1.LINK that contains modules to be placed in SYS1.LINKLIB during system generation.

Col. 10

```
CENPROCS MODEL=145R
CTRLPROG QSPACE=1,TZ=(E,5,5),REAL=16,TRACE=100,        X
      ASCII=INCLUDE,OPTIONS=(DEVSTAT),MAXIO=50,         X
      DEBTSZE=80,DEBTINC=80
CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CHANNEL ADDRESS=(1,2),TYPE=SELECTOR
IODEVICE UNIT=3210,ADDRESS=009
IODEVICE UNIT=2520,MODEL=B1,ADDRESS=(00C)
IODEVICE UNIT=3211,ADDRESS=00E
IODEVICE UNIT=2401,ADDRESS=(180,2),MODEL=1,             X
      FEATURE=9-TRACK
IODEVICE UNIT=2314,ADDRESS=(130),OPTCHAN=2
CKPTREST    NOTELIG=(001,031),ELIGBLE=(100)
DATAMGT  ACSMETH=(BTAM)
EDITOR SIZE=(256,64)
LOADER   LIN=SYSLIN,LIB=SYSLIB,PRINT=SYSLOUT,           X
      SIZE=164,PARM=(MAP,LET,CALL,RES)
MACLIB EXCLUDE=(GPS,TCAM,OCR,TSO)
GRAPHICS PORRTNS=INCLUDE,GSP=EXCLUDE
PAGE TYPE=P,SIZE=(BLK,10),LPA=YES
PAGE  UNIT=130
SCHEDULR CONSOLE=009,ALTCONS=(I-00C,O-00E),             X
      START I=MANUAL,REPLY=10,                          X
      HARDCPY=(SYSLOG,ALL,INCMDS),BCLMT=100,            X
      VLMOUNT=AVR,TAVR=200,WTLCLSS=D,ESV=CON,           X
      IOC=009
SECONSLE CONSOLE=(I-00C,O-00E),ALTCONS=009
```

Figure 30 (Part 1 of 2). System generation macro instructions for a complete system generation

```
LPALIB PDS=SYS1.USER,MEMBERS=(IGC0025E)
SVCTABLE SVC-250-D1-FC01,SVC-255-E3-FC00
RESMODS  PDS=SYS1.USER,MEMBERS=(RES1,RES2,IGC250)
LINKLIB  PDS=SYS1.LINK,MEMBERS=(U1,U2)
UCS IMAGE=ALL1403,DEFAULT=(AN,HN)
UNITNAME NAME=SYSDA,                                 X
      UNIT=(130,131,132,133,134,135,136,137)
UNITNAME NAME=SYSSQ,UNIT=(180,181)
GENERATE   GENTYPE=ALL
END
```

Figure 30 (Part 2 of 2). System generation macro instructions for a complete system generation

System Generation Macro Instructions for a Nucleus Generation

Figure 31 shows the macro instructions to add a new nucleus to the system that was generated using the macros shown in Figure 30.

For a nucleus generation, the CENPROCS, CHANNEL, and IODEVICE macros must be respecified exactly as they were for the last complete system generation. The SCHEDULR parameters must also be respecified. DEVSTAT support and the system trace table will be excluded from the system (CTRLPROG). The DEB table and the number of extents will be altered from what was originally specified in the CTRLPROG macro to the default values. An additional page data set is to be included and additional user-written routines are to be added to the nucleus and SYS1.LINKLIB.

```
Col. 10
CENPROCS MODEL=145R
CTRLPROG QSPACE=1,TZ=(E,5,5),REAL=16,              X
      TRACE=0,ASCII=INCLUDE
CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CHANNEL ADDRESS=(1,2),TYPE=SELECTOR
IODEVICE UNIT=3210,ADDRESS=009
IODEVICE UNIT=2520,MODEL=B1,ADDRESS=(00C)
IODEVICE UNIT=3211,ADDRESS=00E
IODEVICE UNIT=2401,ADDRESS=(180,2),MODEL=1,        X
      FEATURE=9-TRACK
IODEVICE UNIT=2314,ADDRESS=(130),OPTCHAN=2
DATAMGT  ACSMETH=(BTAM)
GRAPHICS PORRTNS=INCLUDE,GSP=EXCLUDE
PAGE TYPE=P,SIZE=(BLK,10),LPA=YES
PAGE  UNIT=130
PAGE  UNIT=131     ADD PAGE ON 131
SCHEDULR CONSOLE=009,ALTCONS=(I-00C,O-00E),        X
      STARTI=MANUAL,REPLY=10,                      X
      HARDCPY=(SYSLOG,ALL,INCMDS),BCLMT=100,       X
      VLMOUNT=AVR,TAVR=200,WTLCLSS=D,ESV=CON,      X
      IOC=009
SECONSLE CONSOLE=(I-00C,O-00E),ALTCONS=009
LPALIB PDS=SYS1.USER,MEMBERS=(IGC0025E)
```

Figure 31 (Part 1 of 2). System generation macro instructions for a nucleus generation

```
SVCTABLE SVC-250-D1-FC01,SVC-255-E3-FC00
RESMODS  PDS=SYS1.USER,MEMBERS=(RES1,RES2,RES3,IGC250)
LINKLIB  PDS=SYS1.LINK,                                 X
      MEMBERS=(U1,U2,U3)   ADD MEMBER U3
UNITNAME NAME=SYSSQ,UNIT=(180,181,(130,8))
UNITNAME NAME=SYSDA,UNIT=((130,8))
GENERATE   GENTYPE=(NUCLEUS,1)
END
```

Figure 31 (Part 2 of 2). System generation macro instructions for a nucleus generation

System Generation Macro Instructions for an I/O Device Generation

Figure 32 shows the macro instructions to add additional I/O devices and channels to the system that was generated using the macros shown in Figure 30.

In this example, a 3330 at addresses 150-155 and a 3420 at addresses 280 and 281 are to be added to the system configuration, and channel 3 is to be made available to the system. The unitname specifications were changed so that SYSSQ will be the unitname for all of the sequential and direct-access devices. The number of simultaneous I/O operations that can occur has been increased to 100.

```
┌Col. 10
CENPROCS MODEL=145R
CTRLPROG QSPACE=1,TZ=(E,5,5),REAL=16,TRACE=100,        X
      ASCII=INCLUDE,OPTIONS=(DEVSTAT),MAXIO=100,       X
      DEBTSZE=80,DEBTINC=80
CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CHANNEL ADDRESS=(1,2),TYPE=SELECTOR
CHANNEL  ADDRESS=3,TYPE=BLKMPXR
IODEVICE UNIT=3210,ADDRESS=009
IODEVICE UNIT=2520,MODEL=B1,ADDRESS=(00C)
IODEVICE UNIT=3211,ADDRESS=00E
IODEVICE UNIT=2401,ADDRESS=(180,2),MODEL=1,            X
      FEATURE=9-TRACK
IODEVICE UNIT=2314,ADDRESS=(130),OPTCHAN=2
IODEVICE UNIT=3330,ADDRESS=(150,6),FEATURE=SHARED,     X
      OPTCHAN=(2,3)
IODEVICE UNIT=3420,ADDRESS=(280,2),MODEL=5,            X
      FEATURE=(9-TRACK,SHARABLE),OPTCHAN=3
IODEVICE UNIT=1287,ADDRESS=00A
CKPTREST    NOTELIG=(001,031),ELIGBLE=(100)
DATAMGT  ACSMETH=(BTAM)
GRAPHICS PORRTNS=INCLUDE,GSP=EXCLUDE
PAGE TYPE=P,SIZE=(BLK,10),LPA=YES
PAGE   UNIT=130
```

Figure 32 (Part 1 of 2). System generation macro instructions for an I/O device generation

```
SCHEDULR CONSOLE=009,ALTCONS=(I-00C,O-00E),                            X
      STARTI=MANUAL,REPLY=10,                                          X
      HARDCPY=(SYSLOG,ALL,INCMDS),BCLMT=100,                           X
      VLMOUNT=AVR,TAVR=200,WTLCLSS=D,ESV=CON,IOC=009
SECONSLE CONSOLE=(I-00C,O-00E),ALTCONS=009
LPALIB PDS=SYS1.USER,MEMBERS=(IGC0025E)
SVCTABLE SVC-250-D1-FC01,SVC-255-E3-FC00
RESMODS  PDS=SYS1.USER,MEMBERS=(RES1,RES2,IGC250)
LINKLIB  PDS=SYS1.LINK,MEMBERS=(U1,U2)
UCS IMAGE=ALL1403,DEFAULT=(AN,HN)
UNITNAME NAME=SYSDA,UNIT=((130,7),(150,6))
UNITNAME NAME=SYSSQ,                                                   X
      UNIT=(180,181,280,281,(130,8),(150,6))
GENERATE GENTYPE=(IO,1)
```

Figure 32 (Part 2 of 2). System generation macro instructions for an I/O device generation

APPENDIX A: DEVICE TYPES

A device type is automatically assigned during system generation to each collection of devices for which a type of device is specified by the UNIT parameter of an IODEVICE macro instruction. The names and devices to which they apply follow.

Device Type	Description
	Magnetic Tape Devices
2400	2400 9-track magnetic tape drive having an 800 bits-per-inch (density) capability when the dual-density feature is not installed or an 800 and 1600 bits-per-inch (density) capability when the dual-density feature is installed
2400-1	2400 magnetic tape drive with 7-track capability and without data conversion
2400-2	2400 magnetic tape drive with 7-track capability and data conversion
2400-3	2400 or 2420 9-track magnetic tape drive having only a 1600 bits-per-inch (density) capability
2400-4	2400 9-track magnetic tape drive having an 800 and 1600 bits-per-inch (density) capability
2495	2495 tape cartridge reader
3400-2	3420 magnetic tape drive having 7-track capability and data conversion
3400-3	3410 or 3420 9-track magnetic tape drive having 1600 bits-per-inch (density) capability
3400-4	3410 or 3420 9-track magnetic tape drive having 800 and 1600 bits-per-inch (density) capability
	Direct-Access Devices
2305-1	2305 fixed-head disk storage (Model 1)
2305-2	2305 fixed-head disk storage (Model 2)
2314/2319	2314/2319 direct-access storage facility
3330/3333	3330 disk storage drive/3333 disk storage and control
	Unit Record Equipment and other Devices
1052	1052 printer keyboard (Model 7)
1053	1053 printer (Model 4)
1403	1403 printer or 1404 printer (continuous form only)
1443	1443 printer (Model N1)
2501	2501 card reader
2520	2520 card reader punch
2540-1	2540 card reader punch (read feed)
2540-2	2540 card reader punch (punch feed)
2671	2671 paper tape reader
3066	3066 system console – System/370 Model 165II
3158	3158 display console keyboard – System/370 Model 158

Device Type	Description
	Unit Record Equipment and other Devices (continued)
3210	3210 console printer keyboard
3211	3211 printer
3213	3213 printer
3215	3215 console printer keyboard
3505	3505 card reader
3525	3525 card punch with read feature
	Graphics Devices
2250-1	2250 Model 1 display unit
2250-3	2250 Model 3 display unit
2260-1	2260 Model 1 display station (local attachment)
2260-2	2260 Model 2 display station (local attachment)
2265	2265 display station
3277-1	3277 Model 1 display station
3277-2	3277 Model 2 display station
3284-1	3284 Model 1 printer
3284-2	3284 Model 2 printer
3286-1	3286 Model 1 printer
3286-2	3286 Model 2 printer
	Optical Character Readers
1275	1275 optical reader sorter (available through World Trade branch offices only)
1287	1287 optical reader
1288	1288 optical reader
	Magnetic Character Reader
1419	1419 magnetic character reader
	Audio Response Unit
7770	7770 audio response unit
	Remote Analysis Unit
2955	2955 remote analysis unit

APPENDIX B: DESCRIPTION OF THE STARTER SYSTEM AND DISTRIBUTION LIBRARY TAPES

The starter system and the distribution libraries are distributed on tapes provided by IBM or on tapes that you provide.

The starter system is distributed in a dump/restore format for either a 2314/2319 or 3330 Direct-Access Storage Device.

The distribution libraries are unloaded partitioned data sets that are distributed in the IEBCOPY unload/load format on either one 9-track 1600 BPI tape or two 9-track 800 BPI tapes.

There are 40 files on the distribution library tape. The first file contains the control statements for the IEBCOPY utility program that are used for copying the distribution libraries from tape to disk. Each of the remaining 39 files contains a distribution library. Figure 33 lists the distribution libraries that are on the distribution library tape.

The starter system and the distribution libraries must be on direct-access volumes for Stage I and Stage II. See "Preparation for System Generation" for the procedures for initializing the direct-access volumes for the starter system and distribution libraries, dumping the starter system to a direct-access volume, and copying the distribution libraries to a direct-access volume.

IEBCOPY (file 1)
SYS1.AOS00
SYS1.AOS03
SYS1.AOS04
SYS1.AOS05
SYS1.AOS06
SYS1.AOS07
SYS1.AOS10
SYS1.AOS11
SYS1.AOS12
SYS1.AOS20
SYS1.AOS21
SYS1.AOSB0
SYS1.AOSB3
SYS1.AOSC2
SYS1.AOSC5
SYS1.AOSC6
SYS1.AOSCA
SYS1.AOSCD
SYS1.AOSCE
SYS1.AOSD0
SYS1.AOSD7
SYS1.AOSD8
SYS1.AOSG0
SYS1.AOST3
SYS1.AOST4
SYS1.AOSU0
SYS1.ACMDLIB
SYS1.AGENLIB
SYS1.AHELP
SYS1.ALPALIB
SYS1.AMACLIB
SYS1.AMODGEN
SYS1.APARMLIB
SYS1.APROCLIB
SYS1.ASAMPLIB
SYS1.ATCAMMAC
SYS1.ATSOMAC
SYS1.AUADS
SYS1.AOSA0 (file 40)

Figure 33. Tape distribution of the distribution libraries

APPENDIX C: SYSTEM GENERATION MESSAGES

System generation messages are produced by the assembler program during the expansion of system generation macro instructions. These messages are printed in the assembler listing in the SYSPRINT data set. Messages of three types are produced: error messages, warning messages, and informative messages.

Error Messages

Figure 34 shows the message code and format of system generation error messages. The messages follow.

IEIaaannn text

Explanation: The error indicated by the message text is a coding error in the system generation macro instruction aaa that you coded. The message serial number nnn identifies the message.

For the CHANNEL and IODEVICE macro instructions, the message text begins with either the name field of the macro instruction or, if the name field was omitted, the sequential identification number provided by the system.

Examples of error messages are:

```
5, * * * IEICEN100 MODEL VALUE NOT SPECIFIED
5, * * * IEICHA102 CHANNEL2-ADDRESS VALUE NOT SPECIFIED
5, * * * IEICHA102 CHAN#2 ADDRESS VALUE NOT SPECIFIED
```

The first example illustrates a message for the CENPROCS macro instruction.

The second example illustrates a message for a CHANNEL macro instruction. "CHANNEL 2" is the name field of the macro instruction. The third example illustrates the same message, but in this case the name field of the macro instruction was omitted and "CHAN#2" was supplied by the macro instruction.

System Action: The assembler program did not produce a job stream in the SYSPUNCH data set. The program analyzed all remaining system generation macro instructions and printed any other required messages. Either message IEIGEN113 or IEIGEN116 was printed, followed by the message GENERATION TERMINATED. Then the system generation process was abnormally terminated.

Severity Code: 5

User Response: Correct the error or errors indicated and begin the system generation process from the start of Stage I.

IEIGEN113 QUIT SWITCH PRIOR TO GENERATE MACRO

Explanation: One or more errors, indicated by messages, were detected before the GENERATE macro instruction was expanded.

Severity Code: 7

User Response: Correct the error or errors indicated and begin the system generation process from the start of Stage I.

IEIGEN116 QUIT SWITCH SET IN GENERATE MACRO

Explanation: One or more errors were detected during the expansion of the GENERATE macro instruction.

Severity Code: 7

User Response: Correct the error or errors indicated and begin the system generation process from the start of Stage I.

7, * * * GENERATION TERMINATED * * *

Explanation: The system generation process was abnormally terminated.

Severity Code: 7

User Response: None. This message follows message IEIGEN113 and/or IEIGEN116.

Warning Messages

Figure 34 shows the message code and format of system generation warning messages. The messages follow.

IEIaaannn text

Explanation: The message text indicates a condition in macro instruction aaa that may cause errors in the new system. The message serial number nnn identifies the message. For example:

```
0, * * * IEIGEN940 EDITOR MACRO DEFAULTED
```

Explanation: The EDITOR macro instruction was not specified and the default options were taken.

Severity Code: 0

User Response: None.

0, * * * IEIGEN942 LPA=YES NOT SPECIFIED ON ANY PAGE MACRO -- DEFAULTED TO FIRST PAGE MACRO

Explanation: A device was not specified in a PAGE macro to contain the pageable link pack area.

Severity Code: 0

User Response: None

Informative Messages

*, message text

Explanation: This type of message documents the options selected for the new system through the system generation macro instructions. All options are described, whether the selection was explicit or implicit.

macro namecomponent name***component ID***

Explanation: A message of this type appears before any PUNCH statements that were produced by the macro expansion. The macro name is the name of the macro instruction that produced any succeeding PUNCH statements. The component name is the name of the component group responsbile for maintaining the particular macro expansion. The component ID is the identification to be used in reporting trouble if a problem is isolated to the code produced by the particular macro expansion.

Message Code	**Code**
IEI	s,* * * IEIaaannn text s = Severity code: 0 Warning message; the condition indicated may cause errors in the new system. 5 Error message; user error in coding of a system generation macro instruction. 7 Error message; message is produced by the GENERATE macro instruction. aaa = An abreviation of the system generation macro instruction at which the error was detected: *aaa* *Macro Instruction* CEN CENPROCS CHA CHANNEL CKP CKPTREST CTR CTRLPROG DAT DATAMGT DTS DATASET EDI EDIT EDT EDITOR GEN GENERATE GPH GRAPHICS IOD IODEVICE LDR LOADER LNK LINKLIB

Figure 34 (Part 1 of 2). System generation error and warning messages

Message Code	Code	
	aaa	*Macro Instruction*
	LPA	LPALIB
	MAL	MACLIB
	PAG	PAGE
	RES	RESMODS
	SCH	SCHEDULR
	SCN	SECONSLE
	SVC	SVCTABLE
	TSO	TSO
	UCS	UCS
	UNI	UNITNAME

	nnn = Message serial number
	text = Message text	

Figure 34 (Part 2 of 2). System generation error and warning messages

APPENDIX D: FUNCTIONALLY EQUIVALENT I/O DEVICES

I/O Device	Equivalent to	Usage
2319	2314	The 2319 is a three-drive disk storage unit that attaches to any System/370. The 2319 drives are functionally equivalent to the 2314 drives. As with the 2314, a maximum of eight drives can be used. To use a 2319, specify UNIT=2314 in an IODEVICE macro. A 2319 can be specified as the unitname of the device. To refer to a 2319 as a 2319, specify UNIT=2314 in an IODEVICE macro and NAME=2319 in a UNITNAME macro.
3333	3330	The IBM 3333 Disk Storage and Control is functionally equivalent to the IBM 3330 Disk Storage Drive. To use a 3333, specify UNIT=3330 in an IODEVICE macro. A 3333 can be specified as the unitname of the device. To refer to a 3333 as a 3333, specify UNIT=3330 in an IODEVICE macro and NAME=3333 in a UNITNAME macro.

APPENDIX E: DIAGNOSTIC OVERRIDE

Diagnostic override enables a job stream to be produced during Stage I, even though errors were found in the Stage I input deck. The job stream that is produced may or may not be meaningful, depending on the type of error.

If you want diagnostic override, include the following statements in the Stage I input deck after the Stage I job control language statements:

```
         COPY    SGSYSPAK
&SGMENTB(16)    SETB  1
```

APPENDIX F: CONSOLE SHEET OF THE PROCEDURES FOR STARTING THE STARTER SYSTEM

This console sheet corresponds to the procedures for starting the starter system as described in "Preparing for System Generation." Text in uppercase indicates system commands and responses. Text in lowercase indicates the commands and responses that you must make.

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE 01.0 VS/2
IEA317A  SYS1.PAGE ALLOCATED ON STARTR
IEA332A SPECIFY DUMP OR CANCEL

 IEE140I     SYSLOG      H CMDS             ALL
 IEE140I     01F/009     N ALL      01      1-2
 IEE140I     009/01F     M ALL      02      1-2
 IEE140I     040/060     N INFO     03 Z,A  NONE
 IEE140I     060/0C0     N INFO     04 Z,A  NONE
 IEE140I     0C0/0D0     N INFO     05 Z,A  NONE
 IEE140I     0D0/0E0     N INFO     06 Z,A  NONE
 IEE140I     0E0/0F0     N INFO     07 Z,A  NONE
 IEE140I     0F0/140     N INFO     08 Z,A  NONE
 IEE140I     140/240     N INFO     09 Z,A  NONE
 IEE140I     240/340     N INFO     10 Z,A  NONE
 IEE140I     340/009     N INFO     11 Z,A  NONE
 IEE101A READY
*IEE114A DATE=72.195,CLOCK=08.10.48
*00 IEE114A DATE=72.195,CLOCK=08.10.48,GMT REPLY WITH SET PARM OR U
r 00,'Q=(,F)'
 IEE600I REPLY TO 00 IS;'Q=(,F)'
 IEE118I SET PARAMETER(S) ACCEPTED
*01 IEF423A SPECIFY JOBQUE PARAMETERS
r 1,u
 IEE600I REPLY TO 01 IS;U
mn  jobnames,t
*02 IEC107D E 235,STARTR,MASTER,SCHEDULR,SYS1.SYSVLOGX
 IEE354I SMF PARAMETERS
   PRM=SYSSMF
   ALT=SYSSMF
   SID=H1
   OPI=YES
   JWT=010
   BUF=2000
   MDL=55
   MAN=ALL
   EXT=YES
   OPT=2
*03 IEE357A REPLY WITH SMF VALUES OR U
r 2,u
 IEE600I REPLY TO 02 IS;U
```

```
*04 IEC107D E 235,STARTR,MASTER,SCHEDULR,SYS1.SYSVLOGY
r 3,u
 IEE600I REPLY TO 03 IS;U
 IEE363I SMF SYSSMF DEVICE NOT IN SYSTEM
r 4,u
 IEE351I SMF SYS1.MAN RECORDING NOT BEING USED
 IEE600I REPLY TO 04 IS;U
 IEE041I LOG NOW RECORDING ON SYS1.SYSVLOGX ON 235
s rdr,00c
 IEF403I RDR      STARTED    TIME=08.12.49
s wtr,00E
 IEF403I WTR      STARTED    TIME=08.13.01
s init
 IEF403I  INIT    STARTED    TIME=08.13.12
```

GLOSSARY

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the *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, which was prepared by Subcommittee X3K5 on Terminology and Glossary of American National Standards Committee X3. ANSI definitions are preceded by an asterisk.

ABEND: Abnormal end of task.

AGENLIB: A distribution library that contains the macro definitions of the system generation macro instructions used during Stage I.

AMODGEN: A distribution library that contains the macro definitions of the system generation macro instructions used during Stage II assemblies.

assembler language: A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with the instruction formats and data formats of the computer.

automatic priority group: A group of tasks at a single priority level that are dispatched according to a special algorithm that attempts to provide optimum use of CPU and I/O resources by these tasks.

binary synchronous transmission: Data transmission in which character synchronism is controlled by timing signals generated at the sending and receiving stations.

BLDL table: A list of the track addresses of modules on SYS1.LINKLIB. The purpose of the table is to reduce the time required to find the listed modules on SYS1.LINKLIB.

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 partitioned data set called the procedure library and that can be retrieved by coding the name of the procedure on an execute (EXEC) statement or started by a START command.

communications task: The part of job management responsible for handling communications between the operator console and the system.

complete system generation: The creation of an entirely new System Control Program.

console: That part of a computer used for communication between the operator or maintenance engineer and the computer.

control program: A program that is designed to schedule and supervise the performance of data processing work by a computing system.

control unit: A device that controls the reading, writing, or display of data at one or more input/output devices.

CPU: Central processing unit.

DAT: Dynamic address translation.

data control block (DCB): A control block used by access routines in storing and retrieving data.

data definition statement (DD): A job control statement that describes a data set associated with a particular job step.

data mangement: A major function of the operating system that involves organizing, cataloging, locating, storing, retrieving, and maintaining data.

data organization: The arrangement of information in a data set. For example, sequential organization or partitioned organization.

data set: The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DCB: Data control block.

DD statement: Data definition statement.

ddname: Data definition name.

device type: The general name for a kind of device; for example, 2314, 3330. See also group name, unit address.

direct-access volume initialization: The use of the IBCDASDI or IEHDASDR utility programs for writing a home address, a volume label, and a volume table of contents on a direct-access volume, for checking for defective tracks and assigning alternate tracks for those that are defective, and for writing the IPL program on a new system volume.

display unit: A device which provides a visual representation of data.

distribution libraries: IBM-supplied partitioned data sets on tape containing single VS2 components or combinations of VS2 components that the user restores to disk for subsequent inclusion in a new system.

DSS: Dynamic Support System.

dynamic address translation (DAT): The change of a virtual storage address to a real storage address during execution of an instruction.

dynamic support system (DSS): An interactive debugging facility that allows authorized maintenance personnel to monitor and analyze events and alter data.

EOF: End of file.

EOJ: End of job.

EREP: The environment recording, editing, and printing of a program that makes the data contained on the system recorder file available for further analysis.

ERP: Error recovery procedures.

error recovery procedures (ERP): Standard procedures designed to ensure that all the routines that test particular devices provide a uniform type and quality of information.

execute statement (EXEC): A job control statement that marks the beginning of a job step and identifies the program to be executed or the cataloged or in-stream procedure to be used.

existing VS2 system: A system the user has that is used as a generating system to process Stage I and Stage II. It is also a system that the user has and wants to change to include new or additional I/O devices, additional program options, or updates to include data management and user-written routines.

external page storage: The portion of auxiliary storage that is used to contain pages.

fixed BLDL table: A BLDL table that the user has specified to be fixed in the lower portion of real storage.

fixed link pack area: An extension of the link pack area that occupies fixed pages in the lower portion of real storage.

fixed page: A page in real storage that is not to be paged out.

GPS: Graphic programming services.

graphic programming services (GPS): A number of services provided by the system for use in designing and executing programs that communicate with a user at an IBM 2250 Display Unit, an IBM 2260 Display Unit, or an IBM 2260 Display Station.

group name: A generic name for a collection of I/O devices; for example, DISK or TAPE. See also device type, unit address.

hard-copy: A printed copy of machine output in a visually readable form, for example, printed reports, listings, documents, and summaries.

hard-copy log: In systems with multiple console support or a graphic console, a permanent record of system activity.

hold queue: A waiting list in the SYS1.SYSJOBQE data set for jobs whose initiation is to be delayed until the operator releases them from the queue.

home address: An address written on a direct-access volume, denoting a track's address relative to the beginning of the volume.

IBCDASDI: A program that initializes direct-access volumes, and assigns alternate tracks on direct-access storage volumes.

ICAPRTBL: A program that loads the universal character set (UCS) buffer and forms control buffer (FCB) for an IBM 3211 Printer.

IEBCOPY: A program that copies one or more partitioned data sets or merge partitioned data sets. Specified members of a partitioned data set(s) can be selected for, or excluded from, a copy operation.

IEBEDIT: A program that can create an output data set containing a selection of jobs or job steps. At a later time, the data set can be used as an input stream for job processing.

IEBPTPCH: A program that prints or punches all, or selected portions, of a sequential data set. Records can be printed or punched to meet either standard specifications or user specifications.

IEBUPDTE: A program that incorporates source language modifications into sequential data sets or into partitioned data sets.

IEHDASDR: A program that prepares direct-access volumes for use and ensures that any permanent machine errors (i.e. defective tracks) that are found on a direct-access volume do not seriously degrade the performance of those volumes. Also, this program can dump the entire contents or portions of a direct-access volume onto a volume or volumes of the same direct-access device type, onto a magnetic tape volume or volumes, or onto a system output device.

IEHLIST: A program that can be used to list entries in a catalog, entries in the directory of one or more partitioned data sets, and entries in a volume table of contents.

IEHPROGM: A program that allows the user to modify system control data and maintain data sets at an organizational level. This program can be used to:

- Scratch a data set or member
- Rename a data set or member
- Catalog or uncatalog a data set
- Build or delete an index or an index alias
- Connect or release two volumes
- Build and maintain a generation data group index

IFCDIP00: A program that is used to reinitialize the SYS1.LOGREC system data set in the event it is destroyed.

IFCEREP0: A service aid that edits and writes records that are contained in the SYS1.LOGREC system data set.

***initial program loader (IPL)**: The procedure that causes the initial part of an operating system or other program to be loaded such that the program can then proceed under its own control.

initiator: The function of job management that is responsible for interpreting job control statements, selecting job steps for execution, satisfying resource requests for job steps, initiating job steps, and terminating job steps.

input stream: The sequence of job control statements and data submitted to an operating system on an input unit especially activated for this purpose by the operator. Synonymous with input job stream, job input stream.

input work queue: A queue (waiting list) of job definitions in direct-access storage assigned to a job class and arranged in order of assigned priority. Synonymous with input queue, input job queue, job queue.

installation verification procedure (IVP): A program that tests whether the newly installed System Control Program is operational and supports the installation's machine configuration.

IPL: Initial program loader.

IQADCM00: A program that creates the DSS paging system data set SYS1.DSSVM.

IVP: Installation verification procedure.

I/O device generation: A type of system generation that can be performed against an existing system to add or delete I/O devices or channels, add universal character set support, change I/O device group names, or change console specifications.

job processing: The reading of job control statements and data from an input stream, the initiating of job steps defined in the statements, and the writing of system output messages.

job queue management: The part of job entry central services responsible for maintaining and managing the SYS1.SYSJOBQE data set.

job scheduler: The part of the control program that reads and interprets job definitions, schedules the jobs for processing, initiates and terminates the processing of jobs and job steps, and records job output data.

job stream: In system generation, the output from Stage I, consisting of the expansion of system generation macro instructions into job control language and control statements.

line adapter: An IBM modem which is a feature of a particular device.

*American National Standard Definition

link pack area (LPA): An area of virtual storage containing reenterable routines that are loaded at IPL and can be used concurrently by all tasks in the system.

load module: The output of the linkage editor; a program in a format suitable for loading into virtual storage for execution.

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

LPA: Link pack area.

LSQA: Local system queue area.

machine check handler (MCH): A feature that analyzes errors and attempts recovery by retrying the failing instruction, if possible. If retry is unsuccessful, it attempts to correct the malfunction or to isolate the affected task.

macro library: A library of macro definitions used during macro expansion.

magnetic ink character recognition: The machine recognition of characters printed with magnetic ink.

master console: In a system with multiple consoles, the basic console used for communication between the operator and the system.

master scheduler: A control program routine that responds to operator commands and initiates the requested action.

MCH: Machine check handler.

MICR: Magnetic ink character reader.

***modem:** (MOdulator-DEModulator) A device that modulates and demodulates signals transmitted over communication facilities.

mount attribute: The attribute assigned to a volume that controls when the volume can be demounted; the mount attributes are permanently resident, reserved, and removable.

multiplexor channel: A channel designed to operate with a number of I/O devices simultaneously. Several I/O devices can transfer records at the same time by interleaving bytes of data.

mutually exclusive parameters: Parameters that cannot be coded on the same job control statement.

new system: An entirely new System Control Program that has been generated. Or, an existing system that is changed to include support for new or more I/O devices, additional program options, or updates to data management and user-written routines.

new system volume: The volume that contains system data sets into which modules or combinations of modules are placed during Stage II.

NIP: Nucleus initialization program.

nonpageable dynamic area: An area of virtual storage whose virtual addresses are identical to real addresses; it is used for programs or parts of programs that are not to be paged during execution.

non-switched line: A connection between a remote terminal and a computer that does not have to be established by dialing.

NRZI: Non-return-to-zero-inverted recording.

nucleus initialization program (NIP): The program that initializes the resident control program; it allows the operator to request last minute changes to certain options specified during system generation.

nucleus generation: The creation of a new nucleus for the System Control Program.

***object module:** A module that is the output of an assembler or compiler and is input to a linkage editor.

OBJPDS: A partitioned data set used to store the object modules that are assembled during Stage II of system generation.

***offline:** Pertaining to equipment or devices not under control of the central processing unit.

***online:** Pertaining to equipment or devices under control of the central processing unit.

online test executive program (OLTEP): An operating system facility that schedules and controls activities on the online test system and provides communication with the operator. This program is part of a set of programs that can be used to test I/O devices, control units, and channels concurrently with the execution of programs.

***operating system:** Software which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting,

*American National Standard Definition

compilation, storage assignment, data management, and related services.

output queue: A queue of control information describing system output data sets, which specifies to an output writer the location and disposition of system output.

output stream: Diagnostic messages and other output data issued by the operating system or the processing program on output devices especially activated for this purpose by the operator.

page: (1) A fixed-length block of instructions, data or both, that can be transferred between real and external storage.
(2) To transfer instructions, data, or both between real storage and external page storage.

page data set: A data set in external page storage, in which pages are stored.

paging device: A direct-access storage device on which pages (and possibly other data) are stored.

page frame: A block of real storage that can contain a page.

pageable supervisor area: The area of virtual storage containing supervisor routines that can be paged into and out of real storage.

paging: The process of transferring pages between real and external page storage to assist in allocating real storage among concurrently executing program.

paging supervisor: A part of the supervisor that allocates and releases real storage space (page frames) for pages, and initiates page-in and page-out operations.

partitioned data set: A data set in direct-access storage that is divided into partitions, called members, each of which can contain a program or part of a program. Each partitioned data set contains a directory (or index) that the control program can use to locate a program in the library.

PDS: Partitioned data set directory.

PID: Program Information Department.

point-to-point line: A line that connects a single remote station to the computer; it may be either switched or non-switched.

POR: Problem-oriented routine.

primary paging device: An auxiliary storage device that is used in preference to secondary paging devices for paging operations. Portions of a primary paging device can be used for purposes other than paging operations. The primary paging device contains the link pack area.

program options: Features of the VS2 System Control Program.

qualified name: A data set name that is composed of multiple names separated by periods (e.g., A.B.C.). For a cataloged data set, each name corresponds to an index level in the catalog.

queue: A waiting line or list formed by items in a system waiting for service; for example, tasks to be performed or output to be written by a writer.

reader procedure: A cataloged procedure that controls the input stream reader.

real storage: The storage of System/370 from which the central processing unit can directly obtain instructions and data, and to which it can directly return results.

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

routing code: A code assigned to an operator message and used to route the message to the proper console.

secondary console: In a system with multiple consoles, any console except the master console.

secondary paging device: An auxiliary storage device that is not used for paging operations until the available space on primary paging devices falls below a specified minimum. Portions of a secondary paging device can be used for purposes other than paging operations.

selector channel: A channel designed to operate with only one I/O device at a time. Once the I/O device is selected, a complete record is transferred one byte at a time.

SMF: System management facilities.

SQA: System queue area.

Stage I: A single assembly of user-supplied macro instructions with output consisting of job control language statements and utility control statements for Stage II assemblies, link-edits, and copies.

Stage II: The execution of job control language statements and utility control statements from Stage I to assemble, link-edit, and copy selected modules, specified by the Stage I macro instructions, into the new system.

starter system: An IBM-supplied VS2 system consisting of a control program that supports any of the central processing units needed to perform a system generation, an assembler and linkage editor for Stage I and Stage II, and utility programs for a new system data set and volume initialization and Stage II processing. It is used when there is not an existing VS2 system with which to generate a new system.

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

supervisor: The part of the control program that coordinates the use of resources and maintains the flow of CPU operations.

supervisor call instruction (SVC): An instruction that interrupts the program being executed and passes control to the supervisor so that it can perform a specific service indicated by the instruction.

SVC: Supervisor call instruction.

SVC routine: A control program routine that performs or begins a control program service specified by a supervisor call.

***system**: An organized collection of men, machines, and methods required to accomplish a set of specific functions.

system control program: A group of programs that (1) accept and schedule jobs in a continuous flow (job management); (2) supervise, on a sequential or priority basis, each unit of work to be done (task management); (3) simplify storage, retrieval, and maintenance of data, regardless of the way it is organized and stored (data management).

system data set: A user-allocated data set on a new system volume.

system generation: Obtaining VS2 components on tape, ordering a starter system if there is no existing system, planning the program options for the new system, specifying macros to include the options in the new system including the options by macro execution, executing the macro expansion to generate the new system, and testing the new system.

system initialization: The process of preparing job management for processing, including such functions as locating, cataloging, and formatting system data sets. System initialization is performed by the master scheduler at IPL.

system log: Data sets on which the communication between problem programs, operators, and the system is recorded.

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

system queue area (SQA): A virtual area reserved for system-related control blocks and tables.

system residence volume: The volume that contains the IPL program, the volume index of the SYSCTLG system data set, and the system data sets SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LOGREC. The system residence volume must reside on the I/O device which is addressed when initial program loading is performed.

TCU: Teleprocessing control unit.

teleprocessing control unit (TCU): An input/output control unit that addresses messages to and receives messages from remote terminals.

temporary data set: A data set that is created and deleted in the same job.

time-slicing: A feature that allows each task of specified priority to have control of the CPU for a given interval of time.

transient area: A virtual storage area used for temporary storage of transient routines, such as nonresident SVC or error-handling routines.

UCS: Universal character set.

unit address: The three-character address of a particular device, specified at system generation; for example, 191 or 293.

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

virtual address: An address that refers to virtual storage and must, therefore, be translated into a real storage address when it is used.

***American National Standard Definition**

virtual storage: Addressable space that appears to the user as real storage, from which instructions and data are mapped into real storage locations. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, rather than by the actual number of real storage locations.

volume table of contents (VTOC): A table on a direct-access volume that describes each data set on the volume.

VTOC: Volume table of contents.

writer procedure: A cataloged procedure that controls the output stream writer.

INDEX

Indexes for reference manuals are consolidated in *OS/VS Master Index*, GC28-0602. For additional information about any subject listed below, refer to other publications listed for the same subject in the *Master Index*.

OS/VS2 System Generation Reference
GC26-3792-1

Reader's Comment Form

Your comments about this publication will help us to improve it for you. Comment in the space below, giving specific page and paragraph references whenever possible. All comments become the property of IBM.

Please do not use this form to ask technical questions about IBM systems and programs or to request copies of publications. Rather, direct such questions or requests to your local IBM representative.

If you would like a reply, please provide your name, job title, and business address (including ZIP code).

Fold on two lines, staple, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Fold and Staple

First Class Permit
Number 2078
San Jose, California

Business Reply Mail

No postage necessary if mailed in the U.S.A.

Postage will be paid by:

IBM Corporation
Programming Center – Publishing
Department D58
Monterey and Cottle Roads
San Jose, California 95193

Fold and Staple

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)

GC26-3792-1

OS/VS2 System Generation Reference (File No. S370-34) Printed in U.S.A. GC26-3792-1

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)