GC26-3791-1

Systems

OS/VS1 System Generation Reference

VS1 Release 2

IBM

Second Edition (December 1972)

This edition replaces the previous edition (numbered GC26-3791-0) and makes that edition obsolete.

This edition applies to Release 2 of OS/VS1 and to all subsequent releases unless otherwise indicated in new editions or technical newsletters.

Significant system changes are summarized under "OS/VS1 Summary of Changes" following "How to Use This Book." Each technical change is marked by a vertical line to the left of the change.

Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest *IBM SRL Newsletter,* GN20-0360, that amends *IBM System/360 and System/370 Bibliography,* GA22-6822, to learn which editions and technical newsletters are applicable and current.

Requests for copies of IBM publications should be made to the IBM branch office that serves you.

Forms for readers' comments are provided at the back of the publication. If the forms have been removed, comments may be addressed to IBM Corporation, Programming Center – Publishing, Department D58, Monterey and Cottle Roads, San Jose, California 95114. All comments become the property of IBM.

HOW TO USE THIS BOOK

This publication is intended for system programmers who are to generate an OS/VS1 System Control Program. This publication does not explain the concepts underlying VS1 operations or the options available in VS1. Information of this type is contained in the following publications:

OS/VS1 Planning and Use Guide, GC24-5090

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 seven 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 VS1 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 VS1 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 tapes.

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, "Automatic Device Status Initialization," tells how to include, during system generation, the support that will treat not-ready devices and devices not attached to the system in an offline status.

Appendix G, "Determining the Maximum Number of I/O Devices," gives the formula for determining the maximum number of I/O devices for a system.

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

- DSS (dynamic support system)
- VSAM

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

References

The information contained in the following publications is required to understand and select the options for the VS1 System Control Program to be generated.

OS/VS1 Planning and Use Guide, GC24-5090
OS/VS1 Storage Estimates, GC24-5094
IBM System/370 System Summary, GA22-7001

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

OS/VS Assembler Language, GC33-4010
OS/VS JCL Reference, GC28-0618
OS/VS Utilities, GC35-0005

The following publications are referenced in this manual to provide more detailed information about a particular topic:

OS/VS BTAM, GC27-6980
OS/VS Checkpoint/Restart, GC26-3784
OS/MFT, OS/MVT, and OS/VS1 CRJE Concepts and Facilities, GC30-2012
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/VS1 Debugging Guide, GC24-5093
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, GC26-3803
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: VS1 System Messages, GC38-1001
OS/VS Message Library: VS1 System Codes, GC38-1003
OS/VS1 OLTEP, GC28-0666
Operator's Library: OS/VS Console Configurations, GC38-0120
Operator's Library: OS/VS1 CRJE, GC38-0335
Operator's Library: OS/VS1 Display Consoles, GC38-0255
Operator's Library: OS/VS1 Reference, GC38-0110
Operator's Library: OS/VS TCAM, GC38-0305
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP, GC27-6974
OS/VS1 RES System Programmer's Guide, GC28-6878
OS/VS Service Aids, GC28-0633
OS/VS1 Supervisor Logic, SY24-5155
OS/VS Supervisor Services and Macro Instructions, GC27-6979
OS/VS1 System Data Areas, SY28-0605
OS/VS System Management Facilities (SMF), GC35-0004
OS/VS SYS1.LOGREC Error Recording, GC28-0638
OS/VS Tape Labels, GC26-3795
OS TCAM Concepts and Facilities, GC30-2022
OS/VS TCAM Programmer's Guide, GC30-2034
OS/VS Virtual Storage Access Method (VSAM) Planning Guide, GC26-3799

OS/VS1 SUMMARY OF CHANGES

Release 2

New Devices

Central Processing Units

System/370 Models 155II and 158 – see "Requirements for Generating a New System Control Program," the discussions of the CENPROCS and IODEVICE macro instructions in the chapter "Specifying the New System Control Program," and Appendix A.

Direct-Access Storage Device

3333 Disk Storage and Control – see the discussion of the IODEVICE macro in the chapter "Specifying the New System Control Program," and Appendix A.

Magnetic Tape Units

3410 and 3420 tape subsystem – see "Requirements for Generating a New System Control Program," the discussion of the IODEVICE macro in the chapter "Specifying the New System Control Program," and Appendix A.

Readers and Punches

3505 Reader – see "Requirements for Generating a New System Control Program," the discussion of the IODEVICE macro instructions in the chapter "Specifying the New System Control Program," and Appendix A.

3525 Reader/Punch – see "Requirements for Generating a New System Control Program," the discussion of the IODEVICE macro in the chapter "Specifying the New System Control Program," the discussion of SYS1.IMAGELIB in the chapter "Selecting the New System Data Sets," and Appendix A.

Binary Synchronous Terminals

Refer to the IODEVICE macro instruction in the chapter "Specifying the New System Control Program" for information on specifying the following terminals:

- 2798 Guidance Display Unit
- 3270 Display Station
- 3670 Brokerage Communication Station
- 3735 Programmable Buffered Terminal

Consoles

Refer to the discussions of the IODEVICE, SCHEDULR, and SECONSLE macro instructions in the chapter "Specifying the New System Control Program," and to Appendix A for information on specifying the following consoles:

- 1052 Printer Keyboard
- 3158 Display Console with keyboard – System/370 Model 158
- 3277 Models 1 and 2 Display Station
- 3213 Console-Printer with no keyboard – System/370 Model 158

Telecommunications Control Unit

3705 Communications Control Unit – see the discussion of the IODEVICE macro instruction in the chapter "Specifying the New System Control Program."

Printers

3284 and 3286 as part of the 3270 Display System – see the discussion of the IODEVICE, SCHEDULR, and SECONSLE macro instructions in the chapter "Specifying the New System Control Program," and Appendix A.

New Procedures

Distribution Libraries

The distribution libraries are distributed on two 9-track 800 or 1600 BPI tapes (DLIBT1 and DLIBT2) and are loaded onto either two 2314 volumes or one 3330 volume. Refer to "Preparing for System Generation" for the procedures for copying the distribution libraries from tape to a direct-access volume and to Appendix B for a list of the contents of the distribution library tapes.

Using the Starter System as the Generating System

Using the System/370 Model 135 with the Integrated Communications Adapter (ICA) feature – see "Starter System Special Considerations" in the chapter "Preparing for System Generation."

Refer to "Machine Requirements" in the chapter "Requirements for Generating a New System Control Program" for information on the support for the following new devices in the starter system machine configuration:

- 1052 Console Keyboard
- 3158 Console Keyboard – System/370 Model 158
- 3213 Printer – System/370 Model 158
- 3505 Card Reader
- 3525 Card Punch

Using an Existing VS1 System as the Generating System

If an existing VS1 system that is not a Release 2 level is being used as the generating system, you must include the Release 2 assembler and linkage editor cataloged procedures in your existing system's procedure library (SYS1.PROCLIB). You must also punch the IBCDASDI independent utility and the IPL program from the SYS1.ASAMPLIB distribution library. Refer to "System Generation Using an Existing VS1 System" in the chapter "Preparing for System Generation" for these procedures.

New Programming Features

DEB Validity Checking

Data extent block (DEB) validity checking is a standard feature that prevents a user's data set that is associated with a given DEB from being read or modified, either accidentally or intentionally, by another user program. Refer to the DEBTINC and DEBTSZE parameters in the discussion of the CTRLPROG macro instruction in the chapter "Specifying the New System Control Program" for information on specifying DEB table and to OPTIONS=NODEBCHK in the CTRLPROG macro for information on excluding the input/output supervisor linkage to this feature.

Dynamic Dispatching

Dynamic dispatching is an optional feature that provides for the alteration of the dispatching priorities of selected user tasks as they are being executed. Refer to the DYNINTR and DYNPART parameters in the discussion of the CTRLPROG macro instruction in the chapter "Specifying the New System Control Program."

I/O Load Balancing

I/O load balancing is an optional feature that allocates temporary data sets to devices in such a way as to equalize the amount of I/O contention on each device. Refer to OPTIONS= IOLOADBAL in the discussion of the SCHEDULR macro instruction in the chapter "Specifying the New System Control Program."

Fetch Protect

The fetch protect optional feature provides security for your data by preventing anyone from examining the contents of anyone else's area of storage. Refer to the SECURTY parameter in the discussion of the CTRLPROG macro instruction in the chapter "Specifying the New System Control Program.

Greenwich Mean Time

The Greenwich Mean Time (GMT) optional feature provides a time of day (TOD) clock that is independent of local time. Refer to the TZ parameter in the discussion of the CTRLPROG macro instruction in the chapter "Specifying the New System Control Program."

3270 Display System

Refer to the IODEVICE, SCHEDULR, and SECONSLE macro instructions in the chapter "Specifying the New System Control Program" for information for specifying the 3277, 3284, and 3286 as part of the 3270 Display System.

For information on specifying programmed-function-keyboard (PFK) command entry and light-pen command entry for the 3277, refer to the discussions of the IODEVICE, SCHEDULR, and SECONSLE macro instructions and to the discussion of SYS1.DCMLIB in the chapter "Selecting the New System Data Sets."

Specification Changes

The option for specifying the type of I/O request queuing has been added for the 2305 – see the discussion of the IODEVICE macro instruction in the chapter "Specifying the New System Control Program."

The option for specifying ordered queuing for the 3330 has been added – see the discussion of the IODEVICE macro instruction in the chapter "Specifying the New System Control Program."

Programmed-function-keyboard (PFK) command entry can be specified for the 2250 and 3158 display units – see the discussions in the IODEVICE, SCHEDULR, and SECONSLE macro instructions in the chapter "Specifying the New System Control Program" and to the discussion of SYS1.DCMLIB in the chapter "Selecting the New System Data Sets."

The default value for the loader option of searching the link pack area to resolve external reference has been changed from RES to NORES – see the discussion of the LOADER macro instruction in the chapter "Specifying the New System Control Program."

The option for specifying more than two device types has been added to the PAGE macro instruction – see the discussion of the PAGE macro instruction in the chapter "Specifying the New System Control Program."

User-written routines may be added to or deleted from SYS1.LINKLIB during a nucleus generation and user-written routines may be changed or deleted during an I/O device generation – see the discussion of the LINKLIB macro instruction in the chapter "Specifying the New System Control Program."

User-written routines can be added to or deleted from SYS1.SVCLIB during a nucleus generation – see the discussion of the SVCLIB macro instruction in the chapter "Specifying the New System Control Program."

Automatic Device Status Initialization (SMARTNIP) can be specified during system generation to treat specified devices in an offline status – see Appendix F.

Diagnostic Override enables a job stream to be produced during Stage I even though there are errors in the Stage I input deck – see Appendix E.

The option can be specified for writing operator commands, system commands, and responses, and status displays to the hardcopy log – see the HARDCPY parameter in the discussion of the SCHEDULR macro in the chapter "Specifying the New System Control Program."

The unit address of an integrated operator console must be specified if the new system's primary console is not a 3158, 3210, or 3215. If the primary console is not one of these, the unit address of one of these consoles must be specified as the unit address of an integrated operator console – see the IOC parameter in the discussion of the SCHEDULR macro instruction in the chapter "Specifying the New System Control Program."

The maximum number of I/O devices for a system could be less than 768 – see Appendix G for the formula for determining this maximum value.

Three object module utility data sets (OBJPDS1, OBJPDS2, and OBJPDS3) are required for Stage II processing – see the discussion of the GENERATE macro instruction in the chapter "Specifying the New System Control Program" and to "Stage II: Processing the Job Stream" in the chapter "System Control Program Installation."

A jobclass for Stage II output can be specified – see the discussion of the GENERATE macro instruction in the chapter "Specifying the New System Control Program."

An output class for Stage II output can be specified – see the discussion of the GENERATE macro instruction in the chapter "Specifying the New System Control Program."

Miscellaneous Technical Changes

The Release 1 version of this manual erroneously stated that the value you specify in the BUFSIZE parameter of the JES macro is a decimal number from 436 to 9999. The value that can be specified is a decimal number from 436 to 99999. This error has been corrected in this release.

CONTENTS

FIGURES

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 OS/VS1 System Control Program and received the starter system tape and/or distribution library tapes, but before the actual system generation.

The System Generation Process

System generation is a process that creates an OS/VS1 System Control Program tailored to both the data processing requirements and machine configuration of an installation.

The VS1 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 VS1 System Control Program according to the specifications in the macro instructions (see Figure 1). The new VS1 System Control Program is composed of the standard programs incorporated into every System Control Program and optional programs selected from the distribution libraries. In addition, your own routines can be included in the new VS1 System Control Program.

A VS1 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 VS1 System Control Program.

Types of System Generation

There are three types of system generation: a complete System Control Program (system) generation, a nucleus generation, and an I/O device generation. You specify the type of generation 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 VS1 System Control Program. For a nucleus or I/O device generation, the job stream is used to modify an existing VS1 system.

Complete System Generation

This is the generation of a complete VS1 System Control Program. It is done if you are installing a VS1 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 configurations 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.NUCLEUS, SYS1.PARMLIB, SYS1.SVCLIB, and SYS1.LINKLIB are the system data sets that are modified. (These system data sets are discussed in the chapter "Selecting the New System Data Sets.")

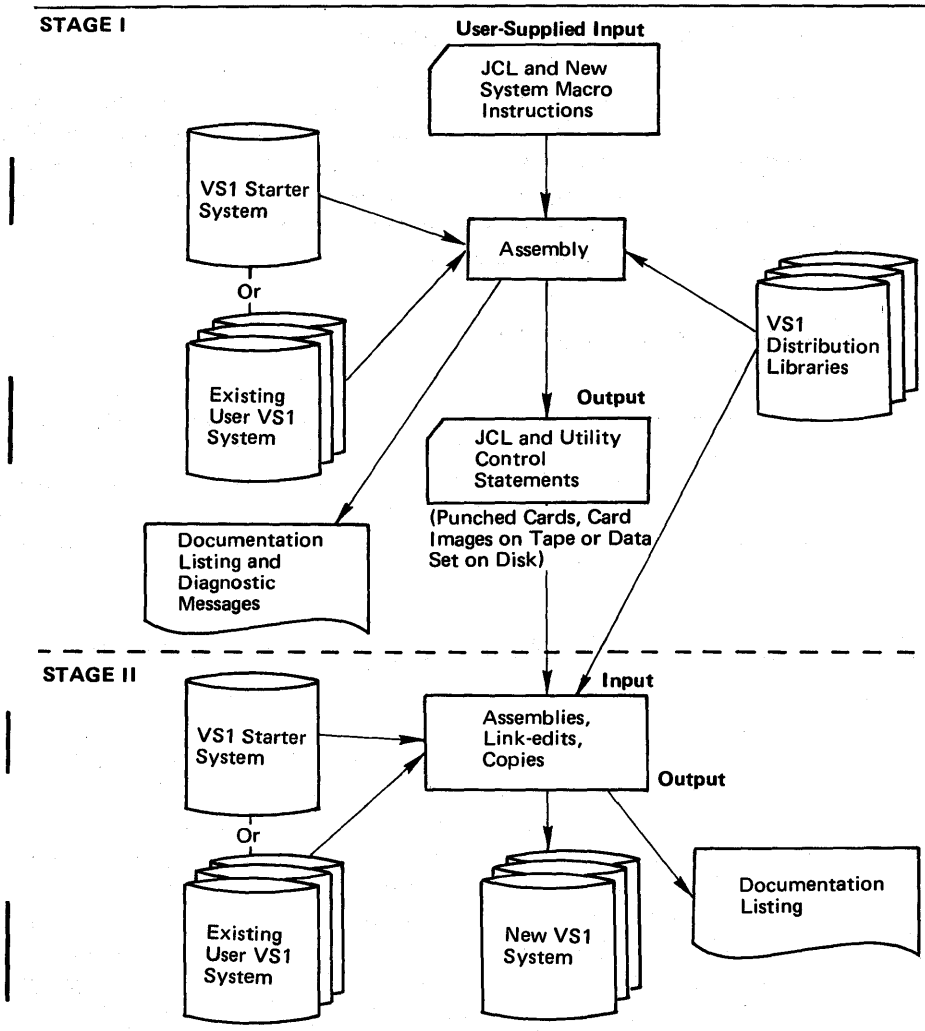


Figure 1. The system generation process. System generation macro instructions are assembled and expanded to form a job stream (Stage I). This job stream assembles, link-edits, and copies selected modules from the distribution libraries and user-supplied modules to the new system volumes to form the new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

I/O Device Generation

An I/O device generation is done if 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, add additional access methods, or create a new nucleus. For an I/O device generation, the system being modified cannot be the generating system. In an I/O device generation, SYS1.IMAGELIB, SYS1.NUCLEUS, SYS1.LINKLIB, SYS1.LOGREC, SYS1.PARMLIB, and SYS1.SVCLIB are the system data sets that are modified. (These system data sets are described in the chapter "Selecting the New System Data Sets.")

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 and how to code them 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 also discussed with an example that may be helpful to refer to.

Allocating Space to the Utility Data Sets Required for Stage II

Three utility data sets are required for Stage II processing and you must allocate space to them before Stage II processing begins. If you are using the IEHPROGM utility program to allocate space to the system data sets and catalog them in the system catalog, you would probably want to allocate space to the utility data sets 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 the DATASET macro to allocate space to the system data sets, you can code the IEHPROGM control statements for allocating space to the utility data sets when you code the job control language statements that are 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 VS1 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 (SYSUT1, SYSUT2, and SYSUT3) and Stage II (OBJPDS1, OBJPDS2, and OBJPDS3).

Restarting System Generation

Coding errors, inadequate space allocation, 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 VS1 System Control Program, you can test it 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 VS1 system using either an existing VS1 system as the generating system or a starter system that is provided by IBM as the generating system. This chapter discusses the real storage requirements for system generation, the machine requirements for performing a system generation using the starter system as the generating system, and the programming requirements for performing a system generation using either an existing VS1 system or a starter system as the generating system.

Real Storage Requirements

To execute system generation using a 3330 starter system, 144K bytes of real storage is required. To execute system generation using a 2314 starter system, 144K bytes of real storage is required or, by loading the second nucleus, system generation can execute in less than 144K.

Machine Requirements

You can perform a system generation using one of the following System/370 central processing units:

- Model 135
- Model 145
- Model 155II
- Model 158

Figure 2 shows the maximum machine configuration allowed for a VS1 system generation using a 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 a 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 discussion on 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 like any other job. The system control program used as the generating system is an existing VS1 system. If this is your first VS1 system generation, you can use a starter system that contains the programs necessary to perform a system generation.

Using an Existing System as the Generating System

For an existing VS1 system to be used as the generating system, it must contain a VS1 level of the following:

- IEBCOPY utility program
- IEBEDIT utility program
- IEBUPDTE utility program

- IEHDASDR utility program
- IEHIOSUP utility program
- IEHLIST utility program
- IEHPROGM utility program
- IFCDIP00 service aid program
- IQADCM00 utility program (for DSS)
- Linkage Editor
- OS/VS Assembler

These programs are described 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."

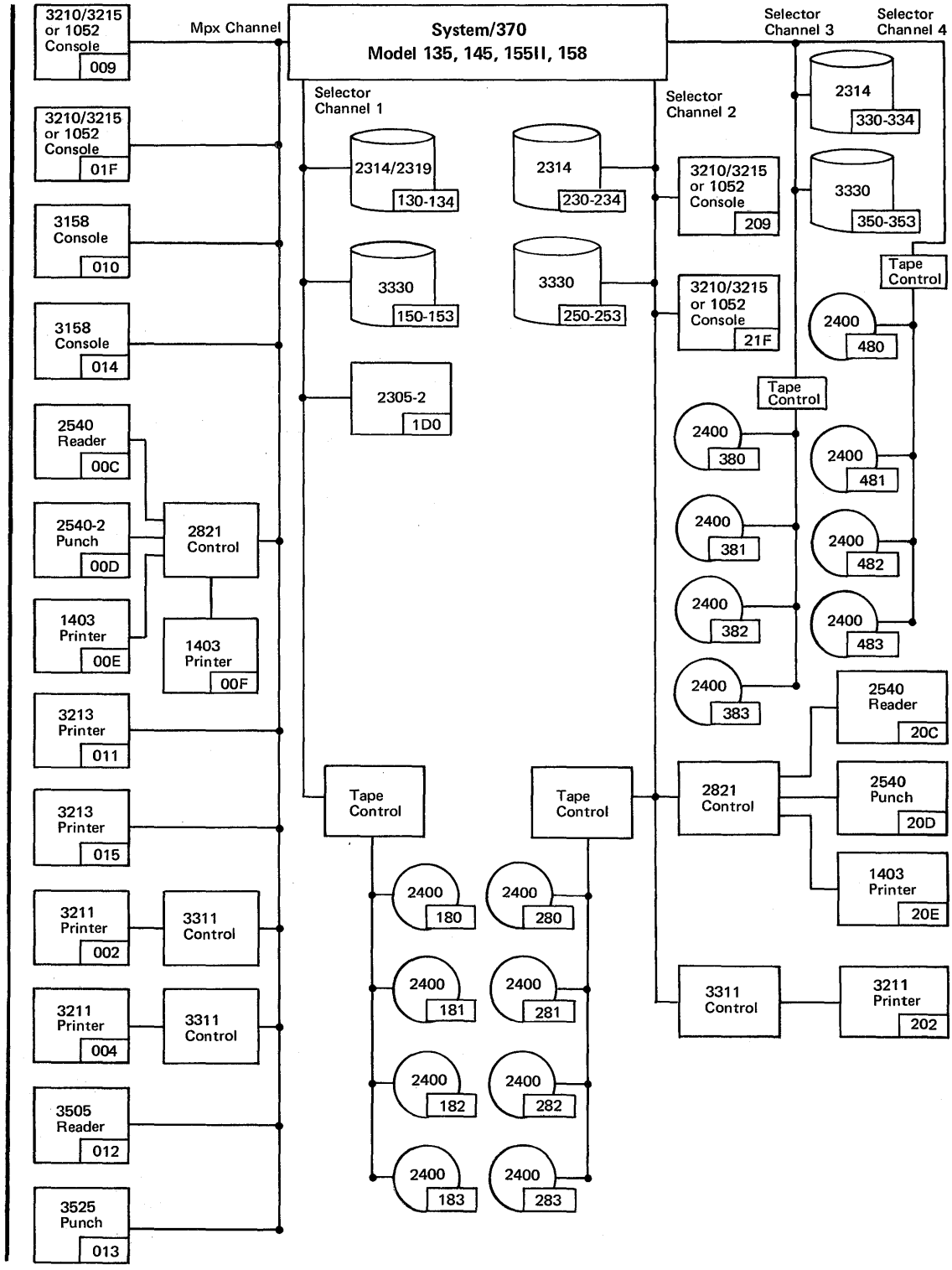


Figure 2. Maximum machine configuration for using the starter system

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		
		3158	010, 014				
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		202		
		1403	00E, 00F		20E		
		3213[3]	011, 015				
		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	2305-2		1D0			
		2314 (or 2319 on channel 1)		130, 131, 132, 133, 134	230, 231, 232, 233, 234	330, 331, 332, 333, 334	
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353	
Starter System	1	2314 (or 2319 on channel 1)		130, 131, 132, 133, 134	230, 231, 232, 233, 234	330, 331, 332, 333, 334	
		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 2305-2, 2314/2319, or 3330.

[2]The three utility data sets required for Stage I and the three object-module utility data sets required for Stage II should not require an additional direct-access device.

[3]Console printer for the 3158.

Figure 3 (Part 1 of 2). Minimum I/O device requirements for performing a system generation using a 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
Distribution Libraries	2 or	2314s (or 2319s on channel 1)		130, 131, 132, 133, 134	230, 231, 232, 233, 234	330, 331, 332, 333, 334	
	1	3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353	
Six Utility Data Sets[2]	1	2305-2		1D0			
		2314 (or 2319 on channel 1)		130, 131, 132, 133, 134	230, 231, 232, 233, 234	330, 331 332, 333, 334	
		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 2305-2, 2314/2319, or 3330.

[2]The three utility data sets required for Stage I and the three object-module utility data sets required for Stage II should not require an additional direct-access device.

[3]Console printer for the 3158.

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

Magnetic Tape Units[1]	
2400	2400 series 9-track magnetic tape unit
2400-2	2400 series 7-track magnetic tape unit with data conversion
Direct-Access Devices	
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)
3158	3158 console keyboard for the System/370 model 158
3210	3210 console printer keyboard
3211	3211 printer
3213	3213 printer for the System/370 model 158
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*: 180 280 380 480 / 181 281 381 481 / 182 282 382 482 / 183 283 383 483 *2305-2*: 1D0 *2314/2319*: 130 230 330 / 131 231 331 / 132 232 332 / 133 233 333 / 134 234 334 *3330*: 150 250 350 / 151 251 351 / 152 252 352 / 153 253 353
SYSDA	Direct-access devices at any of the following addresses: *2305-2*: 1D0 *2314/2319*: 130 230 330 / 131 231 331 / 132 232 332 / 133 233 333 / 134 234 334 *3330*: 150 250 350 / 151 251 351 / 152 252 352 / 153 253 353

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 the *OS/VS Assembler Language* manual.

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 to 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:

- The system generation macro instructions are discussed alphabetically.
- Letters in boldtype, numbers, and punctuation marks must be coded exactly as shown.
- Exceptions to this convention are brackets, [ ]; braces, { }; ellipses, . . . ; and subscripts. These 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 use it 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 of 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.

Several parameters, such as UNIT=name in the IODEVICE macro instruction, require the unit name of a device. The unit name of a device can be one of the following:

- Unit address, such as 192
- Device type, such as 2314
- Group name, or name of a collection of devices, such as TAPE or SYSDA. (See UNITNAME macro instruction in this chapter.)

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	required	required	required
	EDITOR	optional	--	--
	JES	optional	--	--
	LOADER	optional	--	--
	MACLIB	optional	--	--
	PAGE[1]	required	required	required
	PARTITNS	required	required	required
	SCHEDULR	required	required	required
	SECONSLE[1]	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]
	RESMODS	optional	optional[2]	--
	SVCLIB	optional	optional[2]	--
	SVCTABLE[1,4]	optional	optional	optional
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 VS1 system generation.

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

[4] This macro instruction is required under certain conditions. Refer to the SVCTABLE 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 generation* or *I/O device generation,* the CENPROCS macro instruction must be respecified with the same parameters and subparameters that were specified in the last complete system generation.

```
CENPROCS    [INSTSET={STD  }]
                     {UNIV }

            MODEL={135R}
                  {145R}
                  {155R}
                  {158R}

            [SECMODS={ALL                }]
                     {(model[,model] . . .)}
```

Parameter	Subparameter	Explanation
INSTSET=		specifies the instruction set available in the central processing unit.
	STD	specifies the standard instruction set with decimal instructions and the storage protection feature.
	UNIV	specifies the universal instruction set. This is the standard set with the decimal instructions, the storage protection feature, and floating point.
MODEL=	**135R** **145R** **155R** **158R**	specifies the model number of the central processing unit.
SECMODS=		specifies the additional System/370 models for which environmental recording, editing, and printing (EREP) support will be included in the new system. If this parameter is omitted, no secondary models will be supported.

Parameter	Subparameter	Explanation
SECMODS= (continued)	**ALL**	specifies that all supported CPU models are to be included.
	model	specifies the additional CPU models that are to be included.

Example: This example specifies a model 145 central processing unit and additional support for model 135.

```
CPU     CENPROCS     MODEL=145R,SECMODS=(135R)
```

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 in your computing system.

For a *nucleus generation*, the CHANNEL macro instructions must be respecified with the same parameters and subparameters that were specified 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** | **MULTIPLEXOR** | **SELECTOR** }

Parameter	Subparameter	Explanation
ADDRESS=	*address*	specifies the number 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 blocked multiplexor channel that is integrated with a central processing unit. An address must be specified for each block multiplexor channel.
	MULTIPLEXOR	The channel is a multiplexor channel integrated with the central processing unit.
	SELECTOR	The channel is a selector channel integrated with the central processing unit. 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: VS1 System Codes.*

CKPTREST **[ELIGIBLE=(***code***[,***code***] . . .)]**
[NOTELIG=(*code***[,***code***] . . .)]**

Parameter	Subparameter	Explanation
ELIGIBLE=	*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

Required for:
complete
nucleus
I/O device

The CTRLPROG macro instruction is used to specify control program options.

For a *nucleus generation,* the options specified in the DEBTINC, DEBTSZE, DYNPART, DYNINTR, FETCH, RESIDNT, SYSQUE, TMSLICE, and TRACE parameters can be added to or deleted from the system or changed from the values specified in the last complete system generation. (If RESIDNT=TRSVC is specified, then OPTIONS=TRSVCTABLE must also be specified.) All other parameters must be respecified exactly as they were in the last complete system generation.

For an *I/O device generation,* the values specified in the MAXIO and SYSQUE parameters can be changed. All other parameters and subparameters must be respecified exactly as they were in the last complete system generation.

For information on control program options, refer to *OS/VS1 Planning and Use Guide.* For storage estimates, refer to *OS/VS1 Storage Estimates.*

```
CTRLPROG    [ALTSYS=address]

            [ASCII={INCLUDE }]
                   {EXCLUDE}

            [DEBTINC={size}]
                     {80  }

            [DEBTSZE={size}]
                     {80  }

            [DYNPART=(Pn–Pm)] [1]

            [DYNINTR=(a,b,c,d)] [1]

            [FETCH={PCI }]
                   {STD}

            [MAXIO=number]

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

            [OVERLAY={ADVANCED}]
                     {BASIC   }

            [RESIDNT=(value[,value] . . . )]

            [SECURTY={FPROT   }]
                     {NOFPROT}
```

[1] This parameter may be overriden by the operator at IPL.

```
CTRLPROG    [SYSQUE={size}]
(continued)         {4   }

            [TMSLICE=(Pn – Pm,SLC – time)] [1]

            [TRACE={number}]
                   {0     }

            [TZ=({W},hh[,mm])] [1]
                {E}

            [VIRTUAL={size  }] [1]
                     {1024  }

            [VSAM={EXCLUDE }]
                  {INCLUDE }
```

[1]This parameter may be overridden by the operator at IPL.

Parameter	Subparameter	Explanation
ALTSYS=	*address*	specifies the alternate system residence device unit address to be used by the dynamic device reconfiguration (DDR) system residence program. For information on DDR, refer to the *OS/VS1 Planning and Use Guide.* This parameter must be specified unless OPTIONS=NODDRSYS is specified.
ASCII=		specifies the inclusion or exclusion of the ASCII translation routine.
	INCLUDE	specifies that the routine will be included in the system as part of the pageable supervisor area.
	EXCLUDE	specifies that the routine will not be included in the operating system. The default is EXCLUDE. If you plan on using the ASCII utilities, this parameter must be specified.
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, 80 is used. If this parameter is specified and OPTIONS=NODEBCHK is also specified, then this parameter is ignored.
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, 80 is used.

Parameter	Subparameter	Explanation
DEBTSZE= (continued)	*size* **80** (continued)	If this parameter is specified and OPTIONS=NODEBCHK is also specified, then this parameter is ignored. For information on DEB validity checking, refer to *OS/VS1 Storage Estimates.*
DYNPART=	**(P***n***–P***m***)**	specifies the partition or groups of partitions for which dynamic dispatching will be used in the execution of selected tasks. (The partitions are specified in the PARTITNS macro.) The dispatching group must consist of contiguous partitions. If dynamic dispatching is specified, time-slicing cannot be specified. For information on dynamic dispatching, refer to *OS/VS1 Planning and Use Guide.* Uppercase letters, parentheses, and hyphens must be coded as shown. **P***n* is a number that specifies the lowest partition in a dispatching group. **P***m* is a number that specifies the highest partition in a dispatching group.
DYNINTR=	**(***a,b,c,d***)**	specifies the priority level for a group of tasks in a dynamic dispatching group in an attempt to provide optimum use of CPU and I/O resources by the tasks in the group. If this parameter is specified, then the DYNPART parameter must also be specified. For information on dynamic dispatching, refer to *OS/VS1 Planning and Use Guide.* *a* is a decimal number from 1 to 99 that represents, in milliseconds, the delta to be added to or subtracted from the time-slice value at the end of each statistics interval. *b* is a decimal number from 1 to 999 that represents, in milliseconds, the lower bound of a time-slice that may be given to a task. *c* is a decimal number, specified in hundreds, for the ratio of CPU to I/O-bound tasks. *d* is a decimal number from 0 to 99999 that represents, in milliseconds, the length of the statistics interval.
FETCH=		specifies the type of program fetch to be used by the new VS1 system.
	PCI	specifies the use of program controlled interruption while a program is being fetched into storage. For information on PCI, refer to *OS/VS Data Management for System Programmers.*

Parameter	Subparameter	Explanation
FETCH= (continued)	**STD**	specifies the standard program fetch. The default value is STD.
MAXIO=	*number*	specifies the maximum number of I/O operations that can be simultaneously processed by the new VS1 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 system. Refer to the *OS/VS1 Planning and Use Guide* for further information.
	BLDL	specifies a BLDL table to be fixed in real storage. If this option is not specified, the BLDL table will be pageable. For information on the BLDL table, refer to *OS/VS Data Management Services Guide.*
	NODDR	specifies that dynamic device reconfiguration (DDR) is not to be included in the new VS1 system. If NODDR is specified, NODDRSYS should also be specified. If NODDRSYS is not specified, it will be assumed. DDR allows you to move a demountable volume from one device to another and to reposition it, if necessary, without abnormally terminating the job or having to IPL again. A request to move a volume may be initiated by either the system or the operator (see *Operator's Library: OS/VS1 Reference*). If your new system is to have the DDR feature, you should not code specific unit addresses in programs that are to be processed on the system. If the EXCP macro instruction is used to address magnetic tape drives in a program that has DDR, REPOS=Y or N must be coded in the DCB macro instruction to indicate whether an accurate block count is being maintained. For information on the EXCP macro, refer to *OS/VS Data Management for System Programmers.*
	DDRNSL	specifies that DDR for magnetic tape volumes with nonstandard labels is to be included in the new VS1 system. If DDRNSL and NODDR are both specified, NODDR is ignored.

Parameter	Subparameter	Explanation
OPTIONS= (continued)	**DDRNSL** (continued)	If you have nonstandard magnetic tape labels and plan to use DDR, this subparameter must be included. A nonstandard label routine with the name NSLREPOS must be supplied. This routine can be added either during system generation using the SVCLIB macro instruction or it can be link-edited into SYS1.SVCLIB after system generation. (For additional information, refer to *OS/VS Tape Labels*.)
	NODDRSYS	specifies that DDR for the system residence volume containing SVCLIB will be excluded.
	NODAV	specifies exclusion of the code to verify the volume serial number of any 2314, 2319, or 3330 volumes mounted after initial program load (IPL) that have not been specified by a system mount request. If NODAV is specified, NODDR must also be specified. (Volume serial number verification is required for DDR.)
	NODEBCHK	specifies that the input/output supervisor (IOS) linkage to the DEB validity checking feature will not be included in the new system. If this subparameter value is not specified, then full DEB validity checking is assumed. (DEB validity checking is specified in the DEBTINC and DEBTSZE parameters.) For additional information, refer to *OS/VS1 Planning and Use Guide.*
	RDE	specifies inclusion of the reliability data extractor feature. For information on RDE, refer to *OS/VS SYS1. LOGREC Error Recording.*
	RER	specifies that the reduced error recovery procedures for magnetic tape are to be used if requested in the OPTCD parameter of a data definition (DD) statement or in the DCB macro instruction. If this subparameter is not specified, all requests for reduced error recovery will be ignored. For information on error recovery procedures, refer to *OS/VS1 Planning and Use Guide.*
	TRSVCTBL	specifies that a table containing the relative track addresses of all transient supervisor calls (SVCs) is to be stored in the resident portion of the control program. TRSVCTBL must be specified if TRSVC is specified in the RESIDNT parameter of the CTRLPROG macro or if SVCs are made resident in the pageable supervisor area. If type 3 and type 4 SVC routines will be made resident, this subparameter must be specified. (For additional information, refer to *OS/VS1 Storage Estimates.*) During a nucleus generation, this option can be added to or deleted from the options specified during a complete system generation. TRSVCTBL must be specified if TRSVC is specified in the RESIDNT parameter of the CTRLPROG macro instruction.

Parameters	Subparameters	Explanation
OVERLAY=		specifies the overlay supervisor options. (For information on overlay structure, refer to *OS/VS1 Supervisor Logic.*)
	ADVANCED	specifies synchronous overlay with error checking for invalid SEGWT macro instructions.
	BASIC	specifies synchronous overlay without exclusive call checking. The default is BASIC.
RESIDNT=		specifies which of the following types of modules, normally brought into the pageable area each time they are used, are to be made resident in the pageable portion of the supervisor. Detailed information on this parameter, making modules resident in the pageable portion of the supervisor, and on the standard lists in SYS1.PARMLIB can be found in *OS/VS1 Planning and Use Guide.*
	ACSMETH	specifies that access method modules are to be included in the pageable portion of the supervisor when the system is loaded. The names of the access method modules will also be included in the IEAIGG00 list in SYS1.PARMLIB. You must specify this parameter value if the system log function is to be included in the new system (see SCHEDULR macro). This parameter value must also be specified if the checkpoint/restart facility is to be included in the new system. For information on checkpoint/restart, refer to: *OS/VS1 Planning and Use Guide* *OS/VS Checkpoint/Restart*
	ERP	specifies that any error-recovery procedures are to be made resident in the pageable supervisor when the system is loaded. For performance reasons, it is recommended that RESIDNT=ERP be specified for the 1287 and 1288 optical readers. If you specify ERP at system generation, then after system generation, you can update the IEAIGE01 list in SYS1.PARMLIB to include the names of any ERP routines you want resident. For additional information, refer to *OS/VS1 Planning and Use Guide* and *OS/VS1 Storage Estimates.*
	RENTCODE	specifies that reenterable load modules from SYS1.LINKLIB are to be made resident in the pageable portion of the supervisor when the system is loaded.

Parameters	Subparameters	Explanation
RESIDNT= (continued)	**TRSVC**	specifies that type 3 and type 4 SVC routines are to be made resident in the pageable portion of the supervisor when the system is loaded. The names of these modules will also be included in the IEARSV00 and IEARSV01 lists in SYS1.PARMLIB. If this subparameter is specified, then OPTIONS=TRSVCTBL must also be specified. If it is not specified, it is assumed. For a nucleus generation, this option can be added to or deleted from the options specified in a complete system generation. If this option is to be included, OPTIONS=TRSVCTBL must also be specified.
SECURTY=		specifies whether or not partitions are to be fetch-protected.
	FPROT	indicates that all partitions are to be fetch-protected.
	NOFPROT	indicates that no partitions are to be fetch-protected. The default value is NOFPROT.
SYSQUE=	*size* **4**	specifies the size of the system queue area (SQA) that is required to initialize the new VS1 system. The minimum size that may be requested is 4 (which will result in an allocation of 4K). (You code the parameter value to which the system appends a K.) If the value specified is not even, the number specified will be rounded to the next higher even number. This value can be changed during a nucleus or I/O device generation. After the system has been initialized, additional SQA space is dynamically allocated as needed. The default is 4. For information on SQA, refer to *OS/VS1 Storage Estimates.*
TMSLICE=		specifies time-slicing for the jobs scheduled to be executed in a group of consecutive partitions. Uppercase letters and hyphens are coded as shown.
	P*n*–**P***m*	*n* specifies the lowest partition number in a group of consecutive partitions to be included in a time-sliced group. *m* specifies the highest partition number in a group of consecutive partitions to be included in a time-sliced group. The values of *n* and *m* must be a valid subset of the numbers specified in the PARTITNS macro instruction.

Parameters	Subparameters	Explanation
TMSLICE= (continued)	**SLC** – *time*	*time* specifies, as a decimal integer from 20 to 9999, the maximum number of milliseconds for which each ready task in the group of partitions is to have control during one cycle through the group. This value can be changed after IPL. If TMSLICE is not specified, no jobs will be time-sliced. Time-slicing can be added to or deleted from the system during a nucleus generation. If time-slicing is specified, dynamic dispatching cannot be specified. For further information on time-slicing, refer to: *OS/VS1 Planning and Use Guide* *OS/VS1 Storage Estimates*
TRACE=	*number* **0**	specifies the number of system trace table entries to be maintained. The default is 0. A tracing routine aids in debugging and maintenance by storing in the trace table information pertaining to start I/O (SIO) commands, supervisor (SVC) interruptions, dispatcher interruptions, and I/O interruptions. When the table has been completely filled, 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 can be found in the following VS publications: *OS/VS1 Planning and Use Guide* *OS/VS1 Storage Estimates* The trace option and the generalized trace facility (GTF) component of the control program can coexist in the same system. The trace option, however, is inhibited while GTF is active. For information on GTF, refer to the discussion on the PARTITNS macro instruction in this section. In a nucleus generation, entries can be added to or deleted from the trace table.
TZ=		specifies the time zone deviation from Greenwich Mean Time (GMT) which will result in local standard time being produced.
	W **E**	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.

Parameters	Subparameters	Explanation
TZ= (continued)	*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 59. If *mm* is omitted, 0 is used. If the TZ parameter is not specified, a GMT offset of zero is created. This will result in local time equaling the time in the TOD clock (which the system assumes to be Greenwich Mean Time).
VIRTUAL=	*size* **1024**	specifies the maximum size of virtual storage. You specify the number of 1024-byte (1K) blocks to be made available. The value is expressed as an even decimal number without the K. If the value specified is not even, it is rounded to the next highest even number. This value may be changed at IPL. The maximum virtual storage size for VS1 is 16384K (16 megabytes). The default is 1024 (1,048,576) or 16384 1K blocks if a value greater than 16384 is specified. If VSAM is to be included in the system, then at least 2048 must be specified. For additional information on the size of virtual storage, refer to *OS/VS1 Storage Estimates.*
VSAM=	**EXCLUDE** **INCLUDE**	specifies the inclusion or exclusion of VSAM (virtual storage access method). If VSAM is to be included in the new system, at least 2 megabytes (2048K bytes) of virtual storage must be specified in the VIRTUAL parameter of this macro. If less than 2 megabytes is specified, VSAM will not be included in the new system. The default is INCLUDE. For additional information on VSAM, refer to *OS/VS Virtual Storage Access Method (VSAM) Planning Guide.*

Example: This macro specifies the options of a fixed BLDL table, the exclusion of dynamic device reconfiguration, reduced error recovery for tape, and a transient SVC table. Access method modules, reenterable modules, and type 3 and 4 SVCs are to be made part of the nucleus. This macro also specifies advanced overlay, program controlled interrupt (PCI) fetch, a system queue area of 12K bytes, a virtual storage size of 2048K bytes, a maximum of 36 simultaneous input-output requests, a trace table with 500 entries, and the inclusion of VSAM.

```
CTRLPROG   CTRLPROG  OPTIONS=(BLDL,NODDR,NODDRSYS,RER,TRSVCTBL),    X
                     RESIDNT=(ACSMETH,RENTCODE,TRSVC),             X
                     OVERLAY=ADVANCED,                             X
                     FETCH=PCI,                                    X
                     SYSQUE=12,                                    X
                     VIRTUAL=2048,                                 X
                     MAXIO=36,                                     X
                     TRACE=500,                                    X
                     VSAM=INCLUDE
```

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/VS1 Storage Estimates

OS/VS BTAM

OS TCAM Concepts and Facilities

The following standard access methods are always included in the operating 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 for this generation. The same parameters 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 for this generation. The same parameters must be coded.

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 index sequential access method

Parameter	Subparameter	Explanation
ACSMETH= (continued)	*method* (continued)	ACSMETH=TCAM must be specified if the TCAM independent release is going to be included in the System Control Program. The inclusion of VSAM (virtual storage access method) in the new system is specified by the VSAM parameter of the CTRLPROG macro.

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. All of the system data sets that are specified using the DATASET macro are cataloged except SYSCTLG, which contains the system catalog.

Each system data set to be included in the new system must be cataloged (except SYSCTLG) and have space allocated to it. You can use a DATASET macro to do this – one DATASET macro for each new system data set – or use the IEHPROGM utility program (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"). You need not use either method exclusively. The only requirement is that the DATASET macro *must* be used 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 been cataloged. Omission of the SPACE parameter in the DATASET macro will prevent the allocation of space for a system data set.

The information needed to allocate space to the page data sets is specified in the PAGE macro instruction. Refer to the explanation on specifying the PAGE macro in this section.

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

For a *nucleus generation*, if SYS1.LINKLIB and/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*, you must respecify a DATASET macro for any of the following system data sets if they are not on the system residence volume: SYS1.DCMLIB, SYS1.IMAGELIB, SYS1.LINKLIB, or SYS1.PARMLIB. 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 in the DATASET macro 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 to be initialized. You cannot specify DCB information for the system data sets to be initialized 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*

[SPACE=({**CYL** / **TRK** / *blksize*}**,***quantity*)**]**

[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: ACCT BRODCAST DCMLIB DSSVM DUMP IMAGELIB LINKLIB MACLIB MANX MANY NUCLEUS PARMLIB PROCLIB RMTMAC SAMPLIB SVCLIB SYSCTLG SYSJOBQE SYSPOOL TELCMLIB UADS The following system data sets must reside on the system residence volume: NUCLEUS LOGREC[1] SVCLIB SYSCTLG The following system data sets must exist during system generation and if they are not specified, will be assumed to be preallocated on the system residence volume (SYSRES): DSSVM LINKLIB MACLIB NUCLEUS PARMLIB PROCLIB SAMPLIB SVCLIB SYSCTLG IMAGELIB must exist in the system to be generated if the UCS macro instruction is specified.

[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)		TELCMLIB must exist in the system to be generated if BTAM or TCAM is specified in the DATAMGT macro instruction. RMTMAC must exist in the system to be generated if OPTIONS=REMOTE is specifed in the SCHEDULR macro instruction. If you are generating a system with RES (OPTIONS=REMOTE in the SCHEDULR macro), then UADS and BRODCAST should have space allocated to them and they should be cataloged in the system catalog before IPL.
SPACE=		specifies a request that space allocation be based solely on the values given in this subparameter. Estimates for space allocation for the system data sets are in *OS/VS1 Storage Estimates.*
	CYL **TRK** ***blksize***	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 data sets: LINKLIB MACLIB PROCLIB RMTMAC SAMPLIB SVCLIB SYSCTLG TELCMLIB UADS

Parameters	Subparameters	Explanation
SPACE= (continued)	*quantity* (continued)	*directory blocks* specifies the number of 256-byte blocks to be allocated for the directory of a partitioned data set. This must be specified for the following system data sets: DCMLIB IMAGELIB LINKLIB MACLIB NUCLEUS PARMLIB PROCLIB RMTMAC SAMPLIB SVCLIB TELCMLIB UADS The amount of space for the system data set SYS1.LOGREC is always calculated and allocated on the system residence volume for a complete or I/O device generation. For improved system efficiency it is recommended that the following system data sets begin on a cylinder boundary: ACCT LINKLIB MACLIB PROCLIB SVCLIB SYSJOBQE SYSPOOL TELCMLIB If the DATASET macro is used to allocate space to SYS1.DSSVM, the following must be specified in the SPACE parameter: SPACE=(2048,(600))
VOL=		defines the volume serial number and the device type of the volume for the system data set specified.
	serial **SYSRES**	specifies the serial number of the volume. SYSRES is the default value unless the RESVOL parameter in the GENERATE macro instruction specifies another value.
	type **2314**	specifies the device type of the volume. The device types that may be coded are: 2305-2 2314 3330

Parameter	Subparameter	Explanation
VOL= (continued)	*type* **2314** (continued)	The VOL parameter works in conjunction with the RESVOL parameter in the GENERATE macro instruction. 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. More than one volume may be specified for ACCT and SYSPOOL, but the volumes must be specified in separate DATASET macro instructions. If either serial number or device type is specified, both must be specified. The default for the VOL parameter is SYSRES,2314.

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))
```

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*} | **192,64**)]

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 **192**	is a value from 64 to 9999. It specifies the amount of 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 192 is assumed, resulting in 192K bytes.
	size2 **64**	is a value from 6 to 100. It specifies the 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 64 is assumed, resulting in 64K bytes.

Example: This macro specifies that 256K bytes of virtual storage will be available to the linkage editor and 64K 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 volume serial number and device type of the system residence volume of the system to be generated, the output class and job class used during system generation, the type of generation being done, and the partitioned data sets required for Stage II assemblies. This macro instruction must be specified last.

You must allocate space for the partitioned data sets that you specify in the GENERATE macro. Refer to "Stage II Input" in the chapter "System Control Program Installation" for this information.

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 specified in 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. OBJPDS1 and OBJPDS2 are the object module partitioned data sets used during a nucleus generation.

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    }

            [OBJPDS1={SYS1.name     }]
                     {SYS1.OBJPDS1}

            [OBJPDS2={SYS1.name     }]
                     {SYS1.OBJPDS2}

            [OBJPDS3={SYS1.name     }]
                     {SYS1.OBJPDS3}
```

GENERATE (continued)

```
[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. 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 to 9 that identifies the member name (IEANUC0n) of the existing nucleus that will be modified. A new nucleus cannot be created during an I/O device configuration.
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. The default is SYS1.
JCLASS=	*class* **A**	specifies the jobclass (A-O) to be used for output from Stage II of system generation. The default is A.
OBJPDS1=	**SYS1.***name* **SYS1.OBJPDS1**	specifies one of three partitioned data sets 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. The default is SYS1.OBJPDS1.

Parameter	Subparameter	Explanation
OBJPDS2=	**SYS1.*name*** **SYS1.OBJPDS2**	specifies one of three partitioned data sets 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. The default is SYS1.OBJPDS2.
OBJPDS3=	**SYS1.*name*** **SYS1.OBJPDS3**	specifies one of three partitioned data sets 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. The default is SYS1.OBJPDS3.
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. The default is A.
RESVOL=		specifies the volume serial number and device type of the new system residence volume to be generated.
	serial **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 2305-2, 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 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. Default values are used for the three partitioned data sets that are required for Stage II execution.

```
GEN   GENERATE   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 on GPS, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Station* and *OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment).*

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. The default value is EXCLUDE.
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. The default value is INCLUDE.

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 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. This number could be less, based upon the size of the unit control blocks (UCBs) and the number of request queue elements (RQEs) in the system to be generated (see Appendix G for additional information).

Figure 7 shows the valid combinations of values for the UNIT, MODEL, and FEATURE parameters.

Figure 8 shows the values that can be specified for I/O devices in the FEATURE parameter.

Figure 9 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, 2402, 2403, 2415, 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 system generation.

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

           [DEVTYPE=type]

           [ERRTAB=nnn]

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

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

The validity of the following keyword parameters depends on the type of unit specified. See Figure 7 for valid keywords and parameters.

ADAPTER=*adapter*	
[EXPBFR={*number*}**]** {**4096**}	(2250-3 only)
[FEATURE=(*feature*[,*feature*] . . .**)]**	
GCU={**2848-1**} {**2848-2**} {**2848-21**} {**2848-22**}	(2260-2 only)
[NUMSECT={*number*}**]** {**16**}	(2250-3 only)
[MODEL=*model***]**	
OBRCNT=*number*	(BSC1, BSC2, BSC3 only for 2715 as part of the 2790 terminal system)
[OPTCHAN=(*address*[,*address*] . . .**)]**	(2305, 2314, 3330, 2401, 2402, 2403, 2404, 2420, 3420, or DUMMY only)
PCU=*number*	(2250-3 only)
SETADDR=*value*	(TCU=2702 only)
TCU={**2701**} {**2702**} {**2703**}	(Telecommunications devices)

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 telecommunication lines. Figure 9 associates terminal control adapters with transmission control units.
ADDRESS=		specifies the unit addresses of devices or telecommunications 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 number 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=(190,5) is specified, the unit addresses 190, 191, 192, 193, and 194 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 or 3333 in one IODEVICE macro is 8. (See Appendix D for a description of the 3333.) This parameter value is ignored for the 2305-2.
DEVTYPE=	*type*	specifies any additional characteristics of the device. This value specified must be eight hexadecimal characters. This parameter need not be specified for any IBM-supported device. This parameter must be specified if UNIT=DUMMY is specified. For further information about this parameter value, refer to the *OS/VS1 Planning and Use Guide,* and to the description of the UCB in *OS/VS1 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.SVCLIB.

Parameter	Subparameter	Explanation
ERRTAB= (continued)	*nnn* (continued)	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 (or a 2250 model 3 attached to a 2840) that use EXPRESS attention handling routines. The value may be an integer from 1 to 8192. If this parameter is omitted, 4096 bytes are used. For information on the 2250, refer to *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit.*
FEATURE=	*feature*	specifies the optional features that the device has. These values can be written in any order. Refer to Figure 7 for the valid features that may be specified for the devices and to Figure 8 for a summary of features that may be specified.
GCU=	**2848-1** **2848-2** **2848-21** **2848-22**	specifies the type of graphic control unit 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 Display Station (Local Attachment).*
IOREQUE=		specifies the type of I/O request queuing to be provided by the supervisor for the device.
	PRIORITY	specifies queuing according to task priority.
	ORDERED	specifies queuing according to cylinder address. IOREQUE=ORDERED may be specified only for the 2314/2319 or 3330. If it is specified for any other device, FIFO is substituted.
	FIFO	specifies first-in-first-out queuing. The default is FIFO.
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 7).
NUMSECT=	*number* **16**	specifies the number of 256-byte buffer sections in the 2840 display control unit to be guaranteed available to the 2250 model 3. These buffer sections can be used only by the device being specified. If the 2250 model 3 is specified as an operator console and this parameter is omitted 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 Display Unit.*

Parameter	Subparameter	Explanation
OBRCNT=	*number* **800**	is a number from 0 to 800 that specifies the number of area stations connected to one 2715 transmission control in a 2790 data communications system. The total number of area stations cannot exceed 800. If 800 is exceeded, 800 is used.
OPTCHAN=	*address*	specifies the alternate channels through which any of the following devices may be addressed: 2305, 2314, 3330, 2401, 2402, 2403, 2404, 2420, 3420, or DUMMY (a device not supported by IBM). The value of the address subparameter must be greater than the high-order digit of the value in the ADDRESS parameter of this macro instruction. The address of a selector 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).
PCU=	*number*	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

Parameter	Subparameter	Explanation
PCU= (continued)	*number* (continued)	4D3. A 2250 cannot have a unit address that is within the range of addresses of other 2250s. For example, 122, 124 cannot be assigned to any 2250s if 121, 125, and 126 are addresses of other 2250s.
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 1 2 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 9 associates terminal control adapters with transmission control units.
UNIT=	*device type*	specifies the device type. Figure 7 lists and defines the devices that may be specified. (Device types are described also in Appendix A.) In the case of telecommunications lines, the UNIT parameter specifies the device type that is connected to a telecommunications line, or the type of binary synchronous line configuration.
		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 drives (see ADDRESS= in this section). To use a 2319, specify UNIT=2314. Appendix D contains a description of the 2319.
	the following	The 2596 card read punch is functionally equivalent to the 1442 N1 card read punch. If a 2596 is being used, the following must be specified: UNIT=1442; ADDRESS=(2596 device address); and MODEL=N1. The FEATURE=CARDIMAGE parameter may not be specified. Appendix D contains a description of the 2596.
		The IBM 3333 Disk Storage and Control is functionally equivalent to the IBM 3330 Disk Storage Device. To use a 3333, specify UNIT=3330. Appendix D contains a description of the 3333.

Parameter	Subparameter	Explanation
UNIT= (continued)	**DUMMY**	specifies a device that is not supported by IBM. If 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.

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

```
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 is used for I/O request queuing.

```
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 used for I/O request queuing.

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

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Direct-Access Devices				
2314		SHARED	**IOREQUE=** {**PRIORITY** / **ORDERED** / **FIFO**} **OPTCHAN=** {*n* / (*n*,*n*[,*n*])}	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=** {**PRIORITY** / **ORDERED** / **FIFO**} **OPTCHAN=** {*n* / (*n*,*n*[,*n*])}	Disk Storage Drive FEATURE=SHARED and the OPTCHAN parameter are mutually exclusive for the device. 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	2	SHARED	**IOREQUE=** {**PRIORITY** / **FIFO**} **OPTCHAN=** *n*	Fixed Head Disk Storage FEATURE=SHARED and the OPTCHAN parameter are mutually exclusive for the device.
Display Devices				
1053	4		**IOREQUE=** {**PRIORITY** / **FIFO**}	Printer
2250	1	ABSLTVEC ALKYB2250 {BUFFER4K / BUFFER8K} CHARGNTR DESIGNFEAT LIGHTPEN PRGMKYBD	**IOREQUE=** {**PRIORITY** / **FIFO**} **EXPBFR=***n*	Display unit Refer to the "Unit Record" part of this figure for specifying this device as an operator's console.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

Figure 7 (Part 1 of 10). 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=** {**PRIORITY** {**FIFO** **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=** {**PRIORITY** {**FIFO** **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=** {**PRIORITY** {**FIFO** **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 If the 78-key keyboard is not specified, a 66-key keyboard is assumed. 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 {FRCHAR {GRCHAR {UKCHAR {DOCHAR		Display unit for the 3270 Display System 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.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

Figure 7 (Part 2 of 10). 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.)	AUDALRM MAGCDRD NUMLOCK SELPEN {EBKY3277 / ASKY3277 / DEKY3277 / OCKY3277} {KB70KEY / KB78KEY / KB81KEY}		Only one type of keyboard may be specified. If no keyboards are specified, a 66-key keyboard is assumed. The AUDALRM feature can only be specified if a keyboard has been specified.
3284	1 or 2	{FRCHAR / GRCHAR / KACHAR / UKCHAR / <u>DOCHAR</u>}		Printer for the 3277 console
3286	1 or 2	{FRCHAR / GRCHAR / KACHAR / UKCHAR / <u>DOCHAR</u>}		Printer for the 3277 console
Magnetic Tape Units				
2401 or 2402	1, 2, 3	READWRITE MDECOMPAT {7-TRACK / <u>9-TRACK</u>} DATACONV	**IOREQUE=** {**PRIORITY** / **<u>FIFO</u>**} **OPTCHAN=***n*	Magnetic tape unit FEATURE=9-TRACK is assumed if the FEATURE parameter is omitted. Each device of the 2402 Magnetic Tape Unit must be uniquely addressed in an IODEVICE macro instruction.
	4,5,6	READWRITE 9-TRACK DUALDENS		
2401	8	7-TRACK READWRITE	**IOREQUE=** {**PRIORITY** / **<u>FIFO</u>**} **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.

[1]Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

Figure 7 (Part 3 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Magnetic Tape Units (continued)				
2403 or 2404	1,2,3	MDECOMPAT {7-TRACK / 9-TRACK} DATACONV	**IOREQUE=** {**PRIORITY** / **FIFO**} **OPTCHAN=***n*	Magnetic tape unit FEATURE=READWRITE is implicit for the 2404 and must not be specified.
2403	4,5,6	9-TRACK DUALDENS		Magnetic tape unit
2415	1,2,3	{7-TRACK / 9-TRACK} DATACONV	**IOREQUE=** {**PRIORITY** / **FIFO**}	Magnetic tape unit Each device of the 2415 Magnetic Tape Unit must be uniquely addressed in an IODEVICE macro instruction. FEATURE=9-TRACK is assumed if the FEATURE parameter is omitted.
	4,5,6	{7-TRACK / 9-TRACK} DUALDENS DATACONV		
2420			**IOREQUE=** {**PRIORITY** / **FIFO**} **OPTCHAN=***n*	Magnetic tape unit This is a 9-track 1600-BPI drive only, so the FEATURE parameter is not required.
2495			**IOREQUE=** {**PRIORITY** / **FIFO**}	Magnetic tape cartridge reader
3410	1,2,3	{7-TRACK / 9-TRACK} DUALDENS	**IOREQUE=** {**PRIORITY** / **FIFO**}	Magnetic tape unit
3420	3,5,7	{7-TRACK / 9-TRACK} DUALDENS SHARABLE	**IOREQUE=** {**PRIORITY** / **FIFO**} **OPTCHAN=***n*	Magnetic tape unit
Optical Character Readers				
1275 or 1287 or 1288			**IOREQUE=** {**PRIORITY** / **FIFO**}	The ADDRESS parameter must specify the number of the primary control unit; the number must be even. There must be only one IODEVICE macro instruction for each 1275. Two unit addresses

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

Figure 7 (Part 4 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Optical Character Readers (continued)				
1275 or 1287 or 1288 (cont.)				are generated by the one IODEVICE macro instruction. The 1275 is available through IBM World Trade Corporation branch offices.
Magnetic Ink Character Readers				
1419			IOREQUE={PRIORITY / FIFO}	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 instruction for each 1419. Two addresses are generated by the one IODEVICE macro instruction.
Unit Record Devices[2]				
1052	7		IOREQUE={PRIORITY / FIFO}	Printer-keyboard Installed in combination with the 2150 Control Unit.
1403	N1 or N2	UNVCHSET	IOREQUE={PRIORITY / FIFO}	Printer
	7			UNVCHSET is invalid for the 1403-7.
1442	N1	CARDIMAGE	IOREQUE={PRIORITY / FIFO}	Card reader punch
	N2			Card punch only
1443	N1	SELCHSET 24ADDPOS	IOREQUE={PRIORITY / FIFO}	Printer
2250	1	ALKYB2250 {BUFFER4K / BUFFER8K} CHARGNTR LIGHTPEN PRGMKYBD	IOREQUE={PRIORITY / FIFO}	LIGHTPEN and PRGMKYBD are optional; other features are required for use as a console device.

[1]Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter
[2]Figure 10 lists all of the devices that can be used as consoles.

Figure 7 (Part 5 of 10). 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=**{**PRIORITY** **FIFO**} **NUMSECT=***n* **EXPBFR=***n* **PCU=***n*	For use of the 2250 as a console device the ALKYB2250 is required. For use as a console device, if NUMSECT is omitted or less than 16, a value of 16 is used. The PCU parameter is required.
2260	1	LINEADDR ALKYB2260 NODESCUR	**IOREQUE=**{**PRIORITY** **FIFO**}	NODESCUR is optional; other features are required for use of the 2260 as a console device.
2501	B1 or B2	CARDIMAGE	**IOREQUE=**{**PRIORITY** **FIFO**}	Card reader
2520	B1	CARDIMAGE	**IOREQUE=**{**PRIORITY** **FIFO**}	Card reader punch
	B2 or B3			Card punch only
2540R or 2540P	1	CARDIMAGE	**IOREQUE=**{**PRIORITY** **FIFO**}	Card reader punch The 2540R and 2540P are specified for the same 2540 Card Reader Punch. Two IODEVICE macro instructions must be specified.
2671	1		**IOREQUE=**{**PRIORITY** **FIFO**}	Paper tape reader
3158		{EBKY3277 ASKY3277 DEKY3277 OCKY3277} SELPEN KB78KEY		Display console with keyboard — System/370 Model 158. If the 78-key keyboard is not specified, a 66-key keyboard is assumed.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.
[2] Figure 10 lists all of the devices that can be used as consoles.

Figure 7 (Part 6 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Unit Record Device[2] (continued)				
3210 or 3215			IOREQUE= {PRIORITY / FIFO}	Console-printer keyboard
3211			IOREQUE= {PRIORITY / FIFO}	High-speed printer The universal character set is a standard feature.
3213			IOREQUE= {PRIORITY / FIFO}	Console-printer with no keyboard — System/370 Model 155II
3277	2	{ASKY3277 / DEKY3277 / EBKY3277 / OCKY3277} {KB70KEY / KB78KEY / KB81KEY} SELPEN		If the 3277 is being used as a console device with input capability, a keyboard must be specified. If no keyboard is specified, a 66-key keyboard is assumed. The 3277 Model 1 can be used only as an output-only console to display operator messages; optional features or parameters cannot be specified.
3505		CARDIMAGE	IOREQUE= {PRIORITY / FIFO}	Card reader and control unit
3525		CARDIMAGE {TWOLINE / MULTILINE}	IOREQUE= {PRIORITY / FIFO}	Card punch Every 3525 specified must be attached to a 3505.
Telecommunications[3]				
1030		AUTOPOLL	IOREQUE= {PRIORITY / FIFO}	Data collection system
1050		AUTOANSR AUTOCALL AUTOPOLL	SETADDR= *value*	Data communication system AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

[2] Figure 10 lists all of the devices that can be used as consoles.

[3] Figure 9 lists the terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device (UNIT) to a transmission control unit (TCU).

Figure 7 (Part 7 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Telecommunications[3] (continued)				
1050X		AUTOCALL AUTOANSR	**IOREQUE=** {**PRIORITY** | **FIFO**} **SETADDR=** *value*	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 OPTIONS=MSC in the SCHEDULR macro is specified and the 2740 is specified as a console, the 2740 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 2740C or 2740X must be specified if 2740 devices are to be used for CRJE. CHECKING is required.
2740X		CHECKING AUTOCALL AUTOANSR		Communications terminal with PTTC code 2740C or 2740X must be specified if 2740 devices are to be used for CRJE. CHECKING is required.

[1]Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

[3]Figure 9 lists the terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device (UNIT) to a transmission control unit (TCU).

Figure 7 (Part 8 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Telecommunications[3] (continued)				
2741C or 2741P		AUTOANSR	**IOREQUE=** {**PRIORITY** / **FIFO**} **SETADDR=** *value*	Communications terminal with correspondence code or communications terminal with PTTC code
115A				Western Union[6] Terminal
83B3				AT&T[7] Selective Calling Terminal
TWX		AUTOANSR AUTOCALL		Teletype[8] Models 33 or 35
WTTA				IBM World Trade Corporation Telegraph Terminal
BSC1		DUALCODE[5] DUALCOMM	**IOREQUE=** {**PRIORITY** / **FIFO**} **OBRCNT=***n*	BSC[4] station nonswitched point-to-point line.
BSC2		DUALCODE[5] DUALCOMM AUTOANSR AUTOCALL		BSC[4] station switched point-to-point line.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

[3] Figure 9 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
2798 Guidance Display Unit
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.

[6] Trademark of Western Union Telegraph Company.

[7] Trademark of American Telephone and Telegraph Company.

[8] Trademark of the Teletype Corporation.

Figure 7 (Part 9 of 10). Parameter values that may be specified in an IODEVICE macro instruction

UNIT	MODEL	FEATURE[1] (optional)	Optional Parameters	Notes
Telecommunications[3] (continued)				
BSC3		DUALCODE[5] DUALCOMM AUTOPOLL	IOREQUE= {PRIORITY {FIFO OBRCNT=*n*	BSC[4] station nonswitched multipoint line. AUTOPOLL is used.
7770	3		IOREQUE= {PRIORITY {FIFO	Audio response unit
2955			IOREQUE= {PRIORITY {FIFO	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.

[1] Figure 8 lists and defines all of the options that can be specified through the FEATURE parameter.

[3] Figure 9 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
- 2798 Guidance Display Unit
- 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 7 (Part 10 of 10). 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 hardware storage for display regeneration.

Figure 8 (Part 1 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
CARDIMAGE	1442 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 2402 2403 2404 2415	The data conversion feature allows the writing and reading of binary data on 7-track 2400 and 3400 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	United States English 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 2402 2403	The dual density feature allows a program to utilize the tape unit as either an 800 BPI or a 1600 BPI machine.

Figure 8 (Part 2 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
DUALDENS (cont.)	2404 2415 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.
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 if ASKY3277, EBKY3277, or OKCY3277 is specified.
KB81KEY	3277	81-key keyboard on the 3277.
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 2402 2403 2404	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.

Figure 8 (Part 3 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
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	3277 3158	This feature specifies a 78-key operator console keyboard on the 3277 or 3158.
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 2402	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 English character generator.

Figure 8 (Part 4 of 5). Values that can be specified in the FEATURE parameter of an IODEVICE macro instruction

Feature	Unit	Description
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.
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 2402 2403 2404 2415 3410 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.

Figure 8 (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	2701	Synchronous data adapter, type II, and an appropriate modem
	BSC3	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 2702 2703	Telegraph adapter, type II, and an appropriate modem 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 9. Terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device to a transmission control unit (TCU)

JES

Optional for:
complete

Not applicable for:
nucleus
I/O device

The JES macro instruction is used to specify the job entry subsystem requirements in the new VS1 system. This macro instruction is optional. If it is not specified, the default values are assumed.

The parameters specified in the JES macro are reproduced as a member in SYS1.PARMLIB named JESPARMS. JESPARMS also contains the ALCUNIT parameter value. (The value for the ALCUNIT parameter is placed in JESPARMS automatically during system generation. You cannot specify a value for ALCUNIT in the JES macro.) The values in JESPARMS may be changed after system generation by using the IEBUPDTE utility program.

For information on the job entry subsystem (JES) and on the ALCUNIT parameter, refer to *OS/VS1 Planning and Use Guide.* For storage estimates, refer to *OS/VS1 Storage Estimates.*

```
JES     [BUFSIZE={number}]
               {880   }

        [JOUTLIM={number}]
                 {0     }

        [NUMBUFS={number}]
                 {7     }

        [RDR=(R={r},Y={y},B={b})]
              {1}   {5}   {0}

        [SPOLCAP={number}]
                 {80    }

        [SPOLVOL=(serial[,serial] ...)]

        [STEPWTP={number}]
                 {15    }

        [SWDSLMT={number}]
                 {15    }

        [WTLRCDS={number}]
                 {100   }

        [WTR=(W={w},U={u},Z={z},B={b})]
              {1}   {0}   {6}   {0}
```

Parameter	Subparameter	Explanation
BUFSIZE=	*number* **880**	specifies the size of the required buffers in the JES central buffer pool. *Number* is a decimal number from 436 to 99999. This value can be calculated by using the following formula: n*(r+4)+16 where: n is the number of records in the buffer r is the user record size If this parameter is omitted, 880 is used. The maximum size is limited to the smallest track capacity of all the SYS1.SYSPOOL devices. If a specified buffer size is too large, the system will assume the smallest track capacity. If the buffer size is too small, the default value of 880 is used. Buffer size may not be changed during system restart.
JOUTLIM=	*number* **0**	specifies the maximum number of output records. The number must be less than or equal to 16,777,215. When specified (and not zero), this parameter allows you to specify a limit on the maximum number of logical records to be included in the output data set. The specified or defaulted value can be overridden for a specific output data set by specifying the OUTLIM parameter on the DD card defining the output data set. If this parameter is omitted, 0 is assumed. Your output limit exit (IEFUSO) is provided only if SMF=BASIC or FULL is specified in the SCHEDULR macro instruction.
NUMBUFS=	*number* **7**	is a decimal number from 7 to 999 that specifies the number of buffers in the central buffer pool. The number of buffers may be calculated with the following formula: 3r + w + (d * p) = number where: r is the maximum number of readers w is the maximum number of writers d is the number of concurrently open spooled data sets in a partition p is the maximum number of partitions If this parameter is omitted, 7 is used.
RDR=		specifies the maximum number of JES readers that may run concurrently. It is used to calculate the amount of work area that must be reserved for use by the readers. All subparameters must be coded if this parameter is specified.

Parameter	Subparameter	Explanation
RDR= (continued)	**R=***r*	specifies the maximum number of readers.
	1	If this parameter is omitted, 1 is used.
		For a system with RES, the value specified in this subparameter must include the number of remote readers that will be active at the same time.
	Y=*y*	specifies the control interval to be used for unit record input devices (for example, the 2540 card reader).
	5	If this parameter is omitted, 5 is used.
		The control interval determines the speed at which a unit record device will be driven; it is the number of records per buffer. The assumed control interval values (5 for readers, 6 for writers) will drive these devices at near their rated speeds. Because of additional physical I/O operations, a unit record reader will not approach its rated speed when the reader is processing a job card. A smaller control interval will reduce buffer requirements but will degrade performance for unit record devices.
	B=*b*	specifies the sum of the blocksizes of all procedure libraries and input streams, other than unit record input streams, that may be processed concurrently. The value of B is equal to:
	0	b0 + b1 + . . . bn
		where b represents the blocksize of one such input stream or procedure library. The n represents the total number of these blocksizes that may be processed concurrently. For example, two readers are to be active concurrently. One reader is to read a unit record input stream, the other is to read a tape input stream blocked at 1600 bytes, and both can read a procedure library blocked at 800 bytes. The value of B is equal to:
		(1600+800+800)=3200
		If this parameter is omitted, 0 is used.
SPOLCAP=	*number*	is the percentage, expressed in decimal from 40 to 90, of spool capacity that, when reached, will cause the operator to be informed.
	80	If this parameter is omitted, 80 is assumed.
SPOLVOL=	*serial*	specifies the volume serial number(s) of the volume(s) on which the SPOOL data set(s) reside.

Parameter	Subparameter	Explanation
SPOLVOL= (continued)	*serial* (continued)	If this parameter is omitted, the serial number of the system residence volume is used. A maximum of ten volumes may be specified. The volume serial number is only valid for 2305-2, 2314, and 3330 volumes. The definition of spool volumes may be changed at IPL by specifying the SPOOL=CHNG as a SET parameter and responding to the message IEE545A which will follow. (For information on the SET parameter see the *OS/VS Message Library: VS1 System Messages.*)
STEPWTP=	*number* **15**	is a number from 15 to 9999 that specifies the number of messages that the write-to-rogrammer (WTP) routine can issue during a job step. If this parameter is omitted, 15 is used. If the step limit has been exceeded, 15 additional requests for system WTP messages are allowed. No additional processing-program WTP requests are honored.
SWDSLMT=	*number* **15**	specifies the number of 176-byte records to be reserved to end a job if it would overflow its scheduler work area data set. Number is a decimal number from 8 to 99, calculated as follows: $6 + \frac{2*k}{11} + \frac{j}{44} + \frac{a}{4} + (2*b) - (2*a)$ where: k is the maximum number of DD statements per step j is the maximum number of devices required in the step a is the total number of generation data groups (GDG) used during a job b is the total number of generations in all the GDGs in the job. Include in calculations all cataloged and in-stream procedures used. Round fractions to the next highest integer. If this parameter is omitted, 15 is used. The default does not include GDGs. The system operator can change this number whenever a START INIT command is being processed.
WTLRCDS=	*number* **100**	specifies the number of WTL macros and/or LOG commands that will be used before the log data set that is being recorded to is closed and the alternate data set is opened. The number is the number of records from 100 to 9999.

Parameter	Subparameter	Explanation
WTLRCDS= (continued)	*number* **100** (continued)	If this parameter is omitted, 100 is used. If HARDCPY=SYSLOG is specified in the SCHEDULR macro instruction, the minimum specification should be 1000 records. If HARDCPY=SYSLOG is not specified in the SCHEDULR macro instruction, but your installation intends to use the system log as the hardcopy device (using the VARY command to alter the hardcopy status after IPL), you should still use the minimum specification of 1000 records.
WTR=		is used to calculate the amount of work space that must be reserved for use by the writers. All subparameters must be coded if the WTR parameter is specified. The subparameters specified in the RDR and WTR parameters determine the virtual storage size requirement for the job entry peripheral services work area.
	W=*w* **1**	specifies the maximum number of writers. For a system with RES, the value specified in this subparameter must include the number of remote writers that will be active at the same time.
	U=*u* **0**	specifies the number of 1024-byte blocks of virtual storage required for any user-written writer or job separator routines. This number should include the module sizes and work space required. If this parameter is omitted, 0 is used.
	Z=*z* **6**	specifies the control interval to be used for unit record output devices (for example, the 1403 printer). If this parameter is omitted, 6 is used. The control interval determines the speed at which a unit record device will be driven. It is the number of records per buffer. The assumed control interval values (5 for readers, 6 for writers) will drive these devices at near their rated speeds. A smaller control interval will reduce buffer requirements but will degrade performance for unit record devices.
	B=*b* **0**	specifies the sum of the blocksizes of all output streams, other than unit record output streams that may be processed concurrently. The value of B is equal to: b0+b1+...bn

Parameter	Subparameter	Explanation
WTR= (continued)	**B=*b*** **0** (continued)	where b represents the blocksize of one such output stream and n represents the total number of these blocksizes that may be processed concurrently. For example, three writers are to be active concurrently. One writer is to write a unit record output stream, the second writer is to write a tape output stream blocked at 400 bytes, and the third writer is to write a tape output stream blocked at 800 bytes. The value of B is equal to: (400+800)=1200 If this parameter is omitted, 0 is used.

Example: This macro defines the job entry substream requirements. It describes the features of the JES reader and writer. The volume serial number of the SPOOL data set is SYSRSM. The size of the JES central buffer pool is 1376 bytes and the number of buffers in the JES central buffer pool is 28. The maximum number of logical records to be included in the output data set is 100000. Thirty 176-byte records are available if a job would overflow the scheduler work area data set. Fifteen write-to-programmer messages are allowed per job step.

```
JES      JES     RDR=(R=2,Y=5,B=1200),                    X
                 WTR=(W=2,U=3,Z=6,B=2400),                X
                 SPOLVOL=(SYSRSM),                        X
                 BUFSIZE=1376,                            X
                 NUMBUFS=28,                              X
                 JOUTLIM=100000,                          X
                 SWDSLMT=30,                              X
                 STEPWTP=15
```

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) of the new VS1 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.

You may specify, at your option, 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 resident or pageable supervisor area.

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***] . . .)]**

[VIRTUAL=(*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 must be alphabetic.
RESIDNT=	*name*	specifies the member or members to be added to the IEAIGG03 list of modules in SYS1.PARMLIB. These modules will become part of the systems resident supervisor area. If RESIDNT=RENTCODE has not been specified in the CTRLPROG macro instruction, it will be assumed when this parameter is specified.

Parameter	Subparameter	Explanation
VIRTUAL=	*name*	specifies the member or members to be added to the IEAIGG02 list of modules in SYS1.PARMLIB. These modules will become part of the system's pageable supervisor area. A name appearing in the MEMBERS, RESIDNT, or VIRTUAL parameters should not appear in either of the others. The total number of names (MEMBERS, RESIDNT, and VIRTUAL) must not exceed 20. Members added by the RESIDNT and VIRTUAL parameters are also members of the system link library (SYS1.LINKLIB).

Example: This macro specifies the 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 not specified the default values are assumed.

The LOADER macro instruction can appear only once in the system generation input deck.

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=	*lib ddname* **SYSLIB**	specifies the ddname of the library that will be searched to resolve external references. If this parameter is omitted, SYSLIB is assumed.
LIN=	*in 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 assumed.
PARM=		specifies the options that will be used by the loader program. The following options can be listed in any order.
	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.

Parameter	Subparameter	Explanation
PARM= (continued)	**NOPRINT**	specifies that neither diagnostic messages nor the map of external references will be produced. The default value is PRINT.
	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. If you specify this subparameter, the PRINT subparameter of the PARM parameter must also be specified. The default value is NOMAP.
	NOLET	specifies that execution of the loaded program will not be attempted if 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. The default value is NOLET.
	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. The default value is CALL.
	NORES	specifies that the link pack area queue will not be searched.
	RES	specifies that an automatic search of the link pack area queue 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 be specified in the PARM parameter of this macro instruction. The default value is NORES.
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 used.
SIZE=	*size* **128**	specifies the amount of virtual storage required for the loader program, its 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 used.

Example: This macro selects the options for the loader program. 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 used for the CALL and NORES subparameters.

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

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 VS1 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 macro library (SYS1.MACLIB). These values can be listed in any order.
	BTAM	specifies that the macro definition used for BTAM are to be excluded.
	GPS	specifies that the macro definitions used for graphic programming services (GPS) are to be excluded.
	OCR	specifies that the macro definitions used for optical character readers 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 eight page data sets may be defined. You must specify a PAGE macro for each data set required.

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 respecify the UNIT, VOLNO, and SIZE parameters 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 on paging, refer to the *OS/VS1 Planning and Use Guide.* For information on the size of page data sets, refer to *OS/VS1 Storage Estimates.*

PAGE **DEV=***device type*

[SIZE={(**BLK**|**TRK**|**CYL**,*number*)|**MAX**}**]**

[{**UNIT=***address*|**VOLNO=***volume serial*}]

Parameter	Subparameter	Explanation
DEV=	*device type*	specifies the type of device that is to contain the page data set. Valid device types are 2305-2, 2314, and 3330. There is no default value. The device specified in this macro instruction must also be specified in an IODEVICE macro instruction.
SIZE=		specifies how the space for the page data set is to be allocated. The space that will be allocated will be contiguous.
	BLK	specifies that space is to be allocated in 2K 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.

Parameter	Subparameter	Explanation
SIZE= (continued)	**MAX**	specifies the use of the existing page data set on the selected volume (if this is not your first system generation), or the largest contiguous extent available on the volume if no page data set exists. The use of this value may result in a page data set that is too large for the specified virtual storage size. For additional information, refer to *OS/VS1 Storage Estimates.* If SIZE is omitted, MAX is used.
UNIT=	*address*	specifies the unit address of the device to contain the page data set. The device must be specified in the IODEVICE macro instruction.
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 ⩽ 07). The default values specify the first through eighth page data sets.

Example: This macro specifies that a 2314 with the volume serial number of 231X10 is to contain the page data set. The size of the page data set is 8192 2K blocks or enough to hold 16384K (16 megabytes) of virtual storage.

```
PAGE    PAGE    DEV=2314,VOLNO=231X10,SIZE=(BLK,8192)
```

PARTITNS

Required for:
complete
nucleus
I/O device

The PARTITNS macro instruction specifies the number, class, and size of problem program and system task partitions. Partitions may be specified in any order as long as every partition is specified.

The total number of problem program and system task partitions may not exceed 52. At least one, but no more than 15, problem program partitions must be specified. If two problem program partitions are specified and TCAM is also specified, then one additional partition of size 0 is generated to accommodate the generalized trace facility (GTF). If only one problem program partition is specified, it is assumed that the system being generated is minimal and that GTF will not be executable.

For a *nucleus generation,* partitions can be added to or deleted from the system, but every partition must be specified.

For an *I/O device generation,* partitions cannot be added to or deleted from the system. Every partition that was specified in the last complete system generation must be respecified.

For information on problem program and system task partitions, refer to *OS/VS1 Planning and Use Guide.* For size estimates, refer to *OS/VS1 Storage Estimates.*

PARTITNS *operand* [,*operand*] . . .

Parameter	Subparameter	Explanation
operand		Each operand is written in the format shown below. Uppercase letters, parentheses, and hyphens are written exactly as shown. **P***n***(C-***class***,S-***size***)** where:
	P*n*	specifies the partition number. The *n* is an integer from 0 to 15.
	C-*class*	specifies the function of the partition. *Class* has one of the following values: S which specifies a system task partition, or A string of up to fifteen letters (A through O) that indicates the job class that can use the partition.

Parameter	Subparameter	Explanation
operand (continued)	**S-***size*	specifies the size of the partition from 0 to maximum in 64K increments (specified without the K). (K represents 1024 bytes.) A partition size of 0 will reserve a TCB (task control block) for that partition but the TCB cannot be used until the operator redefines its size at IPL.

Example: This macro specifies the number, job class, and size of 5 partitions that are to be included in the VS1 system. Partition 0 defines a problem program partition for GTF.

```
PARTNS    PARTITNS    P0(C-ABCD,S-0),                    X
                      P1(C-ABCD,S-256),                  X
                      P2(C-ABCD,S-256),                  X
                      P3(C-ABCD,S-256),                  X
                      P4(C-ABCD,S-256)
```

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 subparameters 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 SYS1.NUCLEUS. The name is one to eight alphameric characters, the first of which is alphabetic. A maximum of ten members can be included in the nucleus. Members specified as either IEAXYZ1-IEAXYZ5 or beginning with IGC will be made resident in the fixed nucleus and the names will appear in the nucleus map. Members specified with any other name will be made resident in the pageable nucleus. If resident SVC routines are being included, each module can contain more than one SVC routine. The type, number, and SVRB extended save area of each of the resident SVC routines to be included must be specified in the SVCTABLE macro instruction. (For further information on writing your own SVC routines, refer to the *OS/VS1 Planning and Use Guide.)*
PDS=	*name*	specifies the partitioned data set that contains the modules to be included in SYS1.NUCLEUS. The name is one to eight 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,* the values specified in the WTOBFRS and REPLY parameters may be changed and I/O load balancing feature (OPTIONS=IOLOADBAL) may be deleted. 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 primary, master, 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 [AREA=(*nn* [,*nn*] ...)]

[BCLMT={*number* / **20**}]

[ESV={**NO** / **SMV** / **CON**}]

[EVA=(n_1,n_2)]

[IOC=*address*]

[JOBQFMT={*number* / **5**}] [1]

[JOBQLMT=*number*] [1]

[JOBQTMT={*number* / **60**}] [1]

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

[PFK=*number*]

[REPLY={*number* / **5**}]

[1]These subparameters may be overridden by the operator at IPL.

```
[SMF={BASIC       }]
     {FULL        }
     {NOTSUPPLIED}

[STARTI={AUTO  }][1]
        {MANUAL}

[STARTR=(A-address[,V-serial][1]
        [,D-dsname])]

[STARTW=(A-address[,V-serial][1]
        [,D-dsname])]

[TAVR={200}]
      {556}
      {800}

[VLMOUNT=AVR]

[WTLCLSS={classname}]
         {L        }
```

If OPTIONS=MCS not specified:

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

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

[WTOBFRS={number}]
         {4     }
```

If OPTIONS=MCS:

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

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

[1] These subparameters may be overridden by the operator at IPL.

```
[HARDCPY=({address  }[1]
          {SYSLOG   }
          [,{(routing code[,routing code]...)}]
            {ALL                             }
          [,{CMDS  }])]
            {INCMDS}
            {STCMDS}

[OLDWTOR=(routing code[,routing code]...)]

[ROUTCDE=(routing code[,routing code]...)]

[WTOBFRS={number}]
         {4     }
```

[1] These subparameters may be overridden by the operator at IPL.

Parameter	Subparameter	Explanation
ALTCONS=	*address*	specifies the unit address of a device 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. If OPTIONS=MCS is specified, this device must have input and output capability. 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 a SECONSLE macro instruction. If OPTIONS=MCS is specified, the device specified in this parameter must also be specified in the CONSOLE parameter of the SECONSLE macro instruction.
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 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 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/VS1 Display Consoles.*
BCLMT=	*number* **20**	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 20 is used.
CONSOLE=	*address*	specifies the unit address of the primary console or, if OPTIONS=MCS is specified, this parameter specifies the unit address of the master console. One primary console must always be specified. If a graphics device will be active as a console, a device that produces printed output must be specified. If OPTIONS=MCS is specified, this device must have input and output capability.
	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.

Parameter	Subparameter	Explanation
CONSOLE= (continued)	**O-*address*** (continued)	If OPTIONS=MCS is specified, any 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).
ESV=		specifies the destination of volume error statistics (ESV) records.
	NO	specifies that the records are not to be constructed.
	SMF	specifies that the records are to be put in the system management facility (SMF) data sets. If this value is specified, full SMF is defaulted to.
	CON	specifies that an abridged version of the records is to be written to the console. If this keyword parameter is omitted, CON is used.
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.
	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. If this parameter is omitted, EVA is not included in the new VS1 system.
HARDCPY=		specifies that a hardcopy 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 hardcopy log but are in the SYSLOG data set.) When a graphics console is active or if there is more than one active console, a hardcopy log will be required when the system is loaded or during job execution. If OPTIONS=MCS is specified and this parameter is not specified, SYSLOG and ALL are used. 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)		For information about operator communication with the system, the hardcopy log, and the system log, refer to *Operator's Library: OS/VS Console Configurations* and *OS/VS Supervisor Services and Macro Instructions.*
	SYSLOG	specifies that the data that is supposed to go to the hardcopy 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 hardcopy 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 in a SECONSLE macro. A graphics device cannot be specified as the hardcopy log device.
	ALL	specifies that all write-to-operator (WTO and WTOR) messages are to be put on the hardcopy log.
	routing code	is a number from 1 to 16 that designates the routing code that the hardcopy 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 hardcopy log.
	INCMDS	specifies that operator and system commands and responses (but not status displays) are to be written on the hardcopy log.
	STCMDS	specifies that status displays (except time-interval updated status displays) are to be written on the hardcopy log. If neither CMDS, INCMDS, nor STCMDS is specified, no operator or system commands or responses will be put on the hardcopy log.
IOC=	*address*	specifies the unit address of an integrated operator console. The value specified must be the unit address of a 3158, 3210, or 3215 console. This parameter need not be specified if the primary console is one of the consoles listed above. If the primary console is not one of these consoles, this parameter must be specified.

Parameter	Subparameter	Explanation
JOBQFMT=	*number* **5** 10	specifies the number of 176-byte queue records in a logical track for the system job queue. The number specified is an integer from 5 to 255. This number represents the total number of 176-byte records in each logical track. This value can be changed at IPL. If this parameter is omitted, a value of 5 is used.
JOBQLMT=	*number* 50	specifies the number of 176-byte records in the system job queue to be reserved for each initiator started. The number to be specified is an integer from 25 to 9999. This value can be changed at IPL. If this parameter is omitted, the value 2(JOBQFMT) + 15 is used. The quantity 2(JOBQFMT) is an approximation of the minimum job queue space needed to process a job having output in two SYSOUT classes. The default assumes that automatic restart is not being used.
JOBQTMT=	*number* **60** 100	specifies the number of 176-byte records in the system job queue to be reserved to start a writer and an initiator if job queue space becomes critical. The number specified is an integer from 60 to 9999. This value can be changed at IPL. If this parameter is omitted, 60 is used.
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,* and to *OS/VS Supervisor Services and Macros.*
OPTIONS=		specifies job scheduler, console, and system log options. These values may be listed in any order.
	CRJE	specifies the inclusion of conversational remote job entry (CRJE). (Additional information can be found in *OS/MFT, OS/MVT,* and *OS/VS1 CRJE Concepts and Facilities.*) If this subparameter is specified, **BTAM** will be assumed if **BTAM** was not specified in the **DATAMGT** macro instruction.

Parameter	Subparameter	Explanation
OPTIONS= (continued)	**EXIT**	specifies that a user exit is to be taken before the routing code of the write-to-operator (WTO and WTOR) message is used to send the message to the appropriate consoles. This allows you to add your own routine that will add, delete, or change routing and descriptor codes. For additional information, refer to *OS/VS Message Library: Routine and Descriptor Codes* and to *OS/VS Supervisor Services and Macros.*
	MCS	specifies that multiple console support (MCS) is to be included. If MCS is specified, the device specified in the ALTCONS parameter of this macro instruction must also be specified in the CONSOLE parameter of the SECONSLE macro instruction (see the SECONSLE macro instruction). For information on MCS refer to the *OS/VS1 Planning and Use Guide.* If MCS is specified, then the EXIT subparameter may be used.
	NOLOG	specifies that no system log is to be included. If this subparameter is specified and HARDCPY=SYSLOG is also specified, then this subparameter is ignored. If this subparameter is not specified, the system log will be included. In this case, RESIDNT=ASCMETH should be coded in the CTRLPROG macro instruction. If it is not coded, it will be assumed.
	IOLOADBAL	specifies that dynamic load balancing, which is based on the I/O activity per device, is to be included in the system. If this subparameter is not specified, load balancing will be based on the number of data sets per device. For information on load balancing, refer to *OS/VS1 Planning and Use Guide.*
	REMOTE	specifies that the Remote Entry Services (RES) option is to be included in the new system. If this option is specified, an RTAM task control block (TCB) and boundary box will also be included (the TCB gives RTAM a priority immediately above the master scheduler task). Information on RES and RTAM is in *OS/VS1 RES System Programmer's Guide.*
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).

Parameter	Subparameter	Explanation
PFK= (continued)	*number* (continued)	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 feature or a programmed-function keyboard, 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/VS1 Display Consoles.*
REPLY=	*number* **5**	specifies the number of reply queue elements to be used by the WTOR routines. Each reply queue element is 24 bytes long. If this keyword parameter is omitted, 5 is used.
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*).
SMF=		specifies the level of the system management facility (SMF) to be supported. User-written accounting routine exits are optional when BASIC or FULL is requested.
	BASIC	specifies that exits to your own accounting routines are provided with additional JES accounting information supplied at the job purge time exit. (Accounting records are written to the SYS1.ACCT system data set.)
	FULL	specifies that full SMF is to be included in the new VS1 system. Exits to your accounting routines are provided with additional JES accounting information supplied at the job purge time exit. User exits are also provided in the SMF routines. (SMF records are written to the SYS1.MANX and SYS1.MANY system data sets.)
	NOTSUPPLIED	specifies that you do not intend to provide an accounting routine and that SMF is not being included. The default value is NOTSUPPLIED, unless ESV=SMF is specified, in which case SMF=FULL is the default value.

Parameter	Subparameter	Explanation
SMF= (continued)	**NOTSUPPLIED** (continued)	The OUTLIM exit (IEFUSO) is taken only if SMF=BASIC or FULL is specified. For information on SMF, refer to the following publications: *OS/VS1 Planning and Use Guide* *OS/VS Systems Management Facilities (SMF)* *OS/VS1 Storage Estimates*
STARTI=		specifies whether a START INIT command is to be executed automatically each time the new system is loaded. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS1 Reference* for additional information.)
	AUTO	specifies that the command is to be executed automatically.
	MANUAL	specifies that it is not to be executed automatically. If this parameter is omitted, MANUAL is used.
STARTR=		specifies that a START RDR command is to be executed automatically for a device each time the new system is loaded. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS1 Reference* for additional information.)
	A-*address*	is the unit address of the input device to be started.
	V-*serial*	If 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. This command can be overridden by the operator after IPL. (Refer to *Operator's Library: OS/VS1 Reference* for additional information.)
	A-*address*	is the unit address of the output device to be started.
	V-*serial*	If 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 can be from 1 to 8 characters; the first character must be alphabetic.

Parameter	Subparameter	Explanation
TAVR=	**200** **556** **800**	specifies the standard density for 7-track magnetic tape volumes used with automatic volume recognition (AVR). If this parameter is omitted, 800 BPI is used. This parameter may be specified only if the VLMOUNT=AVR parameter is coded.
VLMOUNT=	**AVR**	specifies automatic volume recognition. For information on AVR refer to *OS/VS JCL Services.*
WTLCLSS=	*classname* **L**	specifies the classname to be used as a default for SYSOUT write-to-log messages. Classname is a letter from A through Z or a number from 0 through 9. If this parameter is omitted, L is used.
WTOBFRS=	*number* **4**	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. If the MCS subparameter in this macro instruction is not specified and this parameter is omitted, 4 is used. If OPTIONS=MCS is specified and the number of buffers specified is less than two buffers per console, the specified value is ignored and two buffers per console are assigned. The specified value is assigned if more than two buffers per console are specified. If this parameter is omitted, two buffers are assumed for each operator's console specified in the SECONSLE macro instructions. If the system does not have MCS, each buffer will be 144-bytes long. If the system does have MCS, each buffer will be 168-bytes long. The number of buffers should be at least equal to 4 times the number of initiators expected to be active at one time. For information on communication between the operato and the system, refer to *OS/VS Supervisor Services and Macros.*

Example: This macro specifies the size of each logical track for the system job queue and the number of records to be reserved for each initiator and writer. The system start commands are issued manually by default and the log procedures are also assumed by default. The unit address of the console and alternate console is 009 and 01F, respectively. The unit address of the integrated operator console is 009. The volume error statistics records are to be printed on the console. Full system management facility (SMF) is to be included in the new system. Default options are used for SYSOUT write-to-log messages, and the number of buffers available for write-to-operator routines.

```
SCHEDULR    SCHEDULR    SMF=FULL,JOBQFMT=5,                X
                        JOBQLMT=40,JOBQTMT=60,             X
                        CONSOLE=009,ALTCONS=01F,           X
                        IOC=009
```

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[1]
1442	N1		X			Card-read punch
1443	N1			X	X	Printer[1]
2250		X	X	X		Display unit[2]
2260	1	X	X	X		Display station[1,3]
2501			X			Card reader
2520	B1		X			Card reader punch
2540R			X			Card reader
2740	1	X	X	X	X	Communication terminal[3]
3158		X	X	X		Display unit[1,3]
3210		X	X	X	X	Console print keyboard
3211				X	X	High-speed printer. The universal character set is included[1].
3213				X	X	Printer
3215		X	X	X	X	Console print keyboard
3277	1			X		Display unit. May be specified as an O-address only console and must only be specified in the SECONSLE macro.
3277	2	X	X	X		Display unit[3]
3284	1			X	X	Printer[1]
3284	2			X	X	Printer[1]
3286	1			X	X	Printer[1]
3286	2			X	X	Printer[1]
3505			X			Card reader
3525			X			Card punch. Must have read feature.

[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 10. 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.

SECONSLE

Optional for:
complete
nucleus
I/O device

The SECONSLE macro instruction specifies a secondary console. This macro instruction is used if the multiple console support (MCS) option is specified in the OPTIONS parameter of the SCHEDULR macro instruction. This macro instruction must be specified at least once if MCS is specified. If MCS is not specified, secondary console support cannot be included in the system. Figure 10 lists the devices that may be specified as secondary consoles.

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.

A maximum of 31 secondary consoles can be specified. If more are coded, 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 if MCS is specified.

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

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

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

            [PFK=number]

            [ROUTCDE={ALL                       }]
                     {(routing code[,routing code] ...)}
```

SECONSLE (continued)

[USE={MS|SD}]

[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 10 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). A console specification can be made with the CONSOLE parameter in either the SCHEDULR or SECONSLE macro instruction only once. It can be made more than once with the ALTCONS parameter.
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 bottommost display area on the screen (the bottom lines of the message area). Subsequent *nn*s define areas stacked above the bottommost area. The

Parameter	Subparameter	Explanation
AREA= (continued)	*nn* (continued)	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) 19 if a 3277 Model 2 is used 23 if a 3277 Model 2 is used only for output (USE=SD) 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 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 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/VS1 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 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). A console specification can be made with the CONSOLE parameter in either the SCHEDULR or SECONSLE macro instruction only once. It can be made more than once with the ALTCONS parameter.

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 feature or a programmed-function keyboard, 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/VS1 Display Consoles.*
ROUTCDE=		specifies which routing codes the console will receive.
	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.*) If this parameter is omitted, no routing codes are assigned to this console.
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.
	MS	specifies that the console is to be used as an output-only console to display operator messages.
	SD	specifies that the console is to be used as an output-only console for status displays.

Parameter	Subparameter	Explanation
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/VS1 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)
```

SVCLIB

Optional for:
complete nucleus

Not applicable for:
I/O device

The SVCLIB macro instruction is used to add user-written routines, such as type 3 or type 4 SVC routines or I/O appendages, in load module form, to the SVC library (SYS1.SVCLIB). Additionally, you may specify those user-written routines that are reentrant and that are to be added to the list members that will become a part of the resident or pageable supervisor area.

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

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.SVCLIB. If you want the same members included that were included in the last complete system generation, you must respecify them.

For information on user-written routines, refer to *OS/VS1 Planning and Use Guide.*

SVCLIB **[MEMBERS=(***name***[,***name***] ...)]**

PDS=SYS1.*name*

[RESIDNT=(*name***[,***name***] ...)]**

[VIRTUAL=(*name***[,***name***] ...)]**

Parameter	Subparameter	Explanation
MEMBERS=	*name*	specifies the member or members to be added to the new SVC library.
PDS=	**SYS1.***name*	specifies the partitioned data set that contains the load module routines to be added to the SVC library. The name cannot exceed eight alphameric characters. The first character must be alphabetic.
RESIDNT=	*name*	specifies the member or members that are to be added to the IEARSV01 list of modules in SYS1.PARMLIB. These modules will become part of the system's resident supervisor area.

Parameter	Subparameter	Explanation
RESIDNT= (continued)	*name* (continued)	If RESIDNT=TRSVC, RESIDNT=ERP, and OPTIONS=TRSVCTBL have not been specified in the CTRLPROG macro instruction, they will be assumed when this parameter is specified.
VIRTUAL=	*name*	specifies the member or members that are to be added to the IEARSV00 list of modules in SYS1.PARMLIB. These modules will become part of the system's pageable supervisor area. If OPTIONS=TRSVCTBL has not been specified in the CTRLPROG macro instruction, it will be assumed when this parameter is specified. A name appearing in the MEMBERS, RESIDNT, or VIRTUAL parameters should not appear as a value in either of the others. The total number of names (MEMBERS, RESIDNT and VIRTUAL) must not exceed 20. Members added by the RESIDNT and VIRTUAL parameters are also members of the new VS1 system's SVC library.

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

```
USERSVC    SVCLIB    PDS=SYS1.USERSVC,                    X
                     MEMBERS=(IGC00240,IGC00241)
```

SVCTABLE

Optional for complete nucleus I/O device

The SVCTABLE macro instruction is used to specify the number, type, and supervisor request block (SVRB) extended save area of the user-written supervisor call (SVC) routines that are to be added to the new system.

Type 1 and 2 SVCs: For each type 1 or type 2 SVC, a corresponding member must 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 must be specified in the SVCLIB 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 previously specified type 3 or 4 SVC (using the SVCLIB macro instruction) that is to be supported by the new nucleus.

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

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.

SVC-*nnn*-{D*a*|E*a*}-S*b*

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
- *b* specifies the size of the SVRB associated with the SVC routine. The decimal number indicates the number of doublewords by which the SVRB is to be extended. A type 1 SVC must have a value of 0. Types 2, 3, and 4 can have a value of from 0 to 6.

Parameter	Subparameter	Explanation
operand (continued)		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.") E1 (type 1 SVC entered with interruptions enabled) is an invalid parameter. All type 1 SVCs must be entered disabled. For additional information on SVCs, refer to the *OS/VS1 Planning and Use Guide.*

Example: This macro supplies information about two user-written SVC routines. The SVCs are numbers 240 and 241. SVC 240 is a type 1 SVC that is being entered with interruptions disabled. SVC 241 is a type 2 SVC that is being entered with interruptions enabled. The size of the SVRB associated with the second SVC is extended by one doubleword.

```
SVCTAB    SVCTABLE    SVC-240-D1-S0,SVC-241-E2-S1
```

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. This macro instruction is optional. If it is omitted, it is expected that the IBM or user character-set images will be included by a process other than system generation. If the UCS macro is specified, then SYS1.IMAGELIB must have space allocated to it and it must be cataloged. 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={ALL1403                }
              {ALL3211                }
              {ALL                    }
              {(image[,image]...)     }
```

Parameter	Subparameter	Explanation
DEFAULT=		specifies the UCS images to be used as defaults if 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 images specified in the IMAGE parameter that are to be considered defaults. It is recommended that all UCS images that can produce valid results be specified as defaults.
IMAGE=		specifies, with one of the following values, the IBM standard character set images to be included. These values may be listed in any order.
	ALL1403	specifies that all the 1403 UCS images are to be included.

Parameter	Subparameter	Explanation
IMAGE= (continued)	**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.
	image	specifies particular UCS images to be included; the images are one or more of the values listed in Figure 11. 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	P1/I character set, 60 graphics.
T11	Text printing, 120 graphics.

Figure 11. 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 VS1 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 be the same as those specified in the IODEVICE macro instruction for those devices. For a list of I/O devices, see Appendix A.

If you want to refer to a device that is not supported by IBM using job control language statements, you must generate a unit address for the device by specifying a UNITNAME macro. (The device must also be specified in an IODEVICE macro.)

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.

Parameter	Subparameter	Explanation
UNIT= (continued)	*(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. 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.

```
UNITNAM    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.

```
           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).

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

SELECTING THE NEW SYSTEM DATA SETS

This chapter contains information about the required and optional system data sets that can be specified for the new VS1 system.

"Allocating Space to the New System Data Sets and Cataloging Them in the System Catalog" tells how to allocate space to the system data sets and catalog them using either the DATASET macro or the IEHPROGM utility program. This section also contains examples of coding the parameters for the DATASET macro and the control statements for executing the IEHPROGM utility program.

The information in this section about specifying DD statements and catalog statements applies only if you use IEHPROGM to allocate space to the system data sets and catalog them. If you use the DATASET macro, you only specify values for the SPACE and VOL parameters in the DATASET macro and the default values for the remaining parameters are supplied for you during system generation.

"System Data Set Summary" presents a summary of the required and optional system data sets. This section may be helpful to refer to when you are specifying the new system data sets.

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

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

Introductory material about the system data sets is in *OS/VS1 Planning and Use Guide.* Information about determining the size of the system data sets is in *OS/VS1 Storage Estimates.*

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 1170 tracks on a 2314/2319 or more than 728 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.LINKLIB, SYS1.SVCLIB, and SYS1.SYSJOBQE.

If system generation is to be executed in 144K bytes of real storage, the default values given for the blocksizes for the system data sets that are blocked to a full track should not be specified. A blocksize that is less than a full track should be specified.

The information needed to allocate space to the page data sets is specified in the PAGE macro instruction. Refer to the explanation on specifying the PAGE macro in the section "Specifying the New System Control 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. Refer to the description of the DATASET macro in "Specifying the New System Control Program" for additional information.

Figure 12 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)
DSSVM     DATASET  DSSVM,SPACE=(2048,(600))
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))
SVCLIB    DATASET  SVCLIB,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))
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 12. 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 for the system data sets and catalog them in the system catalog. If you use IEHPROGM instead of using the DATASET macro, space must be allocated to the system data sets and they must be cataloged before you begin Stage II.

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

Figure 13 illustrates the control statements for IEHPROGM. Sapce requirements for the system data sets are determined by several factors, especially by the type of device used and the characteristics of the system control program to be generated. Exact storage requirements for various types of direct-access devices are in *OS/VS1 Storage Estimates.*

```
//NEWSYS JOB
//        EXEC PGM=IEHPROGM
//SYSPRINT   DD SYSOUT=A
//CATALOG    DD DSNAME=SYSCTLG,VOLUME=(,RETAIN,SER=111111),       X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(2)),              X
//           LABEL=EXPDT=99350
//SVCLIB     DD DSNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=111111),   X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(15,1,187)),       X
//           LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,            X
//           BLKSIZE=2048)
//LINKLIB    DD DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=111111), X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(50,10,200)),      X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//PARMLIB    DD DSNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=111111), X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(1,,17),,CONTIG), X
//           LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//PROCLIB    DD DSNAME=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=111111), X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(2,1,50)),         X
//           LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//MACLIB     DD DSNAME=SYS1.MACLIB,VOLUME=(,RETAIN,SER=111111),   X
//           UNIT=2314,LABEL=EXPDT=99350,SPACE=(CYL,(20,5,50)),   X
//           DISP=(,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3360)
//TELCLIB    DD DSNAME=SYS1.TELCMLIB,VOLUME=(RETAIN,SER=111111), X
//           UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(20,2,20)),        X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//JOBQE      DD DSNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=111111),X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(5),,CONTIG)
//IMAGELIB   DD DSNAME=SYS1.IMAGELIB,VOLUME=(,RETAIN,SER=111111),X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(1,,17)),          X
//           LABEL=EXPDT=99350
//SAMPLIB    DD DSNAME=SYS1.SAMPLIB,VOLUME=(,RETAIN,SER=111111), X
//           UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(12,2,17)),        X
//           LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294,         X
//           LRECL=80)
//MANX       DD DSNAME=SYS1.MANX,VOLUME=(,RETAIN,SER=111111),     X
//           UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(10)),             X
//           LABEL=EXPDT=99350
```

Figure 13 (Part 1 of 2). Initializing system data sets using IEHPROGM. All of the system data sets in this example are to reside on one 2314 direct-access volume.

```
//MANY        DD DSNAME=SYS1.MANY,VOLUME=(,RETAIN,SER=111111),     X
//               UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(10)),          X
//               LABEL=EXPDT=99350
//NUCLEUS     DD DSNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=111111), X
//               SPACE=(CYL,(6,,2),,CONTIG),LABEL=EXPDT=99350,     X
//               UNIT=2314,DISP=(,KEEP)
//SYSIN DD    *
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.SVCLIB
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.LINKLIB
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.SYSJOBQE
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.PROCLIB
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.MACLIB
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.PARMLIB
    CATLG      COVL=2314=111111,VOL=2314=111111,DSNAME=SYS1.SAMPLIB
    CATLG      COVL=2314=111111,VOL=2314=111111,DSNAME=SYS1.IMAGELIB
    CATLG      COVL=2314=111111,VOL=2314=111111,DSNAME=SYS1.TELCMLIB
    CATLG      COVL=2314=111111,VOL=2314=111111,DSNAME=SYS1.MANX
    CATLG      COVL=2314=111111,VOL=2314=111111,DSNAME=SYS1.MANY
    CATLG      CVOL=2314=111111,VOL=2314=111111,DSNAME=SYS1.NUCLEUS
```

Figure 13 (Part 2 of 2). Initializing system data sets using IEHPROGM. All of the system data sets in this example are to reside on one 2314 direct-access volume.

The input deck for IEHPROGM must contain:

- A JOB statement with any parameters required by your installation.
- A 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),DISP=(,KEEP),                X
//          DCB=(see Figure 13)
```

- 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,VOL=unit=serial
```

The DD and CATLG statements and examples of allocation 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 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 device type 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/VS1 Storage Estimates.*
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. (Refer to "System Data Set Summary.")

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 device type 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 device type 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.

System Data Set Summary

Figure 14 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	seq.	required	yes	none	no
SYS1.DSSVM	seq.	optional	no	none[6]	yes
SYS1.LINKLIB[1]	PDS	optional	yes	RECFM=U,BLKSIZE=7294[2]	yes
SYS1.LOGREC[4]	seq.	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=U,BLKSIZE=7294[2]	optional
SYS1.SVCLIB[1]	PDS	required	yes	DSORG=POU,RECFM=U,BLKSIZE=2048	yes
SYS1.SYSJOBQE	seq.	optional	no	none	yes
SYS1.SYSPOOL	seq.	optional	no	none	no
Optional System Data Sets[8]					
SYS1.ACCT	seq.	optional	no	RECFM=U,BLKSIZE=7294[2]	no
SYS1.BRODCAST[5]	direct	optional	no	none	yes
SYS1.DCMLIB	PDS	optional	no	none	yes
SYS1.DUMP	seq.	optional	no	none	optional
SYS1.IMAGELIB	PDS	optional	no	RECFM=U,BLKSIZE=1024	yes
SYS1.MANX	seq.	optional	no	none	optional
SYS1.MANY	seq.	optional	no	none	optional
SYS1.RMTMAC[5]	PDS	optional	yes	RECFM=FB,LRECL=80,BLKSIZE=7280[3]	yes
SYS1.TELCMLIB	PDS	optional	yes	RECFM=U,BLKSIZE=7294[2]	yes
SYS1.UADS[5,7]	PDS	optional	yes	DSORG=PO,RECFM=FB	yes

[1] Space should be allocated in cylinders.

[2] BLKSIZE=7294 applies if the system data set resides on a 2314 or 2319. 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, 14640 for a 2305-2, 12960 for a 3330.

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

[5] Required for a system with RES (Remote Entry Services).

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

[7] The BLKSIZE specification should correspond to the needs of your installation.

[8] The optional system data sets may be required under certain conditions. Refer to the explanation of data sets in this section.

Note: The blocksize found in the DSCB for the data set whose record form is specified as U (RECFM=U) will be the maximum blocksize for the device being used. This is not necessarily the size of the current record.

Figure 14. 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 VS1 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 serial number of the new system residence volume must be specified for this data set.

Secondary space can be allocated.

CATLG statement:

This statement must not be coded for this data set.

SYS1.ACCT

Required if user-written accounting routines are in the system.
Sequential

Contents

This system data set contains accounting information from accounting routines that you write.

Requirements for specification

Location:

This data set must be on a permanently resident direct-access volume. It can be on 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:

$$\text{RECFM=U,BLKSIZE=}\begin{cases}14660 & \text{for a 2305-2}\\ 7294 & \text{for a 2314}\\ 13030 & \text{for a 3330}\end{cases}$$

CATLG statement:

This statement must not be coded for this data set.

Notes

You can allocate two SYS1.ACCT data sets as long as both are on permanently mounted direct-access volumes.

This data set must be defined if SMF=BASIC is specified in the SCHEDULR macro instruction.

SYS1.BRODCAST

Required for systems with RES.
Direct

Contents

This system data set stores two types of RES 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 this system data set, refer to *OS/VS1 RES System Programmer's 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.

Secondary space cannot be allocated.

DCB parameters cannot be specified.

CATLG statement:

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

Notes

With each use of the ACCOUNT utility SYNC subcommand:

- The data set is formatted and initialized
- Each user-identification in SYS1.UADS becomes an entry in this data set
- All notice and mail messages that were contained in the data set are lost.

SYS1.DCMLIB

Required if programmed-function-key (PFK) command entry 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

Required if you want a virtual storage image dump recorded in case of system task failure.
Sequential

Contents

This system data set is used by the damage assessment routine (DAR) to record a virtual storage image dump in case of system task failures.

For further information about virtual storage image dumps, refer to *OS/VS1 Debugging Guide.* For information on service aids, 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:

2305-2 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 at IPL.

Secondary space cannot be allocated.

CATLG statement:

This data set must be cataloged in the system catalog (SYSCTLG) only if it is going to be on a direct-access volume.

Note

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

SYS1.IMAGELIB

Required if a 1403 printer with the universal character set feature (UNVCHSET), a 3211 printer, or a 3525 card punch with the read feature is in the system.
Partitioned

Contents

This data set contains the universal character set (UCS) and forms control buffer (FCB) image modules. For information on 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:

Secondary space cannot be allocated.

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. Nonresident VS1 system routines are contained in this system data set, as well as the machine-check handler nucleus, initialization modules, and dynamically allocated modules. This data set also contains an assembler-language processor, a linkage editor, the utility programs, and service aids. If VSAM is to be included in the new system, this data set contains the Access Method Services program.

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 can be allocated.

The following DCB subparameters must be specified:

RECFM=U,BLKSIZE=(14660) for a 2305-2
(7294) for a 2314
(13030) for a 3330

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 record statistical data about machine failures (CPU failures, I/O device errors, and channel errors).

Requirements for specification

Location:

During system generation, space is automatically allocated for this data set on the system residence volume. This data set must be 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 IFCDIP00 program. Information on this program is in *OS/VS LOGREC Error Recording.* For information on space allocation for this data set, refer to *OS/VS1 Storage Estimates.*

SYS1.MACLIB

Required
Partitioned

Contents

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

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 can be allocated.

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
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 system management facility (SMF) will be included in the new system.
Sequential

Contents

Both of these system data sets contain the data collected by the SMF routines. (For information on SMF, refer to *OS/VS1 Planning and Use Guide* and *OS/VS System Management Facilities (SMF).*)

Requirements for specification

Location:

Both data sets should be defined on a direct-access device type. Both 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.

If these data sets are to be used, space must be allocated before the system is loaded.

CATLG statement:

These data sets must 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 allocations cannot be made.

CATLG statement:

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

SYS1.PARMLIB

Required
Partitioned

Contents

This system data set contains the RAM, BLDL, and SMFDFLT system parameter lists that are used by the nucleus initialization program (NIP), the PRESRES list that is used by the master scheduler, the JES parameters, the remote teleprocessing access method (RTAM) default parameters, the LNKST00 list that is used to concatenate data sets to SYS1.LINKLIB, the IEASYS00 list that contains the PAGE parameters, the IEAIGE00 list, the IEAIGG02 list, the IEAIGG03 list, the IEARSV00 list, and the IEARSV01 list. Optionally, this system data set may contain a list of parameters for use if automated system initialization is to be used.

For information on this system data set, refer to *OS/VS1 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 either system tasks or steps invoked by the operator or by the programmer. For information on this system data set, refer to *OS/VS1 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 can be allocated.

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
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) if it does not reside on the system residence volume. You should, however, always catalog this data set.

SYS1.RMTMAC

Required for systems with RES.
Partitioned

Contents

This system data set contains the source and macro definitions for RES modules IFSRTAM and IFSPREIN. This data set also contains the source and standard options for remote work stations. For information on this data set, refer to *OS/VS1 RES System Programmer's Guide.*

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 can be allocated.

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
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.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 accounting routines that can be used if the system management facility (SMF) is to be included in the system. (For information on punching the independent utilities, IPL text, and SMF sample programs, 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 can be allocated.

The following DCB subparameters must be specified:

RECFM=U,BLKSIZE={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 need not be cataloged in the system catalog (SYSCTLG).

Notes

This system data set is not used by the System Control Program. Therefore, after the newly installed System Control Program has been tested using the installation verification procedure, this system data set may be deleted.

SYS1.SVCLIB

Required
Partitioned

Contents

This system data set contains nonresident SVC routines, the data management access methods, the system's error recovery routines, and the machine-check handler nonresident modules.

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 can be allocated.

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=2048
```

CATLG statement:

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

Notes

The blocksize found in the DSCB will be the maximum blocksize for the device being used. This is not necessarily the size of the current record.

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 on this system data set, refer to *OS/VS1 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 1170 tracks can be on a 2314 or 2319, and no more than 728 on a 3330. A full volume can be allocated for the 2305-2.

Secondary space cannot be allocated.

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.

Notes

If this data set is date protected, the operator will have to reply

xx,'U'

to the message IEC107D before the data set can be formatted.

This data set must be formatted when the new VS1 system is loaded for the first time. (For detailed information, refer to *Operator's Library: OS/VS1 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 new system is loaded. For convenience, it is recommended that you catalog it and allocate space for it with the other system data sets.

SYS1.SYSPOOL

Required
Sequential

Contents

This system data set is used as a work area by the job scheduler. It contains data sets for spooling, system messages, write-to-programmer messages, the system log, and job control statements from any cataloged procedures that are executed. For information on this system data set, refer to *OS/VS1 Planning and Use Guide.*

Requirements for specification

Location:

This data set must be on a direct-access volume. It may be on the system residence volume but this may degrade performance.

DD statement:

For any SYS1.SYSPOOL data set, only the first extent is referenced and that extent must contain at least two tracks.

Secondary space should not be allocated.

CATLG statement:

This statement must not be coded for this data set.

Notes

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

From one to ten volumes containing SYS1.SYSPOOL data sets may be mounted at IPL. The total amount of space to be specified is limited as follows:

- The total number of tracks on all volumes may not exceed 64K tracks.
- The master cylinder map (MCM), which is used to manage the allocation of the work area, may not exceed 2048 bytes. For information on determining the size of the master cylinder map, refer to *OS/VS1 Storage Estimates.*

The SET parameter SPOOL=([CHNG],F) should be specified during the first IPL of the new system. This will cause the formatting of all SYS1.SYSPOOL data sets. As long as the extents of all the data sets do not change and there is no BUFSIZE change, the format request should be bypassed for any subsequent IPL.

For additional information, refer to the discussion on the JES macro instruction and to the *Operator's Library: OS/VS1 Reference.*

If a spool data set is date protected, the operator will have to reply

xx,'U'

to the message IEC107D before the data set can be formatted.

The minimum spool is 20 logical cylinders. The formula for calculating the logical cylinder definitions can be found in *OS/VS1 Storage Estimates.*

SYS1.TELCMLIB

Required if BTAM is going to be in the VS1 system or if TCAM is going to be included after system generation.
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 can be allocated.

The following DCB subparameters must be specified:

RECFM=U,BLKSIZE= { 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 RES.
Partitioned

Contents

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

For further information about this data set, refer to *OS/VS1 RES System Programmer's Guide.*

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 can be allocated.

The following DCB subparameters must be specified:

```
DSORG=PO,RECFM=FB
```

The BLKSIZE specification should correspond to the needs of your installation.

CATLG statement:

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

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 VS1 system as the generating system
- Adding your own routines to the new VS1 system to be generated

Initializing Direct-Access Volumes

Before a VS1 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 a system control program. The IEHDASDR utility program operates under the control of a system control program. 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 VS1 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 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 this chapter.

System Generation Using the Starter System

IBM provides a starter system that can be used for your first system generation. 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. 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 of 800 BPI. This applies whether the tape volumes mounted on the drives have standard labels or no labels.

If, however, you want to use 1600 BPI for the 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 135 With the ICA Feature

The Integrated Communication Adapter (ICA) feature on the System/370 Model 135 uses 1 address for each of up to 8 teleprocessing lines, beginning at address 001 on channel 0. Therefore, if you are using the starter system, you cannot have a 3211 printer on channel 0 at address 002 or 004.

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 (DLIBA1)
- Starting the starter system
- Copying the distribution libraries from the tape volumes (DLIBT1 and DLIBT2) to either two 2314 volumes (DLIBA2 and DLIBA3) or one 3330 volume (DLIBA2).
- Punching the independent utilities and IPL text on DLIBA2
- Listing the volume table of contents (VTOC) on the starter system and distribution library volumes.
- Initializing the volume that is to contain the new VS1 system

This section describes a procedure that can be followed to prepare for system generation. In this example, the direct-access volumes that are to contain the starter system, distribution libraries, and the new system to be generated are 2314 volumes. (Two 2314 volumes are required to contain the distribution libraries.) This procedure can also be followed if 2319 or 3330 volumes are used. (One 3330 volume is required to contain the distribution libraries.) If a 3330 is being used, the parameters that specify the device will have to 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
2314 direct-access storage device	starter system – DLIBA1	130
2314 direct-access storage device	distribution library volume – DLIBA2	131
2314 direct-access storage device	distribution library volume – DLIBA3	132
2314 direct-access storage device	new system (SYSRES)	133
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 tapes – DLIBT1 and DLIBT2	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 and IBCDMPRS independent utility programs.
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:

```
INITVOL1  JOB
          MSG     TODEV=1403,TOADDR=00E
          DADEF   TODEV=2314,TOADDR=130,                    X
                  VOLID=SCRATCH,FLAGTEST=NO
          VLD     NEWVOLID=DLIBA1,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 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,          X
                    TODEV=2314,TOADDR=130,              X
                    VOLID=DLIBA1
          END
```

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.

8. The direct-access volume that is to contain the starter system's page data set must be mounted. If there is enough space, this volume can be the starter system volume, as in this example. For a 2314/2319, the device address can be 130, 230, or 330. For a 3330, the device address can be 150, 250, or 350. (Approximately 10 cylinders are required for a 2314/2319 volume, 6 cylinders for a 3330 volume.)
9. Set the load selector switches on the control panel to the channel, control unit, and device of the volume that contains the starter system. Then press the LOAD key.
10. If your real storage size is less than 384K bytes, signal EOB to the SPECIFY VIRTUAL STORAGE SIZE message. If your real storage size is greater than or equal to 384K bytes, reply R 00, '2048' and signal EOB.
11. Signal EOB to the SPECIFY SYSTEM PARAMETERS message.
12. Signal EOB to the request for a SYS1.DUMP data set.
13. Reply R 00, 'NO' to the CHANGE PARTITIONS message and signal EOB.
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 01, 'Q=(,F)' and signal EOB. If the date is incorrect, type

R 01, 'Q=(,F),DATE=yy.ddd,CLOCK=hh.mm.ss' where yy is the year (00-99), ddd is the day (001-366), hh are the hours (00-23), mm are the minutes (00-59), and ss are the seconds (00-59), and set the TOD switch on the console.

15. Reply R 02, 'U' to the message SPECIFY JOB QUEUE PARAMETERS and signal EOB.
16. When the starter system is ready to begin working, it sends you the message: READY.
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.P0,,,classname
```

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

18. Reply R 03, 'U' to the message SPECIFY UCS PARAMETERS and signal EOB.
19. Reply R 04, 'U' to the message SPECIFY SWADS RESERVE VALUE FOR P0 and signal EOB. The system will enter the wait state.

Initialize the Volumes That Will Contain the Distribution Libraries

20. The volumes to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device. The volumes will be initialized. After IEHDASDR has been executed, the initialized volumes must be varied online.

```
// VARY (131,132),OFFLINE
//INIT         JOB  ACCT123,PROGRAMMER,MSGLEVEL=1
//DLIBS       EXEC  PGM=IEHDASDR,PARM='N=1'
//SYSPRINT      DD  SYSOUT=A
//SYSIN         DD  *
   ANALYZE          TODD=(131),VTOC=00004,EXTENT=00010, X
                    NEWVOLID=DLIBA2,OWNERID=ID
   ANALYZE          TODD=(132),VTOC=00004,EXTENT=00010, X
                    NEWVOLID=DLIBA3,OWNERID=ID
/*
```

Copy the Distribution Libraries to a Direct-Access Volume(s)

The IEBCOPY utility program is used to load the distribution libraries from tape to a direct-access volume(s). When IEBCOPY is executed, space will be allocated to each of the distribution libraries and they will be cataloged in the system catalog of the generating system (in this case, the starter system).

The IEBCOPY control statements are in the first two data sets on the distribution library tapes. The control statements for copying to a 2314/2319 volume are in the first data set (LD231420) of each tape. The control statements for copying to a 3330 volume are in the second data set (LD333020) of each tape.

21. Mount the first distribution library tape (DLIBT1).
22. To access the IEBCOPY control statments for copying to DLIBA2 and DLIBA3, type the following:

```
START RDR,182,LABEL=(1,SL),DSNAME=LD231420,
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 and the distribution libraries from DLIBT1 will be copied to DLIBA2 and DLIBA3.

23. Mount the second distribution library tape (DLIBT2).
24. To access the IEBCOPY control statements for copying to DLIBA3, type the following:

```
START RDR,182,LABEL=(1,SL),DSNAME=LD231420,
VOL=SER=DLIBT2
```

(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 and the distribution libraries from DLIBT2 will be copied to DLIBA3.

Completion of the preceding steps provides operable direct-access volumes with backup tapes.

Note: If the distribution libraries are to be loaded to one 3330 volume (DLIBA2), you would follow the same procedure for copying to the 2314 volumes. To access the IEBCOPY control statements for copying the distribution libraries from the first tape to DLIBA2, type the following:

```
START RDR,182,LABEL=(2,SL),DSNAME=LD333020,
VOL=SER=DLIBT1
```

To access the IEBCOPY control statements for copying the distribution libraries from the second tape to DLIBA2, type the following:

```
START RDR,182,LABEL=(2,SL),DSNAME=LD333020,
VOL=SER=DLIBT2
```

Punch the Utility Programs and IPL Text

25. *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 a system control program 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 distribution library SYS1.ASAMPLIB. By punching them at this time, they are available and you need only include them in your input deck whenever you want to use these utilities or whenever you initialize a system residence volume.

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,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 routines that are contained in SYS1.ASAMPLIB for use if the system management facility (SMF) is to be included in the new VS1 system. By listing the cards that have been punched out from SYS1.ASAMPLIB, sample coding is provided that you can use as a reference while you are writing your own routine. (For information on SMF, refer to *OS/VS System Management Facilities (SMF).*) The accounting 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)

26. Before Stage I, you must allocate space to the three utility data sets that are used during Stage I and the three utility data sets that are used during Stage II by the assembler. If you choose to allocate this space on either the starter system volume or distribution library volumes, you may want to list the VTOCs of the three volumes at this point to determine if enough space is available. (The allocation of the utility data sets is discussed in "System Control Program Installation.")

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

```
//JOB2       JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1     EXEC PGM=IEHLIST
//SYSPRINT    DD SYSOUT=A
//SYSRES      DD UNIT=2314,DISP=OLD,VOLUME=SER=DLIBA1
//DLIBA2      DD UNIT=2314,DISP=OLD,VOLUME=SER=DLIBA2
//DLIBA3      DD UNIT=2314,DISP=OLD,VOLUME=SER=DLIBA3
//SYSIN       DD *
       LISTVTOC  VOL=2314=DLIBA1,FORMAT
       LISTVTOC  VOL=2314=DLIBA2,FORMAT
       LISTVTOC  VOL=2314=DLIBA3,FORMAT
/*
```

If JOB2 is run immediately after JOB1 (in step 25) eliminate the JOB2 card.

Initialize the Volume That Will Contain the New VS1 System

27. Mount the disk volume that is to contain the VS1 system that is to be generated.
28. 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 133,OFFLINE
//INIT      JOB  ACT123,PROGRAMMER,MSGLEVEL=1
//SYSRES    EXEC PGM=IEHDASDR
//SYSPRINT   DD  SYSOUT=A
//ASAMPLIB   DD  DISP=OLD,                                    X
//               DSNAME=SYS1.ASAMPLIB(IEAIPL00)
//SYSIN      DD  *
     ANALYZE TODD=(133),VTOC=2,EXTENT=8,                      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 VS1 System

After your first system generation using the starter system, an existing VS1 system can be used as the generating system. When you use an existing VS1 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

If you use the first method, you can modify the nucleus of the generating system. If you use the second method, you cannot modify the generating system.

When you use an existing VS1 system as the generating system, the START command for the initiator should include the SPACE keyword to increase the size of the scheduler work area data set (SWADS) from 250 blocks to 600 blocks (see "Starting the Starter System" in this chapter).

Preparation for System Generation Using an Existing VS1 System as the Generating System

This section describes a procedure that can be followed to prepare for system generation when an existing VS1 system is used as the generating system. In this example, the direct-access volumes that are to contain the distribution libraries and the new system to be generated are 2314 volumes. (Two 2314 volumes are required to contain the distribution libraries.) This procedure can also be followed if 2319 or 3330 volumes are used. (One 3330 volume is required to contain the distribution libraries.) If a 3330 is being used, the parameters that specify the device will have to be changed.

Initialize the Volumes That Will Contain the Distribution Libraries

1. Mount the volumes onto which the contents of the distribution library tapes are to be copied.
2. The volumes 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 volumes will be initialized. After IEHDASDR has been executed, the initialized volumes must be varied online.

```
// VARY (131,132),OFFLINE
//INIT      JOB   ACCT123,PROGRAMMER,MSGLEVEL=1
//DLIBS     EXEC  PGM=IEHDASDR,PARM='N=1'
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
   ANALYZE  TODD=(131),VTOC=00004,EXTENT=00010,    X
              NEWVOLID=DLIBA2,OWNERID=ID
   ANALYZE  TODD=(132),VTOC=00004,EXTENT=00010,    X
              NEWVOLID=DLIBA3,OWNERID=ID
/*
```

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

Copy the Distribution Libraries to a Direct-Access Volume(s)

The IEBCOPY utility program is used to load the distribution libraries from tape to a direct-access volume(s). When IEBCOPY is executed, space will be allocated to each of the distribution libraries and they will be cataloged in the system catalog of the generating system (in this case, an existing system).

The IEBCOPY control statements are in the first two data sets on the distribution library tapes. The control statements for copying to a 2314/2319 volume are in the first data set (LD231420) of each tape. The control statements for copying to a 3330 volume are in the second data set (LD333020) of each tape.

3. Mount the first distribution library tape (DLIBT1).
4. To access the IEBCOPY control statements for copying to DLIBA2 and DLIBA3, type the following:

```
START RDR,182,LABEL=(1,SL),DSNAME=LD231420,
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 and the distribution libraries from DLIBT1 will be copied to DLIBA2 and DLIBA3.

5. Mount the second distribution library tape (DLIBT2).
6. To access the IEBCOPY control statements for copying to DLIBA3, type the following:

```
START RDR,182,LABEL=(1,SL),DSNAME=LD231420,
VOL=SER=DLIBT2
```

(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 and the distribution libraries from DLIBT2 will be copied to DLIBA3.

Completion of the preceding steps provides operable direct-access volumes with backup tapes.

Note: If the distribution libraries are to be loaded to one 3330 volume (DLIBA2), you would follow the same procedure for copying to the 2314 volumes. To access the IEBCOPY control statements for copying the distribution libraries from the first tape to DLIBA2, type the following:

```
START RDR,182,LABEL=(2,SL),DSNAME=LD333020,
VOL=SER=DLIBT1
```

To access the IEBCOPY control statements for copying the distribution libraries from the second tape to DLIBA2, type the following:

```
START RDR,182,LABEL=(2,SL),DSNAME=LD333020,
VOL=SER=DLIBT2
```

Include the VS1 Release 2 Level Assembler and Linkage Editor Cataloged Procedures in the Generating System

7. If your generating system is not a release 2 level, you will need to include the release 2 level assembler (ASMS) and linkage editor (LINKS) cataloged procedures in your generating system's procedure library for Stage II execution.

   The two cataloged procedures can be included in your procedure library by either obtaining them from the SYS1.APROCLIB distribution library or by coding them yourself.

   To obtain the ASMS and LINKS cataloged procedures from the SYS1.APROCLIB distribution library, place the following control statements in the input device and enter a START RDR,00C command:

```
//UPDATE   JOB ACCOUNT123,PROGRAMMER,MSGLEVEL=1
//PROC    EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT  DD SYSOUT=A
//SYSUT1    DD DSN=SYS1.APROCLIB,UNIT=2314,DISP=OLD
//SYSUT2    DD DSN=SYS1.proclib,UNIT=unit,              X
//             VOLUME=SER=volume serial,DISP=OLD
//SYSIN     DD *
./       REPRO NAME=ASMS,LIST=ALL
./       REPRO NAME=LINKS,LIST=ALL
./       ENDUP
/*
```

If you choose to include the two cataloged procedures in your procedure library yourself, place the following control statements and the ASMS and LINKS cataloged procedures in the input device and enter a START RDR, 00C command:

```
//UPDATE   JOB  ACCOUNT123,PROGRAMMER,MSGLEVEL=1
//PROC    EXEC  PGM=IEBUPDTE,PARM=MOD
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSN=SYS1.proclib,UNIT=unit,                    X
//              VOLUME=SER=volume serial,DISP=OLD
//SYSUT2    DD  DSN=SYS1.proclib,UNIT=unit,                    X
//              VOLUME=SER=volume serial,DISP=OLD
//SYSIN     DD  *
./          ADD  NAME=ASMS,LIST=ALL
//ASMS    PROC
//A       EXEC  PGM=ASMBLR,COND=(4,LT)
//SYSLIB    DD  DSN=SYS1.AMODGEN,DISP=(SHR,PASS)
//            DD  DSN=SYS1.AMACLIB,DISP=(SHR,PASS)
//SYSUT1    DD  DISP=(NEW,DELETE),DSN=&&SYSUT1,                X
//              SPACE=(1700,(2400,200)),UNIT=2314
//SYSUT2    DD  DISP=(NEW,DELETE),DSN=&&SYSUT2,                X
//              SPACE=(1700,(1800,200)),UNIT=2314
//SYSUT3    DD  DISP=(NEW,DELETE),DSN=&&SYSUT3,                X
//              SPACE=(1700,(600,100)),UNIT=2314
//SYSPRINT  DD  SPACE=(121,(1500,200),RLSE),                   X
//              DCB=(RECFM=FB,LRECL=121,                       X
//              BLKSIZE=3509),SYSOUT=&CLASS
//SYSPUNCH  DD  DISP=OLD,VOLUME=(,RETAIN),                     X
//              DSN=SYS1.&OBJ.(&MOD)
./          ADD  NAME=LINKS,LIST=ALL
//LINKS   PROC
//LK      EXEC  PGM=IEWL,COND=(8,LT)
//SYSUT1    DD  DISP=(NEW,DELETE),DSN=&&SYSUT1,                X
//              SPACE=(1700,(400,50)),UNIT=2314
//SYSPRINT  DD  SPACE=(121,(850,50),RLSE),                     X
//              DCB=(RECFM=FB,LRECL=121,                       X
//              BLKSIZE=1210),SYSOUT=&CLASS
//SYSPUNCH  DD  DISP=SHR,VOLUME=(,RETAIN),                     X
//              DSN=SYS1.&OBJ
//SYSLMOD   DD  DISP=OLD,UNIT=&UNIT,                           X
//              VOLUME=SER=&SER,                               X
//              DSN=&N..&NAME&P1&MOD&P2
//
./        ENDUP
/*
```

Punch the Utility Programs and IPL Text

8. The independent utilities operate outside the control of a system control program 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 to be generated. 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.

   If your generating system is not the same release as the new VS1 system to be generated, you will need to punch the IBCDASDI independent utility program from SYS1.ASAMPLIB in order to initialize the volume that will be the new system volume and to include the IPL text. If your generating system is the same release as the new system to be generated, you may use the IEHDASDR utility program in your generating system to initialize the new system volume and to include the IPL text (see "Initialize the Volume That Will Contain the New VS1 System" in the section "System Generation Using the Starter System").

   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 statements 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 VS1 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.

Initialize the Volume That Will Contain the New VS1 System

9. Mount the volume that is to contain the VS1 system that is to be generated.
10. Load the IBCDASDI independent utility program that was punched from SYS1.ASAMPLIB (step 8) by placing it in the input device, setting the load selector switches and pressing the console LOAD key. Then, place the following IBCDASDI control statements along with the IPL text that was punched from SYS1.ASAMPLIB in the input device:

```
INIT    JOB
        MSG     TODEV=1403,TOADDR=00E
        DADEF   TODEV=2314,TOADDR=133,VOLID=SCRATCH, X
                FLAGTEST=NO
        VLD     NEWVOLID=SYSRES,OWNERID=DEPT54
        VTOCD   STRTADR=2,EXTENT=8
      IPLTXT
        .
        .    (IPL text)
        .
END
```

Note: If the volume is being initialized for the first time, FLAGTEST=NO must be specified in the DADEF statement.

11. 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.
12. 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.

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 and for copying the distribution library tapes 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 VS1 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.

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.
- Initialize the volume(s) onto which the distribution library tapes are to be copied. (See "Processing the Starter System and Distribution Library Tapes" in this chapter.)
- Copy the contents of the distribution library tapes to a direct-access volume(s). (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 SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LINKLIB.

Routines that are to be added to SYS1.NUCLEUS, SYS1.SVCLIB, or SYS1.LINKLIB 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 one to eight alphameric characters, the first character alphabetic. The name of the partitioned data set and the names 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 SYS1.NUCLEUS during the first 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 SVCLIB macro instruction specifies load modules, such as type 3 or type 4 SVC routines and nonstandard label routines, to be added to SYS1.SVCLIB during an IEBCOPY step in Stage II (see "System Control Program Installation").

If SVCs are added to SYS1.NUCLEUS or SYS1.SVCLIB, 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 15 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 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-S0
```

The RESMODS, LINKLIB, SVCLIB, 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,LOAD'
//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,              X
//                DISP=(,PASS),SPACE=(CYL,(5,1,1,)),    X
//                DCB=BLKSIZE=400
//SYSIN       DD *
IGC255    CSECT
          ENTRY IGC255
     source statements
               .
               .
               .
             END
/*
//STEP2     EXEC PGM=IEWL,PARM='UNIT,NCAL,XREF,RENT'
//SYSPRINT    DD SYSOUT=A
//SYSUT1      DD SPACE=(1024,(200,20)),UNIT=SYSDA
//SYSLMOD     DD DSN=SYS1.USERLIB(IGC255),              X
//                DISP=(,CATLG),VOL=SER=PACK01,         X
//                SPACE=(CYL,(2,1,10)),                 X
//                DCB=(RECFM=U,BLKSIZE=7294),           X
//                UNIT=SYSDA
//SYSLIN      DD DSN=&&LOADSET,DISP=(OLD,DELETE)
/*
//
```

Figure 15. Adding a user-written routine to the system

Figure 16 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. 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'
//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,                  X
//                   SPACE=(80,(200,50)),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
/*
//STEP2     EXEC     PGM=IEWL,PARM=(LIST,NCAL)
//SYSLIN      DD     DSNAME=&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD     DD     DSNAME=SYS1.USER(IEAXYZ1),UNIT=2314,         X
//                   DISP=(,CATLG),VOLUME=(,RETAIN,SER=PACK01),   X
//                   SPACE=(1024,(50,20,5))
//SYSUT1      DD     UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),           X
//                   SPACE=(1024,(50,20,5))
//SYSPRINT    DD     SYSOUT=A
/*
//
```

Figure 16. 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 VS1 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. (See "Specifying the New System Control Program.") The sequence of the cards in the deck and the job control language statements is shown in Figure 17.

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

Figure 17. Input deck for Stage I

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 initialization 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 named 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

The input deck for Stage I may also contain the macro statements that will include, in the system to be generated, the additional support required to treat not-ready devices and devices that are not currently attached to the system in an offline status. Refer to Appendix F for this information.

Figure 18 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 device 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.")

Device Type	DD Statement					
	SYSUT1	SYSUT2	SYSUT3	OBJPDS1[1]	OBJPDS2[1]	OBJPDS3[1]
2305 or 3330	3,2	5,2	8,2	5,1,12	3,1,10	5,1,12
2314 or 2319	15,2	10,2	16,2	7,1,12	5,1,10	7,1,12

[1]The OBJPDS1, OBJPDS2, and OBJPDS3 utility data sets are used during Stage II

Figure 18. 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 generating 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 a direct-access volume 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 Stage 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:

- The 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 system data sets
- Utility programs complete the construction and initialization of the system data sets that make up the new VS1 system.

The completion of Stage II results in a VS1 System Control Program that reflects what was specified in the system generation macro instructions.

The Stage II job stream consists of several jobs so multijobbing may be used. If multijobbing is used to execute Stage II, you must specify three partitions when the generating system is loaded. If multijobbing is not used, the jobs will execute sequentially.

Stage II Input

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

Allocating Space for and Cataloging the Utility Data Sets Required for Stage II

Before executing Stage II, you must allocate space to the three utility data sets, OBJPDS1, OBJPDS2, and OBJPDS3, that are used by the assembler during Stage II. Space can be allocated to these utility data sets at the same time you allocate space to the system data sets, if you are using IEHPROGM. (See "Preparing for System Generation.") If you are using the DATAŞET macro to allocate space to the system data sets, you will need to allocate space to the Stage II utility data sets before Stage II is executed. Figure 19 shows the job control statements used to catalog and allocate space for the OBJPDS1 utility data set. Figure 18 gives the space requirements for the OBJPDS1, OBJPDS2, and OBJPDS3 utility data sets.

```
//OBJPDS1  JOB MSGLEVEL=1
//         EXEC PGM=IEHPROGM
//OBJPDS1   DD DSNAME=SYS1.OBJPDS1,SPACE=(space),    X
//             DISP=(,CATLG),UNIT=(unit address),    X
//             VOLUME=(,RETAIN,SER=serial number)
//SYSPRINT  DD DUMMY
//SYSIN     DD DUMMY
```

Figure 19. Allocation of the OBJPDS1 utility data set

If Stage II immediately follows Stage I, the JOB card is not specified.

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 20 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'.

```
//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  DATA
./         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 20. 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 IEBCOPY, IFCDIP00, IEHIOSUP, IEHPROGM, IEHLIST, and IEBUPDTE. 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 new System Control Program. Figure 21 in this section lists 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.

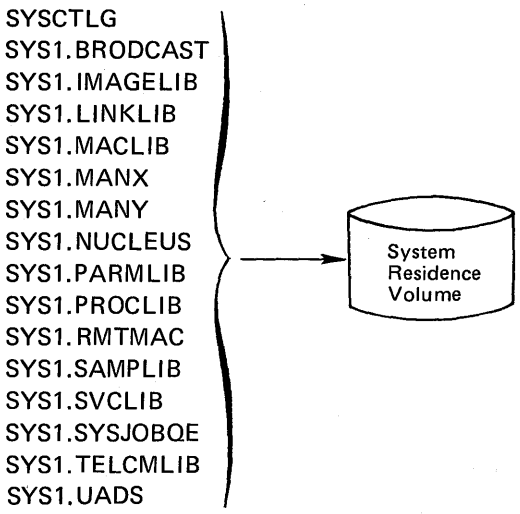
During Stage II execution, the job stream (job control language and utility control statements) assembles, link-edits, and copies specific modules from the distribution libraries and user libraries into system data sets that were allocated on the new system volume(s) to form the new VS1 system.

Execution of the job stream occurs in fifteen jobs for a complete system generation. For a nucleus or I/O device generation, job stream execution occurs in less than fifteen 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 options were specified in the macro instructions.

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

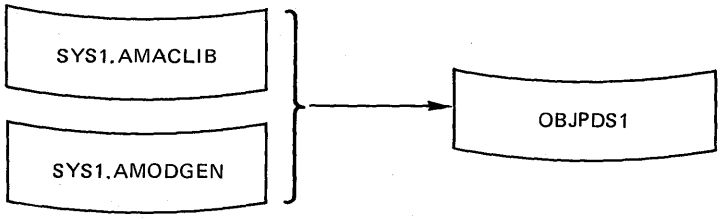
Job 1: If DATASET macros were specified, the IEHPROGM utility program catalogs and allocates space for the new system data sets. If the system data sets were cataloged and space was allocated before Stage II, this job step does not occur.

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



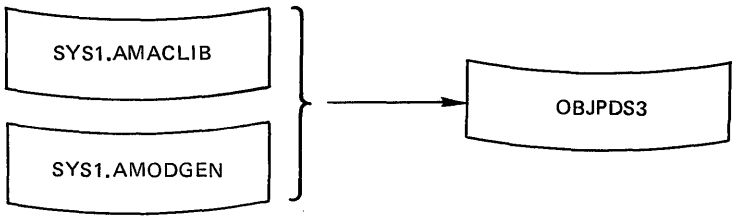
Job 2: The nucleus is assembled. Modules are assembled from the macros in the SYS1.AMODGEN and SYS1.AMACLIB distribution libraries and placed in the OBJPDS1 utility data set.

```
//SYSGEN2  JOB
//SG1      EXEC  ASMS
                .
                .
                .
//SG13     EXEC  ASMS
```



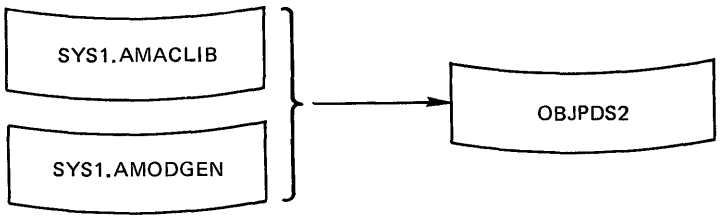
Job 3: Modules are assembled from the macros in the SYS1.AMODGEN and SYS1.AMACLIB distribution libraries and placed in the OBJPDS3 utility data set.

```
//SYSGEN3  JOB
//SG1      EXEC  ASMS
                .
                .
                .
//SG18     EXEC  ASMS
```



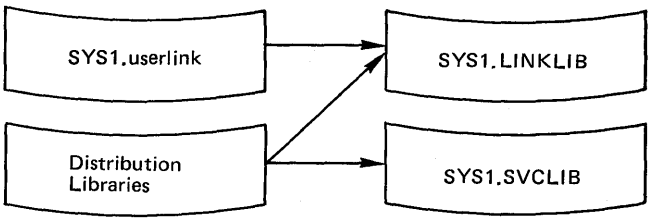
Job 4: Modules are assembled from the macros in the SYS1.AMACLIB and SYS1.AMODGEN distribution libraries and placed in the OBJPDS2 utility data set.

```
//SYSGEN4  JOB
//SG1      EXEC  ASMS
                .
                .
                .
//SG10     EXEC  ASMS
```



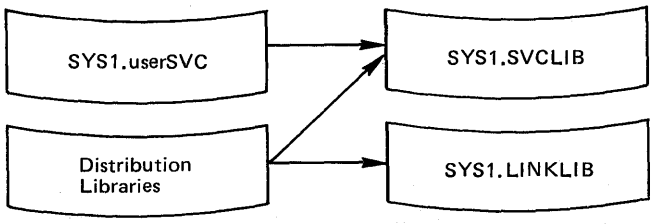
Job 5: The IEBCOPY utility program copies modules from the distribution libraries and user-written routines from the user-link library into SYS1.LINKLIB and modules from the distribution libraries into SYS1.SVCLIB.

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



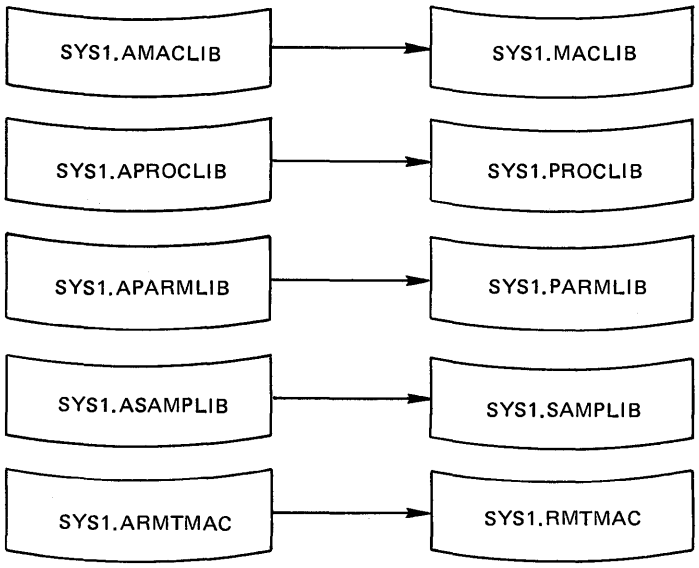
Job 6: The IEBCOPY utility program copies modules from the distribution libraries and user-written routines from the user-SVC library into SYS1.SVCLIB and modules from the distribution libraries into SYS1.LINKLIB.

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



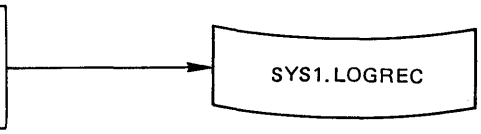
Job 7: The IEBCOPY utility copies modules from SYS1.AMACLIB to SYS1.MACLIB in the new system. It also copies SYS1.APROCLIB, SYS1.APARMLIB, SYS1.ASAMPLIB, and SYS1.ARMTMAC to SYS1.PROCLIB, SYS1.PARMLIB, SYS1.SAMPLIB, and SYS1.RMTMAC (if RES is to be included in the new system).

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



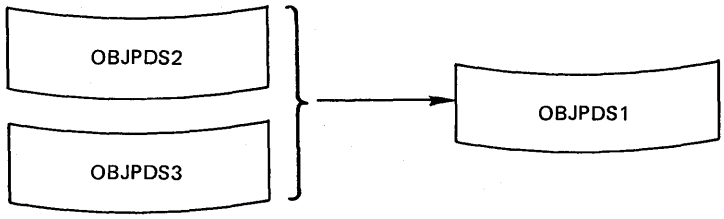
Job 8: This job allocates space to SYS1.LOGREC and catalogs it on the new system residence volume.

```
//SYSGEN8   JOB
//SG1       EXEC  IFCDIP00
//SG2       EXEC  IEHPROGM
```



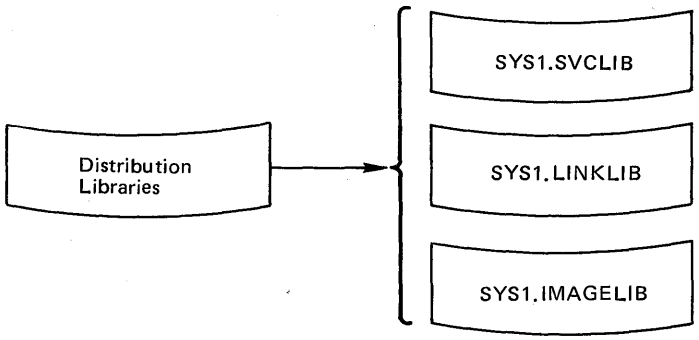
Job 9: The IEBCOPY utility program copies the modules that were assembled and placed in OBJPDS2 and OBJPDS3 (jobs 3 and 4) to OBJPDS1.

```
//SYSGEN9    JOB
//SG1        EXEC   IEBCOPY
```



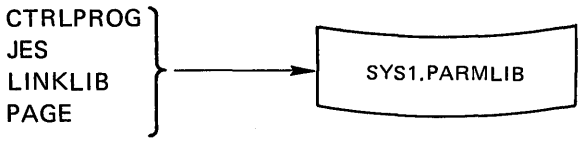
Job 10: The IEBCOPY utility program copies modules from the distribution libraries into SYS1.SVCLIB, SYS1.LINKLIB, and modules from the distribution libraries are link-edited into SYS1.IMAGELIB.

```
//SYSGEN10   JOB
//SG1        EXEC   IEBCOPY
//SG2        EXEC   LINKS
//SG3        EXEC   LINKS
```



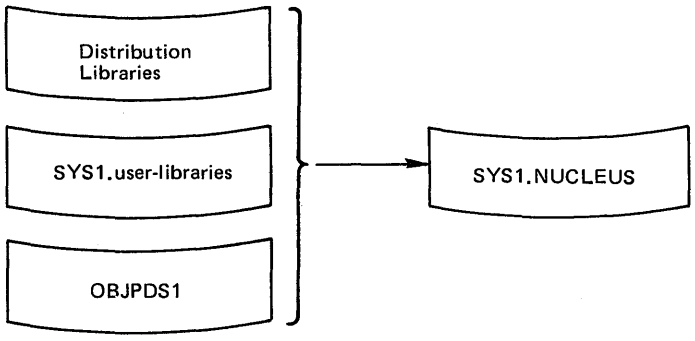
Job 11: The IEBUPDTE utility program adds members to SYS1.PARMLIB. The members that are added were specified in the CTRLPROG, JES, PAGE, and LINKLIB macro instructions.

```
//SYSGEN11   JOB
//SG1        EXEC   IEBUPDTE
                .
                .
                .
//SG3        EXEC   IEBUPDTE
```



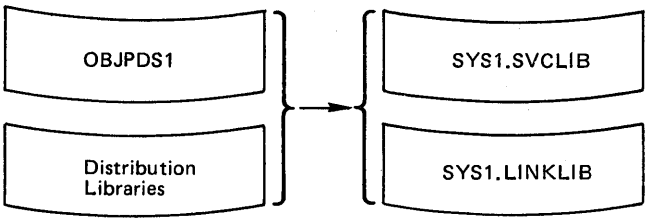
Job 12: Modules from the distribution libraries, the OBJPDS1 utility data set, and user-written routines from the user-specified libraries are link-edited and placed in SYS1.NUCLEUS.

```
//SYSGEN12   JOB
//SG1        EXEC   LINKS
```



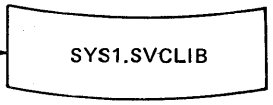
Job 13: Modules from the OBJPDS1 utility data set and modules from the distribution libraries are link-edited and placed in SYS1.SVCLIB and SYS1.LINKLIB.

```
//SYSGEN13   JOB
//SG1        EXEC   LINKS
                    .
                    .
                    .
//SG26       EXEC   LINKS
```



Job 14: The IEHIOSUP utility program builds the XCTL tables in SYS1.SVCLIB for type 4 SVCs.

```
//SYSGEN14   JOB
//SG1        EXEC   IEHIOSUP
```



Job 15: 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. The IEHLIST utility program lists the system catalog, the directories of the partitioned data sets in the new VS1 system, and the VTOC.

```
//SYSGEN15   JOB
//SG1        EXEC   IEHPROGM
//SG2        EXEC   IEHLIST
```

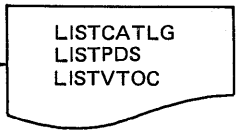


Figure 21 lists the components of the distribution libraries, the names of the distribution libraries from where the components are obtained, and where the components of the distribution libraries are placed during Stage II execution.

Component	Obtained from	Placed in
Access Method Services	SYS1.AOSU0	SYS1.LINKLIB
Assembler	SYS1.AOS03	SYS1.LINKLIB
BDAM	SYS1.AOSD0	SYS1.NUCLEUS SYS1.SVCLIB
BTAM	SYS1.AOS20	SYS1.NUCLEUS SYS1.SVCLIB SYS1.TELCMLIB
Conversational Remote Job Entry	SYS1.AOS0A	SYS1.LINKLIB SYS1.TELCMLIB
Data Management Routines (Primary Routines)	SYS1.AOSD0	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Graphic Programming Services	SYS1.AOSG0	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Graphic Subroutine Package	SYS1.AOS07	SYS1.LINKLIB
Generalized Trace Facility	SYS1.AOS11 SYS1.AOSC5 SYS1.AMACLIB	SYS1.LINKLIB SYS1.MACLIB SYS1.NUCLEUS
Installation Verification Procedure	SYS1.ASAMPLIB	SYS1.SAMPLIB
Independent Utilities, IPL Text, and SMF sample accounting routines	SYS1.ASAMPLIB	SYS1.SAMPLIB
IOS	SYS1.AOSCA SYS1.AOSC5 SYS1.AOSCD SYS1.AOSD7	SYS1.NUCLEUS SYS1.SVCLIB
ISAM	SYS1.AOSD8	SYS1.NUCLEUS SYS1.SVCLIB
Job Entry Subsystem	SYS1.AOSB0 SYS1.AOS00 SYS1.AOSD7 SYS1.AOSG0 SYS1.AOSD0	SYS1.LINKLIB SYS1.SVCLIB
Job Management	SYS1.AOSB0 SYS1.AOSB3 SYS1.AOSBB SYS1.AOSD0	SYS1.NUCLEUS

Figure 21 (Part 1 of 3). The components of the distribution libraries and where they are placed in the system

Component	Obtained from	Placed in
Linkage Editor	SYS1.AOS04	SYS1.LINKLIB
Loader Program	SYS1.AOS05	SYS1.LINKLIB
Macro Library	SYS1.AMACLIB	SYS1.MACLIB
Multiple Console Support	SYS1.AOSB3	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
On-Line Test Executive Program	SYS1.AOS06 SYS1.AOSC5	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Open/Close/End-of-Volume	SYS1.AOSD0	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Overlay Supervisor	SYS1.AOSC2	SYS1.LINKLIB SYS1.NUCLEUS
Parameter Library	SYS1.APARMLIB	SYS1.PARMLIB
Procedure Library	SYS1.APROCLIB	SYS1.PROCLIB
Recovery Management	SYS1.AOSCA SYS1.AOSCD SYS1.AOSCE	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Remote Entry Services (RES)	SYS1.AOSBB SYS1.AOST4 SYS1.ACMDLIB SYS1.ATSOMAC SYS1.ARMTMAC	SYS1.NUCLEUS SYS1.SVCLIB SYS1.RMTMAC
Service Aids	SYS1.AOS12 SYS1.AOSCD SYS1.AMACLIB	SYS1.LINKLIB SYS1.MACLIB SYS1.NUCLEUS
Scheduler	SYS1.AOSC6 SYS1.AOSB3 SYS1.AOSC2 SYS1.AOS00 SYS1.AOSC5 SYS1.AOSB0 SYS1.AOS21	SYS1.LINKLIB SYS1.NUCLEUS
Stage I System Generation Macro Definitions	SYS1.AGENLIB	Not applicable
Stage II System Generation Macro Definitions	SYS1.AMODGEN	Not applicable

Figure 21 (Part 2 of 3). The components of the distribution libraries and where they are placed in the system

Component	Obtained from	Placed in
Supervisor	SYS1.AOSC5 SYS1.AOSCD	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
System Management Facility	SYS1.AOS00 SYS1.APARMLIB	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Teleprocessing	SYS1.AOS21	SYS1.TELCMLIB
Utilities	SYS1.AOSU0	SYS1.LINKLIB SYS1.SVCLIB
VSAM	SYS1.AOSA0	SYS1.SVCLIB

Figure 21 (Part 3 of 3). 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 VS1 system. The new VS1 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 VS1 system should be tested to eliminate any possible malfunctions. Your VS1 system can be tested by the installation verification procedure (IVP). For the information necessary to use IVP, see "Testing the System Control Program".

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

After you have successfully installed your new VS1 system using an existing VS1 system or after you have performed a nucleus generation against the generating system you may want to remove the names of the distribution libraries from the catalog of the generating system. This can be done using IEHPROGM. Place the following IEHPROGM control deck 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.)

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

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.

Renaming the New Nucleus

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

- Rename your old primary nucleus and rename your new 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 were 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 from a direct-access volume.

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 22 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 22. Punching the job stream

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 direct-access volume 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 a direct-access volume, 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 volume. 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 from a Direct-Access Volume

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 a direct-access volume. 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 23 shows the control statements required by IEBEDIT.

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

Figure 23. 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 tape or direct-access volume is mounted. Unless the job stream 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 a direct-access volume, 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 volume. 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 a direct-access volume, 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=SYSGEN4,TYPE=INCLUDE,STEPNAME=(SG3,SG6-SG9)
```

  indicates that job steps 3, 6, 7, 8, and 9 of job 4 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 24 shows an IEBEDIT input deck for restarting Stage II, jobs 13, 14, and 15 from tape. In this example, space allocation for SYS1.LINKLIB was not sufficient, causing the subsequent job steps to fail. Since job 13 included link-edits with output to SYS1.SVCLIB and SYS1.LINKLIB that did not execute, they need to be restarted. Also, although job 14 and 15 executed, they need to be restarted to obtain accurate results. The restart job stream consists of job 13, steps 4 through 24, job 14 (IEHIOSUP), and job 15 (IEHLIST).

```
//RESTART   JOB
//         EXEC  PGM=IEBEDIT
//SYSPRINT   DD  SYSOUT=A
//SYSUT1     DD  UNIT=2400,LABEL=(,NL),DSN=STAGE1, X
//                VOL=SER=JOBSTM,DCB=(RECFM=F,      X
//                BLKSIZE=80,DEN=2),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=SYSGEN13,TYPE=EXCLUDE,     X
                   STEPNAME=(SG4-SG24)
      EDIT         START=SYSGEN14
      EDIT         START=SYSGEN15
/*
```

Figure 24. 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.

Restarting Job 1: IEHPROGM

Restart the entire job.

Restarting Jobs 2, 3, and 4: Assemblies

To restart any assembly step, place the JOB card in front of the EXEC card for that assembly, or use the IEBEDIT utility program if the job stream is on tape or disk.

Restarting Jobs 5 and 6: IEBCOPY

Space for the system data sets copied to in these steps does not have to be reallocated unless the reason for the failure was not enough space for the system data sets. 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 at the beginning of the job or
- Reallocate the system data set affected and restart the job

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 7: IEBCOPY

To restart, space for SYS1.MACLIB, SYS1.SAMPLIB, SYS1.PROCLIB, and SYS1.PARMLIB must be reallocated if space was the problem. Scratch the space for the system data sets on its original volume using the IEHPROGM utility program. Reallocate the space, increasing the amount, for the system data sets on their original volume. Then restart Stage II at the beginning of the job.

Restarting Job 8: IEHPROGM, IFCDIP00

To restart, place the JOB card in front of the job step that failed, or use the IEBEDIT program if the job stream is on tape or disk.

Restarting Job 9: IEBCOPY

Jobs 2, 3, and 4 must execute successfully before this step is executed.

Space for the utility data set copied to does not have to be reallocated unless the reason for the failure was not enough space for the utility data set. If the problem was not enough space, copy the utility data set to a scratch volume using the IEBCOPY utility program. Scratch the space for the utility data set on its original volume using IEHPROGM. Reallocate the space, increasing the amount of space, for the utility data set on its original volume. Use IEBCOPY to copy the utility data set from the scratch volume to its original volume. Restart Stage II at the beginning of the job.

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 10: IEBCOPY, Link-Edits

Space for the system data sets copied to in the IEBCOPY step does not have to be reallocated unless the reason for the failure was not enough space for the system data sets. If the problem was not enough space, you can carry out one of the following 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 the IEHPROGM utility program. 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 at the beginning of the job or
- Reallocate the system data sets affected and restart with the first job that copied that system data set.

If the problem is not insufficient space, restart the job, as any members already copied will not be copied again. (See "Reallocating Space for System Data Sets.")

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 system data set and copy the system 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 insufficient 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 11: 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 12: Nucleus Link-Edit

Jobs 2, 3, 4, and 9 must execute successfully before this step will start. To restart the link-edit step, scratch the system data set and reallocate space for it. Then restart Stage II at the beginning of the job.

Restarting Job 13: Link-Edits

Jobs 2, 3, 4, and 9 must execute successfully before this step 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 14: IEHIOSUP

Jobs 5, 6, and 10 must execute successfully before job 14. Restart job 14 at the beginning.

Restarting Job 15: 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 15.

Reallocating Space for Data Sets

The following sections discuss the reallocation of the utility data sets OBJPDS1-3 and the reallocation of space for the system data sets.

Reallocating space for data sets includes:

- Scratching the space allocated to the data set during the preparation for system generation
- Allocating new space to the system 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 this order 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 Sets

Figure 25 shows the statements used to reallocate the utility data set specified by the OBJPDS1 DD statement in the input deck before Stage II. In this example, the DSNAME parameter of the first OBJPDS1 DD statement must contain the name given to that utility data set in the original allocation.

```
//OBJPDS1   JOB
//STEP1     EXEC  PGM=IEHPROGM      SCRATCH OBJPDS1
//OBJPDS1   DD    DSNAME=SYS1.name,DISP=(OLD,DELETE)
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    DUMMY
//STEP2     EXEC  PGM=IEHPROGM      REALLOCATE OBJPDS1
//OBJPDS1   DD    (New parameters for reallocation
//                of OBJPDS1)
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    DUMMY
/*
//
```

Figure 25. Reallocating the OBJPDS1 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 that needs to be saved (for example, the data set will be recopied completely when system generation is restarted), the

IEHPROGM utility program can be used to scratch and reallocate space for 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, space for the original data set will have to be reallocated, and the contents of the data set will be copied from the temporary data set into the reallocated data set.

The input deck for scratching and reallocating space for 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,      X
//             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),                 X
            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 new system data sets to be reallocated. This DD statement must be the same as the one used in the input deck for the original allocation:

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

  If you used the DATASET macro instructions, this information can be found in JOB 1 of the Stage I documentation listing. If you initialized the system data sets before system generation using IEHPROGM, this information can be found in your IEHPROGM listing.

- A DD * statement (SYSIN).
- A /* statement.

As an example of reallocating space for system data sets that do not contain meaningful data, suppose SYS1.SVCLIB and SYS1.LINKLIB must have their space reallocated. The original space allocation for these two data sets was:

```
//SVCLIB   DD DSNAME=SYS1.SVCLIB,                             X
//            VOLUME=(,RETAIN,SER=AAA111),                    X
//            UNIT=3330,DISP=(,KEEP),                         X
//            SPACE=(CYL,(20,5,75)),                          X
//            LABEL=EXPDT=99350,DCB=(DSORG=POU,               X
//            RECFM=U,BLKSIZE=2048
//LINKLIB  DD DSNAME=SYS1.LINKLIB,
//            VOLUME=(,RETAIN,SER=AAA112),                    X
//            UNIT=3330,DISP=(,KEEP),                         X
//            SPACE=(CYL,(15,5,100)),                         X
//            LABEL=EXPDT=99350,                              X
//            DCB=(RECFM=U,BLKSIZE=13030)
```

The generating system's system residence volume is a 3330. Figure 26 shows the input deck for reallocating space for these system data sets:

```
//SAME      JOB
//STEP1    EXEC PGM=IEHPROGM          -SCRATCH-
//NEWRES     DD UNIT=3330,VOLUME=SER=AAA111,          X
//              DISP=OLD
//LINVOL     DD UNIT=3330,                            X
//              VOLUME=(,RETAIN,SER=AAA112),          X
//              DISP=OLD
//SYSPRINT   DD SYSOUT=A
//SYSIN      DD *
       SCRATCH  DSNAME=SYS1.SVCLIB,VOL=3330=AAA111,  X
                PURGE
       SCRATCH  DSNAME=SYS1.LINKLIB,                 X
                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,   X
//              DISP=OLD
//SYSPRINT   DD SYSOUT=A
//SVCLIB     DD DSNAME=SYS1.SVCLIB,                   X
//              VOLUME=(,RETAIN,SER=AAA111),          X
//              UNIT=3330,DISP=(,KEEP),               X
//              SPACE=(CYL,(20,5,75)),                X
//              LABEL=EXPDT=99350,                    X
//              DCB=(RECFM=U,BLKSIZE=2048,DSORG=POU)
//LINKLIB    DD DSNAME=SYS1.LINKLIB,                  X
//              VOLUME=(,RETAIN,SER=AAA112),          X
//              UNIT=3330,DISP=(,KEEP),               X
//              SPACE=(CYL,(15,5,100)),               X
//              LABEL=EXPDT=99350,                    X
//              DCB=(RECFM=U,BLKSIZE=13030)
//SYSIN      DD DUMMY
/*
//
```

Figure 26. Reallocating SYS1.SVCLIB and SYS1.LINKLIB on the same space

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 space 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 27 illustrates space reallocation for 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,                    X
//             VOLUME=(,RETAIN,SER=SYSTEM),            X
//             UNIT=2314,DISP=(,KEEP),                 X
//             SPACE=(TRK,(7,3),,CONTIG),              X
//             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,                   X
               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,                 X
               VOL=2314=SYSTEM,PURGE
/*
//
```

Figure 27. Reallocating the SYS1.PARMLIB system data set

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 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 VS1 configuration.

The IVP Job Stream

The IVP job stream is contained in your new VS1 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: VS1 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 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,UNIT=132,VOL=SER=SYSRES,LABEL=(,SL),
DSNAME=SYS1.SAMPLIB(IVPJOBS),
DCB=(BLKSIZE=80,LRECL=80,RECFM=F)
```

- 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 assembly/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 VS1 system.

EXAMPLES OF SYSTEM GENERATION

This chapter contains examples of system generation. Included in this chapter are:

- A diagram of a system to be generated (Figure 28)
- 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 28 shows the machine configuration used in this example.

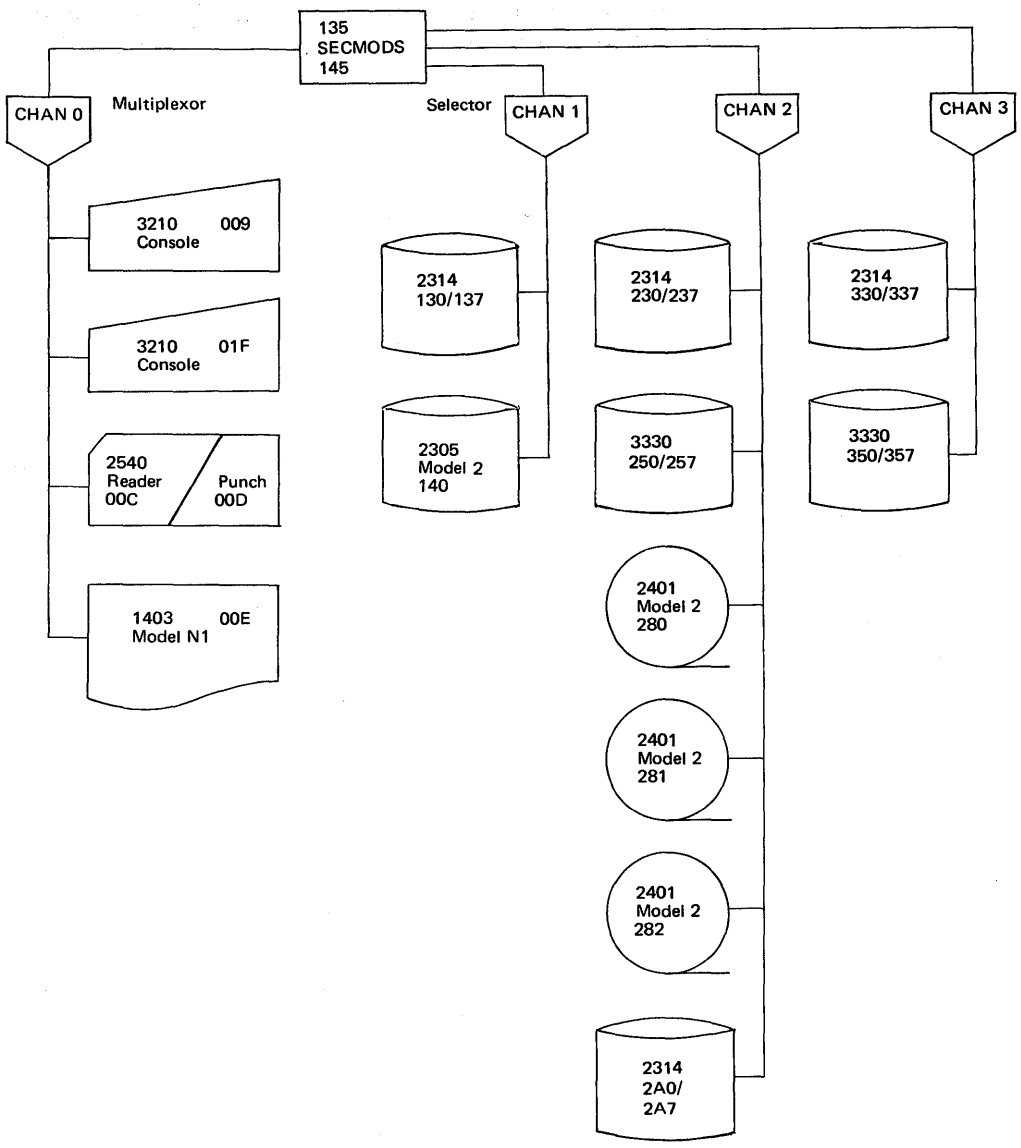


Figure 28. Machine configuration

Job Control Language for Stage I

Figure 29 shows the job control language for Stage I. These statements are placed at the beginning of the job, before the system generation macro instructions.

```
//          JOB    MSGLEVEL=1
//ASM      EXEC    PGM=ASMBLR,PARM='DECK,NOLOAD'
//SYSPRINT  DD     SYSOUT=A,SPACE=(CYL,(20,3))
//SYSLIB    DD   DSN=SYS1.AGENLIB,DISP=SHR,                X
//                VOL=SER=DLIBA2,UNIT=SYSDA
//SYSUT1    DD  UNIT=2314,SPACE=(CYL,(20,5))
//SYSUT2    DD  UNIT=2314,SPACE=(CYL,(20,5))
//SYSUT3    DD  UNIT=2314,SPACE=(CYL,(20,5))
//SYSPUNCH  DD     SYSOUT=B
//SYSIN     DD    *
```

Figure 29. Job control language for executing Stage I

System Generation Macro Instructions for a Complete System Generation

Figure 30 shows the system generation macro instructions for Stage I. The DATASET macro is used, so space will be allocated to the system data sets and they will be cataloged during Stage II. The new system will support the devices shown in Figure 28. The linkage editor and loader programs, ISAM, and SMF will be included. The new system will have five partitions. Programming support for BTAM, GPS, and OCR will be excluded and UCS images will be included. The paging device will be a 2305-2.

```
CPU       CENPROCS   MODEL=135R,SECMODS=(145R)
CTRLPROG  CTRLPROG   OPTIONS=(BLDL,NODDR,NODDRSYS,RER,TRSVCTBL), X
                     RESIDNT=(ACSMETH,RENTCODE,TRSVC),          X
                     SYSQUE=12,                                 X
                     VIRTUAL=2048,                              X
                     VSAM=INCLUDE,                              X
                     DYNPART=(P1-P3),                           X
                     DYNINTR=(10,64,50,256),                    X
                     MAXIO=36,                                  X
                     TRACE=500,                                 X
                     OVERLAY=ADVANCED,                          X
                     FETCH=PCI,                                 X
                     ASCII=INCLUDE
ACESS     DATAMGT    ACSMETH=(ISAM)
EDITOR    EDITOR     SIZE=(192,64)
JES       JES        RDR=(R=1,Y=5,B=800),                       X
                     WTR=(W=2,U=0,Z=6,B=1600),                  X
                     SPOLVOL=(SYSRSM),                          X
                     BUFSIZE=436,                               X
                     NUMBUFS=10,                                X
                     JOUTLIM=0,                                 X
                     SWDSLMT=15,                                X
                     STEPWTP=15,                                X
                     WTLRCDS=100
SCHEDULR  SCHEDULR   SMF=FULL,                                  X
                     JOBQFMT=5,                                 X
                     JOBQLMT=40,                                X
                     JOBQTMT=60,                                X
                     REPLY=8,WTOBFRS=16,                        X
                     CONSOLE=009,                               X
                     IOC=009,                                   X
                     ALTCONS=01F
PARTNS    PARTITNS   P0(C-S,S-128),    SYSTEM PARTITION        X
                     P1(C-ABC,S-256),                           X
                     P2(C-ABCDFGKLM,S-256),                     X
                     P3(C-ABCDFGKLM,S-256),                     X
                     P4(C-FGHI,S-128)
LOADER    LOADER     LIN=SYSLIN,                                X
                     LIB=SYSLIB,                                X
                     PRINT=SYSLOUT,                             X
                     SIZE=128,                                  X
                     PARM=(PRINT,MAP)
MACLIB    MACLIB     EXCLUDE=(BTAM,GPS,OCR)
PAGE      PAGE   DEV=2305-2,VOLNO=23051,SIZE=(CYL,50)
UCS       UCS    IMAGE=(AN,HN,PCAN,PCHN),DEFAULT=(HN,AN),
          DATASET    SYSCTLG,SPACE=(CYL,(5,1))
          DATASET    LINKLIB,SPACE=(CYL,(30,2,100))
          DATASET    SVCLIB,SPACE=(CYL,(25,2,100))
          DATASET    PROCLIB,SPACE=(CYL,(5,2,25))
          DATASET    SYSJOBQE,SPACE=(CYL,(30))
```

Figure 30 (Part 1 of 2). System generation macro instructions for a complete system generation

```
         DATASET    MACLIB,SPACE=(CYL,(10,2,55))
         DATASET    IMAGELIB,SPACE=(CYL,(2,,5))
         DATASET    MANX,SPACE=(TRK,(25))
         DATASET    MANY,SPACE=(TRK,(25))
         DATASET    NUCLEUS,SPACE=(CYL,(10,,2))
         DATASET    PARMLIB,SPACE=(CYL,(2,,3))
         DATASET    SYSPOOL,VOL=(SYSRSM,2314)  PRE ALLOCATED
         DATASET    SAMPLIB,SPACE=(CYL,(10,,3))
CHAN0    CHANNEL    ADDRESS=(0),TYPE=MULTIPLEXOR
CHAN13   CHANNEL    ADDRESS=(1,2,3),TYPE=SELECTOR
C009     IODEVICE   UNIT=3210,ADDRESS=009
C01F     IODEVICE   UNIT=3210,ADDRESS=01F
DA130    IODEVICE   UNIT=2314,ADDRESS=(130,8)
DA230    IODEVICE   UNIT=2314,ADDRESS=(230,8)
DA2A0    IODEVICE   UNIT=2314,ADDRESS=(2A0,8)
DA330    IODEVICE   UNIT=2314,ADDRESS=(330,8)
Z140     IODEVICE   UNIT=2305,MODEL=2,ADDRESS=(140)
M250     IODEVICE   UNIT=3330,ADDRESS=(250,8)
M350     IODEVICE   UNIT=3330,ADDRESS=(350,8)
READER   IODEVICE   UNIT=2540R,ADDRESS=(00C),MODEL=1,                X
               FEATURE=(CARDIMAGE)
P00D     IODEVICE   UNIT=2540P,ADDRESS=(00D),                        X
               FEATURE=(CARDIMAGE),MODEL=1
P00E     IODEVICE   UNIT=1403,ADDRESS=00E,                           X
               FEATURE=(UNVCHSET),MODEL=N1
T2803    IODEVICE   UNIT=2401,ADDRESS=(280,3),MODEL=2,               X
               FEATURE=(9-TRACK)
SYSDA    UNITNAME   NAME=SYSDA,                                      X
               UNIT=((130,8),(230,8),(2A0,8),(330,8),               X
               (250,8),140,(350,8))
Z2305    UNITNAME   NAME=2305,                                       X
               UNIT=(140)
SYSSQ    UNITNAME   NAME=SYSSQ,                                      X
               UNIT=((130,8),(230,8),(2A0,8),(330,8),               X
               (250,8),140,(350,8))
SYSIN    UNITNAME   NAME=SYSIN,                                      X
               UNIT=((130,2),(230,2),(330,2))
SYSOUT   UNITNAME   NAME=SYSOUT,                                     X
               UNIT=((130,2),(230,2),(330,2))
PUNCH    UNITNAME   NAME=PUNCH,                                      X
               UNIT=(00D)
TAPE     UNITNAME   UNIT=((280,3)),                                  X
               NAME=TAPE
TAPE9    UNITNAME   UNIT=((280,3)),                                  X
               NAME=TAPE9
GEN      GENERATE   GENTYPE=ALL,                                     X
               OBJPDS1=SYS1.OBJPDS1,OBJPDS2=SYS1.OBJPDS2,           X
               OBJPDS3=SYS1.OBJPDS3,RESVOL=(SYSTEM,3330)
         END
/*
```

Figure 30 (Part 2 of 2). System generation macro instructions for a complete system generation

System Generation Macro Instructions for a Nucleus Generation

A new nucleus is to be added to the new System Control Program. Figure 31 shows the system generation macro instructions needed to add a new nucleus to the system specified in Figure 30.

For a nucleus generation, the CENPROCS, CHANNEL, and IODEVICE parameters and subparameters must be specified exactly as they were for the last complete system generation. The SCHEDULR parameters and subparameters must be respecified exactly as they were for the last complete system generation except for the REPLY and WTOBFRS parameters and subparameters. These may be changed (as they were in this example). Except for some of the subparameters in the RESIDNT parameter of the CTRLPROG macro instruction, and the DEBTINC, DEBTSZE, DYNPART, DYNINTR, FETCH, SYSQUE, TMSLICE, and TRACE parameters, the CTRLPROG macro instruction must be specified exactly the same as in the last complete system generation. Also in this nucleus generation, one additional page data set has been added to the system and the number of partitions has been changed from five to three.

Except for SYS1.SYSPOOL, which is not on the system residence volume and therefore must be specified in a DATASET macro, the other system data sets need not be respecified for a nucleus generation. The DATASET macro instructions need not be removed from the deck because they will be ignored.

```
CPU       CENPROCS   MODEL=135R,SECMODS=(145R)
CTRLPROG  CTRLPROG   OPTIONS=(BLDL,NODDR,NODDRSYS,RER,                X
               TRSVCTBL),RESIDNT=(RENTCODE,TRSCV),                   X
               SYSQUE=12,                                            X
               VIRTUAL=2048,                                         X
               VSAM=INCLUDE,                                         X
               MAXIO=36,                                             X
               TRACE=100,                                            X
               OVERLAY=ADVANCED,                                     X
               FETCH=PCI,                                            X
               ASCII=INCLUDE
ACESS     DATAMGT    ACSMETH=(ISAM)
PAGE      PAGE  DEV=2305-2,VOLNO=23051,SIZE=(CYL,50)
PAGE2     PAGE  DEV=3330,VOLNO=111111,SIZE=(CYL,30)
PARTNS    PARTITNS   P0(C-S,S-128),     SYSTEM PARTITION            X
               P1(C-ABC,S-128),                                      X
               P2(C-ABCDFGKLM,S-128)
SCHEDULR  SCHEDULR   SMF=FULL,                                       X
               JOBQFMT=5,                                            X
               JOBQLMT=40,                                           X
               JOBQTMT=60,                                           X
               REPLY=2,WTOBFRS=2,                                    X
               CONSOLE=009,                                          X
               IOC=009,                                              X
               ALTCONS=01F
          DATASET    SYSCTLG,SPACE=(CYL,(5,1))
          DATASET    LINKLIB,SPACE=(CYL,(30,2,100))
          DATASET    SVCLIB,SPACE=(CYL,(25,2,100))
          DATASET    PROCLIB,SPACE=(CYL,(5,2,25))
          DATASET    SYSJOBQE,SPACE=(CYL,(30))
          DATASET    MACLIB,SPACE=(CYL,(10,2,55))
          DATASET    IMAGELIB,SPACE=(CYL,(2,,5))
          DATASET    MANX,SPACE=(TRK,(25))
          DATASET    MANY,SPACE=(TRK,(25))
          DATASET    NUCLEUS,SPACE=(CYL,(10,,2))
          DATASET    PARMLIB,SPACE=(CYL,(2,,3))
          DATASET    SYSPOOL,VOL=(SYSRSM,2314)  PRE ALLOCATED
          DATASET    SAMPLIB,SPACE=(CYL,(2,,3))
CHAN-0    CHANNEL    ADDRESS=(0),TYPE=MULTIPLEXOR
CHAN1-3   CHANNEL    ADDRESS=(1,2,3),TYPE=SELECTOR
C009      IODEVICE   UNIT=3210,ADDRESS=009
C01F      IODEVICE   UNIT=3210,ADDRESS=01F
DA-130    IODEVICE   UNIT=2314,ADDRESS=(130,8)
DA-230    IODEVICE   UNIT=2314,ADDRESS=(230,8)
DA-2A0    IODEVICE   UNIT=2314,ADDRESS=(2A0,8)
DA-330    IODEVICE   UNIT=2314,ADDRESS=(330,8)
Z-140     IODEVICE   UNIT=2305,MODEL=2,ADDRESS=(140)
M-250     IODEVICE   UNIT=3330,ADDRESS=(250,8)
```

Figure 31 (Part 1 of 2). System generation macro instructions for a nucleus generation

```
M-350     IODEVICE    UNIT=3330,ADDRESS=(350,8)
READER    IODEVICE    UNIT=2540R,ADDRESS=(00C),MODEL=1,                X
                FEATURE=(CARDIMAGE)
P-00D     IODEVICE    UNIT=2540P,ADDRESS=(00D),                        X
                FEATURE=(CARDIMAGE),MODEL=1
P-00E     IODEVICE    UNIT=1403,ADDRESS=00E,                           X
                FEATURE=(UNVCHSET),MODEL=N1
T-280-3   IODEVICE    UNIT=2401,ADDRESS=(280,3),MODEL=2,               X
                FEATURE=(9-TRACK)
GEN       GENERATE    GENTYPE=(NUCLEUS,2),                             X
                OBJPDS1=SYS1.OBJPDS1,OBJPDS2=SYS1.OBJPDS2,             X
                OBJPDS3=SYS1.OBJPDS3,                                  X
                RESVOL=(SYSTEM,3330)
          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

I/O devices and channels can be added to or deleted from the system. This example shows the macro instructions required to modify the machine configuration for the system that was generated using the macros shown in Figure 30. The new machine configuration will include an additional channel, a 3215 console-printer keyboard at address 01F, a 3211 printer at address 004, a 2501 card reader at address 014, a 2520 card reader at address 015, 2420 tape drives at addresses 380-386, a 2401 model 5 tape drive at address 389, a 2401 model 3 tape drive at address 38A, and 2401 model 1 tape drives with the unit name TAPE7 at addresses 38B and 38C. Support for the 3210 console-printer keyboard at address 01F will be deleted.

The CENPROCS, CTRLPROG, and SCHEDULR macros in this example must be respecified exactly as they were in the last complete system generation.

Except for SYS1.SYSPOOL, which is not on the system residence volume and therefore must be respecified in a DATASET macro, the other system data sets need not be respecified for an I/O device generation.

```
CPU       CENPROCS   MODEL=135R,SECMODS=(145R)
CTRLPROG  CTRLPROG   OPTIONS=(BLDL,NODDR,NODDRSYS,RER,TRSVCTBL),  X
                RESIDNT=(ACSMETH,RENTCODE,TRSVC),                X
                SYSQUE=12,                                       X
                VIRTUAL=2048,                                    X
                VSAM=INCLUDE                                     X
                DYNPART=(P1-P3),                                 X
                DYNINTR=(10,64,50,256),                          X
                MAXIO=36,                                        X
                TRACE=500,                                       X
                OVERLAY=ADVANCED,                                X
                FETCH=PCI,                                       X
                ASCII=INCLUDE
ACESS     DATAMGT    ACSMETH=(ISAM)
SCHEDULR  SCHEDULR   SMF=FULL,                                   X
                JOBQFMT=5,                                       X
                JOBQLMT=40,                                      X
                JOBQTMT=60,                                      X
                REPLY=8,WTOBFRS=16,                              X
                CONSOLE=009,                                     X
                IOC=009,                                         X
                ALTCONS=01F
PARTNS    PARTITNS   P0(C-S,S-128),    SYSTEM PARTITION          X
                P1(C-ABC,S-256),                                 X
                P2(C-ABCDFGKLM,S-256),                           X
                P3(C-ABCDFGKLM,S-256),                           X
                P4(C-FGHI,S-128)
PAGE      PAGE   DEV=2305-2,VOLNO=23051,SIZE=(CYL,50)
UCS       UCS    IMAGE=(AN,HN,PCAN,PCHN),                        X
                DEFAULT=(HN,AN)
          DATASET    SYSPOOL,VOL=(SYSRSM,2314) PRE ALLOCATED
CHAN0     CHANNEL    ADDRESS=(0),TYPE=MULTIPLEXOR
CHAN14    CHANNEL    ADDRESS=(1,2,3,4),TYPE=SELECTOR
C009      IODEVICE   UNIT=3210,ADDRESS=009
C01F      IODEVICE   UNIT=3215,ADDRESS=01F
P004      IODEVICE   UNIT=3211,ADDRESS=004
R014      IODEVICE   UNIT=2501,ADDRESS=014,MODEL=B1
R015      IODEVICE   UNIT=2520,ADDRESS=015,MODEL=B1
DA130     IODEVICE   UNIT=2314,ADDRESS=(130,8)
DA230     IODEVICE   UNIT=2314,ADDRESS=(230,8)
DA2A0     IODEVICE   UNIT=2314,ADDRESS=(2A0,8)
DA330     IODEVICE   UNIT=2314,ADDRESS=(330,8)
Z140      IODEVICE   UNIT=2305,MODEL=2,ADDRESS=(140)
M250      IODEVICE   UNIT=3330,ADDRESS=(250,8)
M350      IODEVICE   UNIT=3330,ADDRESS=(350,8)
READER    IODEVICE   UNIT=2540R,ADDRESS=(00C),MODEL=1,          X
               FEATURE=(CARDIMAGE)
```

Figure 32 (Part 1 of 2). System generation macro instructions for an I/O device generation

```
P00D     IODEVICE   UNIT=2540P,ADDRESS=(00D),MODEL=1,                    X
                    FEATURE=(CARDIMAGE)
P00E     IODEVICE   UNIT=1403,ADDRESS=00E,                               X
                    FEATURE=(UNVCHSET),MODEL=N1
T280     IODEVICE   UNIT=2401,ADDRESS=(280,3),MODEL=2,                   X
                    FEATURE=(9-TRACK)
T380     IODEVICE   UNIT=2420,ADDRESS=(380,7),OPTCHAN=4,MODEL=5
T389     IODEVICE   UNIT=2401,ADDRESS=389,MODEL=5,OPTCHAN=4,             X
                    FEATURE=(9-TRACK,DUALDENS)
T38A     IODEVICE   UNIT=2401,ADDRESS=38A,OPTCHAN=4,MODEL=3
T38B     IODEVICE   UNIT=2401,ADDRESS=(38B,2),MODEL=1,OPTCHAN=4,         X
                    FEATURE=(7-TRACK,MDECOMPAT)
SYSDA    UNITNAME   NAME=SYSDA,                                          X
                    UNIT=((130,8),(230,8),(2A0,8),(330,8),              X
                    (250,8),140,(350,8))
Z2305    UNITNAME   NAME=2305,
                    UNIT=(140)
SYSSQ    UNITNAME   NAME=SYSSQ,                                          X
                    UNIT=((130,8),(230,8),(2A0,8),(330,8),              X
                    (250,8),140,(350,8))
SYSIN    UNITNAME   NAME=SYSIN,                                          X
                    UNIT=((130,2),(230,2),(330,2))
SYSOUT   UNITNAME   NAME=SYSOUT,                                         X
                    UNIT=((130,2),(230,2),(330,2))
PUNCH    UNITNAME   NAME=PUNCH,                                          X
                    UNIT=(00D)
TAPE     UNITNAME   UNIT=((280,3),(380,7),(389,4)),                      X
                    NAME=TAPE
TAPE9    UNITNAME   UNIT=((280,3),(380,7),(389,2)),                      X
                    NAME=TAPE9
TAPE7    UNITNAME   UNIT=((38B,2)),                                      X
                    NAME=TAPE7
GEN      GENERATE   GENTYPE=(IO,1),                                      X
                    OBJPDS1=SYS1.OBJPDS1,OBJPDS2=SYS1.OBJPDS2,          X
                    OBJPDS3=SYS1.OBJPDS3,RESVOL=(SYSTEM,3330)
         END
```

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 Drives	
2400	2400 9-track magnetic tape drive having an 800 bits-per-inch (density) capability when the dual-density feature is not installed or a 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
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	
2305-2	2305 fixed-head disk storage (Model 2)
2314/2319	2314/2319 direct-access storage facility
3330/3333	3330 disk storage drive
Unit Record Equipment	
1052	1052 printer keyboard (Model 7)
1053	1053 printer (Model 4)
1403	1403 printer or 1404 printer (continuous form only)
1442	1442 card read punch
1443	1443 printer (Model N1)
2495	2495 tape cartridge reader
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
3158	3158 display console keyboard – System/370 Model 158

Device Type	Description
Unit Record Equipment (continued)	
3210	3210 console printer keyboard
3211	3211 printer
3213	3213 console printer – System/370 Model 155II
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	
7770	7770 audio response unit
Remote Analysis	
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 9-track tapes provided by IBM or on 9-track tapes that you provide.

The starter system is distributed in a dump/restore format for either a 2314/2319 or 3330/3333.

The distibution libraries are unloaded partitioned data sets that are distributed in the IEBCOPY unload/load format on either two 9-track 1600 BPI tapes or two 9-track 800 BPI tapes.

There are 38 distribution libraries. The first file on each tape contains the IEBCOPY utility program for copying the distribution libraries from tape to a direct-access volume. The remaining files on the tapes each contain a distribution library. Figure 33 lists the distribution libraries.

SYS1.AOS00	SYS1.AOSCA
SYS1.AOS03	SYS1.AOSCD
SYS1.AOS04	SYS1.AOSCE
SYS1.AOS05	SYS1.AOSD0
SYS1.AOS06	SYS1.AOSD7
SYS1.AOS07	SYS1.AOSD8
SYS1.AOS0A	SYS1.AOSG0
SYS1.AOS10	SYS1.AOST4
SYS1.AOS11	SYS1.AOSU0
SYS1.AOS12	SYS1.ACMDLIB
SYS1.AOS20	SYS1.AGENLIB
SYS1.AOS21	SYS1.AMACLIB
SYS1.AOSA0	SYS1.AMODGEN
SYS1.AOSB0	SYS1.APARMLIB
SYS1.AOSB3	SYS1.APROCLIB
SYS1.AOSBB	SYS1.APVTMACS (optional)
SYS1.AOSC2	SYS1.ARMTMAC
SYS1.AOSC5	SYS1.ASAMPLIB
SYS1.AOSC6	SYS1.ATSOMAC

Figure 33. Contents of the distribution library tapes

The starter system and the distribution libraries must be on direct-access volumes for Stage I and Stage II execution. See "Preparing for System Generation" for the procedures for initializing the volumes for the starter system and the distribution libraries, dumping the starter system to a direct-access volume, and copying the distribution libraries to a direct-access volume(s).

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 the figure.

Message Code	Code	
IEI	*s*,***	IEI*aaannn 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 abbreviation of the system generation macro instruction at which the error was detected:

aaa	*Macro Instruction*	*aaa*	*Macro Instruction*
CEN	CENPROCS	LDR	LOADER
CHA	CHANNEL	LNK	LINKLIB
CKP	CKPTREST	MAL	MACLIB
CTR	CTRLPROG	PAG	PAGE
DAT	DATAMGT	PTN	PARTITNS
DTS	DATASET	RES	RESMODS
EDT	EDITOR	SCN	SECONSLE
GEN	GENERATE	SCH	SCHEDULR
GPH	GRAPHICS	SVC	SVCTABLE
IOD	IODEVICE	SVL	SVCLIB
JES	JES	UCS	UCS
		UNI	UNITNAME

nnn = Message serial number

text = Message text

Figure 34. System generation error and warning messages

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. "CHANNEL2" 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 processs 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,* * * IEICTR902 DYNAMIC DEVICE RECONFIGURATION ASSUMED WITH OPTIONS=DDRNSL--OPTIONS=NODDR IGNORED

Explanation: The user specified conflicting subparameters in the OPTIONS parameter in the CTRLPROG macro instruction, so the DDRNSL option was taken and the NODDR option is ignored.

Severity Code: 0

User Response: The user may want to correct this condition; otherwise, there is no user response.

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 name***component 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 responsible 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.

APPENDIX D: FUNCTIONALLY EQUIVALENT I/O DEVICES

I/O Device	Equivalent to	Usage
2319	2314	The 2319 direct-access storage facility 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.
2596	1442	The 2596 card read punch is functionally equivalent to the 1442 N1 card read punch. If the 2596 is specified as a 1442 N1 at system generation time, it becomes directly accessible by an assembler language program which enables it to read and punch 96-column cards. For information on the data management macros used to access a 2596, see *OS/VS Data Management Services Guide*. *Use:* • The device may only be used for direct input from or direct output to an assembler language program. • The device may not be assigned as a job stream (SYSIN/SYSOUT) device. • A data set on the device may not be opened as both an input and output data set at the same time. • If both the 2596 and the 1442 devices are being used in the same system, the device address cannot be included in a device class at system generation. • If both the 2596 and the 1442 are being used in the same system, they must be assigned by their respective device addresses. • If the system contains DDR, the 2596 cannot be replaced with the 1442 by the SWAP command. A 2596 can be specified as the unitname of the device. To refer to a 2596 as a 2596, specify UNIT=1442 in an IODEVICE macro and NAME=2596 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: AUTOMATIC DEVICE STATUS INITIALIZATION

Support can be included during system generation that will treat not-ready devices and devices that are not currently attached to the system in an offline status. To include this additional support, place the following statements before the system generation macro instructions in the Stage I input deck:

```
          COPY     SGGBLPAK
&SGDEVSA(1)        SETC     'xx'
```

The value you specify will determine what devices are to be treated as being offline. The value you specify, in hexadecimal notation, can be one or any combination of the following:

Value	Devices
08	for unit record devices
10	for graphics devices
20	for direct-access devices
40	for communications devices
80	for magnetic tape drives
F8	all devices

APPENDIX G: DETERMINING THE MAXIMUM NUMBER OF I/O DEVICES

The maximum number of I/O devices could be less than 768. The maximum number is based upon the size of the unit control blocks (UCBs) and the number of request queue elements (RQEs) in the system. The following formula can be used to calculate the maximum number of I/O devices:

$$\frac{A - B - (C*D)}{E}$$

where

A = the maximum address of the RQEs in the system. The maximum address must not exceed X'7FFF'

B = the first level interrupt handler requirement. In this calculation, assume the maximum value of X'FFF'

C = the number of RQEs in the system

D = the length of one RQE which is X'14'

E = the length of the largest UCB in the system. UCB lengths can be found in *OS/VS1 System Data Areas.*

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.

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.

buffer management: The part of job management that provides a central buffer-handling facility for the job entry subsystem, maintaining unassigned buffers in a buffer pool, assigning them to requestors, and maintaining a table showing the status of each buffer.

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.

concatenated data sets: A group of logically connected data sets that are treated as one data set for the duration of a job step.

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.

conversational remote job entry (CRJE): An operating system facility for entering job control language statements from a remote terminal, and causing the scheduling and execution of the jobs described. The terminal user is prompted for missing operands or corrections.

CPU: Central processing unit.

CRJE: Conversational remote job entry.

damage assessment routine (DAR): A routine that attempts recovery from system failure.

DAR: Damage assessment routine.

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 management: 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.

DDR: Dynamic device reconfiguration.

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 VS1 components or combinations of VS1 components that the user restores to disk for subsequent inclusion in a new system.

dynamic address translation (DAT): The change of a virtual storage address to a real storage address during execution of an instruction.

dynamic device reconfiguration (DDR): A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

EOF: End of file.

EOJ: End of job.

EREP: The environmental 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 VS1 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 page: A page in real storage that is not to be paged out.

GDG: Generation data group.

generalized trace facility (GTF): A system service that can be invoked to assist in problem determination and diagnosis.

generation data group (GDG): A collection of data sets that are kept in chronological order; each data set is called a generation data set.

generation data set: One generation of a generation data group.

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.

GTF: Generalized trace facility.

hard copy: A printed copy of machine output in a visually readable form, for example, printed reports, listings, documents, and summaries.

hardcopy 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.

*American National Standard Definition

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.

JES: Job entry subsystem.

JES reader: The part of the job entry subsystem that controls the input streams and its associated job control statements.

JES writer: The part of the job entry subsystem that controls the output of specified data sets.

job entry subsystem (JES): A system facility for spooling, job queueing, and managing the scheduler work area.

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 and the scheduler work area data sets (SWADS).

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.

load module: The output of the linkage editor; a program in a format suitable for loading into virtual storage for execution.

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 actions.

MCH: Machine check handler.

MCS: Multiple console support.

***MICR**: Magnetic ink character recognition.

***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.

multiple console support (MCS): An optional feature of the VS1 configuration of the control program that permits selective message routing to up to thirty-two operator's consoles.

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.

*American National Standard Definition

new system data set initialization: The process of allocating space for data sets and cataloging the data sets in the new system catalog.

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.

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 generation: The creation of a new nucleus for the System Control Program.

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.

***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, 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.

*American National Standard Definition

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.

overlay supervisor: A routine that controls the proper sequencing and positioning of segments of computer programs in limited storage during their execution.

page: A fixed-length block of instructions, data, or both, that can be transferred between real storage and external page storage.

page data set: A data set in external page storage, in which pages are stored.

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 programs.

paging supervisor: A part of the supervisor that allocates and releases real storage space 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.

program options: Features of the VS1 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, in systems with multiple console support (MCS), to route the message to the proper console.

scheduler work area data set (SWADS): A data set that contains most of the job management control blocks; there is one SWADS for each initiator.

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

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.

spool management: The function of the job entry central services (JECS) responsible for reading and writing input and output streams on the SYS1.SYSPOOL data set.

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

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.

*American National Standard Definition

starter system: An IBM-supplied VS1 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 new system data set and volume initialization and Stage II processing. It is used when there is not an existing VS1 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.

SWADS: Scheduler work area data set.

***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 VS1 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: A data set maintained as part of the SYS1.SYSPOOL system data set that is used by programmers and operators to record information.

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

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 a 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 printer feature that permits the use of a variety of character arrays.

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.

A

B

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

123

READER'S COMMENT FORM

OS/VS1 System Generation Reference GC26-3791-1

Your comments about this publication will help us to produce better publications for your use. If you wish to comment, please use the space provided below, giving specific page and paragraph references.

Please do not use this form to ask technical questions about the system or equipment or to make requests for copies of publications. Instead, make such inquiries or requests to your IBM representative or to the IBM Branch Office serving your locality.

Reply requested

Yes ☐

No ☐

Name _____

Job Title _____

Address _____

_____ Zip _____

No postage necessary if mailed in the U S A

YOUR COMMENTS, PLEASE . . .

This publication is one of a series which serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

fold fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Programming Center—Publishing
Department D58
Monterey and Cottle Roads
San Jose, California 95114

fold fold

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)

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)