GC26-3792-8
File No. S370-34

Systems

OS/VS2 System Programming Library: System Generation Reference

Release 3.8

IBM

This publication was produced using the
IBM Document Composition Facility
(program number 5748-XX9)
and the master was printed on the
IBM 3800 Printing Subsystem.

Ninth Edition (October 1980)

This is a major revision of, and makes obsolete, GC26-3792-7, its technical newsletters GN26-0946 and GN26-0957, and the System Library Supplements GD26-6015-0, GD26-6016-0, GD26-6020-0, and GD26-6024-1.

This edition applies to Release 1.0 of Data Facility/Device Support, Program Product 5740-AM7, as well as to Release 3.8 of OS/VS2 MVS, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Amendments" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

This publication is intended for system programmers who are to install an MVS (multiple virtual storage) system control program.

This publication provides you with the information necessary to install an MVS system control program. The information contained in this publication is divided into chapters.

"Introduction" briefly defines system generation and discusses the types of system generation. It also provides a list and explanation of the procedures for generating an MVS system control program.

"Requirements for Generating an MVS 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.

"Specifying the MVS 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 when you code them to specify the MVS system control program. Included are:

- The required and optional macro instructions for the 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 and Defining the System Data Sets" describes the procedures for allocating space for the system data sets and cataloging them in the master 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 of the system data sets.

"Defining the Page and Swap Data Sets" describes the procedures for allocating space to the page and swap data sets and cataloging them in the master catalog.

"Preparing for System Generation" presents a general discussion about the procedures that need to be done to prepare for 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 Stage I or Stage II.

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

"Verifying New Device Allocation Tables" discusses the Eligible Device Table verification program, IEFEB400.

"Example of System Generation" presents an example that illustrates macro specification for a complete 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 and distribution library tapes and lists the contents of the distribution library tapes.

"Appendix C. Installation Device Preference Table" lists the devices that may be specified to create a device preference table during system generation.

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

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

"Appendix F. System Generation Messages" lists and explains the messages that are produced by the macros during Stage I.

"Appendix G. Using the Master Catalog from the Old Production System" provides an example of and a procedure for including the master catalog, the page data sets, and the SYS1.STGINDEX data set in your new system.

The following publications provide detailed explanations of OS/VS2 operations and the options available in OS/VS2:

Introduction to Virtual Storage in System/370, GR20-4260

OS/VS2 Release 3 Guide, GC28-0700

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

OS/VS2 System Programming Library: Initialization and Tuning Guide, GC28-0681

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

OS/VS2 System Programming Library: Supervisor, GC28-0628

OS/VS2 System Programming Library: TSO, GC28-0629

The following publication provides information on converting from an OS/MVT or VS2 Release 1 system to MVS:

OS/VS2 Conversion Notebook, GC28-0689

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

OS/VS2 Access Method Services, GC26-3841

OS/VS—DOS/VS—VM/370 Assembler Language, GC33-4010

OS/VS2 MVS Utilities, GC26-3902

OS/VS2 JCL, GC28-0692

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

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

An Introduction to the IBM 3250 Graphic Display System, GA33-3035

IBM System/370 Principles of Operation, GA22-7000

IBM 3800 Printing Subsystem Programmers Guide, GC26-3846

Introduction to 3375 Direct Access Storage, GA26-1666

Introduction to 3380 Direct Access Storage, GA26-1662

Introduction to JES3, GC28-0607

OS/VS—VM/370 Assembler Programmer's Guide, GC33-4021

OS/VS BTAM, GC27-6980

OS/VS2 MVS Checkpoint/Restart, GC26-3877

OS/VS2 Data Areas, SYB8-0606

OS/VS2 MVS Data Management Macro Instructions, GC26-3873

OS/VS2 MVS Data Management Services Guide, GC26-3875

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/VS2 MVS JES2 Logic, SY24-6000

Network Job Entry Facility for JES2 Logic, LY24-6001

OS/VS2 MVS System Programming Library: JES2, GC23-0002

System Programming Library: Network Job Entry Facility for JES2, SC23-0003

OS/VS Mass Storage Control Table Create, GC35-0013

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

OS/VS Message Library: VS2 Routing and Descriptor Codes, GC38-1102

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

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

OS/VS2 MVS Multiprocessing: An Introduction and Guide to Writing Operating and Recovery Procedures, GC28-0952

Operator's Library: OS/VS2 MVS System Commands, GC38-0229

Operator's Library: OS/VS2 MVS JES2 Commands, GC23-0007

Operator's Library: OS/VS TCAM, GC30-3037

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

OS/VS2 Supervisor Services and Macro Instructions, GC28-0683

OS/VS2 System Programming Library: System Management Facilities (SMF), GC28-0706

OS/VS System Modification Program (SMP) System Programmers Guide, GC28-0673

OS/VS2 System Programming Library: Debugging Handbook, Volume 1, GC28-0708

OS/VS2 System Programming Library: Debugging Handbook, Volume 2, GC28-0709

OS/VS2 System Programming Library: Debugging Handbook, Volume 3, GC28-0710

OS/VS2 System Programming Library: JES3, GC28-0608

OS/VS2 System Programming Library: OLTEP, GC28-0675

OS/VS2 System Programming Library: Service Aids, GC28-0674

OS/VS2 System Programming Library: SYS1.LOGREC Error Recording, GC28-0677

OS/VS Tape Labels, GC26-3795

OS TCAM Concepts and Facilities, GC30-2022

OS/VS TCAM Programmer's Guide, GC30-2051

OS/VS2 TSO Command Language Reference, GC28-0646

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

OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide, GC26-3838

Reference Manual for IBM 3830 Storage Control Model 2, GA26-1617

VTAM Concepts and Planning, GC27-6998

OS/VS2 VIO Logic, SY26-3834

3350/3344 Installation and Conversion Guide, GC20-1780

SUMMARY OF AMENDMENTS

DATA FACILITY/DEVICE SUPPORT (DF/DS) RELEASE 1.0 PROGRAM PRODUCT

NEW PROGRAMMING SUPPORT

Information to support the IBM 3375 Direct Access Storage and IBM 3380 Direct Access Storage is now included. For additional information, see Introduction to 3375 Direct Access Storage, or Introduction to 3380 Direct Access Storage.

Entries for the 3375 and 3380 have been added to the following:

- VOL parameter of the DATASET macro
- RESVOL parameter of the GENERATE macro
- List of burst devices in the IODEVICE macro
- AP parameter of the IODEVICE macro
- Figure 13
- Figure 23
- Figure 32
- Figure 36
- Figure 38
- Appendix A. "Device Types"
- Appendix C. "Installation Device Preference Table"
- Appendix G. "Using the Master Catalog from the Old Production System"

An entry for BLKSIZE for the 3380 has been added to each system data set description where appropriate.

OTHER CHANGES

Obsolete DCM alias names must be deleted from SYS1.LPALIB(CONSOLE macro).

The DEVTYPE parameter of the IODEVICE macro has been updated concerning DUMMY devices.

The description of the ALTCTRL feature of the IODEVICE macro has been updated.

OS/VS2 MVS/SYSTEM PRODUCT-JES3 (5740-XYN)

OS/VS2 MVS/SYSTEM PRODUCT-JES2 (5740-XYS)

NEW PROGRAMMING SUPPORT

Information to support the OS/VS2 MVS/System Product is now included. For additional information see OS/VS2 MVS/System Product General Information, GC28-1025-0.

CHANGES IN MACRO INSTRUCTIONS

You can now specify the IBM 3278 Model 1, 2, 2A, 3, and 4 Display Stations, and the IBM 3279 Model 2A, 2B, 3A, and 3B Display Stations using the IODEVICE macro instruction.

You can now specify any devices that are connected to a selector or block multiplexer channel in the OPTCHAN parameter of the IODEVICE macro.

OTHER CHANGES

For the IBM 3278 and 3279 Display Stations, information has been added to the CONSOLE macro.

Information on restricted devices has been added to the IODEVICE macro and the glossary.

Entries for the IBM 3278 and 3279 Display Stations have been added to the following:

- Figure 8
- Figure 14 and Figure 19
- Figure 23
- The Appendix "Device Types"

JANUARY 1980

In the CONSOLE macro, the statement concerning deleting obsolete DCM ALIAS names has been updated. The CONSOLE examples have also been updated.

In the APFLIB parameter of the CTRLPROG macro, SYS1.IMAGELIB and SYS1.LPALIB have been added to the authorized data sets.

In the IODEVICE macro, the amount of storage available for UCBs is 60K-1.

The IBM 3540 notes in the figure "Parameter values that may be specified in an IODEVICE macro instruction" have been updated to state that 3540 addresses do not have to be contiguous.

The PTREAD feature has been added to Figure 23.

The SYSLOG parameter of the SCHEDULR macro has been updated to state that data goes to the system log if a routecode is specified in the WTO or WTOR.

The recommended DCB parameters for SYS1.IMAGELIB have been updated and Figure 32 updated to show SYS1.IMAGELIB is a required data set.

SYS1.TCOMMAC is a new data set. Entries for the data set have been included in the following:

- List of system data set names (DATASET macro)
- Figure 11
- Figure 32

The recommended DCB parameters for SYS1.VTAMLIB have been updated.

OS/VS2 MVS CRYPTOGRAPHIC UNIT SUPPORT

Information to support the IBM 3848 Cryptographic Unit is now included. For additional information, see IBM 3848 Cryptographic Unit Product Description and Operating Procedures, GA22-7072.

An entry for the IBM 3848 has been added to the following:

- Figure 13 through Figure 22
- "Appendix A. Device Types"
- "Appendix C. Installation Device Preference Table"

OS/VS2 MVS 3800 ENHANCEMENTS

With 3800 Enhancements, library character set modules are included in SYS1.IMAGELIB, and secondary extents can be specified for SYS1.IMAGELIB.

NEW PROGRAMMING SUPPORT

The system generation information to support the IBM 3203 model 5 printer is now included. You specify the IBM 3203 model 5 as 3203-4.

CHANGES IN MACRO INSTRUCTIONS

You may now specify the IBM 3203 printer in the UNIT parameter of the IODEVICE macro and the value 4 in the MODEL parameter.

OTHER CHANGES

In the DATAMGT macro, universal character set (UCS) images are automatically included during system generation if an IBM 3203-4 printer is included in the system.

The universal character set (UCS) and the forms control buffer (FCB) module for the IBM 3203-4 are contained in SYS1.IMAGELIB.

An entry for the IBM 3203-4 has been added to the following:

- Figure 8
- Figure 9
- Figure 13 through Figure 22
- Figure 25

- "Appendix A. Device Types"
- "Appendix C. Installation Device Preference Table"

MAINTENANCE CHANGES

The minimum requirement for the Starter System and Distribution Libraries for IBM 3340-35 is 5 and for IBM 3330 is 3.

The OLDWTOR parameter of the CONSOLE macro has been updated concerning replies to routing code 9.

An example has been added to the CONSOLE macro for the IBM 3033, 3032, and 3031 consoles. The example in the figure "Macro instructions for a complete system generation" has had similar information added.

The SQA parameter of the CTRLPROG macro has been updated concerning the minimum SQA size.

The IODEVICE macro has been updated concerning specification of control units for the IBM 3270 and 3791L.

A note has been added to the ADDRESS parameter of the IODEVICE macro concerning the IODEVICE statements for uniprocessor and multiprocessor systems.

A note has been added to the ALTCTRL feature in Figure 23.

A caution has been added concerning checking the return code prior to using the Stage I output as Stage II input under "Stage I Execution" in the chapter "System Control Program Installation."

The IBM 3250, 3278, 3287, 3288, and 3289 have been added to "Appendix D. Functionally Equivalent I/O Devices."

OS/VS2 MVS/SYSTEM EXTENSIONS RELEASE 2 (5740-XE1)

The information to support MVS/Extensions Release 2 is now included. For additional information concerning this program product, see OS/VS2 MVS/System Extensions Release 2 General Informatin Manual, GC28-0872-4. You may now specify multiple SYS1.MANn system data sets (SYS1.MANA through SYS1.MANZ and SYS1.MAN0 through SYS1.MAN9).

RELEASE 3.8

I/O DEVICE GENERATION

Many parameter values which previously could only be modified by another complete system generation can now be modified by an I/O device generation. The macros affected are AFFINITY, CKPTREST, CTRLPROG, EDIT, SCHEDULR, and TSO. For more information, see the macro descriptions.

DYNAMIC SUPPORT SYSTEM (DSS)

The support for DSS has been removed from this manual.

STARTER SYSTEM

The starter and distribution libraries are distributed on 1600 or 6250 BPI tapes for loading to 3330/3333 or 3340. The distribution libraries are in SMP Release 4 relfile format, and SMP Release 4 is used to load the libraries from the distribution tape to direct-access devices. For detailed description of the distribution tape format see "Program Directory for use with Release 3.8 of 5752-VS2." This has caused changes to the following:

- "The Generating System" and "The Distribution Libraries" in the chapter "Requirements for Generating an MVS System Control Program"
- Figure 3
- The IOC parameter of the CTRLPROG macro has been deleted.
- "Processing the Starter System and Distribution Library Tapes," "Load the Contents of the Distribution Tape to a Direct-Access Volume," and "Preparing for a System Generation Using an Existing MVS System as the Generating System" in the chapter "Preparing for System Generation"
- "Appendix B. Description of the Starter System and Distribution Library Tapes"

OTHER CHANGES

The total number of names has been qualified, and the NUMBER parameter has been updated in the AFFINITY macro. An example has also been added.

You must generate a dummy device for channels that do not appear as the primary channel for any device (CHANNEL macro).

The CONSOLE macro has been updated concerning multiple console support. The maximum number of secondary consoles has been increased to 98. The IOC parameter has been deleted.

The CTRLPROG macro has been updated concerning adding OPTIONS=RDE to an I/O device generation.

The industry subsystems require VTAM as the access method.

If you specify VTAM as an access method in the DATAMGT macro, the IND parameter is not required.

In the DATASET macro SYS1.IMAGELIB has been added as a required data set for an I/O device generation and Figure 11 has been updated correspondingly.

Figure 12 has been updated.

The IODEVICE macro has been updated concerning the specification of the IBM Model 168 Processor and an example of this specification has been added. The ADDRESS parameter range of the IODEVICE macro is 000-FFE. In the OPTCHAN parameter a warning has been added concerning the varying online of devices which are not installed.

Figure 13 through Figure 22 has been updated concerning specification of the IBM 3330 MODEL parameter and the addresses required for the IBM 3350 Model B2.

The DEVPREF parameter of the SCHEDULR macro has been updated.

The SVCTABLE macro has been updated to include the type 5 SVC.

A restriction concerning the unitname, SYSALLDA, has been added to the UNITNAME macro. Also added is a restriction concerning UNITNAME macros with the same names. The address range in the UNIT parameter is 000-FFE.

Figure 32 has been updated.

SYS1.IMAGELIB is a required data set.

SYS1.PARMLIB specification requires the additional DCB parameter LRECL=80.

A note has been added to the SYS1.UADS data set concerning the SYS1.UADS supplied by IBM.

Figure 42 has been updated.

The IBM 3330 Model 2 has been added to "Appendix A. Device Types."

In "Appendix B. Description of the Starter System and Distribution Library Tapes" SYS1.AOSH2 has been changed to SYS1.AOSH3.

CONTENTS

FIGURES

INTRODUCTION

This chapter presents an overview of the system generation process and a discussion of the types of system generation. Also in this chapter is a brief discussion about how each chapter in this section can be best utilized in performing a system generation.

THE SYSTEM GENERATION PROCESS

System generation is the process that creates an OS/VS2 MVS system control program tailored to both the data processing requirements and machine configuration of an installation.

The MVS system control program to be installed is specified with system generation macro instructions that you code. During system generation, IBM-supplied components and user-written components are combined in a variety of ways to build a new MVS system control program according to the specifications in the macro instructions (see Figure 1). The new MVS system control program is composed of the standard programs incorporated into every MVS system control program, optional programs that you select, and your own routines.

TYPES OF SYSTEM GENERATION

There are three types of system generation: a complete system generation, an I/O device generation, and an Eligible Device Table generation. You specify the type of generation in the GENERATE or EDTGEN macro instruction.

For a complete system generation, the job stream produced during Stage I is processed during Stage II to create an entirely new MVS system control program. For an I/O device generation, the job stream is used to modify an existing MVS system control program. For an Eligible Device Table generation, the job stream is used to build new device allocation tables.

COMPLETE SYSTEM GENERATION

This is the generation of a complete MVS system control program. It is done when you are installing an MVS system control program for the first time or when you must modify an existing MVS system control program.

I/O DEVICE GENERATION

An I/O device generation is done when changes need to be made to the machine configuration or certain other parameters. In an I/O device generation, for example, you can add or delete I/O devices or channels from an installation, change I/O device group names, or change console specifications. You cannot add the programming support for telecommunications, MSS, or graphics, or add additional access methods, or add devices requiring updates to any system data sets other than SYS1.DCMLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.LOGREC, SYS1.LPALIB, and SYS1.NUCLEUS (this includes OCR and MICR devices requiring updates to SYS1.MACLIB). For an I/O device generation, additions are made to the SYS1.DCMLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.LOGREC, SYS1.LPALIB, and SYS1.NUCLEUS system data sets.

SYSTEM MODIFICATION PROGRAM (SMP) USERS: The SMPCDS data set should be updated after an I/O device generation to reflect the current status of the system. The output of STAGE I from a complete system generation is used by the JCLIN function to create a new SMPCDS data set. The output of STAGE I from an I/O device generation or partial generation is used by the JCLIN function to update the SMPCDS data set. For additional information on SMP, see OS/VS System Modification Program (SMP).

ELIGIBLE DEVICE TABLE GENERATION

An Eligible Device Table generation (EDTGEN) is done when various versions of device allocation tables are needed for use on different processors or with different job mixes.

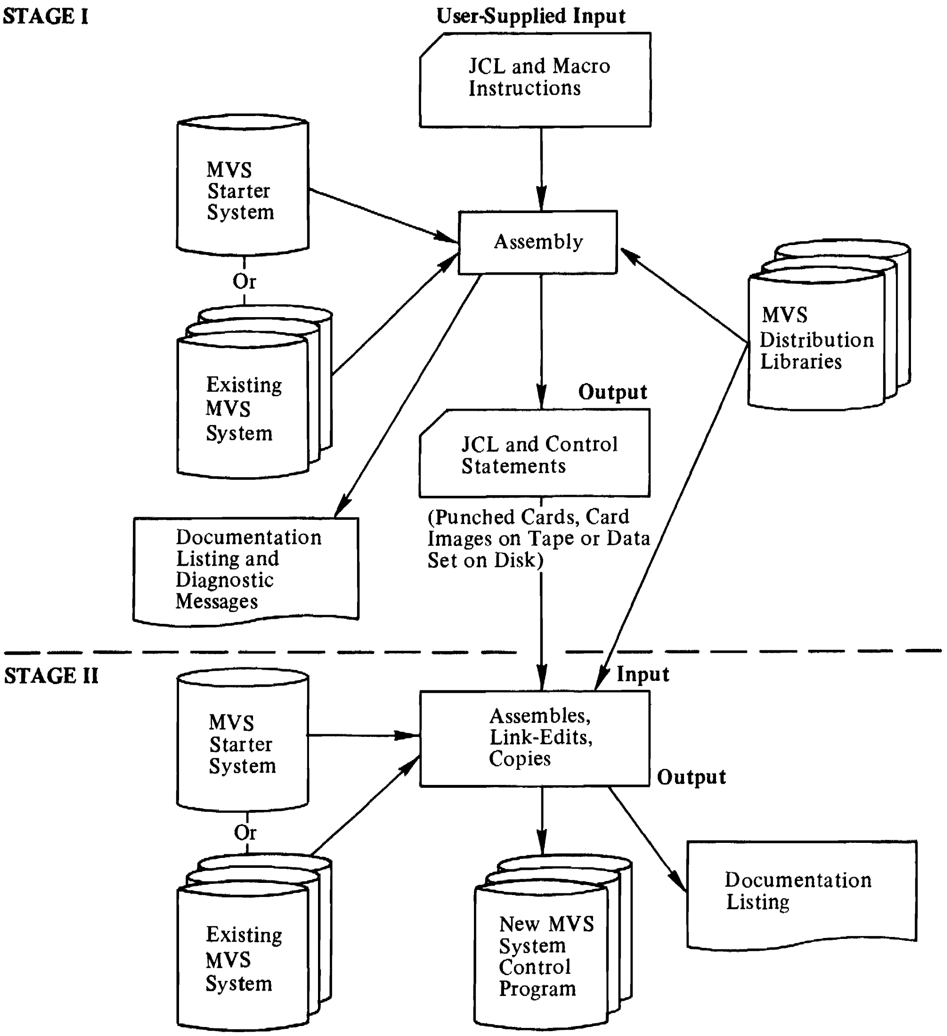
STAGE I
User-Supplied Input
JCL and Macro Instructions
MVS Starter System
Or
Existing MVS System
Assembly
MVS Distribution Libraries
Output
JCL and Control Statements
(Punched Cards, Card Images on Tape or Data Set on Disk)
Documentation Listing and Diagnostic Messages
STAGE II
Input
MVS Starter System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

Figure 1. The System Generation Process: In Stage I, system generation macro instructions are assembled and expanded to form a job stream. In Stage II, this job is used to assemble, link-edit, and copy selected modules from the distribution libraries and user-supplied components from user data sets to system data sets on system volumes to form the MVS system control program.

OVERVIEW OF SYSTEM GENERATION PROCEDURES

This section describes the sequence of procedures to be done to install an MVS system control program. 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 any execution begins.

CODING SYSTEM GENERATION MACRO INSTRUCTIONS

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

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

DEFINING THE SYSTEM DATA SETS

The system data sets (including the page data sets) should be considered next. They must have space allocated for them and they must be cataloged in the master catalog of the system to be generated. In this publication, this process is referred to as defining.

System data sets can be defined either by using the DATASET macro to define them as part of system generation or by using JCL and/or the Access Method Services to define them before system generation.

If you are using the DATASET macro, you should refer to "Selecting and Defining the System Data Sets" when you code the DATASET macro as well as to the description of the DATASET macro in "Specifying the MVS System Control Program."

If you are using JCL and/or the Access Method Services to define the system data sets before system generation, you should also refer to "Selecting and Defining the System Data Sets" for information on specifying the job control language and command statements required.

If you are defining the system data sets before system generation, you must allocate space for the object module utility data sets that are required for Stage II processing. This information is in "System Control Program Installation" in the section "Stage II Input."

PREPARING TO PERFORM A SYSTEM GENERATION

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

EXECUTING SYSTEM GENERATION

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

RESTARTING SYSTEM GENERATION

Coding errors, machine malfunctions, or improper space allocations for system data sets 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 an MVS system control program, you can test it by using the installation verification procedure (IVP). IVP is discussed in the chapter "Testing the System Control Program." IVP can only be used to test the MVS system control program after you have successfully installed the JES2 job entry subsystem. (Refer to _OS/VS2 MVS System Programming Library: JES2_ or _System Programming Library: Network Job Entry Facility for JES2_ for a description of JES2 generation.)

VERIFYING NEW DEVICE ALLOCATION TABLES

New device allocation tables may be verified by the Eligible Device Table verification program, IEFEB400. The verification program is discussed in the chapter "Verifying New Device Allocation Tables."

REQUIREMENTS FOR GENERATING AN MVS SYSTEM CONTROL PROGRAM

You must generate an MVS system control program using either an existing MVS system or an MVS starter system that is provided by IBM. This chapter discusses the real-storage requirements for system generation, the machine requirements for system generation using the starter system, and the programming requirements for system generation using either an existing MVS system or MVS starter system. The existing MVS system must be a Release 2 or later release.

REAL-STORAGE REQUIREMENTS

To generate an MVS system control program using a starter system, 768K bytes of real storage are required.

MACHINE REQUIREMENTS

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

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

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

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

PROGRAMMING REQUIREMENTS

To perform a system generation, you must have an MVS system, the generating system, to "drive" the system generation process and to perform the procedures that need to be done before the MVS system control program can be installed. You also need the IBM-supplied distribution libraries.

THE GENERATING SYSTEM

System generation is performed under the control of an existing MVS system control program and is executed as any other job.

For your first system generation, you must use an MVS starter system that is provided by IBM as the generating system. The starter system is a minimum MVS system control program that contains all of the routines and programs necessary to perform a system generation.

Once you have used the starter system for your first system generation, subsequent system generations may be performed using an existing MVS system control program as the generating system.

The generating system must include SMP Release 4. If a starter system or an existing MVS Release 3.7 or older system is used as the generating system, the SMP Release 4 load module must be added to the system prior to processing the distribution libraries.

THE DISTRIBUTION LIBRARIES

The distribution libraries are distributed on tape in SMP Release 4 relfile format. The distribution relfile tape contains the required MVS system subsets; each subset contains a number of MVS components. SMP Release 4 is used to load the system subsets into the distribution libraries. Some of the distribution libraries contain one or more MVS modules that are assembled and/or link-edited during system generation into the appropriate system data sets. These modules contain:

- The MVS program options
- Utility programs
- Data management routines
- Error recovery routines
- Job management routines
- Task management routines
- Problem determination routines
- Teleprocessing routines

Additionally, some of the distribution libraries contain modules that are copied during system generation in their entirety into the appropriate system data sets. These modules contain:

- System parameter lists
- Cataloged procedures
- Macro definitions for the system macros that are used by the assembler

The distribution libraries also contain modules that are used during the system generation process. These modules contain:

- The macro definitions used during Stage I assemblies
- The macro definitions used during Stage II assemblies

"Appendix B. Description of the Starter System and Distribution Tapes" lists all of the distribution libraries.

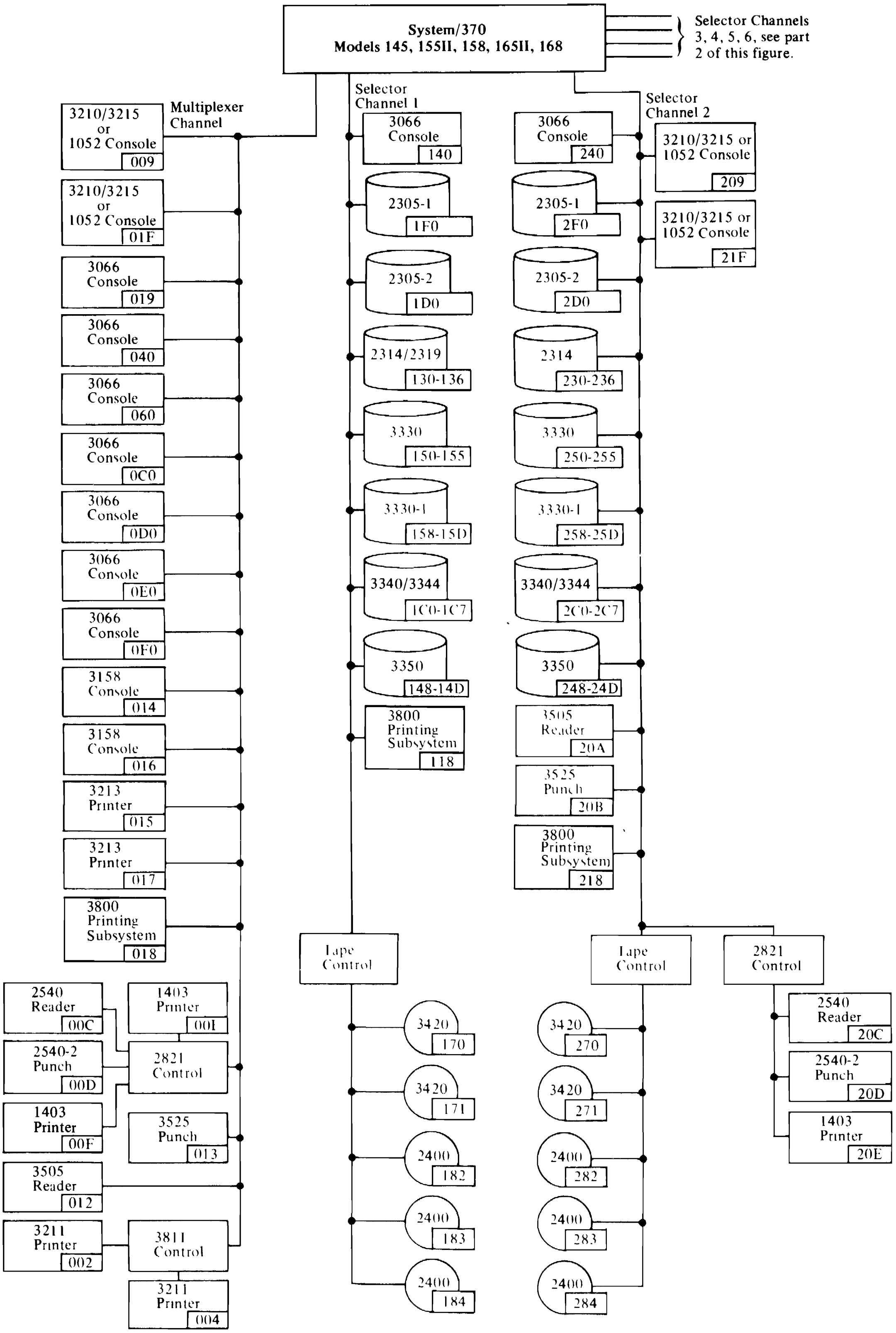

Figure 2 (Part 1 of 2). Maximum machine configuration for using the starter system

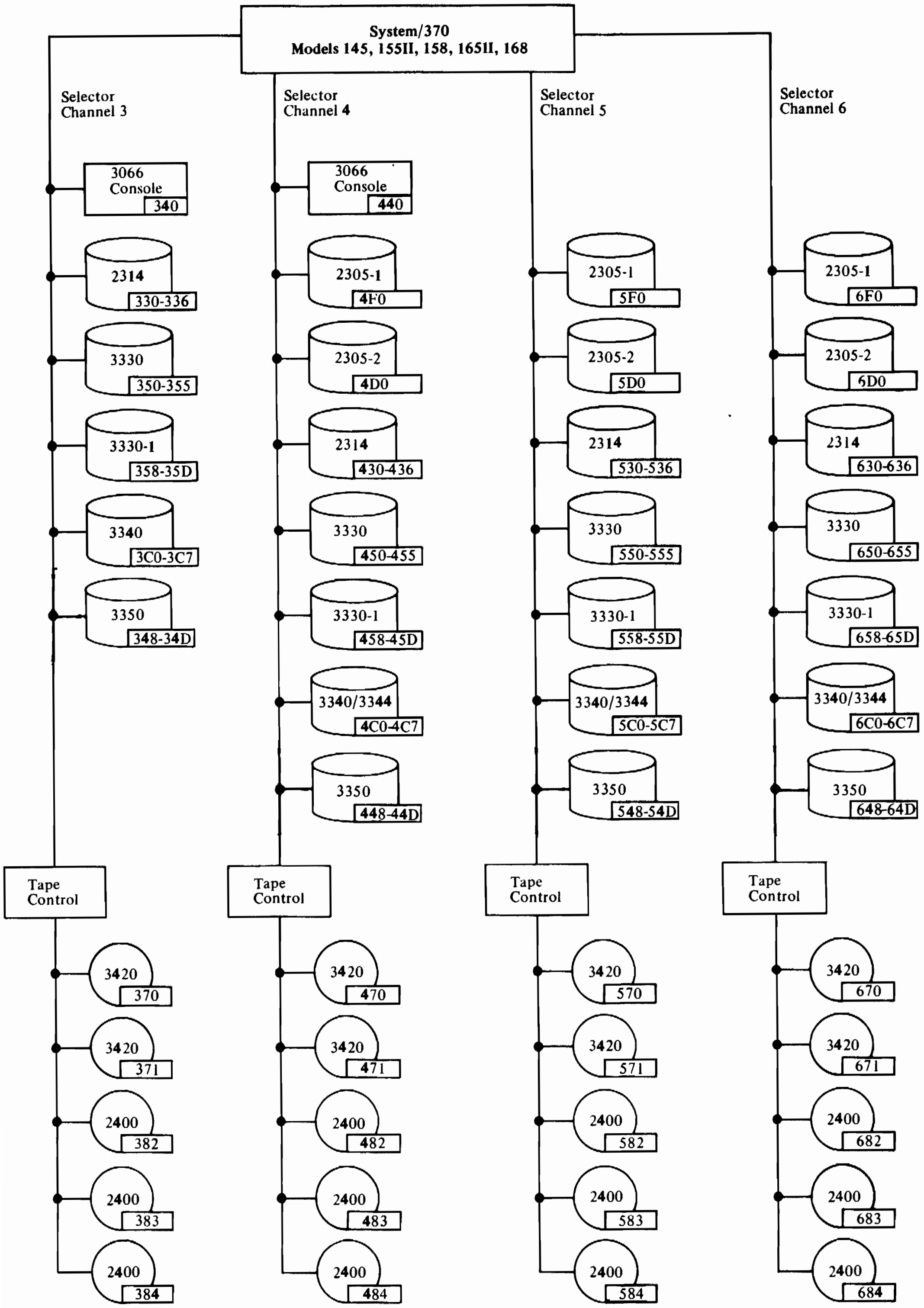

Figure 2 (Part 2 of 2). Maximum machine configuration for using the starter system

				Unit Address (choose one)					
Use	Minimum Requirement[1]	Device (choose one)	Multiplexer Channel	Selector Channel 1	Selector Channel 2	Selector Channel 3	Selector Channel 4	Selector Channel 5	Selector Channel 6
System Console[4]	1	3210/3215 or 1052	009, 01F		209, 21F				
		3066	019, 040, 060, 0C0 0D0, 0E0 0F0	140	240	340	440		
		3158	014, 016						
System Input[2]	1	2540 Reader	00C		20C				
		3505 Reader	012		20A				
	1	3420-8		170, 171	270, 271	370, 371	470, 471	570, 571	670, 671
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383, 384	482, 483, 484	582, 583, 585	682, 683, 684
Punched Output	1	2540-2 Punch	00D		20D				
		3525 Punch	013		20B				
		3420-8		170, 171	270, 271	370, 371	470, 471	570, 571	670, 671
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383, 384	482, 483, 484	582, 583, 584	682, 683, 684
Printed Output	1	3211	002, 004						
		1403	00E, 00F		20E				
		3213[3]	015, 017						
		3420-8		170, 171	270, 271	370, 371	470, 471	570, 571	670, 671
		2400 (9-track)		182, 183, 184	282, 283, 284	382, 383, 384	482, 483, 484	582, 583, 584	682, 683, 684
		3800	018	118	218	318	418	518	618
New System	1 or 2	2305-1		1F0	2F0		4F0	5F0	6F0
		2305-2		1D0	2D0		4D0	5D0	6D0
		2314 (or 2319 on Channel 1)		130, 131, 132, 133, 134, 135, 136	230, 231, 232, 233, 234, 235, 236	330, 331, 332, 333, 334, 335, 336	430, 431, 432, 433, 434, 435, 436	530, 531, 532, 533, 534, 535, 536	630, 631, 632, 633, 634, 635, 636
		3330		150, 151, 152, 153, 154, 155	250, 251, 252, 253, 254, 255	350, 351, 352, 353, 354, 355	450, 451, 452, 453, 454, 455	550, 551, 552, 553, 554, 555	650, 651 652, 653, 654, 655
		3330-1		158, 159, 15A, 15B, 15C, 15D	258, 259, 25A, 25B, 25C, 25D	358, 359, 35A, 35B, 35C, 35D	458, 459, 45A, 45B, 45C, 45D	558, 559, 55A, 55B, 55C, 55D	658, 659, 65A, 65B, 65C, 65D
		3340/3344		1C0, 1C1, 1C2, 1C3, 1C4, 1C5, 1C6, 1C7	2C0, 2C1, 2C2, 2C3, 2C4, 2C5, 2C6, 2C7	3C0, 3C1, 3C2, 3C3, 3C4, 3C5, 3C6, 3C7	4C0, 4C1, 4C2, 4C3, 4C4, 4C5, 4C6, 4C7	5C0, 5C1, 5C2, 5C3, 5C4, 5C5, 5C6, 5C7	6C0, 6C1, 6C2, 6C3, 6C4, 6C5, 6C6, 6C7
		3350		148, 149, 14A, 14B, 14C, 14D	248, 249, 24A, 24B, 24C, 24D	348, 349, 34A, 34B, 34C, 34D	448, 449, 44A, 44B, 44C, 44D	548, 549, 54A, 54B, 54C, 54D	648, 649, 64A, 64B, 64C, 64D

[1] Select the minimum requirement from the list of devices given in each section. For example, the *one* console that is required for system generation can be a 3210/3215, 1052, 3066, or 3158.

[2] One tape drive *and* one card reader are required as input devices.

[3] Console printer for the 3158.

[4] Check the appropriate hardware manual to determine if a console can be placed at the addresses specified for the multiplexer channel.

Figure 3 (Part 1 of 2). Minimum I/O device requirements using the starter system

				Unit Address (choose one)					
Use	Minimum Requirement[1]	Device (choose one)	Multiplexer Channel	Selector Channel 1	Selector Channel 2	Selector Channel 3	Selector Channel 4	Selector Channel 5	Selector Channel 6
Starter System and Distribution Libraries	5	3340-35[3]		130, 131, 132, 133, 134, 135, 136	230, 231, 232, 233, 234, 235, 236	330, 331, 332, 333, 334, 335, 336	430, 431, 432, 433, 434,435, 436	530, 531, 532, 533, 534, 535, 536	630, 631, 632, 633, 634, 635, 636
	4	3330		150, 151, 152, 153, 154, 155	250, 251, 252, 253, 254, 255	350, 351, 352, 353, 354, 355	450, 451, 452, 453, 454, 455	550, 551, 552, 553, 554, 555	650, 651, 652, 653, 654, 655
		3330-1		158, 159, 15A, 15B, 15C, 15D	258, 259, 25A, 25B, 25C, 25D	358, 359, 35A, 35B, 35C, 35D	458, 459, 45A, 45B, 45C, 45D	558, 559, 55A, 55B, 55C, 55D	658, 659, 65A, 65B, 65C, 65D
		3340/3344		1C0, 1C1, 1C2, 1C3, 1C4, 1C5, 1C6, 1C7	2C0, 2C1, 2C2, 2C3, 2C4, 2C5, 2C6, 2C7	3C0, 3C1, 3C2, 3C3, 3C4, 3C5, 3C6, 3C7	4C0, 4C1, 4C2, 4C3, 4C4, 4C5, 4C6, 4C7	5C0, 5C1, 5C2, 5C3, 5C4, 5C5, 5C6, 5C7	6C0, 6C1, 6C2, 6C3, 6C4, 6C5, 6C6, 6C7
		3350		148, 149, 14A, 14B, 14C, 14D	248, 249, 24A, 24B, 24C, 24D	348, 349, 34A, 34B, 34C, 34D	448, 449, 44A, 44B, 44C, 44D	548, 549, 54A, 54B, 54C, 54D	648, 649, 64A, 64B, 64C, 64D
Utility Data Sets[2]	1	2305-1		1F0	2F0		4F0	5F0	6F0
		2305-2		1D0	2D0		4D0	5D0	6D0
		2314 (or 2319 on Channel 1)		130, 131, 132, 133, 134, 135, 136	230, 231, 232, 233, 234, 235, 236	330, 331, 332, 333, 334, 335, 336	430, 431, 432, 433, 434, 435, 436	530, 531, 532, 533, 534, 535, 536	630, 631, 632, 633, 634, 635, 636
		3330		150, 151, 152, 153, 154, 155	250, 251, 252, 253, 254, 255	350, 351, 352, 353, 354, 355	450, 451, 452, 453, 454, 455	550, 551, 552, 553, 554, 555	650, 651, 652, 653, 654, 655
		3330-1		158, 159, 15A, 15B, 15C, 15D	258, 259, 25A, 25B, 25C, 25D	358, 359, 35A, 35B 35C, 35D	458, 459, 45A, 45B, 45C, 45D	558, 559, 55A, 55B, 55C, 55D	658, 659, 65A, 65B, 65C, 65D
		3340/3344		1C0, 1C1, 1C2, 1C3, 1C4, 1C5, 1C6, 1C7	2C0, 2C1, 2C2, 2C3, 2C4, 2C5, 2C6, 2C7	3C0, 3C1, 3C2, 3C3, 3C4, 3C5, 3C6, 3C7	4C0, 4C1, 4C2, 4C3, 4C4, 4C5, 4C6, 4C7	5C0, 5C1, 5C2, 5C3, 5C4, 5C5, 5C6, 5C7	6C0, 6C1, 6C2, 6C3, 6C4, 6C5, 6C6, 6C7
		3350		148, 149, 14A, 14B, 14C, 14D	248, 249, 24A, 24B, 24C, 24D	348, 349, 34A, 34B, 34C, 34D	448, 449, 44A, 44B, 44C, 44D	548, 549, 54A, 54B, 54C, 54D	648, 649, 64A, 64B, 64C, 64D

[1] Select the minimum requirement from the list of devices given in each section. For example, the *one* console that is required for system generation can be a 3210/3215, 1052, 3066, or 3158.

[2] The object module utility data sets required for Stage II do not require an additional direct-access volume if sufficient space is available on the volumes that contain the new system or the starter system.

Figure 3 (Part 2 of 2). Minimum I/O device requirements using the starter system

Magnetic Tape Drives[1]

2400	2400 series 9-track magnetic tape drive
3420-8	3420 series 9-track,1600/6250 BPI tape drive

Direct-Access Devices

2305-1	2305 fixed head disk storage model 1
2305-2	2305 fixed head disk storage model 2
2314	2314/2319 disk storage facility
3330	3330 disk storage drive model 1 or 3333 disk storage and control model 1
3330-1	3330 disk storage model 11 or 3333 disk storage and control model 11
3340	3340/3344 disk storage drive
3350	Direct Access Storage

Unit Record Devices

1052	1052 console keyboard
1403	1403 printer
2540	2540 reader punch (read feed)
2540-2	2540 reader punch (punch feed)
3066	3066 system console
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
3800	Printing Subsystem

[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 with the starter system

Group Name **Devices identified by group name**

SYSSQ Sequential or direct-access devices at any of the following addresses:

2400 series					
182	282	382	482	582	682
183	283	383	483	583	683
184	284	384	484	584	684

3420					
170	270	370	470	570	670
171	271	371	471	571	671

2314/2319					
130	230	330	430	530	630
131	231	331	431	531	631
132	232	332	432	532	632
133	233	333	433	533	633
134	234	334	434	534	634
135	235	335	435	535	635
136	236	336	436	536	636

2305-1	2350-2
1F0	1D0
2F0	2D0
4F0	4D0
5F0	5D0
6F0	6D0

3330					
150	250	350	450	550	650
151	251	351	451	551	651
152	252	352	452	552	652
153	253	353	453	553	653
154	254	354	454	554	654
155	255	355	455	555	655

3330-1		
158	258	358
159	259	359
15A	25A	35A
15B	25B	35B
15C	25C	35C
15D	25D	35D

3340/3344					
1C0	1C0	3C0	4C0	5C0	6C0
1C1	2C1	3C1	4C1	5C1	6C1
1C2	2C2	3C2	4C2	5C2	6C2
1C3	2C3	3C3	4C3	5C3	6C3
1C4	2C4	3C4	4C4	5C4	6C4
1C5	2C5	3C5	4C5	5C5	6C5
1C6	2C6	3C6	4C6	5C6	6C6
1C7	2C7	3C7	4C7	5C7	6C7

3350					
148	248	348	448	548	648
149	249	349	449	549	649
14A	24A	34A	44A	54A	64A
14B	24B	34B	44B	54B	64B
14C	24C	34C	44C	54C	64C
14D	24D	34D	44D	54D	64D

SYSDA Direct-access devices at any of the following addresses:

2305-1	2305-2
1F0	1D0
2F0	2D0
4F0	4D0
5F0	5D0
6F0	6D0

2314/2319					
130	230	330	430	530	630
131	231	331	431	531	631
132	232	332	432	532	632
133	233	333	433	533	633
134	234	334	434	534	634
135	235	335	435	535	635

3330					
150	250	350	450	550	650
151	251	351	451	551	651
152	252	352	452	552	652
153	253	353	453	553	653
154	254	354	454	554	654
155	255	355	455	555	655

3330-1		
158	258	358
159	259	359
15A	25A	35A
15B	25B	35B
15C	25C	35C
15D	25D	35D

3340/3344					
1C0	2C0	3C0	4C0	5C0	6C0
1C1	2C1	3C1	4C1	5C1	6C1
1C2	2C2	3C2	4C2	5C2	6C2
1C3	2C3	3C3	4C3	5C3	6C3
1C4	2C4	3C4	4C4	5C4	6C4
1C5	2C5	3C5	4C5	5C5	6C5
1C6	2C6	3C6	4C6	5C6	6C6
1C7	2C7	3C7	4C7	5C7	6C7

3350					
148	248	348	448	548	648
149	249	349	449	549	649
14A	24A	34A	44A	54A	64A
14B	24B	34B	44B	54B	64B
14C	24C	34C	44C	54C	64C
14D	24D	34D	44D	54D	64D

Figure 5 (Part 1 of 2). Group names that can be used to identify I/O devices

Group Name	Devices identified by group name
SYSRDR	Card readers at any of the following addresses: 2540 3505 00C 012 20C 20A
TAPE	Tape units at any of the following addresses: 2400 series: 3420: 182 282 382 482 582 682 170 270 370 470 570 670 183 283 383 483 583 683 171 271 371 471 571 671 184 284 384 484 584 684

Figure 5 (Part 2 of 2). Group names that can be used to identify I/O devices

SPECIFYING THE MVS 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, format, parameters and defaults of each macro instruction are described, and a coding example of each macro instruction is given.

CODING SYSTEM GENERATION MACRO INSTRUCTIONS

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

RULES FOR CODING SYSTEM GENERATION MACRO INSTRUCTIONS

The rules for coding system generation macro instructions are those of the assembler language. The following paragraphs are a summary of these rules as stated in OS/VS—DOS/VS—VM/370 Assembler Language.

System generation macro instructions have the following standard format:

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

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

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

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

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

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

DESCRIBING SYSTEM GENERATION MACRO INSTRUCTIONS

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

- Letters in boldtype, numbers, and punctuation marks must be coded exactly as shown. Exceptions to this convention are brackets, [ ]; braces { }; ellipses, . . .; and superscripts, which are never coded.

- Lowercase letters 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.

- Items enclosed in braces represent alternative items. Exactly one of the items should be coded.

- If an alternative item is underlined, it is the default value—the system will automatically assume it is your choice if none of the items is coded.

- An ellipsis indicates that the preceding item or group of items can be coded two or more times in succession.

- Parentheses must enclose subfields, if more than one is specified. If only one subfield is specified, omit the parentheses. In the example below, the ADDRESS=1 parameter could be specified as ADDRESS=(1,2,3) indicating three selector channels.

A typical system generation macro instruction might appear as:

CHAN1	CHANNEL	ADDRESS=1,TYPE=SELECTOR

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

MACRO INSTRUCTION SUMMARY

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

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 or EDTGEN macro instruction must be the last macro instruction in the input deck for system generation. All other system generation macro instructions can be issued in any order.

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

Group	**Macro Instruction**	**Complete**	**I/O Device**	**Eligible Device Table**
Machine Configuration	CHANNEL[1]	required	required	required[4]
	IODEVICE[1]	required	required	required[4]
	UNITNAME[1]	required	required	required
	CONSOLE[1]	required	required	—
Control Program	AFFINITY	optional	optional[2]	—
	CKPTREST	optional	—	—
	CTRLPROG	optional	optional	—
	EDIT	optional	optional	—
	JES	optional[3]	—	—
	SCHEDULR	optional	optional	—
	TSO	optional	optional[2]	—
Data Management	DATAMGMT	optional	optional[2]	—
User-written SVC Routines	SVCTABLE[1]	optional	optional[2]	—
Generation	DATASET[1]	required	required	—
	GENERATE	required	required	—
	EDTGEN	—	—	required

Figure 6. Required and optional system generation macro instructions

Notes to Figure 6:

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

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

[3] This macro instruction is required if JES3 is to be included in the system.

[4] For an Eligible Device Table generation, the specifications of these macro instructions should not be changed from those used for the original complete or I/O device generation. Changing the

specifications incurs the risk of generating an invalid table. That is, the device-type specifications in the new table may not match those in the UCBs and IOS look-up tables in the target system.

Note: The macros in this figure are ordered in the sequence in which they should be considered.

AFFINITY

Optional for: Complete
I/O Device

Optional for:	Complete I/O Device
Not Applicable for:	Eligible Device Table

The AFFINITY macro instruction is used to specify those problem programs which may be executed only under the control of certain central processing units. You specify the name of the problem program and the identification number of the CPU(s) under which the problem program may be executed. This macro may be specified only once.

For an I/O device generation, if this macro is included, the values specified replace those values specified during the last complete or I/O device generation. In this case, however, the Program Properties Table (PPT) is re-created in its standard form, and any user modifications to this table must be reapplied. If this macro is not included, neither the previous AFFINITY values or the Program Property Table is affected.

[symbol]	**AFFINITY**	operand[,operand]...

operand

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

AFF-name-(number,...)

name

is the name of the problem program. The name you specify cannot exceed eight alphameric characters. The first character must be alphabetic.

You can specify a name only once; any additional entries of that name will be ignored. Up to 64 names can be specified; however, the maximum number of characters of all the program names must not exceed a total of 253, including one comma per name.

number

is a number from 0 to 15, corresponding to the CPU address number of the CPU, that designates the CPU(s) under which the problem program may be executed. The number used to identify the CPU is converted into a value between 1 and 65,535. This value is used as a mask, and becomes part of a subparameter list which is limited to a maximum of 253 characters, including the commas.

EXAMPLE 1: This macro specifies that problem program AFFPGM00 can be executed only under the control of CPU 0, AFFPGM01 can be executed only under the control of CPU 1, and AFFPGM02 can be executed only under the control of CPU 0.

```
AFF   AFFINITY   AFF-AFFPGM00-0,AFF-AFFPGM01-1,          X
                 AFF-AFFPGM02-0
```

EXAMPLE 2: This example describes the procedure used when computing the total number of program name characters which may be used.

```
AFF   AFFINITY   AFF-AFFPGM00-0,AFF-AFFPGM01-1,          X
                 AFF-PGM02-0
```

The number of characters used is:

LP+LP+LP+...LP+(n-1)=Pt which gives 8+8+5+(3-1)=23 where

LP = length of the name of AFFPGM00
LP = length of the name of AFFPGM01
LP = length of the name of PGM02
Pt = total number of characters, including commas
n = total number of name entries

Note: Pt must be less than 254, which is true in this example.

EXAMPLE 3: This example describes the method to be used when computing the total number of CPU mask characters. This involves two steps:

1. Convert the CPU number into a mask value using the table below.

2. Apply the mask value to the formula.

```
  AFF   AFFINITY  AFF-AFFPGM00-0,AFF-AFFPGM01-(0,1,3),      X
                  AFF-PGM02-12
```

AFFPGM00 is executed by CPU 0   mask value = 32768

AFFPGM01 is executed by CPU 0,   mask value = 53248
CPU 1, or CPU 3

PGM02 is executed by CPU 12   mask value = 8

To calculate the total number of CPU characters:

LM+LM+LM+...LM+(n-1)=Mt

where

LM = length of the mask for AFFPGM00
LM = length of the mask for AFFPGM01
LM = length of the mask for PGM02
Mt = total number of characters, including commas
n = total number of name entries

Note: Mt must be less than 254, which is true in this example.

Mask Values

CPU 0=32,768
CPU 1=16,384
CPU 2=8192
CPU 3=4096
CPU 4=2048
CPU 5=1024
CPU 6=512
CPU 7=256
CPU 8=128
CPU 9=64
CPU 10=32
CPU 11=16
CPU 12=8
CPU 13=4
CPU 14=2
CPU 15=1

Required for: Complete
I/O Device
Eligible Device Table

The CHANNEL macro instruction describes the channel characteristics. There must be a CHANNEL macro instruction for each channel type in your computing system. You must generate a dummy device for a channel that does not appear as the primary channel for any device.

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.

[symbol]	**CHANNEL**	**ADDRESS=(**address[**,**address]...**)** **TYPE={BLKMPXR|HISPEEDMPXR|MULTIPLEXOR|SELECTOR}**

ADDRESS=address
specifies the address assigned to the channel. The value must be alphameric, 0 through 9 or A through F. Multiple addresses may be specified for channels of the same type as specified in the TYPE parameter.

TYPE=
specifies the type of channel defined by this macro instruction.

BLKMPXR
The channel is a block multiplexer channel that is integrated either with a central processing unit or with a 2880 channel frame. An address must be specified for each block multiplexer channel.

HISPEEDMPXR
The channel is a 2870 channel frame and there are one or more subselector channels on the 2870. You cannot use separate CHANNEL macro instructions or specify different addresses for the subselector channels.

MULTIPLEXOR
The channel is a byte multiplexer channel integrated with either the central processing unit or a 2870 channel frame (if there are no subselector channels on the 2870).

SELECTOR
The channel is a selector channel integrated with either the central processing unit or a 2860 channel frame. An address must be specified for each selector channel.

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

```
CHAN123    CHANNEL    ADDRESS=(1,2,3),TYPE=SELECTOR
```

CKPTREST

Optional For: Complete
I/O device

Not applicable for: Eligible Device Table

The CKPTREST macro instruction is used in a complete or I/O generation 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 not specified in a complete generation, or if it is not specified in an I/O device generation, the standard set of ABEND codes will be included in the new system.

For information on checkpoint/restart and for the list of standard ABEND codes, refer to OS/VS2 MVS Checkpoint/Restart. For information on ABEND codes, refer to OS/VS Message Library: VS2 System Codes.

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

ELIGBLE=code
specifies ABEND codes written by you which you want eligible for automatic restart. The code specified is a decimal integer from 0 to 4095.

A maximum of ten values may be specified.

NOTELIG=code
specifies those standard ABEND codes which you do not want to be eligible for automatic restart.

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

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

Required for:	Complete I/O Device
Not Applicable for:	Eligible Device Table

The CONSOLE macro instruction specifies the master and secondary consoles and console options for multiple console support in MVS. This macro can also be used to specify only the master console if multiple console support is not going to be used. This macro must be specified at least once. A maximum of 98 secondary consoles may be specified. The master and secondary consoles cannot be specified in the same CONSOLE macro.

A console may be specified only once as either the master console or as a secondary console, but it may be specified more than once as an alternate console.

You need not specify an alternate console to the master console but, if one is specified, it must also be specified as a secondary console. If an alternate console is not specified for a secondary console, the master console is used as the alternate console.

Each console that is specified must also be specified with the same unit address in an IODEVICE macro. For JES3, each console should be generated as a device on each processor (where physically possible) for dynamic system interchange (DSI). For information on DSI, see Introduction to JES3. In addition, each JES3 console with input capability should be generated as a multiple console support (MCS) secondary console on each processor. MCS secondary consoles are specified in the SECONS parameter. If it is not possible to generate MCS secondary consoles on each processor, dummy secondary consoles should be defined. Dummy secondary consoles are generated by specifying TYPE=JES.

When you are using multiple console support, a table is built based on the order of your CONSOLE macro cards. This table is searched under the following conditions:

- When all the alternate consoles of the master console are inactive, the table is searched for a console to assume master console function.
- When the master console and its alternates are ineligible to receive hardcopy, the table is searched for a hardcopy unit.
- When a no-consoles condition is found, the table is searched for a device possessing an alarm to sound.
- When a no-consoles condition is found, the table is searched for an IBM 3036, in order to sound the power and cooling alarm.

Display devices (2250, 2260, 3036, 3066, 3158, 3277, 3278, and 3279) cannot be specified as the input device for a composite console.

For an I/O device generation, all consoles that are to be supported by the system must be specified. There can be additions, deletions, changes, or the same specifications as those in the last complete system generation.

Obsolete DCM ALIAS names must be deleted from the PDS directory, SYS1.LPALIB, using a system utility such as IEHPROGM. If you do not delete these names and you make a change of processors console conflicts can result.

Refer to Figure 8 for a list of valid console devices. For information on master and secondary consoles, refer to Operator's Library: OS/VS2 MVS System Commands.

<table>
<tr><td>[symbol]</td><td>CONSOLE</td><td>{MCONS={address|(I-address,O-address)}|
SECONS={address|(I-address,
O-address)|O-address}}

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

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

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

[PFK=number]

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

[TYPE=JES]

[USE={MS|SD}]

[VALDCMD=(command-group
[,command-group]...)]</td></tr>
</table>

ALTCONS=

specifies the unit address or addresses of an alternate console for either the master console or a secondary console.

If this parameter is not specified for a secondary console, the master console will be used as the alternate console.

Each unit address specified must be the same as that specified for the device in an IODEVICE macro. The devices that can be used are listed in Figure 8.

The device or devices specified as an alternate console must also be specified as either the master console or a secondary console.

address

is the unit address of a console device having input and output capability.

I-address

is the unit address of an input device for a composite console.

O-address

is the unit address of an output device for a composite console. Specification of an output address is only valid for a secondary console; an ouput address is not valid when specifying an alternate console for the master console.

A device specified as part of a composite console cannot be specified any other way (as a single device or as part of a composite console with a different companion device). However, the same combination may be repeated more than once in another CONSOLE macro.

AREA=number

specifies the dimensions of the display areas to be set aside for status displays on the display screen of the console specified in this macro.

The value specified must be a decimal number equal to the number of display screen lines to be in the display area. Each number specified defines one display area. The first number defines the bottom-most display area on the screen (the bottom lines of the message area). Subsequent numbers define areas stacked above the bottom-most area. The minimum specification is 4 lines. The maximum specification and the default value for all areas are shown in Figure 7.

		Full Capability		Status Display (USE=SD)	
Device Type	Model	Maximum Area Size	Default Area Size	Maximum Area Size	Default Area Size
2250	1,3	47	14	(Note 1)	(Note 1)
2260	1	8	8	11	11
3036		19	14	23	(13,10)
3066		30	14	(Note 1)	(Note 1)
3158		19	14	23	(13,10)
3277	2	19	14	23	(13,10)
3278	2	20	14	24	(13,10)
3278	2A	16	14	20	(12,8)
3278	3	28	14	32	(13,10)
3278	4	39	14	43	(13,10)
3279	2A,2B	20	14	24	(13,10)
3279	3A,3B	28	14	32	(13,10)

Note 1. This device cannot be specified as a limited capability console.

Figure 7. Maximum and default display area sizes

This parameter is invalid for any other console and is also invalid for a 2260, 3158, 3277 Model 2, 3278 Model 2, 2A, 3, 4, 3279 Model 2A, 2B, 3A, or 3B when USE=MS has been specified for that device. If limited usage is specified, none of these consoles can be specified as the master console. For further information about display consoles, refer to Operator's Library: MVS System Commands.

MCONS=

specifies the unit address or addresses of a console, having input and output capability, to be used as the master console. This parameter is required and can only be specified once. If it is specified, the SECONS parameter cannot be specified in the same macro.

Each unit address specified must be the same as that specified for the device in an IODEVICE macro. The devices that can be used are listed in Figure 8.

address

is the unit address of a console device having input and output capability.

I-address

is the unit address of an input device for a composite console.

O-address

is the unit address of an output device for a composite console.

A device specified as 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 as an alternate console.

If a graphics device will be active as a console, a device that produces printed output must also be specified.

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, all consoles will receive the write-to-operator messages that do not have routing and descriptor codes.

If this parameter is specified, it can only be specified in the CONSOLE macro with the MCONS parameter also specified. It will then affect all consoles based on the routing code(s) specified here and the routing code(s) specified on the individual CONSOLE macros.

Note that, when you elect to use default routing codes, replies to routing code 9 WTOR messages require a valid password to prevent suppression of the reply.

For information about routing and descriptor codes, refer to OS/VS Message Library: Routing and Descriptor Codes. For information on specifying WTO and WTOR macros, refer to OS/VS2 Supervisor Services and Macro Instructions.

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, 3036, 3158, 3277 Model 2, 3278 Model 2, 2A, 3, 4, 3279 Model 2A, 2B, 3A, or 3B that is specified as a console with input/output capability. The number specified is a decimal number from 1 to 32 that indicates:

- The number of PF keys that the operator can associate with commands after IPL, or
- The number of light-pen-detectable numeric indicators in the PFK line of the screen that the operator can associate with commands after IPL (2250, 3158, or 3277 Model 2). On the 2250, only the first 12 keys have indicators available on the PFK line.

The maximum number of program function keys provided for each device is:

PF Keys	Device
32	2250 Model 1 and 3
12	3036
12	3158
12	3277 Model 2
24	3278 Model 2, 3, and 4
12	3278 Model 2A
24	3279 Model 2A, 2B, 3A, and 3B

If the specified console (2250, 3158, and 3277 Model 2 only) 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 and it must be cataloged in the master catalog.

If this parameter is specified for a 2250, either the LIGHTPEN or PRGMKYBD feature, or both, must be specified for that device in an IODEVICE macro.

For information about the programmed-function keyboard, selector pen, and light pen, refer to Operator's Library: MVS System Commands.

ROUTCDE={ALL|routing code}

specifies the routing codes the console is to receive. You specify a number or numbers from 3 through 16 for the master console or 1 through 16 for a secondary console or, by specifying ALL, the console is to receive all routing codes. If this parameter is not specified for the master console, it receives routing codes 1 and 2. If this parameter is not specified for a secondary console, no routing codes are assigned.

For VTAM: When ACSMETH=VTAM is specified in the DATAMGT macro instruction, the master console and any secondary consoles that will be used to enter VTAM commands must have a routing code of 8 and an operator command group of 2 specified.

For information about routing codes, refer to OS/VS Message Library: VS2 Routing and Descriptor Codes.

SECONS=

specifies the unit address or addresses of a console, having input and output capability, to be used as a secondary console. This parameter is optional unless an alternate console is assigned to the master console. In this case, the alternate console must also be specified as a secondary console. A maximum of 98 secondary consoles can be specified. If this parameter is specified, the MCONS parameter cannot be specified in the same macro.

If a device is specified as the master console, it cannot be specified as a secondary console; it can, however, be specified again as an alternate console.

Each unit address specified must be the same as that specified for the device in an IODEVICE macro. The devices that can be used are listed in Figure 7.

address

is the unit address of a console device having input and output capability.

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. (An output-only console may have an alternate console with input and output capability.)

A device specified as 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 as an alternate console.

If a graphics device is to be active as a console, a device that produces printed output must also be specified.

TYPE=JES

specifies that a JES3 dummy secondary console is being defined. For information about JES3 consoles, refer to OS/VS2 System Programming Library: JES3. Only the VALDCMD parameter may be specified when TYPE=JES is used.

USE=

specifies the intended use of a 2260, 3036, 3158, 3277 Model 2, 3278 Model 2, 2A, 3, 4, 3279 Model 2A, 2B, 3A, or 3B console device that has either input/output capability or that functions as the output portion of a composite console. If this parameter is not specified, full capability is assumed. This parameter can only be specified for a secondary console (SECONS).

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.

VALDCMD=command-group

is a number from 1 to 3 that specifies which operator command groups can be entered from this console. One or more command groups may be specified. A value of 1 indicates that the SYS command group can be entered, a value of 2 indicates that the I/O command group can be entered, and a value of 3 indicates that the CONS command group can be entered. See Operator's Library: OS/VS2 MVS System Commands for the commands included in each command group. This parameter can only be specified for a secondary console (SECONS) or when TYPE=JES is specified.

For VTAM: When ACSMETH=VTAM is specified in the DATAMGT macro instruction, any console that will be used to enter VTAM commands must have an operator command group of 2 and a routing code of 8 specified.

EXAMPLE: This macro identifies the master console with the following options:

- The address of the master console is 01F.
- All routing codes (1 through 16) are to be assigned to the master console.
- The OLDWTOR parameter was omitted so this master console will receive all write-to-operator messages that are not assigned routing and descriptor codes.
- An alternate console was not specified so multiple console support will not be used.

```
CONM    CONSOLE   MCONS=01F,ROUTCDE=ALL
```

EXAMPLE:: This macro identifies a secondary console with the following options:

- The console is an output-only console at address 015.
- No routing codes are assigned.
- The master console is the alternate console (default value).

```
CONS    CONSOLE   SECONS=O-015
```

EXAMPLE: The following is an example of a 3033, 3032, or 3031 console configuration which includes:

- Two 3036 CRT/Keyboards
- 7443 Service Record File
- 2955 Remote Analysis Unit

```
AD3036A     IODEVICE     UNIT=3036,                 X
                         ADDRESS=006,...

AD7443A     IODEVICE     UNIT=7443,                 X
                         ADDRESS=007,...

AD3036B     IODEVICE     UNIT=3036,                 X
                         ADDRESS=606,...

AD7443B     IODEVICE     UNIT=7443,                 X
                         ADDRESS=607,...
```

```
         AD2955      IODEVICE     UNIT=2955,                   X
                                  ADDRESS=608,...

         ADMCON      CONSOLE      MCONS=006,.......,           X
                                  ALTCONS=606

         ADSECON     CONSOLE      SECONS=606,.......,          X
                                  ALTCONS=006
```

Figure 8 lists the devices that can be used as consoles and alternate consoles.

Dev.	Mdl.	Console Address	Composite Console I-address	Composite Console O-address	O-address Only	Notes
1052	7	X	X	X		Console keyboard-printer. Can only be specified when attached through the 2150 adapter.
1403				X	X	Printer
1443	N1			X	X	Printer
2250	1 3	X		X		Display unit (note 1 and note 2)
2260	1	X		X	X	Display station (note 1)
2501			X			Card reader
2520	B1		X			Card reader punch
2540R			X			Card reader
2740		X	X	X		Communication terminal (note 1)
3036		X		X		Display unit
3066		X		X		System console. May only be specified in a S370/165II or 168 system configuration (note 1).
3158		X		X		Display unit
3203	4				X	Printer
3210		X	X	X		Console printer keyboard
3211				X	X	High-speed printer. The universal character set is included.
3213				X	X	Printer
3215		X	X	X		Console printer keyboard
3277	1				X	Display unit.

Figure 8 (Part 1 of 2). Console and alternate console support

Dev.	Mdl.	Console Address	Composite Console I-address	Composite Console O-address	O-address Only	Notes
3277	2	X		X	X	Display unit (note 1)
3278	1				X	Display unit (note 1)
3278	2	X		X	X	Display unit (note 1)
3278	2A	X		X	X	Display unit (note 1)
3278	3	X		X	X	Display unit (note 1)
3278	4	X		X	X	Display unit (note 1)
3279	2A	X		X	X	Display unit (note 1)
3279	2B	X		X	X	Display unit (note 1)
3279	3A	X		X	X	Display unit (note 1)
3279	3B	X		X	X	Display unit (note 1)
3284	1			X	X	Printer
3284	2			X	X	Printer
3286	1			X	X	Printer
3286	2			X	X	Printer
3505			X			Card reader
3525			X			Card punch. Must have read feature.

Figure 8 (Part 2 of 2). Console and alternate console support

Notes to Figure 8:

1. Refer to Figure 13 through Figure 22 for IODEVICE specifications and parameters required when this device is used as an operator's console.

2. A 3250 device can be used as an operator console if it is specified as a 2250 Model 3.

Optional for: Complete
I/O Device

Not Applicable for: Eligible Device Table

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

For an I/O device generation, the ACRCODE, ASCII, OPTIONS(CRH), OPTIONS(RDE), and WARN parameters must be specified either explicitly or by default to match the value specified (either explicitly or by default) in the last complete system generation (note, however, that the OPTIONS(RDE) parameter may be added). The remaining parameters may be respecified with or without change. If these parameters are not specified, the defaults are used. Note, however, that any changes to the remaining parameters are not reflected in PARMLIB (members IEASYS00, IEABLD00, and IEAAPF00), and any user modifications to these members are preserved. If you desire such changes to be also reflected in PARMLIB, the appropriate members should be updated directly.

<table>
<tr><td>[symbol]</td><td>CTRLPROG</td><td>[ACRCODE={NO|YES}]
[APFLIB=(dsname,volid [,dsname,volid]...)]
[ASCII={EXCLUDE|INCLUDE}]
[CSA={number|100}][1]
[OPTIONS=([BLDL][,CRH][,DEVSTAT] [,RDE][,RER])]
[PAGNUM={(pp,s)|(1,1)}]
[REAL={size|76}][1]
[SQA={size|1}][1]
[STORAGE={address|0}]
[TZ=({W|E},hh[,mm])]
[VRREGN={size|64}][1]
[WARN={0|n}]</td></tr>
</table>

[1] These parameters may be changed when the system is initialized.

ACRCODE={NO|YES}
specifies the inclusion or exclusion of alternate CPU recovery (ACR) and interprocessor communication (IPC) routines in SYS1.NUCLEUS. If YES is used (the default value), the routines are included. If NO is specified, the routines are not included in the nucleus and fixed storage requirements for the system are reduced by approximately 5000 bytes, plus 1280 bytes per CPU.

Note: You must specify or default ACRCODE=YES for the attached processor system. If ACRCODE=NO is specified, a uniprocessor system initialization occurs. If the resulting system is initialized on a half-duplex or multiprocessing configuration, it will be brought up as a uniprocessor. No multiprocessing capabilities will exist.

Also, if ACRCODE=NO is specified and the system is initialized on a multiprocessing or half-duplex configuration, the operator must ensure that at least 768K bytes of storage is dialed online and is contiguous. If not, NIP issues a warning message and may load a wait state.

APFLIB=dsname, volid

specifies the name and volume identification of data sets that are to be authorized. The name you specify can be from 1 through 44 alphameric, national (@, #, $), or the two special characters (hyphen and the 12-0 overpunch). Names containing more than eight characters must be segmented between periods; one to eight characters may be specified between periods. The first character of any name or name segment must be an alphabetic or national character. Up to sixteen data sets can be specified at system generation as being authorized (SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.LPALIB, and SYS1.SVCLIB are automatically considered authorized). The names of authorized data sets are placed in the IEAAPF00 list in SYS1.PARMLIB.

After system generation, you can increase the number of authorized data sets by including their name directly in the IEAAPF00 list yourself. The number of data sets that may be designated as authorized is dependent on the length of the data set names, up to a total of 250. The table is 2040 bytes, minus a 2-byte header, minus 37 bytes each for SYS1.LINKLIB and SYS1.SVCLIB, leaving 2001 bytes for other entries. The size of each entry in the table is calculated as follows: number of bytes in the name + 6 bytes (for the VOLID) + 1 byte (for the length).

For information about authorized data sets and the authorized program facility, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

For VTAM: When ACSMETH=VTAM is specified in the DATAMGT macro instruction, this parameter must be specified. The dsname used must be SYS1.VTAMLIB and the user defined Network Control Program load module library.

ASCII={**EXCLUDE**|**INCLUDE**}

specifies the inclusion or exclusion of the ASCII translate routine as part of the pageable supervisor. If you plan to use ASCII utilities, this parameter must be specified.

CSA={number|**100**}

is a decimal number that specifies the size of the common service area (CSA). The number you specify is the number of 1K byte blocks of virtual storage to be set aside for the CSA. If you specify zero, the default value (100K) is used. For explicit information about the calculations used to determine the final value of CSA, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

OPTIONS=

specifies which of the following control program options are to be included in, or excluded from, the new system.

BLDL

specifies a BLDL table that is to be fixed in real storage. If this option is not specified, the BLDL table will be pageable.

CRH

specifies the inclusion of channel reconfiguration hardware (CRH) support on a System/370 Model 168 CPU with multiprocessing or channel set switching (CHS) on a processor with channel set switching capability. ACRCODE=YES must also be coded in the CTRLPROG macro in order to obtain CRH/CHS support.

If a failure occurs on one processor, CRH/CHS support allows the other processor to maintain the channel and device activity of both processors.

DEVSTAT

specifies inclusion of the nucleus initialization program (NIP) support that will regard all direct-access devices and magnetic tape drives that are not ready or are not currently attached to the system as offline. In MFT and MVT, this capability was called "SMARTNIP."

RDE

specifies inclusion of the reliability data extractor feature. For information on RDE, refer to OS/VS2 System Programming Library: 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.

PAGNUM={pp,s|1,1}

specifies the number of page and swap data sets, respectively, that can be added to the system after IPL. The pp is specified as either one or two decimal digits 0 to 64. The s is specified as a decimal digit 0 to 9. Realistic numbers should be specified, since SQA (system queue area) space will be reserved for these data sets (64 bytes per page data set and 48 bytes per swap data set). Note that initialization will create space for the specified data sets and then reserve space according to this parameter up to a maximum of 64 page data sets and 25 swap data sets.

REAL={size|76}

specifies the number of 1K byte blocks to be reserved for the virtual equals real (V=R) address area of virtual storage. You specify a value from 0 to 9999. The value you specify is rounded upward to a multiple of 4K bytes. If you specify a value of 0, no jobs will be executed in a V=R region.

SQA={size|1}

specifies the number of 64K byte segments of virtual storage to be added to the minimum SQA size (3 segments) to be reserved for the system queue area. You specify a value from 0 to 126, where the maximum you may specify is calculated as follows:

```
126-((4((P+3)/4)+C)+63)/64.
```

P=size of the PLPA in 1K blocks
C=size of the CSA in 1K blocks

Note: Ignore the remainder when dividing.

STORAGE={address|0}

is a decimal number from 0 to 16777216 that specifies the highest block of real storage you ever plan on having online at one time. When the system is initialized, either this address or the actual amount of storage online will be used to build the page frame tables.

In a multiprocessing system, if this amount is not online when the system is initialized, you can increase the amount of storage after system initialization up to the address specified in this parameter. If this parameter is not specified, the amount of real storage online when the system is initialized is considered the maximum value. To increase the amount of storage, you must reinitialize the system.

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.

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 is created. This will result in local time equaling the time in the TOD clock (which the system assumes to be on Greenwich Mean Time).

VRREGN={size|64}
specifies the amount of real storage that will be allocated to a job requiring virtual equals real space (V=R) when a region size is not specified for the job. You specify a value from 0 to 9999 which is the number of 1K byte blocks to be used as the default value.

The value specified in this parameter may exceed the value specified in the REAL parameter of this macro. Specifying a greater size means that the system resources are not to be made available to a V=R job for which a region has not been specified.

WARN={0|n}

0
specifies that the power warning feature is not supported.

n
is a value specified in milliseconds and may be a number from 1 to 16777215.

When a number is specified for this parameter, this indicates that the power warning feature is supported. A value of 1 indicates that there is no time delay.

EXAMPLE: This macro specifies the control program options as follows:

- NIP support is to be included.
- The BLDL table is to be pageable.
- The ASCII translate routine is to be included.
- The system is to be a uniprocessor system.
- The size of the common service area is to be one hundred 1K byte blocks (the default value).
- The system is to contain seventy-six 1K byte blocks of virtual=real address space (the default value).

- The size of the system queue area in virtual storage is to be one 64K byte segment (the default value).

- The address of the highest block of real storage to be online is 0 (the default value).

- Greenwich Mean Time is to be used.

- Sixty-four 1K byte blocks of V=R space is to be the default value for jobs requiring V=R space when a region size is not specified (the default value).

- Two data sets are authorized and included in APFLIB.

```
CTRLPG   CTRLPROG   OPTIONS=DEVSTAT,ASCII=INCLUDE,         X
                 ACRCODE=NO,APFLIB=(SYS1.XREG,SYSRES,      X
                 A.B.C.DATASET,PACK01)
```

EXAMPLE: This macro specifies the control program options as follows:

- NIP support is to be included.

- The reliability data extractor (RDE) feature is to be included.

- Reduced error recovery support for magnetic tape is to be included.

- The BLDL table is to be fixed.

- The size of the virtual=real address space is to be four 1K byte blocks.

- The time zone deviation from GMT is five hours west of the Greenwich Meridian.

- The ASCII translate routine is to be included.

- The address of the highest block of real storage to be online is 2,097,152 (X'200000').

- Alternate CPU recovery and interprocessor communications routines in a multiprocessing configuration are to be included (the default value).

- The common service area is to be one hundred 1K byte blocks (the default value).

- One 64K byte segment of virtual storage is to be reserved as the system queue area (the default value).

- Sixty-four 1K byte blocks of V=R storage is to be the default value for jobs requiring V=R space when a region size is not specified for a job (default value).

```
CTRL   CTRLPROG   OPTIONS=(DEVSTAT,RDE,RER,BLDL),         X
               REAL=64,TZ=(W,5),ASCII=INCLUDE,            X
               STORAGE=2097152
```

DATAMGT

Optional for: Complete
I/O Device

Not Applicable for: Eligible Device Table

The DATAMGT macro instruction is used to specify the inclusion of optional access methods, graphic programming services, industry subsystem support, the character arrangement table modules, or the universal character set (UCS) images to be used as defaults. (UCS images are automatically included during system generation if a 1403 printer with the UCS feature, a 3203-4 printer, a 3211 printer, a 3886 optical character reader, or a 3890 document processor is included in the system.) This macro instruction is optional. If it is not specified, none of the access method or graphics options will be included, and all of the UCS images will be included. If the 3800 printer is included in the system, a basic set of character arrangement table modules, a graphic character modification module, and an FCB module will automatically be included in SYS1.IMAGELIB. Additional character arrangement table modules and a graphic character modification module can be included by specifying the TABLE parameter. With 3800 Enhancements, library character set modules are included in SYS1.IMAGELIB.

For additional information on the character arrangement table, graphic character modification modules, FCB modules, and library character sets see _IBM 3800 Printing Subsystem Programmer's Guide_.

The following access methods are always included in the system:

BDAM	basic direct access method
BPAM	basic partitioned access method
BSAM	basic sequential access method
QSAM	queued sequential access method
VSAM	virtual storage access method

For an I/O device generation, if the ACSMETH, IND, and/or GRAPHCS parameters were specified during the last complete generation, they must be respecified without change. The UCSDFLT parameter, however, can be changed. It does not have to be respecified if it has already been specified in a previous system generation.

For information on the optional access methods, refer to:

OS/VS2 MVS Data Management Services Guide
OS/VS BTAM
OS TCAM Concepts and Facilities
VTAM Concepts and Planning

For information on graphic programming services, refer to _OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit_ and _OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment)_.

For information on the industry subsystem support, see _IBM System/370 Subsystem Support Services User's Guide_.

For information on UCS images, refer to _OS/VS2 System Programming Library: Data Management_.

[symbol]	**DATAMGT**	[ACSMETH=(method[,method]...)] [GRAPHCS=([GSP][,PORRTNS])] [IND={NO|YES}] [TABLE={(imageset[,imageset]...)|ALL}] [UCSDFLT={(image[,image]...)|ALL}]

ACSMETH=method
specifies one or more of the following access methods:

BTAM	basic telecommunication access method
GAM	graphics access method
ISAM	basic and queued indexed sequential access methods
TCAM	telecommunications access method
VTAM	virtual telecommunications access method

TCAM and/or VTAM must be specified in the DATAMGT macro. If neither access method is specified, a diagnostic msssage is issued and the system generation is terminated.

If VTAM is specified, the APFLIB parameter must also be specified in the CTRLPROG macro.

GRAPHCS=
specifies the inclusion of graphic programming services. The values specified can be coded in any order.

GSP
specifies that the graphic subroutine package is to be included in SYS1.LINKLIB.

PORRTNS
specifies that problem oriented routines (PORs) are to be included in SYS1.LINKLIB.

IND={NO|YES}
specifies the inclusion or exclusion of industry subsystem support. If IND=YES is specified, one or more industry subsystems can be applied to the new system and VTAM will automatically be included in the system. The SYS1.VTAMLIB and SYS1.INDMAC data sets must be defined before Stage II. If you specify VTAM as an access method, this parameter is not required.

The industry subsystems are supported through SNA (Systems Network Architecture) and require the specification of VTAM as the access method. The industry subsystems are:

- IBM 3270 Information Display System (SNA)
- IBM 3600 Finance Communication System
- IBM 3650 Retail Store System
- IBM 3660 Supermarket System
- IBM 3767 Communication Terminal
- IBM 3770 Data Communication System (System/32)
- IBM 3790 Communication System

TABLE={imageset|**ALL**}

specifies the sets of character arrangement table modules that are to be included in SYS1.IMAGELIB for the 3800. See Figure 10 for the tables that can be specified.

If the 3800 is included in the system, the following modules will automatically be included in SYS1.IMAGELIB as separate members in addition to any specified:

- Character arrangement table modules

Table Name	**Character Set**
GS10	Gothic-10
GS12	Gothic-12
GS15	Gothic-15
GSC	Gothic-15 Condensed
GF10	Gothic-10—folded
GF12	Gothic-12—folded
GF15	Gothic-15 folded
GFC	Gothic-15 Condensed—folded
GU10	Gothic-10 underscored
GU12	Gothic-12 underscored
GU15	Gothic-15 underscored
GUC	Gothic-15 Condensed underscored
TU10	Text 1 & 2 underscored
DUMP	Gothic-15 and underscored Gothic-15

- Graphic character modification module, GRAF_bbbb_. This module contains the World Trade National Use Graphic characters.

- Forms control buffer module, FCB3STD3. This module contains the code to cause the vertical spacing of an 11-inch form to be 8 lines per inch.

If the imageset specified is either T3211 or T1403, the graphic character modification module GRAFSPC1 will be automatically included in SYS1.IMAGELIB. This module contains the graphic characters used by the G11 and GN character arrangement tables to substitute the open bracket ([) for the cent sign (¢), the close bracket (]) for the exclamation point (!), and extends the character arrangement by adding the backward slash (\).

If ALL is specified, all the character arrangement table modules in Figure 10 for the 3800 will be included in SYS1.IMAGELIB as well as modules GRAFbbbb, GRAFSPC1, and FCB3STD3. Note that the tables for the 3800 are not compatible with the standard UCS images that can be specified for the 1403 and 3211 printers.

For additional information on the character arrangement table, graphic character modification, and FCB modules, see _IBM 3800 Printing Subsystem Programmer's Guide_.

UCSDFLT={image|ALL}

specifies the UCS images that are to be used as defaults when a job does not specify an image through its job control language statements. If this parameter is not specified, all UCS images will be used as defaults.

It is recommended that all UCS images that can produce valid results as defaults be specified. The images that can be specified as defaults for the 1403, 3203-4, and 3211 printers are listed in Figure 9. Note that these UCS images are not compatible with the character arrangement tables that can be specified for the 3800 Printing Subsystem.

Image	**IBM Standard Character-Set Image** **1403 or 3203-4 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.
Q	PL/I scientifically preferred character set, 60 graphics.
R	FORTRAN/COBOL commercial, 52 graphics.
S	Text printing, commercial, 84 graphics.
T	Text printing, scientific, 120 graphics.
X	High speed alphameric, 1403 Model 2, 40 graphics.
Y	High speed alphameric, 1403 Model N1, 42 graphics.

Image	**IBM Standard Character-Set Image** **3211 Printer**
A11	Standard commercial character set, 48 graphics.
G11	ASCII character set, 63 graphics.
H11	Scientific character set, 48 graphics.
P11	PL/I character set, 60 graphics.
T11	Text printing, 120 graphics.

Figure 9. Standard character-set images

EXAMPLE: This macro specifies the following options:

- The inclusion of BTAM, TCAM, and ISAM.
- The inclusion of the graphic subroutine package and problem oriented routines in SYS1.LINKLIB.
- All UCS images are to be used as defaults when a job does not specify an image (the default value).

IBM Standard Character Arrangement Table Modules

IBM 3800 Printing Subsystem

Imageset	Table Names	Character Set	Pitch	No. of Graphic Characters
T3211	A11	Gothic	10	48
	G11	Gothic	10	63
	H11	Gothic	10	48
	P11	Gothic	10	60
	T11	Text 1 & 2	10	120[1]
T1403	AN	Gothic	10	48
	GN	Gothic	10	63
	HN	Gothic	10	48
	PCAN	Gothic	10	48
	PCHN	Gothic	10	48
	PN	Gothic	10	60
	QN	Gothic	10	60
	QNC	Gothic	10	60
	RN	Gothic	10	52
	XN	Gothic	10	40
	YN	Gothic	10	42
	SN	Text 1 & 2	10	84[1]
	TN	Text 1 & 2	10	120[1]
TOCR	AOA	OCR-A, Gothic	10	48[1]
	AOD	OCR-A, Gothic	10	48[1]
	AON	OCR-A, Gothic	10	48[1]
	OAA	OCR-A, Gothic	10	48[1]
	ODA	OCR-A, Gothic	10	48[1]
	ONA	OCR-A, Gothic	10	48[1]
	BOA	OCR-B, Gothic	10	48[1]
	BON	OCR-B, Gothic	10	48[1]
	OAB	OCR-B	10	48
	ONB	OCR-B, Gothic	10	48[1]
TKAT	2773	Katakana, Gothic	10	62[1]
	2774	Katakana, Gothic	10	108[1]
	KN1	Katakana, Gothic	10	127[1]
TFMT	FM10	Format	10	36
	FM12	Format	12	36
	FM15	Format	15	36

[1] Uses two WCGMs (writable character generation modules).

Figure 10. Standard character arrangement table modules

The T3211 and TFMT image sets for the 3800 are to be included in SYS1.IMAGELIB if a 3800 is present in the system.

```
DATAMGT   DATAMGT   ACSMETH=(BTAM,TCAM,ISAM),                X
                    GRAPHCS=(PORRTNS,GSP),TABLE=(T3211,TFMT)
```

Required for: Complete
I/O Device

Not Applicable for: Eligible Device Table

The DATASET macro instruction can be used to define system data sets (including the master catalog) during system generation. Additionally, it can be used to include your own routines in the system data sets as part of system generation. Also, some of the optional system data sets must be cataloged, if certain conditions exist, even if space is not allocated for them. The DATASET macro can be used for this. Although there are conditions that make the DATASET macro required, as discussed later, you need not use the DATASET macro exclusively. Prior to system generation you can use JCL to allocate space for the nonVSAM system data sets and the Access Method Services to catalog them and you can use the Access Method Services to allocate space for and catalog the VSAM system data sets.

Page and swap data sets can also be defined using the DATASET macro. At least three page data sets must be defined before or during system generation. No swap data sets are required; however, it is recommended that swap data sets be supplied for maximum system performance. Up to 25 page data sets and up to ten swap data sets may be defined, and additional page and swap data sets may be defined after system generation. When a page data set is defined using a DATASET macro, space is allocated for it, it is cataloged, and its name is included in the IEASYS00 system parameter list in SYS1.PARMLIB. Like VSAM system data sets, page or swap data sets can also be defined using the Access Method Services. If you use the Access Method Services, the page or swap data sets are not automatically included in the IEASYS00 list. Refer to the chapter "Defining the Page and Swap Data Sets" for additional information.

For a complete system generation, the DATASET macro can be used to include your own routines in the system data sets. By specifying them in DATASET, your routines, as members of cataloged partitioned data sets, are link-edited or copied into the system data sets during the Stage II part of system generation. Figure 11 lists all of the system data sets to which routines can be added.

Your own routines, macros, parameters, procedures, and authorized EXCP appendages can be included by specifying the name of the cataloged partitioned data set in the PDS parameter and the name or names of the members to be included in the MEMBERS parameter of the DATASET macro that defines the system data set. Additionally, you can specify those routines you want included in SYS1.LINKLIB, SYS1.LPALIB, or SYS1.SVCLIB that are to be made resident in real storage when the system is loaded. The names you specify will also be included in the IEAFIX01 list in SYS1.PARMLIB.

EXCP appendages that you write for use by unauthorized programs can be included by specifying the name or names of the appendages in the ABEAPP, CHEAPP, EOEAPP, PCIAPP, or SIOAPP parameters when you define the system data set into which they are to be included. During system generation, the EXCP appendages are included in the specified system data sets and their names and types are placed in the IEAAPP00 list in SYS1.PARMLIB. If you plan to include the appendages at a later time, you can still include the names of the appendages in the DATASET macro. During system generation, the name and type of appendage will be included in SYS1.PARMLIB although nothing will be placed in the system data set.

For examples of including your own routines, refer to "Adding User-Written Routines to the MVS System Control Program" in the chapter "Preparing for System Generation."

For a complete system generation, the unique name of the new master catalog must be specified in a DATASET macro. Some system data sets must be cataloged even if space is not going to be allocated for them. All of the above are detailed in Figure 11 in this chapter and in the chapter "Selecting and Defining the System Data Sets." A DATASET macro must be specified for the optional system data sets and for those system data sets that are not required to reside on the system residence volume or are not assumed to reside on it. (See Figure 11 for a list of the optional system data sets.)

If a system data set has been predefined and the space parameter has been omitted or SPACE=(CYL,(0)) has been specified in the DATASET macro, the data set will not be cataloged and space will not be allocated for it.

For an I/O device generation, SYS1.LINKLIB, SYS1.IMAGELIB, and SYS1.LPALIB must be specified in a DATASET macro. Also, the optional system data set SYS1.DCMLIB must be specified, if it exists and is to be updated. The SPACE parameter need not be specified. If it is specified, it will be ignored. However, with 3800 Enhancements, you can use the space parameter to specify secondary extents for SYS1.IMAGELIB.

Information about EXCP appendages can be found in OS/VS2 System Programming Library: Data Management.

<table>
<tr><td>[symbol]</td><td><b>DATASET</b></td><td>{system data set name|<b>PAGEDSN</b>=page data set name|<b>SWAPDSN</b>=swap data set name}
[<b>SPACE</b>=({blocksize|<b>CYL</b>|<b>TRK</b>},(quantity))]
[<b>VOL</b>={({volser|SYSRES},{devtype|3330})}]</td></tr>
</table>

The validity of the following parameters depends on the type of system data set being defined.

<table>
<tr><td></td><td></td><td>[<b>ABEAPP</b>=(name[,name]...)]
[<b>CHEAPP</b>=(name[,name]...)]
[<b>EOEAPP</b>=(name[,name]...)]
[<b>MEMBERS</b>=(name[,name]...)]
[<b>NAME</b>={master catalog name|duplex data set name}]
[<b>PCIAPP</b>=(name[,name]...)]
[<b>PDS</b>=name]
[<b>RESIDNT</b>=(name[,name]...)]
[<b>SIOAPP</b>=(name[,name]...)]</td></tr>
</table>

ABEAPP=name

specifies a user-written abnormal-end appendage. You specify a value from WA to Z9 which is the last two characters of the appendage name; the first six characters of the name (IGG019) are automatically assigned. Up to 84 appendages may be specified in this parameter. This parameter can only be specified for SYS1.LPALIB or SYS1.SVCLIB. If this parameter is specified, then the PDS parameter must be specified.

CHEAPP=name

specifies a user-written channel-end appendage. You specify a value from WA to Z9 which is the last two characters of the appendage name; the first six characters of the name (IGG019) are automatically assigned. Up to 84 appendages may be specified in this parameter. This parameter can only be specified for SYS1.LPALIB or SYS1.SVCLIB. If this parameter is specified, then the PDS parameter must be specified.

EOEAPP=name

specifies a user-written end-of-extent appendage. You specify a value from WA to Z9 which is the last two characters of the appendage name; the first six characters of the name (IGG019) are automatically assigned. Up to 84 appendages may be specified in this parameter. This parameter can only be specified for SYS1.LPALIB or SYS1.SVCLIB. If this parameter is specified, then the PDS parameter must be specified.

MEMBERS=name

specifies the member or members of a cataloged partitioned data set you are defining. Each name cannot exceed eight alphameric characters. The first character must be alphabetic. Up to 20 members may be specified.

If this parameter is specified, the PDS parameter must be specified. During Stage II, each member is selectively copied into the system data set. For SYS1.NUCLEUS, if the PDS parameter is specified and this parameter is not specified, the members of the partitioned data set will be copied into SYS1.NUCLEUS as individual members. If both the PDS and MEMBERS parameters are specified, the members will be link-edited into the nucleus being generated (IEANUC01).

A name appearing as a value in the MEMBERS parameter should not appear as a value in the RESIDNT parameter.

Figure 11 lists the system data sets for which this parameter may be specified and the form that the members must be in for inclusion in the system data sets.

For an I/O device generation, this parameter is ignored.

NAME=name

specifies the unique name of the master catalog or the unique name of the duplex page data set. The name cannot exceed 44 characters. The characters can be alphameric, national (@, #, $), or the two special characters (hyphen and 12-0 overpunch). Names containing more than eight characters must be segmented between periods; one to eight characters may be specified between periods. The first character of any name must be an alphabetic or national character.

This parameter must be specified for the master catalog, even if it was predefined. Refer to Figure 11 for additional information.

PAGEDSN=name

specifies the name of a page data set. The name cannot exceed 44 characters. The characters can be alphameric, national (@, #, $) or the two special characters (hyphen and 12-0 overpunch). Names containing more than eight characters must be segmented between periods; one to eight characters may be specified between periods. The first character of any name must be an alphabetic or national character.

Note: The index value you specify in the GENERATE macro is not appended to the page data set name specified in this parameter.

At least three page data sets must be defined before or during system generation.

Refer to Figure 11 and to the chapter "Defining the Page and Swap Data Sets" for additional information.

PCIAPP=name
specifies a user-written program-controlled interrupt (PCI) appendage. You specify a value from WA to Z9 which is the last two characters of the appendage name; the first six characters of the name (IGG019) are automatically assigned. Up to 84 appendages may be specified in this parameter. This parameter can only be specified for SYS1.LPALIB or SYS1.SVCLIB. If this parameter is specified, then the PDS parameter must be specified.

PDS=name
specifies the complete, fully qualified name of a cataloged partitioned data set that contains your own routines that you want included in a system data set during system generation. The name you specify cannot exceed 44 characters (including SYS1.) The characters can be alphameric, national (@, #, $), or the two special characters (hyphen and 12-0 overpunch). Names containing more than eight characters must be segmented between periods; one to eight characters may be specified between periods. The first character of any name or name segment must be an alphabetic or national character. The partitioned data set specified must have been cataloged in the generating system.

Except for SYS1.NUCLEUS, if this parameter is specifed, the MEMBERS or RESIDNT parameters or both must be specified. Each member of the partitioned data set is selectively copied into the system data set. For SYS1.NUCLEUS, if this parameter is specified and the MEMBERS parameter is not specified, the members of the partitioned data set are copied into SYS1.NUCLEUS as individual members. If both parameters are specified, the members are link-edited into the nucleus being generated (IEANUC01).

Figure 11 lists the system data sets for which this parameter may be specified and the form that the members must be in for inclusion in the system data sets.

For an I/O device generation, this parameter is ignored.

RESIDNT=name
specifies the member or members of a cataloged partitioned data set that are to be copied into the system data set you are defining. Each name you specify cannot exceed eight alphameric characters. The first character must be alphabetic. Up to ten members may be specified. The names you specify are included in the IEAFIX01 list of members in SYS1.PARMLIB during Stage II and will become part of the resident portion of the link pack area when the system is initialized. If this parameter is specified, the PDS parameter must be specified.

A name appearing as a value in the RESIDNT parameter should not appear as a value in the MEMBERS parameter.

Figure 11 lists the system data sets for which this parameter may be specified and the form that the members must be in for inclusion in the system data sets.

For an I/O device generation, this parameter is ignored.

SIOAPP=name
specifies a user-written start I/O appendage. You specify a value from WA to Z9 which is the last two characters of the appendage name; the first six characters of the name (IGG019) are automatically assigned. Up to 84 appendages may be specified in this parameter. This parameter can only

be specified for SYS1.LPALIB or SYS1.SVCLIB. If this parameter is specified, the PDS parameter must be specified.

SPACE=
specifies a request that space allocation be based solely on the values given in this parameter.

{blocksize|**CYL**|**TRK**}
specifies the unit of space to be allocated. Space can be allocated in the DATASET macro in maximum block length, cylinders, or tracks. Some data sets must be allocated in specific units. Refer to Figure 11 for this information.

quantity
specifies how many units of space (blocks, cylinders, or tracks) are to be allocated. Depending on the data set, 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 are to be allocated.

secondary quantity
specifies how many more blocks, cylinders, or tracks are to be allocated if additional space is required. Figure 11 lists the data sets for which secondary space may be allocated. If no secondary allocation is permitted, omit the field; do not code a zero for the secondary quantity.

Caution: When you elect to use secondary extents for data sets which have nucleus resident DEBs, and they are extended and accessed before re-IPL, an abend may occur because the updated DEB is not available for use until re-IPL.

directory blocks
specifies the number of 256-byte blocks to be allocated for the directory of a partitioned data set. Figure 11 lists the system data sets for which this applies.

The amount of space for the SYS1.LOGREC system data set is always calculated and allocated by the generating system on the residence volume of the system being generated.

SWAPDSN=name
specifies the name of a swap data set. The name cannot exceed 44 characters. The characters can be alphameric, national (@,#,$), or the two special characters (hyphen and 12-0 overpunch). Names containing more than eight characters must be segmented between periods; one to eight characters may be specified between periods. The first character of any name must be an alphabetic or national character.

Note: The index value you specify in the GENERATE macro is not appended to the swap data set name specified in this parameter.

Refer to Figure 11 and to the chapter "Defining the Page and Swap Data Sets" for additional information.

system data set name
specifies the system data set to be defined during system generation. Except for SYS1.LOGREC, which is automatically allocated by the generating system, all of the system data sets can be defined using the DATASET macro. The system data sets that can be specified are:

BRODCAST
CMDLIB
DCMLIB
DUMPnn
DUPLEXDS
HELP
IMAGELIB
INDMAC
JES3LIB
LINKLIB
LPALIB
MACLIB
MANn
MANX
MANY
NUCLEUS
PARMLIB
PROCLIB
SAMPLIB
STGINDEX
SVCLIB
TCOMMAC
TELCMLIB
UADS
VSCATLG (master catalog)
VTAMLIB

Some system data sets are required and some system data sets are required to, at least, be cataloged. If certain system data sets are not specified in a DATASET macro, they are assumed to reside on the system residence volume. Refer to Figure 11 for this information.

VOL=

specifies the volume serial number of the volume that is to contain the system data set or page data set and the type of device on which the volume is to reside.

{volser|SYSRES}

specifies the volume serial number. The value you specify cannot exceed six alphameric characters.

{devtype|3330}

specifies the device type. The device types that may be specified are: 2305-1, 2305-2, 2314, 3330, 3330-1, 3340, 3350, 3375, and 3380.

The VOL parameter works in conjunction with the RESVOL parameter in the GENERATE macro. The default value for RESVOL is the same as that for VOL. If RESVOL is changed, the default value for VOL becomes the changed value.

If either the volume serial number or device type is specified, both must be specified.

EXAMPLE: This macro allocates and catalogs SYS1.MANX.

```
MANX    DATASET    MANX,SPACE=(CYL,(10))
```

EXAMPLE: This macro identifies SYS1.LPALIB to the generating system. SYS1.LPALIB has been predefined on a 3340 volume labeled 222222.

```
LPALIB    DATASET    LPALIB,VOL=(222222,3340)
```

EXAMPLE: This macro defines the master catalog. The name of the master catalog is SYS1.VSCATLG and it is to reside on the system residence volume.

```
VSCATLG    DATASET    VSCATLG,SPACE=(CYL,(50,5)),                 X
                  NAME=SYS1.VSCATLG
```

EXAMPLE: This macro defines the page data set, PAGE1 on a 2314 volume labeled PAGE.

```
  PAGE1    DATASET   PAGEDSN=PAGE1,VOL=(PAGE,2314),                 X
                   SPACE=(CYL,(134))
```

EXAMPLE: This macro defines the swap data set, SWAP5 on a 3330 volume labeled SWAP.

```
  SWAP5    DATASET    SWAPDSN=SWAP5,VOL=(SWAP,3330),                X
                   SPACE=(CYL,(20))
```

EXAMPLE: This macro defines the duplex page data set, DUPLEX1 on a 3330 volume labeled DUPLX.

```
  DUPLEX1    DATASET    DUPLEXDS,NAME=DUPLEX1,                      X
                   VOL=(DUPLX,3330),SPACE=(CYL,(10))
```

EXAMPLE: This macro defines the SYS1.NUCLEUS data set. Note that no secondary space allocation is allowed in the SPACE parameter.

```
  NUCLEUS    DATASET   NUCLEUS,SPACE=(CYL,(30,,10))
```

Data Set	SPACE= (unit,	(pri.,	sec.,	dir. blks.))	PDS=	MEMBERS=	RESIDNT=	User- Written Rtns Format	Notes
BRODCAST	x	x							16
CMDLIB	x	x	x	x	x	x		Load Module	1
DCMLIB	x	x		x					21
DUMP*nn*	x	x							4, 21
HELP	x	x	x	x					17, 21
IMAGELIB	x	x	x	x	x	x		FCB, UCS Images, 3800 Support Modules	5, 15
INDMAC	x	x	x	x				Finance Macros	21, 25
JES3LIB	x	x	x	x	x	x			27
LINKLIB	x	x	x	x	x	x	x	Load Module	6, 7, 15, 18
LOGREC									8, 9, 15
LPALIB	x	x	x	x	x	x	x	Load Module	6, 7, 10, 15, 18
MACLIB	x	x	x	x	x	x		System Macros	7, 15, 18
MAN*n*	x	x							29
MANX	x	x							12
MANY	x	x							12
NUCLEUS	x	x		x	x	x		Load Module	9, 11, 15
page data sets	x	x							13, 14, 22, 26
PARMLIB	x	x		x	x	x		System Parameters	15, 18
PROCLIB	x	x	x	x	x	x		System Procedures	7, 15, 18
SAMPLIB	x	x	x	x	x				15, 18
STGINDEX	x	x							14, 15, 22, 23
SVCLIB	x	x	x	x	x	x	x	Load Modules	7, 9, 15
swap data sets	x	x							2, 13, 14, 22
TCOMMAC	x	x	x	x	x				7, 21, 28
TELCMLIB	x	x	x	x	x	x		Load Modules	19, 21
UADS	x	x	x	x					20
VSCATLG	x	x	x						3, 15, 22
VTAMLIB	x	x	x	x	x	x		Load Module	24

Figure 11. Defining the system, page, and swap page data sets using the DATASET macro

Notes to Figure 11:

1. This system data set must be cataloged and space must be allocated unless the TSO macro is specified to include a limited version of the TSO command system (CMDS=NO).

2. Swap data sets cannot be defined on 2314 devices.

3. A DATASET macro must be specified. If the system data set has been predefined, the SPACE parameter must not be specified.

4. From one to ten DUMPnn (DUMP00 through DUMP09) system data sets may be specified.

5. Forms control buffer (FCB) modules, universal character set (UCS) images, 3800 character arrangement table modules, 3800 copy modification modules, and graphic character modification modules can be included. With 3800 Enhancements, library character set modules can be included, and secondary allocation can be specified.

6. A name appearing as a value in the MEMBERS parameter should not appear as a value in the RESIDNT parameter, and vice versa. The total number of names (MEMBERS and RESIDNT) must not exceed 20.

7. For improved system efficiency, it is recommended that space be allocated on a cylinder boundary.

8. The amount of space for LOGREC is always calculated and allocated on the system residence volume. The system generation process defines (allocates and catalogs) SYS1.LOGREC; you must not specify a DATASET macro for SYS1.LOGREC.

9. This system data set must reside on the system residence volume.

10. If a type 3 SVC routine is to be included, the name of the SVC routine must be of the form: IGC00nnn where nnn is a unique decimal number assigned in descending order from 255 to 200. This name must be the name of a member of a cataloged partitioned data set.

    If a type 4 SVC routine is to be included, the name of the SVC routine must be of the form: IGCssnnn where nnn is a unique decimal number assigned in descending order from 255 to 200 and ss is the number of the load module minus 1. This name must be the name of a member of a cataloged partitioned data set.

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

11. A maximum of ten load modules can be included in the nucleus. If type 1 or type 2 SVC routines are being included in the nucleus, each load module can contain more than one SVC routine.

    If a type 1 or type 2 SVC routine is to be included, the name of the SVC routine must be of the form: IGCnnn, where nnn is a unique decimal number assigned in descending order from 255 to 200.

12. This system data set must be cataloged. Space is required only if SMF or MF/1 recording is going to be done.

13. If the SPACE parameter is specified, the VOL parameter must also be specified. If SPACE is not specified, VOL need not be specified.

14. Space must be allocated in cylinders.

15. This system data set is required.

16. This system data set must be cataloged. Space is required only if time-sharing messages are to be written.

17. This system data set is required if the time-sharing HELP command is to be used.

18. If no DATASET macro is coded for this data set, it will be assumed that the data set resides on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

19. This system data set is required if BTAM and/or TCAM is included in the system.

20. This system data set must be cataloged. Space is required only if terminal sessions are to be initiated.

21. If this optional system data set is to be part of the system control program, a DATASET macro must be specified, even if it has been predefined. If it has been predefined, omit the SPACE parameter in DATASET.

22. It is strongly recommended that during your first system generation the master catalog, page data sets, SYS1.STGINDEX, and swap data sets be allocated space on a single volume. After the initial IPL, performance considerations may require a different configuration.

23. The volume on which this data set resides must be permanently mounted or reserved while the data set is in use.

24. This system data set is required if VTAM is included in the system. It must also be specified in the APFLIB parameter of the CTRLPROG macro instruction.

25. This system data set is required if the industry subsystem support is included in the system.

26. At least three data sets will have to be specified. If the system data set has been predefined, the SPACE parameter must not be specified.

27. This system data set is required if JES3 is included in the system.

28. This system data set is required if ACF/TCAM is included in the system.

29. Valid only for System Extension Release 2. Space is required only if recording is going to be done by SMF or other measurement facilities.

Optional for: Complete
I/O Device

Not Applicable for: Eligible Device Table

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

[symbol]	**EDIT**	**BLOCK=([n],[n],....)**
		CHECKER=([name],[name],...)
		CONVERT=({ASIS|CAPS|CAPSONLY},...)
		DATEXIT([name],[name],...)
		DSTYPE=(name[,name]...)
		FIXED=([d-m],[d-m],...)
		FORMAT=({FIXED|FXDONLY|VAR},...)
		PRMPTR=([name],[name],...)
		USEREXT=([name],[name],...)
		USERSRC=({DATASET|INCORE|INLIST},...)
		VAR=([d-m],[d-m],...)

Commas are required to denote missing operands in all parameters except DSTYPE. If this macro is not included in a complete generation, or if it is not included in an I/O device generation, or if operands are omitted, defaults are used. These defaults are listed in Figure 11.

BLOCK=n
specifies, in bytes, the default length of blocks for the data sets created by the TSO EDIT command.

The n is an integer from 8 to 32760 that specifies the default block size for any data set of the type specified in the corresponding name field of the DSTYPE parameter.

CHECKER=name
specifies the processor that is to be used by the TSO EDIT command to check lines in the data set for proper syntax. The selected processor must be installed prior to system generation.

The name is the 1- to 8-character module name that the TSO EDIT command will use to invoke the CHECKER for data sets of the type specified by the corresponding name field in the DSTYPE parameter.

CONVERT=
specifies whether input data to the TSO EDIT command should be converted to uppercase.

ASIS
specifies that the ASIS operand of the TSO EDIT command, which requests that the data remain as

entered (for example, upper and lower case), is to be the default for data sets of the type specified by the corresponding name field of the DSTYPE parameter.

CAPS

specifies that the CAPS operand of the TSO EDIT command, which requests uppercase conversion of data, is to be the default for data sets of the type specified by the corresponding name field of the DSTYPE parameter.

CAPSONLY

specifies that the CAPS operand of the TSO EDIT command is the only permitted operand. The ASIS operand may not be specified on the TSO EDIT command.

DATEXIT=name

specifies the name of the exit routine invoked for the EDIT RENUM subcommand. If this parameter is not specified, the default is a null field for all types except VSBASIC. The default value for VSBASIC is ICDQRNME. One to eight characters may be specified.

DSTYPE=name

specifies IBM-standard data set types or new data set types defined by you that are to have their attributes modified or defined for use by the TSO EDIT command. (See Figure 1.)

The name specified is the 1 to 8 character data set type as it will appear in the TSO EDIT command. Any data set type permitted by the TSO EDIT command or defined by the installation can be specified. Up to ten new data set types defined and up to sixteen standard data set types can be specified in this parameter.

FIXED=d-m

specifies, in bytes, the default and maximum allowable length of logical fixed-length records for data sets created by the TSO EDIT command.

d is an integer from 1 to 255 that specifies the default logical record length for any fixed format data set of the type specified in the corresponding name field of the DSTYPE parameter. The corresponding n value, if specified in a BLOCK parameter, must be a multiple of this value. If the variable record format is the default for the data set type (VAR=), a value of 0 should be coded for d when this subparameter is used to define the maximum size of a fixed-length record.

m is an integer from 1 to 255 that specifies the maximum logical record length of any fixed-format data set of the type specified in the corresponding name field of the DSTYPE parameter. The corresponding n value, if specified in a BLOCK parameter, must be a multiple of this value, and the m value must be greater than or equal to the d value.

The hyphen must be coded as shown.

FORMAT=

specifies the allowable record format for new data sets created by the TSO EDIT command.

FIXED

specifies that fixed-record format is the default value for data sets of the type specified in the corresponding name field of the DSTYPE parameter.

FXDONLY

specifies that only the fixed-record format is valid for data sets of the type specified in the corresponding name field of the DSTYPE parameter. Specifying the VAR parameter for this data set type is invalid.

VAR

specifies that variable-record format is the default value for data sets of the type specified by the corresponding name field of the DSTYPE parameter.

PRMPTR=name

specifies the installation-supplied processor that is to be used by the RUN subcommand of the TSO EDIT command. RUN invokes an installation-supplied processor to execute programs under the TSO EDIT command.

The name is the 1- to 8-character prompter name for the data sets of the type specified in the corresponding name field of the DSTYPE parameter.

USEREXT=name

specifies the installation-supplied user exit routine that is to be invoked by the TSO EDIT command to interpret the parameters specified in the subfield of the DSTYPE operand of the TSO EDIT command. The interpreted parameters will be passed as part of the parameter list to the processor that is specified in the CHECKER parameter. This parameter cannot be used with any of the standard data set types.

The name, which must be 1 to 8 characters long, is associated with data sets of the type specified in the corresponding name field of the DSTYPE parameter.

USERSRC=

specifies the type of input acceptable to the processor specified in the PRMPTR parameter in the TSO EDIT command.

DATASET

specifies that a sequential data set is the only allowable input to the specified processor.

INCORE

specifies that the TSO EDIT command may pass the input source to the specified processor in the form of an in-storage data set, if the source does not exceed 4096 bytes, or in the form of a sequential data set.

INLIST

specifies that the INLIST keyword construction is used in executing the specified data set type.

VAR=d–m

specifies, in bytes, the default and maximum allowable length of logical, variable-length records with variable-record format for new data sets created by the TSO EDIT command.

d is an integer from 5 to 255 that specifies the default logical record length for any variable-format data set of the type specified in the corresponding name field of the DSTYPE parameter. The value specified must be less than or equal to n - 4 of the corresponding n value specified in the BLOCK parameter.

m is an integer from 5 to 255 that specifies the maximum logical record length for any variable-format data set of the type specified in the corresponding name field of the DSTYPE parameter. The value specified must be less than or equal to n- 4 of the corresponding n value specified in the BLOCK parameter.

The hyphen must be coded as shown.

EXAMPLE: This example specifies that a new data set type, USER01, is defined for the TSO EDIT command. Its default record format is fixed and all other system defaults are used (see Figure 12). The standard data set type, DATA, will have a default block size of 2400 instead of 3120 (see Figure 12). All other standard data set types will use the default values that are specified in Figure 12.

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

Data			Default		Maximum					
Set	Block-	Record	LRECL		LRECL		Data	Checker	Prompter	Prompter
Type	Size	Format	F	V	F	V	Conversion	Name	Name	Input
Standard Data Set Types										
FORTH	400	FXDONLY	80	0	80	0	CAPSONLY	--------	--------	DATASET
ASM	3120	FXDONLY	80	0	80	0	CAPSONLY	--------	ASM*	DATASET
TEXT	3120	VAR	0	255	255	255	ASIS	--------	--------	DATASET
DATA	3120	FIXED	80	0	255	255	CAPS	-------	--------	DATASET
CLIST	3120	VAR	0	255	255	255	CAPSONLY	--------	--------	DATASET
CNTL	3120	FXDONLY	80	0	80	0	CAPSONLY	--------	--------	DATASET
COBOL	400	FXDONLY	80	0	80	0	CAPSONLY	--------	COBOL*	DATASET
PLI	400	VAR	0	104	100	104	CAPS	--------	PLIC	DATASET
GOFORT	3120	VAR	0	255	255	255	CAPS	--------	GOFORT*	INCORE
FORTGI	400	FXDONLY	80	0	80	---	CAPSONLY	--------	FORT*	DATASET
VSBASIC	3120	VAR	0	255	80	255	CAPSONLY	--------	VSBASIC	INLIST
PL1F	400	FXDONLY	80	0	100	0	CAPSONLY	PL1FSCAN	--------	DATASET
Any User Defined Data Set Types										
----	3120	FIXED	80	255	255	255	CAPS	-------	--------	--------

*IBM Program Product.

--Null Value

Figure 12. Attribute defaults for parameters omitted from the EDIT macro

EDTGEN (OS/VS2 MVS PROCESSOR SUPPORT 2 (5732-864))

Required for:	Eligible Device Table
Not Applicable for:	Complete I/O Device

The EDTGEN macro instruction is used to specify the building of new device allocation tables. The macro parameters specify the Stage I output data sets and the Stage II destination libraries that will contain the new device allocation tables. This macro must be specified last, but before the assembler END statement in the Stage I input deck.

The new tables may be put into a user-specified data set. The data set must be defined as an MLPA entry in the appropriate PARMLIB members. The PARMLIB members must be specified at IPL to activate the new tables. For information about creating IEALPAxx and other PARMLIB members needed in the creation of an MLPA, see OS/VS2 System Programming Library: Initialization and Tuning Guide.

The new tables may also be put into SYS1.LPALIB, in which case an IPL with CLPA will activate the tables. However, the use of SYS1.LPALIB is not recommended. The new tables will overlay the original device allocation tables and make recovery difficult.

[symbol]	EDTGEN	[DSN={indexed-dsn|SYS1.MLPALIB}] [DSNVOL={(volser,device-type)|(SYSRES,3330)}] [OBJDSN={SYS1.dsname|SYS1.OBJDEV}] [ID={user-id|EDTGEN}] [CHECK={YES|NO}] [DIAGNOS={YES|NO}] [JCLASS={class|A}] [OCLASS={class|A}]

DSN={indexed-dsn|SYS1.MLPALIB}
specifies the data set into which the new device allocation tables are linkedited in Stage II. The data set name must be indexed to conform with system generation services and usage. The total length of the name must not exceed 17 characters. The data set must be preallocated prior to Stage II.

DSNVOL={volser,device-type|SYSRES,3330}
specifies the volume serial number and the device type of the data set specified in DSN. The device type must be specified in the IODEVICE statement for the new tables.

OBJDSN={SYS1.dsname|SYS1.OBJDEV}
specifies the name of the data set that will contain the object modules of the new device allocation tables. The data set must be cataloged, and the total length of the name must not exceed 13 characters.

ID={user-id|EDTGEN}
specifies the table identification that is included as part of the table header. The user-id should be eight characters or less; a longer user-id is truncated to eight characters.

Note: For a complete system generation, the user-id is 'SYSGEN'; for an I/O device generation, it is 'IOGEN'.

CHECK=

YES
specifies that the job step to check the EDT against the UCBs in this system be included in the Stage II job.

NO
specifies that the job step not be included in the Stage II job.

DIAGNOS=

YES
specifies that the diagnostic override facility be established. (Refer to "Appendix E. Diagnostic Override" for information on diagnostic override.)

NO
specifies that the diagnostic override facility not be established.

JCLASS={class|A}
is a letter from A through O that specifies the job class to be used for the output from Stage I.

OCLASS={class|A}
is a letter from A through Z or a number from 0 through 9 that specifies the output class to be used for output from Stage II.

EXAMPLE: This example specifies an Eligible Device Table generation. The linkedit data set and object module data set are uniquely named; a unique user-id is specified; validity checking is not to be performed; diagnostic override is in effect; the job class and output class for Stage II are A (the default).

```
EDTGEN    DSN=EDT1.168A,DSNVOL=(EDTAB,3330),                        X
          OBJDSN=SYS1.EDTAB1,ID=EDT1,CHECK=NO
```

Required for:	Complete I/O Device
Not Applicable for:	Eligible Device Table

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 name of the temporary object module utility data sets required for Stage II assemblies. This macro must be specified last, but before the assembler END statement in the Stage I input deck.

Three object module partitioned data sets are required for Stage II. You specify a single name for them in the GENERATE macro to which a number, 01, 02, or 03 is appended during Stage II. For example, if you specify OBJPDS=SYS1.OBJPDS in the GENERATE macro then, during Stage II, the data sets will be named SYS1.OBJPDS01, SYS1.OBJPDS02, and SYS1.OBJPDS03. You must allocate space for the three utility data sets and catalog them in the catalog of the generating system before Stage II begins. Refer to "Stage II Input" in the chapter "System Control Program Installation" for this information.

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. If the generating system is not the starter system, the index of the system data sets cannot be SYS1.

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 serial number of the system residence volume of the generating system. If you specify a value for the INDEX parameter other than SYS1, each of the system data sets updated during an I/O device generation is renamed by the system during Stage II to this value. At the conclusion of the job that updates a system data set, the system renames the index for that system data set back to SYS1.

[symbol]	GENERATE	[**GENTYPE=**{(**IO**,n)|**ALL**}] [**INDEX=**{name|**SYS1**}] [**JCLASS=**{class|A}] [**OBJPDS**{SYS1.name|**SYS1.OBJPDS**}] [**OCLASS=**{class|A}] [**RESVOL=**([{volser|**SYSRES**}] [,{devtype|3330}])]

GENTYPE=
specifies the type of system generation.

IO,n
specifies an I/O device generation. The n is a number from 1 through 9 that specifies the number of the member of SYS1.NUCLEUS to be created or modified. The member name will be IEANUC0n.

ALL
specifies a complete system generation.

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 that was generated during the last complete generation.

INDEX={name|SYS1}
specifies the qualifier for the system data sets. The qualifier may be from 1 to 6 alphameric characters, the first character alphabetic. If a system other than the starter system is being used for a complete system generation, you must specify an index value other than SYS1; the qualifier of the starter data sets affected is changed to SYS1 during Stage II.

If any system data sets are predefined, their index value must be the same as the value specified in this parameter.

This parameter does not apply to the page data sets, the swap data sets, the duplex data set, or the master catalog.

JCLASS={class|A}
is a letter from A through O that specifies the job class to be used for the output from Stage I.

OBJPDS={SYS1.name|SYS1.OBJPDS}
specifies the general name of the three partitioned data sets to be used to store the object modules that are assembled during Stage II. During Stage II, the numbers 01, 02, and 03 are appended to the name you specify in this parameter. The three temporary data sets must have space allocated for them and be cataloged in the master catalog of the generating system before Stage II begins.

OCLASS={class|A}
is a letter from A through Z or a number from 0 through 9 that specifies the output class to be used for output from Stage II.

RESVOL=
specifies the volume serial number and device type of the system residence volume of the system to be generated.

{volser|SYSRES}
specifies the volume serial number of the system residence volume.

{devtype|3330}
specifies the unit address, device type, or group name of the system residence volume. Valid device types are 2305-1, 2305-2, 2314, 3330, 3330-1, 3340, 3350, 3375, or 3380.

If the volume serial number subparameter is omitted, the devtype subparameter must be preceded by a comma.

EXAMPLE: This example specifies a complete system generation. The system residence volume is a 3330 volume labeled SYSTEM. The index of the system data sets is SG2, which is renamed to SYS1 during Stage II. The job class and output class for Stage II is K. The default value, SYS1.OBJPDS, is used as the general name of the three utility data sets required for Stage II. The numbers 01, 02, and 03 are appended to this name by the system.

```
GEN   GENERATE   GENTYPE=ALL,RESVOL=(SYSTEM,3330),           X
               INDEX=SG2,JCLASS=K,OCLASS=K
```

EXAMPLE: This example specifies an I/O device generation. The new nucleus is to replace the old member (IEANUC01). The index of those system data sets that are modified during Stage II is renamed to SG1 during Stage II by the system and then renamed back to SYS1 by the system at the conclusion of the job that modifies the system data set. Default values are used for the job and output classes for Stage II, the name of the utility data sets used during Stage II, and for the system residence volume.

```
GEN   GENERATE   GENTYPE=(IO,1),INDEX=SG1
```

IODEVICE

Required for:	Complete I/O Device Eligible Device Table

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 your machine configuration must be specified in an IODEVICE macro instruction. This macro instruction is required.

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 automatically assigned 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 1917 I/O devices can be specified during system generation. However, the total number of UCBs for the I/O devices must not exceed what will fit in 60K-1 bytes of storage.

Figure 13 through Figure 22 show the valid combinations of values for the UNIT, MODEL, and FEATURE parameters.

Figure 23 shows the values that can be specified for I/O devices in the FEATURE parameter.

Figure 24 shows the valid combinations of values for telecommunications devices for the UNIT, ADAPTER, and TCU parameters.

Burst devices cannot be attached through multiplexer channels. Burst devices are: tape (2401, 2420, 3410, and 3420), direct-access (2305, 2314, 3330, 3330-1, 3340, 3350, 3375, 3380), and the 2250 display unit. The following burst devices cannot be attached through a high-speed multiplexer channel: 2305, 2314, 3330, 3330-1, 3340, 3350, 3375, and 3380. The following burst devices can be attached through a high-speed multiplexer channel on subselector channels C through F: 2401, 2420, 3410, 3420, and 2250.

An IODEVICE macro must be specified if a 2955 Remote Analysis Unit is to be attached to the system. An IODEVICE macro instruction must also be specified for the 2955 if you have a 3158 Processing Unit. (The 2955 is a standard feature on the 3158 Processing Unit.)

To specify the 168 Service Processor, you must also include an IODEVICE macro with UNIT=DUMMY in the UNIT field. Specify two consecutive even/odd addresses for the 2955 Remote Analysis Unit, and DEVTYPE=50000205. You must also specify a UNITNAME macro with both the NAME and UNIT fields specifying the address of the Service Processor. See the example at the end of the IODEVICE macro discussion.

During system generation, devices may be defined by the system as restricted. Jobs cannot allocate a restricted device using JCL or dynamic allocation (SVC99).

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.

MSS: For the Mass Storage System (MSS), IODEVICE macro instruction cards will be generated for each 3330V as part of the Mass Storage Control Table Create run. The IODEVICE cards should be inserted in the deck which is used as input to Stage I. For more information on the Mass Storage Control Table Create run, see *OS/VS Mass Storage Control Table Create*.

Telecommunications: For telecommunication devices there must be one IODEVICE macro instruction for each telecommunications line or 3704, 3705, or 3791L Communications Controller. The IODEVICE macro instruction applies to a telecommunication line or a 3704, 3705, or 3791L 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.

Note that if you intend to use the 3704 or 3705 Communication Controller with the emulation program, only one 3704 or 3705 device needs to be generated. Each line to be operated in emulation mode must be defined with an IODEVICE macro instruction.

In a VTAM telecommunications network, an IODEVICE macro instruction is required only for locally attached IBM 3270 Information Display Systems and locally attached IBM 3704 and 3705 Communications Controllers. All other devices are supported through the VTAM network definition process and are not operating system generation considerations. However, if a locally attached 3704 or 3705 Communications Controller, used with VTAM, will also support a nonVTAM network through a Network Control Program with the Partitioned Emulation Programming (PEP) Extension, the lines to be operated in emulation mode must be defined with IODEVICE macro instructions.

For locally attached IBM 3270 Information Display System control units (3274 Model 1A) which have Systems Network Architecture (SNA) support for the control of multiple devices through one device address, specify each 3270 control unit as UNIT=3791L. Where each device attached to a 3270 control unit has its own device address, specify only each attached device and omit the control unit.

If more than 16 devices are attached to the IBM 3272 Control Unit, the control unit address must be an even number.

[symbol]	**IODEVICE**	**ADDRESS=**{address| (address,number of units)} [**DEVTYPE=**type] [**ERRTAB=**nnn] [**OFFLINE=**{**YES**|NO}] **UNIT=**{device|**DUMMY**}

The validity of the following parameters depends on the type of unit specified. See Figure 13 through Figure 22 for valid parameters and subparameters.

		ADAPTER=adapter [**AP**={**YES**|**NO**}] [**EXPBFR**={number|4096}] [**FEATURE**=(feature[,feature]...)] **GCU**={**2848-1**|**2848-2**|**2848-21**|**2848-22**} [**MODEL**=model] [**NUMSECT**={number|16}] **OBRCNT**=number [**OPTCHAN**=number] **PCU**=n **SETADDR**=value **TCU**={**2701**|**2702**|**2703**}

ADAPTER=adapter

specifies either the terminal control or transmission adapter used to connect a telecommunications line to a transmission control unit or the type of channel adapter that connects a 3704 or 3705 communications controller to a System/370 channel. This parameter is required for all telecommunication devices. Figure 24 associates terminal control adapters with transmission control units.

ADDRESS=

specifies the unit address or addresses of a device or telecommunications line. For each address assembled, a unit control block (UCB) is created.

address

specifies the unit address for the device. You specify three hexadecimal digits, 000 through FFE. The high-order digit is the address of the channel (specified in a CHANNEL macro), the second digit specifies the control unit, and the low-order digit is a value from 0—F. For a high-speed multiplexer channel, the second digit must be C, D, E, or F.

For the IBM 2821 Control Unit with the two-channel switch feature, the low-order digit of the address must be as follows:

printer (first)	0 or 8
card reader	1 or 9
card punch	2 or A
printer (second)	3 or B
printer (third)	4 or C

If more than 16 devices are attached to the IBM 3272 Control Unit, the control unit address must be an even number.

For the IBM 3838, the low-order digit must be zero.

number of units

specifies the number of sequential addresses that are to be assembled for that device or telecommunications line. You specify a 1, 2, or 3-digit decimal value. For example, if ADDRESS=(130,5) is specified, unit addresses 130, 131, 132, 133, and 134 would be assembled.

If the number of units subparameter is omitted, a value of 1 is assumed for all devices except the 2314, which has a default value of 8, and 3330, 3330-1, 3340/3344, or 3350, which have a default value of 2. This subparameter is ignored for the 2305 (or the IBM 3838) because 8 units are automatically assigned.

The maximum value that can be specified for a 3330, 3330-1, 3340/3344, or 3350 is 32. The maximum value that can be specified for the 2250 is 8.

Notes:

1. If the 32 device address feature is installed, it is advisable to generate all possible addresses, or at least one more than the addresses present. Generate the extra addresses as dummies. For further information see "CSW Function—Bit 2" and "3333/3330/3340 Configurations" in Reference Manual for IBM 3830 Storage Control Model 2.

2. Whether you are preparing IODEVICE statements for a uniprocessor (UP) system or a tightly coupled multiprocessor (MP) system, device addresses should be generated for all paths available to the operating system being generated. These addresses should reflect the actual hardware device attachment address. A channel can only be attached to one processor of an MP system. Therefore, you should use the OPTCHAN parameter to generate alternate addresses (that is, addresses differing only by the channel digit (Cxx)) for channel addresses attached to the same processor side of an MP system. The operating system assumes device address symmetry in a tightly coupled MP system, that is, the address of a device is the same from both sides. However, asymmetric addresses are resolved by the IPL process, which flags the addresses of channels found to be unavailable.

   For alternate addresses of devices attached to different processor sides of an MP system, generate the addresses explicitly and specify the SHAREDUP parameter of the IODEVICE macro. The SHAREDUP parameter prevents unnecessary RESERVE/RELEASE processing and permits the shared device protocol (RESERVE/RELEASE) when an MP system is operating in the UP mode. SHAREDUP should only be used if an MP system may be run as two loosely coupled UP systems. Whenever the device is to be shared by another system that is loosely coupled, specify the SHARED parameter of the IODEVICE macro. For more information, see OS/VS2 MVS Multiprocessing: An Introduction and Guide to Writing Operating and Recovery Procedures.

AP=

YES

specifies that the device has an uninterruptible power supply.

NO

specifies that the device does not have an uninterruptible power supply.

This parameter is valid for the following devices: 2314/2319, 3330/3333 (Model 1 or 11), 3340/3344, 3350, 3375, 3380, 2401, 2415, 2420, 3410, and 3420.

If you specify this parameter, the WARN parameter of the CTRLPROG macro must also be specified.

DEVTYPE=type
specifies any additional characteristics of the device. The value specified must be 8 hexadecimal digits. This parameter must not be specified for any IBM-supported device. This parameter should be specified if a non-IBM device and UNIT=DUMMY are specified.

Note: This parameter should conform with the standard format of the UCBTYP field of the UCB. For further information about the UCB, see OS/VS2 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. After unpacking, this value becomes the suffix of the name IGE00, under which the error routine is stored in SYS1.LPALIB.

The ERRTAB parameter should be specified if UNIT=DUMMY is specified.

EXPBFR={number|4096}
is a decimal number from 1 to 8192 that specifies, in bytes, the amount of buffer space required by a 2250-3 to execute programs written for a 2250-1 that use EXPRESS attention handling routines. This parameter is valid for 2250-3 only.

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 13 through Figure 22 for the features that may be specified for the devices and to Figure 23 for a summary of features that may be specified.

p83

GCU={**2848-1**|**2848-2**|**2848-21**|**2848-22**}
specifies the type of graphic control unit (GCU) to which a 2260 Model 2 is attached. One of these graphic control units must be specified for each 2260 Model 2. This parameter is valid for 2260 Model 2 only.

For information on the 2260, refer to OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment).

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 13 through Figure 22).

NUMSECT={number|16}
specifies the number of 256-byte buffer sections in a 2840 display control unit to be assigned to a 2250 Model 3. These buffer sections can be used only by the device being specified.

The value of the number specified is from 1 to a maximum value that is determined by the following formula:

$$\frac{A}{256} - B + 1$$

where:

A is the size of the 2840 buffer.

B is the number of devices attached to the 2840.

The total number of buffer sections guaranteed to the devices attached to a 2840 must not exceed the number of sections in the buffer of that 2840.

If this parameter is not specified, the device uses the 2840 control unit buffer sections that are not assigned to other devices attached to the 2840. However, there must be at least one section available for assignment to each device attached to the 2840.

If a 2250 Model 3 is specified as an operator console and this parameter is not specified or its value is less than 16, a value of 16 is used. This parameter is valid for 2250 Model 3 only.

For information on the 2250, refer to OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit.

OBRCNT=number
is a number from 0 to 800 that specifies the number of area stations(BSC1, BSC2, BSC3 only) connected to the 2715 transmission control in a 2790 data communications system.

The total of all values specified in this parameter cannot exceed 800 during system generation. If 800 is exceeded, 800 is assumed.

OFFLINE={YES|NO}
specifies whether the device is to be considered online or offline at the time the system is initialized. If YES is specified, the device will be considered offline at system initialization. If NO is specified or if this operand is not coded, the device will be considered online at system initialization.

Note: If FEATURE=SHARABLE is specified for 3420 devices, OFFLINE=YES is forced if not already specified.

OPTCHAN=number
is a one- or two-digit hexadecimal number that indicates the alternate channel or subchannel, respectively, through which the device specified in this macro may be addressed. Only one alternate channel or subchannel may be specified for a device. OPTCHAN does not support byte multiplexer channels.

It is necessary to generate a DUMMY device for channels that are specified only through the OPTCHAN parameter, that is, the channel does not appear as the primary channel for any device.

For a device at any channel except a high-speed multiplexer channel, the one-digit number you specify indicates an alternate channel. The value specified must be greater than the high-order digit of the primary address specified for the device in the ADDRESS parameter. For example, if ADDRESS=59A,OPTCHAN=6 is specified, the alternate channel address 69A is generated in addition to the primary address. Valid devices for this parameter are those that connect to a selector or block multiplexer channel.

If a device is connected to a high-speed multiplexer channel, you would specify a two-digit number to indicate an alternate subchannel. The first digit must be the same as the high-order digit of the primary address given for the device in the ADDRESS parameter. The second digit must be the hexadecimal digit D, E, or F, that is greater than the second digit (C, D, or E) specified in the ADDRESS parameter. For example, if ADDRESS=(4C8,2),OPTCHAN=4D is specified for a device, alternate subchannel addresses 4D8 and 4D9 are generated. Valid devices are: 2401, 2420, 3420, 3704, and 3705.

A subchannel can be specified as an alternate only on the same high-speed multiplexer channel.

Except for devices connected to selector subchannels of a high speed multiplexer channel, only block multiplexer and selector channels can be alternate channels.

There is a maximum of 1023 optional channel paths for each configuration. Each value specified is the address of a channel that was specified in a CHANNEL macro.

There must be no more than one IODEVICE macro for a device, even if an alternate address is given to the device.

Note that devices sharing a control unit are assigned addresses within sets of contiguous numbers. The size of such a set is equal to the maximum number of devices that can share the control unit or 16, whichever is smaller. Control units designed to accommodate more than 16 devices may be assigned nonsequential sets of addresses, each set consisting of 16, or the number required to bring the total number of assigned addresses equal to the maximum number of devices attachable to the control unit, whichever is smaller. (This does not apply to the 3830 Model 2 or the Integrated Storage Control with the 32 drive expansion feature. In this case, the addresses must be 32 contiguous numbers.) The control unit does not respond to any address outside its assigned set or sets. If no control unit responds to an address, the I/O device appears not operational. If a control unit responds to an address for which no device is installed, the absent device appears in the not-ready state. If the operator varies such a device online, the system hangs when it tries to use the device and the job may have to be cancelled or the system reinitialized. See "Input/Output Device Addressing" in _IBM System/370 Principles of Operation_ for more information.

For the 3033, 3032, and 3031 Processors, channels are attached to the processor in groups. It is possible for a hardware error to occur that leaves a group of channels unusable. For increased availability of the 3033 Processor, it is recommended that the OPTCHAN be in a different group than the primary channel.

PCU=n

specifies the number assigned to the physical control unit (2840) to which this 2250 model 3 is attached. The n is a number from 1 to 99. Each physical 2840 must be uniquely identified by this parameter. A 2250 on any 2840 cannot have a unit address that is within the range of addresses of any 2250 on another 2840 or any other unit address. For example, 122 and 124 cannot be assigned to any 2250s if 121, 125, and 126 are addresses of 2250s on another 2840 or if 123 is the address of another device. This parameter is valid for 2250 model 3 only.

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 transmission control unit for a telecommunications line. This parameter is required for all telecommunications lines that are not serviced by the 3704 or 3705. Figure 24 associates terminal control adapters with transmission control units.

UNIT=device
specifies the device. Figure 13 through Figure 22 lists and defines the devices that may be specified. (Device types are described in "Appendix A. Device Types.")

In the case of telecommunications devices, the UNIT parameter specifies the device that is connected to a telecommunications line, a 3704 or 3705, or the type of binary synchronous 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 parameter in this section). To use a 2319, specify UNIT=2314. "Appendix D. Functionally Equivalent I/O Devices" contains a description of the 2319.

The IBM 3333 Disk Storage and Control, Model 1 is functionally equivalent to the IBM 3330 Disk Storage Drive, Model 1. To use a 3333, specify UNIT=3330. "Appendix D. Functionally Equivalent I/O Devices" contains additional information about specifying the 3333.

The IBM 3330 Disk Storage, Model 11 and the IBM 3333 Disk Storage and Control, Model 11 are also functionally equivalent. To specify either of these devices, specify UNIT=3330 and MODEL=11. "Appendix D. Functionally Equivalent I/O Devices" contains additional information about specifying these devices.

The IBM 3344 Disk Storage is functionally equivalent to the IBM 3340 Disk Storage. To use a 3344, specify UNIT=3340. See "Appendix D. Functionally Equivalent I/O Devices" for additional information about specifying the 3344.

DUMMY
specifies a device that is not supported by IBM. When DUMMY is specified, a 32-byte UCB with all its standard fields is generated. It is assumed that you provide your own I/O support routines for the device. For 5752-864 only, the I/O support routines for unit record devices are generated for all DUMMY devices.

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.

Since the UNITNAME macro maximum specification is 100, no more than 100 DUMMY devices may be specified.

EXAMPLE: This macro defines a 3210 console with a unit address of 009.

```
  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.

```
  P00D   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.)

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

EXAMPLE: This macro specifies a 3330 Model 1 at addresses 290 and 291. (The 3330 defaults to two addresses. For a 3330-1, you specify UNIT=3330, MODEL=1 in an IODEVICE macro.)

```
IODEV   IODEVICE   UNIT=3330,MODEL=1,ADDRESS=290
```

EXAMPLE: This macro defines a 168 Service Processor. The IODEVICE macro specifying UNIT=DUMMY must be included. You must specify two consecutive even/odd addresses for the 2955 Remote Analysis Unit and DEVTYPE=50000205. Also, the UNITNAME macro with both the NAME and UNIT fields specifying the Service Processor device address must be included.

```
SVP1   IODEVICE  UNIT=2955,ADDRESS=0DE
SVP2   IODEVICE  UNIT=DUMMY,ADDRESS=0DF,DEVTYPE=50000205

SVP3   UNITNAME  NAME=0DF,UNIT=0DF
```

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
2305	1 or 2	SHARED SHAREDUP		Fixed Head Disk Storage The 2305 Model 1 can only be specified in a CPU Model 165II or 168 system configuration. The SHARED, SHAREDUP, and OPTCHAN parameters are mutually exclusive. The SHARED and SHAREDUP features are valid only for the 2305 Model 2.
2314		ALTCTRL SHARED SHAREDUP	[AP={YES|NO}]	2314 or 2319 Direct-Access Storage Facility. The SHARED and SHAREDUP features are mutually exclusive.
3330	1 or 11	ALTCTRL SHARED SHAREDUP	[AP={YES|NO}]	3330 Disk Storage Drive, 3333 Disk Storage and Control, or 3330 compatibility mode. Specify MODEL=1 for 3330 Disk Storage Models 1 and 2 and for 3333 Disk Storage and Control Model 1. Specify MODEL=11 for 3330 Model 11 and for 3333 Model 11. The SHARED and SHAREDUP features are mutually exclusive. For information on 3330 compatibility mode, see 3350/3344 Installation and Conversion Guide.

Figure 13 (Part 1 of 2). IODEVICE macro parameter values—Direct-Access Devices

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3330V		ALTCTRL SHARED SHAREDUP		Virtual volume for MSS The 3330V is functionally equivalent to the 3330/3333 Model 1. If you specify 3330V, you must also specify the 3851. The SHARED and SHAREDUP features are mutually exclusive.
3340		ALTCTRL RPS SHARED SHAREDUP	[AP={YES|NO}]	3340 Disk Storage Drive or 3340 Compatibility Mode The SHARED and SHAREDUP features are mutually exclusive. For 3340 compatibility mode, RPS must be specified. See "Appendix D. Functionally Equivalent I/O Devices" for information on 3340 compatibility mode.
3350		ALTCTRL SHARED SHAREDUP	[AP={YES|NO}]	Disk Storage Drive The SHARED and SHAREDUP features are mutually exclusive.
3375		ALTCTRL SHARED SHAREDUP	[AP={YES|NO}]	Disk Storage Drive The SHARED and SHAREDUP features are mutually exclusive.
3380		ALTCTRL SHARED SHAREDUP	[AP={YES|NO}]	Disk Storage Drive The SHARED and SHAREDUP features are mutually exclusive.

Figure 13 (Part 2 of 2). IODEVICE macro parameter values—Direct-Access Devices

Note to Figure 13:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Paramaters	Notes
1053	4			Printer
2250	1	ABSLTVEC ALKYB2250 {BUFFER4K| BUFFER8K} CHARGNTR DESIGNFEAT LIGHTPEN PTGMKYBD		Display unit Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console.
	3	ALKYB2250 PRGMKYBD	[NUMSECT=n] [EXPBFR=n] PCU=n	Display unit The parameter PCU is required for the 2250 Model 3. Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console.
2260	1 or 2	{ALKYB2260| DEKYB2260} LINEADDR NODESCUR NMKYB2260	GCU=control unit	Display station The GCU parameter is required for the 2260 Model 2. Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console.

2260 or 2265			ADAPTER=IBM3
			TCU=2701
				Display station
				When the unit is specified as a remote device, the MODEL and FEATURE parameters are not specified.

3036				Display Unit The AUDALRM and DOCHAR features are assumed and should not be specified. Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console.
3158				Display Unit The AUDALRM and DOCHAR features are assumed and should not be specified. Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console.

Figure 14 (Part 1 of 4). IODEVICE macro parameter values—Display Devices

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3277	1 or 2	{ASCACHAR| ASCBCHAR| KACHAR| DOCHAR| FRCHAR| GRCHAR| UKCHAR} AUDALRM MAGCDRD NUMLOCK PTREAD SELPEN {ASKY3277| DEKY3277| EBKY3277| OCKY3277} {KB70KEY| KB78KEY| KB81KEY}		Display unit for the 3270 Display System Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console. Only one type of character generator may be specified. If none is specified, DOCHAR is used as the default. Only one type of keyboard may be specified. If none is specified, a 66-key keyboard is assumed. The AUDALRM feature can only be specified if a keyboard type has been specified. KACHAR is valid only if EBKY3277 and KB81KEY or DEKY3277 and KB70KEY are specified. KB70KEY is valid only if DEKY3277 and KACHAR are specified. KB78KEY is valid only if ASKY3277, EBKY3277, or OCKY3277 is specified. KB81KEY is valid only if EBKY3277 and KACHAR are specified.

Figure 14 (Part 2 of 4). IODEVICE macro parameter values—Display Devices

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3278	1,2,2A, 3,4	{ASCACHAR| ASCBCHAR| KACHAR| DOCHAR| FRCHAR| GRCHAR| UKCHAR} AUDALRM MAGCDRD NUMLOCK PTREAD SELPEN {ASKY3277| DEKY3277| EBKY3277} {KB70KEY| KB78KEY| KB81KEY}		Display unit for the 3270 Display System Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console. Only one type of character generator may be specified. If none is specified, DOCHAR is used as the default. Only one type of keyboard may be specified. If none is specified, a 77-key keyboard is assumed. The AUDALRM feature can only be specified if a key board has been specified. KACHAR is valid only if EBKY3277 and KB81KEY or DEKY3277 and KB70KEY are specified. KB70KEY is valid only if DEKY3277 and KACHAR are specified. KB78KEY is valid only if ASKY3277 or EBKY3277 is specified. KB81KEY is valid only if EBKY3277 and KACHAR are specified.

Figure 14 (Part 3 of 4). IODEVICE macro parameter values—Display Devices

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3279	2A,2B, 3A,3B	{ASCACHAR| ASCBCHAR| KACHAR| DOCHAR| FRCHAR| GRCHAR| UKCHAR} AUDALRM MAGCDRD NUMLOCK PTREAD SELPEN {ASKY3277| DEKY3277| EBKY3277} {KB70KEY| KB78KEY| KB81KEY}		Display unit for the 3270 Display System Refer to the "Unit Record Devices" part of this figure for specifying this device as an operator console. Only one type of character generator may be specified. If none is specified, DOCHAR is used as the default. Only one type of keyboard may be specified. If none is specified, a 77-key keyboard is assumed. The AUDALRM feature can only be specified if a key board has been specified. KACHAR is valid only if EBKY3277 and KB81KEY or DEKY3277 and KB70KEY are specified. KB70KEY is valid only if DEKY3277 and KACHAR are specified. KB78KEY is valid only if ASKY3277 or EBKY3277 is specified. KB81KEY is valid only if EBKY3277 and KACHAR are specified.
3284	1 or 2	{DOCHAR| FRCHAR| GRCHAR| KACHAR| UKCHAR} PTREAD		Printer for the 3277 Only one type of character generator may be specified. If none is specified, DOCHAR is used as the default.
3286	1 or 2	{DOCHAR| FRCHAR| GRCHAR| KACHAR| UKCHAR} PTREAD		Printer for the 3277 Only one type of character generator may be specified. If none is specified, DOCHAR is used as the default.

Figure 14 (Part 4 of 4). IODEVICE macro parameter values—Display Devices

Note to Figure 14:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
2401	1,2,3	ALTCTRL READWRITE MDECOMPAT {7-TRACK| 9-TRACK} DATACONV	[AP={YES|NO}]	2401, 2402, or 2403 Magnetic Tape Unit FEATURE=9-TRACK is assumed if the FEATURE parameter is omitted.
	4,5,6	ALTCTRL READWRITE 9-TRACK DUALDENS		
2401	8	ALTCTRL 7-TRACK READWRITE	[AP={YES|NO}]	2401, 2402, or 2403 Magnetic Tape Unit The DATACONV feature is standard for the 2401-8. READWRITE is required and valid only if the 2401-8 is attached to a 2804-3.
2420		ALTCTRL	[AP={YES|NO}]	Magnetic tape unit This is a 9-track 1600 BPI drive only, so the FEATURE parameter is not required.
2495				Magnetic tape cartridge reader
3410	1,2,3	ALTCTRL DUALDENS {7-TRACK| 9-TRACK}	[AP={YES|NO}]	Magnetic tape drive If DUALDENS is not specified, the 3410 will be utilized as a 1600 BPI tape drive.
3420	3,5,7	ALTCTRL DUALDENS SHARABLE {7-TRACK| 9-TRACK}	[AP={YES|NO}]	Magnetic tape drive The default is 1600 BPI if FEATURE=DUALDENS is not specified. DUALDENS=1600 or 800 BPI for Models 3, 5, and 7.
3420	4,6,8	ALTCTRL 9-TRACK OPT1600 SHARABLE	[AP={YES|NO}]	Magnetic tape drive The default is 6250 BPI if FEATURE=OPT1600 is not specified. OPT1600=6250 or 1600 BPI for Models 4, 6, and 8.

Figure 15. IODEVICE macro parameter values—Magnetic Tape Units

Note to Figure 15:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
1275 or 1287 or 1288				The address specified must be the address of the primary control unit and must be an even number. There must be only one IODEVICE macro for each 1275. Two addresses are generated by the one IODEVICE macro. The 1275 is available through IBM World Trade Corporation branch offices.
3886				Optical character reader

Figure 16. IODEVICE macro parameter values—Optical Character Readers

Note to Figure 16:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3895				Document Reader/Inscriber

Figure 17. IODEVICE macro parameter values—Reader/Inscribers

Note to Figure 17:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
1419				The address specified must be the address of the primary control unit and must be an even number. There must be only one IODEVICE macro for each 1419. Two addresses are generated by the one IODEVICE macro.
3890				Document Processor

Figure 18. IODEVICE macro parameter values—Magnetic Ink Character Readers

Note to Figure 18:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
1052	7			Printer-keyboard Installed in combination with the 2150 Control Unit.
1403	N1 or 2 7	UNVCHSET		Printer UNVCHSET is invalid for the 1403-7.
1443	N1	SELCHSET 24ADDPOS		Printer
2250	1	ALKYB2250 {BUFFER4K| BUFFER8K} CHARGNTR LIGHTPEN PRGMKYBD		Display unit LIGHTPEN and PRGMKYBD are optional; other features are required if the 2250 Model 1 is used as a console device.
2250	3	ALKYB2250 PRGMKYBD	[NUMSECT=n] [EXPBFR=n] PCU=n	Display unit If the 2250 Model 3 is being used as a console device: • ALKYB2250 is required. • If NUMSECT is omitted or less than 16, 16 is used. • The PCU parameter is required.
2260	1	LINEADDR ALKYB2260 NODESCUR		Display Station If the 2260 Model 1 is being used as a console device, LINEADDR and ALKYB2260 are required.
2501	B1 or B2	CARDIMAGE		Card reader
2520	B1 B2 or B3	CARDIMAGE		Card Reader punch Card punch only. Specify the 2520 Models B2 or B3 as 2520 Model B1.
2540R or 2540P	1	CARDIMAGE		Card reader punch The 2540R and 2540P are specified for the same 2540 Card Reader Punch. Two IODEVICE macro instructions must be specified.

Figure 19 (Part 1 of 4). IODEVICE macro parameter values—Unit Record Devices[2]

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
2671	1			Paper tape reader
3036				Display console for 3031, 3032, 3033 processors
3066				The system console for the System/370 Models 165II and 168.
3158				Display console with 66-key operator console keyboard
3203	4			Printer
3210 or 3215				Console-printer keyboard
3211				High-speed printer The universal character set is a standard feature.
3213				Console-printer with no keyboard - System/370 Model 155II.
3277	2	{ASKY3277| DEKY3277| EBKY3277| OCKY3277} SELPEN AUDALRM KB78KEY		Display station If a keyboard is not specified for the 3277 Model 2 for use as a console device, the 66-key keyboard is assumed. KB78KEY is optional and valid only if an ASKY3277, EBKY3277, or OCKY3277 keyboard is specified. The 3277 Model 1 can be used only as an output-only console to display operator messages; optional features or parameters are ignored. The AUDALRM feature can only be specified if a keyboard type has been specified. The DOCHAR feature is used as the default type of character generator. The 3277 is supported only with local attachment and not as a remote terminal device.

Figure 19 (Part 2 of 4). IODEVICE macro parameter values—Unit Record Devices[2]

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3278	2,2A,3,4	{ASKY3277| DEKY3277| EBKY3277} SELPEN AUDALRM KB78KEY		Display station If a keyboard is not specified for the 3278 Model 2, 2A, 3, or 4, for use as a console device the 77-key keyboard is assumed. KB78KEY is optional and valid only if an ASKY3277 or EBKY3277 keyboard is specified. The 3278 Model 1 can be used only as an output-only console to display operator messages: optional features or parameters are ignored. The AUDALRM feature can only be specified if a keyboard type has been specified. The DOCHAR feature is used as the default type of character generator. The 3278 is supported only with local attachment and not as a remote terminal device.
3279	2A,2B, 3A,3B	{ASKY3277| DEKY3277| EBKY3277} SELPEN AUDALRM KB78KEY		Display station If a keyboard is not specified for the 3279 Model 2A, 2B, 3A, or 3B for use as a console device, the 77-key keyboard is assumed. KB78KEY is optional and valid only if an ASKY3277 or EBKY3277 keyboard is specified. The AUDALRM feature can only be specified if a keyboard type has been specified. The DOCHAR feature is used as the default type of character generator. The 3279 is supported only with local attachment and not as a remote terminal device.
3505		CARDIMAGE		Card reader and control unit

Figure 19 (Part 3 of 4). IODEVICE macro parameter values—Unit Record Devices[2]

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3525		CARDIMAGE {TWOLINE| MULTILINE}		Card punch The 3525 attaches to a system through the control unit of a 3505, or attaches directly through an integrated attachment.
3540				Diskette I/O Unit You must generate two addresses if you are using the 3540 Model B2. You do not have to specify contiguous addresses, but the addresses must refer to the same control unit.
3800		BURSTER {CGS1| CGS2}		Printing Subsystem.
3838				Array Processor
3848				Cryptographic Unit
7443				Service Record File (SRF)

Figure 19 (Part 4 of 4). IODEVICE macro parameter values—Unit Record Devices[2]

Note to Figure 19:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

[2] Figure 8 lists all of the devices that can be used as consoles.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
3851				Mass Storage Control This device must be specified for MSS. The first time the 3851 and 3330V are introduced to your system you must do a complete system generation.

Figure 20. IODEVICE macro parameter values—Control Units

Note to Figure 20:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
CTC		370		Channel-to-Channel Adapter This feature must be specified in a JES3 multiprocessing configuration when the CTC adapter is used. For example, a global processor is connected to a local processor through the CTC adapter. For more information on how the CTC adapter is used, see Introduction to JES3.

Figure 21. IODEVICE macro parameter values—Special Features

Note to Figure 21:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
1030		AUTOPOLL	SETADDR=value	Data collection system
1050		AUTOANSR AUTOCALL AUTOPOLL		Data communication system AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified.
1050X		AUTOCALL AUTOANSR		Refers to the 1050 with the time-out suppression feature.
1060		AUTOPOLL		Data communication system

Figure 22 (Part 1 of 3). IODEVICE macro parameter values—Telecommunications[2]

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
2740		AUTOANSR AUTOCALL AUTOPOLL CHECKING INTERRUPT OIU SCONTROL XCONTROL	SETADDR=value	Communications terminal AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified. If the 2740 is specified as a console, then it must have the record checking feature, and must not have the station control feature. The communications line must be nonswitchable. 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 #S30031 is installed, FEATURE= INTERRUPT may be specified.
2740C		AUTOANSR CHECKING		Communications terminal with correspondence code. CHECKING is required.
2740X		AUTOANSR AUTOCALL CHECKING		Communications terminal with PTTC code. CHECKING is required.
2741C or 2741P		AUTOANSR		Communications terminal with correspondence code or communications terminal with PTTC code.
115A				Western Union[3] Terminal
83B3				AT&T[4] Selective Calling Terminal
TWX		AUTOANSR AUTOCALL		Teletype[5] Models 33 or 35
WTTA				IBM World Trade Corporation Telegraph Terminal
BSC1		DUALCODE[7] DUALCOMM	OBRCNT=n	BSC[6] station nonswitched point-to-point line.
BSC2		DUALCODE[7] DUALCOMM AUTOANSR AUTOCALL		BSC[6] station switched point-to-point line. AUTOANSR and AUTOCALL is assumed if BSC2 is specified.
BSC3		DUALCODE[7] DUALCOMM AUTOPOLL	OBRCNT=n	BSC[6] station nonswitched multipoint line. AUTOPOLL is assumed.

Figure 22 (Part 2 of 3). IODEVICE macro parameter values—Telecommunications[2]

Unit	Model	Feature[1] (optional)	Optional and/or Required Parameters	Notes
7770	3			Audio response unit
2955				Remote analysis unit If this device is to be attached, an IODEVICE macro instruction must be specified.
3791L				Controller This must be specified if the 3791 is used as a local channel attachment.
3704 or 3705			ADAPTER= {CA1|CA2}	Communications control unit The only channel adapter that can be specified for the 3704 is CA1. Channel adapter type 3 is specified as a type 2 (CA2). Channel adapter type 4 is specified as a type 1 (CA1).

Figure 22 (Part 3 of 3). IODEVICE macro parameter values—Telecommunications[2]

Notes to Figure 22:

[1] Figure 23 lists and defines all of the options that can be specified through the FEATURE parameter.

[2] Figure 24 lists the terminal control or transmission adapters (ADAPTER) used to connect a telecommunications I/O device (UNIT) to a transmission control unit (TCU).

[3] Trademark of Western Union Telegraph Company.

[4] Trademark of American Telephone and Telegraph Company.

[5] Trademark of the Teletype Corporation.

[6] BSC (Binary Synchronous Communications) stations can be any of the following:

System/3 Processor Station
System/360 Processor Station (including Model 25 Integrated Communications Adapter)
System/370 Processor Station (including Model 125 ICA and Model 135 ICA)
System/360 Model 20 Processor Station
1130 Computing System Processor
1800 Data Acquisition and Control System Processor (BTAM support only)
2770 Data Communications System
2780 Data Transmission Terminal
2790 Data Communications System/2715 Transmission Control Unit Model 2
2972 General Banking System Models 8 and 11 (BTAM support only)
3275 Display Station (BSC2 or BSC3 only)
3277 Display Station (BSC3 only)
3284 Printer Controller (BSC3 only)
3286 Printer Controller (BSC3 only)
3670 Brokerage Communication System (TCAM support only)

3735 Programmable Buffered Terminal (BSC2 and BSC3 only)
3780 Data Transmission Terminal
6670 Information Director, supported as a 2770 (BSC1 and BSC2 only)

[7] 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.

Feature	Device	Description
ABSLTVEC	2250 (Model 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.
ALTCTRL	2314/2319 2401,2420 3330/3333 3330V 3340/3344 3350 3375 3380 3410 3420	These devices can be accessed through an alternate control unit. ALTCTRL should only be coded when there is a separate physical control unit path to the device; that is, a separate physical path such that an alternate path may be tried when a control unit busy condition exists for the first path. If there is an alternate channel path, but no separate control unit path, use the OPTCHAN parameter to specify the alternate channel path. Two control units may be attached to the same device on the same channel; however, this configuration requires one of the addresses be offline. The ALTCTRL feature and the OPTCHAN parameter may be used together when separate control units on different channels are required.
ASCACHAR	3277 3278 3279	ASCII A character generator.
ASCBCHAR	3277 3278 3279	ASCII B character generator.
ASKY3277	3277 3278 3279	ASCII typewriter keyboard.
AUDALRM	3277 3278 3279	Audible alarm feature.
AUTOANSR	BSC2 TWX 1050 1050X 2740 2740C 2740X 2471C 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.

Figure 23 (Part 1 of 6). IODEVICE macro FEATURE parameter values

Feature	Device	Description
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 (Model 1)	This feature provides the display unit with either 4096 bytes or 8192 bytes of virtual storage for display regeneration.
BURSTER	3800	Specifies that the Burster-Trimmer-Stacker feature is installed on the 3800.
CARDIMAGE	2501 2520 2540P (Model 1) 2540R (Model 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.
CGS1	3800	This feature specifies one character generation storage which contains two writable character generation modules (WCGMs) sufficient for 128 characters.
CGS2	3800	This feature specifies two character generation storages which contain a total of four writable character generation modules (WCGMs) sufficient for 255 characters.
CHARGNTR	2250 (Model 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 Communications Terminal is equipped with the record checking feature.
DATACONV	2401	The data conversion feature allows the writing and reading of binary data on 7-track 2400 tape units.
DEKYB2260	2260	This feature specifies an alphameric keyboard with numeric inset for the 2260. The numeric keys are inset in the keyboard in a block arrangement for rapid numeric data entry.
DEKY3277	3277 3278 3279	EBCDIC data entry keyboard.
DESIGNFEAT	2250 (Model 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.

Figure 23 (Part 2 of 6). IODEVICE macro FEATURE parameter values

Feature	Device	Description
DOCHAR	3277 3278 3279 3284 3286	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 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	3277 3278 3279	EBCDIC typewriter keyboard.
FRCHAR	3277 3278 3279 3284 3286	French character generator.
GRCHAR	3277 3278 3279 3284 3286	German character generator.
INTERRUPT	2740	This feature indicates that the 2740 is a Model 1 with RPQ #S30031 added to allow improved performance when the 2740 is being used as a console.
KACHAR	3277 3278 3279 3284 3286	Katakana character generator.
KB70KEY	3277 3278 3279	70-key keyboard (3277) or 76-key keyboard (3278 and 3279)
KB78KEY	3277 3278 3279	78-key keyboard (3277) or 87-key keyboard (3278 and 3279). The feature can only be specified if ASKY3277, EBKY3277, is specified.
KB81KEY	3277 3278 3279	81-key keyboard (3277) or 88-key keyboard (3278 and 3279)
LIGHTPEN	2250 (Model 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.

Figure 23 (Part 3 of 6). IODEVICE macro FEATURE parameter values

Feature	Device	Description
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 3278 3279	This feature specifies a magnetic card reader adapter.
MDECOMPAT	2401	The mode compatibility feature enables NRZI tape units (Models 1, 2, and 3) to operate with phase-encoding (PE) tape controllers (2803-2 or 2804-2).
MULTILINE	3525	This feature allows the 3525 Card Punch with the print feature to print up to 25 lines on a card.
NMKYB2260	2260	The numeric keyboard feature specifies that the keyboard is organized like a 10-key adding machine.
NODESCUR	2260	The nondestructive cursor is a feature on the 2848 control unit for the 2260. The feature allows the operator to move the cursor anywhere on the display without changing displayed information.
NUMLOCK	3277 3278 3279	Numeric lock feature.
OCKY3277	3277	This feature specifies a 78-key operator console keyboard.
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.
OPT1600	3420	This feature specifies that the 3420 will be utilized as a 6250 or 1600 BPI tape drive.
PRGMKYBD	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.
PTREAD	3277 3278 3279 3284 3286	This feature indicates the device is attached to a 3274-1D control unit.
READWRITE	2401	This 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.

Figure 23 (Part 4 of 6). IODEVICE macro FEATURE parameter values

Feature	Device	Description
RPS	3340/3344	Rotational position sensing. This feature must be specified for 3340 compatibility mode.
SCONTROL	2740	The 2740 Communications 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	3277 3278 3279	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. It is the user's responsibility to partition shared tape drives between processors. Care must be taken to insure that a shared tape drive is not allocated and/or unloaded. When the SHARABLE feature is used, OFFLINE=YES is forced if it is not specified.
SHARED	2305 (Model 2) 2314/2319 3330/3333 3330V 3340/3344 3350 3375 3380	This feature allows the system to share direct-access storage devices with other systems and causes the hardware device reserve/release logic to be used when required. This feature and SHAREDUP are mutually exclusive. For the 2305, the SHARED, SHAREDUP, and OPTCHAN parameters are mutually exclusive.
SHAREDUP	2305 (Model 2) 2314/2319 3330/3333 3330V 3340/3344 3350 3375 3380	This feature indicates that the hardware device reserve/release logic is to be used only when the system is operating in uniprocessing mode. The use of this feature will eliminate the overhead of the device reserve/release logic when a device is attached only to both processors of a tightly coupled multiprocessing system. The SHAREDUP feature must not be specified if the device is attached to a processor other than the tightly coupled multiprocessing system. This feature will allow for the sharing of these devices when the system is reconfigured for uniprocessing mode. This feature and SHARED are mutually exclusive. For the 2305, the SHARED, SHAREDUP, and OPTCHAN parameters are mutually exclusive.
TWOLINE	3525	This feature allows the 3525 card punch with the print feature to print 1 or 2 lines on a card.

Figure 23 (Part 5 of 6). IODEVICE macro FEATURE parameter values

Feature	Device	Description
UKCHAR	3277 3278 3279 3284 3286	United Kingdom character generator.
UNVCHSET	1403 (Model 2) 1403 (Model 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.
XCONTROL	2740	The 2740 Communications 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.
370	CTC	System/370.
7-TRACK or 9-TRACK	2401 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 730. 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 23 (Part 6 of 6). IODEVICE macro FEATURE parameter values

Adapter	Device	TCU	Description
BSCA	BSC1 BSC2 BSC3	2701	Synchronous data adapter, type II, and an appropriate modem.
		2703	Synchronous terminal control and an appropriate modum.
IBM1	1050 1050X 1060 2740 2740C 2740X 2741C 2741P	1701	IBM terminal adapter, type I, and an appropriate modem or an IBM line adapter.
		2702 2703	IBM terminal control, type I, and either a data set line adapter and an appropriate modem or an IBM line adapter.
IBM2	1030	2701	IBM terminal adapter, type II, and either an appropriate modem or an IBM line adapter.
		2702 2703	IBM terminal control, type II, and either a data set line adapter and an appropriate modem or an IBM line adapter.
IBM3	2260 2265	2701	IBM terminal adapter, type III, and an appropriate modem.
IBMT	1050 1050X	2701	IBM telegraph adapter.
		2703	IBM terminal control, type I, and a telegraph line adapter.
TELE1	115A 83B3	2701	Telegraph adapter, type I.
		2702 2703	Telegraph terminal control, type I, and a telegraph line adapter.
TELE2	TWX	2701	Telegraph adapter, type II, and an appropriate modem.
		2702 2703	Telegraph terminal control, type II, and an appropriate modem.
TELEW	WTTA	2701	IBM World Trade Corporation telegraph adapter.
		2702 2703	IBM World Trade Corporation adapter and a telegraph line adapter.

Figure 24. Terminal control or transmission adapters (ADAPTER)

Optional for: Complete

Not Applicable for: I/O Device
Eligible Device Table

The JES macro instruction is used to specify the configuration for the JES3 subsystem. The JES macro supplies information to generate a basic JES3 initialization deck which is added as the JES3INxx member of SYS1.PARMLIB. In addition, the JES macro causes JCL to be created which, during Stage II, will be used to rename the JES3 member in SYS1.PROCLIB to the name specified in the PROCID parameter and to copy and/or link edit the appropriate object modules into SYS1.LPALIB and SYS1.JES3LIB. This macro is valid for JES3 only.

[symbol]	**JES**	[CNS=(({01F|address},{3215|device type})[,(...)]])][1] [D2314=({address|(address,n)}[,...])] [D3330=({address|(address,n)}[,...])] [LOCLJES={NO|YES}] [PARMID={00|xx}] [PROCID={JES3|procname}] [PRT=(({00E|address},{1403|device type},{PN|image})[,(...)]])][1] [PUN=({00D|address},...)][1] [RDR=({00C|address},...)][1] [T2400=({address|(address,n)}[,...])] [T3400({address|(address,n)}[,...])]

[1] JES3 device names are assigned to unit record devices; for example, RDR=(00C,01C) would cause the names RD00C and RD01C to be assigned to 00C and 01C respectively.

CNS=

specifies the console devices to be managed by JES3. JES3 will assign names to each console device for the initialization deck.

({01F|address},{3215|device type})

If the CNS parameter is not specified, 01F is the default address and 3215 is the default type. If you code the CNS parameter, you must provide an address and type for each device specified. The maximum number of console devices is 23. This is because of the Assembler limitation of 255 characters per operand.

D2314={address|(address,n)}

specifies the 2314 or 2319 DASD devices to be managed by JES3. A string of devices can be specified by (address,n) where n is the number of sequential addresses to be assigned. For example, if D2314=((130,7)) is specified, the 2314 or 2319 DASD devices at addresses 130-136 will be managed by JES3. If D2314=((130,20)) is specified, the 2314 or 2319 DASD devices at addresses 130-13F and 140-143 will be managed by JES3.

n can be a value from 1 to 63 and can have one or two digits (for example, 5 or 05).

D3330={address|(address,n)}

specifies the 3330 or 3333 DASD devices to be managed by JES3. A string of devices can be specified by (address,n) where n is the number of sequential addresses to be assigned. For example, if D3330=((150,5)) is specified, the 3330 or 3333 DASD devices at addresses 150-154 will be managed by JES3. If D3330=((150,20)) is specified, the 3330 or 3333 DASD devices at addresses 150-15F and 160-163 will be managed by JES3.

n can be a value from 1 to 63 and can have one or two digits (for example, 5 or 05).

LOCLJES=

specifies whether or not the system being generated will be used only as a JES3 local processor.

NO

specifies that a JES3 initialization deck will be added to SYS1.PARMLIB, the JES3 object modules will be copied and/or link edited from the distribution library to SYS1.JES3LIB and SYS1.LPALIB, and the JES3 procedure will be copied from the distribution library to SYS1.PROCLIB.

YES

specifies that a JES3 initialization deck will not be created; however, the JES3 object modules will still be copied and/or link edited from the distribution library to SYS1.JES3LIB and SYS1.LPALIB, and the JES3 procedure will be copied from the distribution library to SYS1.PROCLIB.

Note: If you are running in a multiprocessing environment and are using a shared SYS1.JES3LIB, you must alter the Stage II deck so that the object modules copied from the distribution library will be copied only once.

PARMID={00|xx}

specifies the suffix to the SYS1.PARMLIB member name into which the JES3 initialization deck will be added. The value of xx must be 2 alphameric characters.

PROCID={JES3|procname}

specifies the name of the JES3 procedure in SYS1.PROCLIB. This name must correspond with the name specified in the PRISUB parameter of the SCHEDULR macro and must not exceed four alphameric characters.

PRT=

specifies the printers to be managed by JES3. JES3 will assign names to each printer for the initialization deck.

{00E|address}

specifies the address of the printer.

{1403|device type}

specifies the type of printer.

{PN|image}

specifies the character-set image which will be used for the printer. See Figure 9 for the values that can be specified.

If the PRT parameter is not specified, 00E will be the default address, 1403 will be the default type, and PN will be the default character-set image. If you specify the PRT parameter, you must specify an address, type, and image for the printer.

PUN=

specifies the punches to be managed by JES3. JES3 will assign names to each punch for the initialization deck.

{00D|address}
If the PUN parameter is not specified, 00D is the default address.

RDR=

specifies the readers to be managed by JES3. JES3 will assign names to each reader for the initialization deck.

{00C|address}
If the RDR parameter is not specified, 00C is the default address.

T2400={address|(address,n)}
specifies the 2400 tape devices to be managed by JES3. A string of devices can be specified by (address,n) where n is the number of sequential addresses to be assigned. For example, if T2400=((182,3)) is specified, the 2400 tape devices at addresses 182-184 will be managed by JES3. If T2400=((182,20)) is specified, the 2400 tape devices at addresses 182-18F and 190-195 will be managed by JES3.

n can be a value from 1 to 63 and can have one or two digits (for example, 5 or 05).

T3400={address|(address,n)}
specifies the 3400 tape devices to be managed by JES3. A string of devices can be specified by (address,n) where n is the number of sequential addresses to be assigned. For example, if T3400=((170,2)) is specified, the 3400 tape devices at addresses 170-171 will be managed by JES3. If T3400=((170,20)) is specified, the 3400 tape devices at addresses 170-17F and 180-183 will be managed by JES3.

n can be a value from 1 to 63 and can have one or two digits (for example, 5 or 05).

Optional for: Complete
I/O Device

Not Applicable for: Eligible Device Table

The SCHEDULR macro instruction specifies the job scheduler and master scheduler options. This macro is optional. If it is not specified in a complete generation or if it is not specified in an I/O device generation, the default values are used. Note that if the HARDCOPY parameter is modified in an I/O device generation, the change is not reflected in PARMLIB member IEASYS00, and any user modifications to this member are preserved. If you desire such changes to be reflected in PARMLIB, the appropriate members should be updated directly.

[symbol]	**SCHEDULR**	[BCLMT={number|100}] [DEVPREF=(generic name [,generic name]...)] [HARDCPY=({address|SYSLOG} {,(routing code[,routing code]...)|,ALL}{,CMDS|,INCMDS|,NOCMDS|,STCMDS})] [PRISUB={name|JES2|JES3}] [SUBSYS={(name[,name]...)|PRISUB}] [TAVR={200|556|800}]

BCLMT={number|100}
is an integer from 1 to 1000 that specifies the number of 130-byte records that will be set aside for broadcast messages in the SYS1.BRODCAST system data set. If this parameter is specified, the SYS1.BRODCAST system data set should be specified in a DATASET macro.

DEVPREF=generic name
is a list of up to thirty device types that is used to create an installation device preference table. The extended device allocation function uses the order in which the devices are specified as a preference when allocating a device within a job step or when an esoteric group name (such as DISK) is specified for a device. If this parameter is not specified, an installation device preference table is created for the devices in your system in the order of device performance. Refer to "Appendix C. Installation Device Preference Table" for the order in which the devices are allocated, according to device performance, if this parameter is not specified. When you specify this parameter, the devices listed in "Appendix C. Installation Device Preference Table" are appended to the devices you coded to build the device preference table.

HARDCPY=
specifies that a hardcopy log is to be used to record operator commands, system commands, and responses, and write-to-operator (WTO and WTOR) messages. (Some control (K) commands, which control console functions rather than system functions, are not recorded on the hardcopy log. These commands are Control D, Control E, and Control with no operands.) If this parameter is not specified, SYSLOG, ALL, and NOCMDS are used. For information about operator communication with the system, the hardcopy log, and the system log, refer to Operator's Library: OS/VS2 MVS System Commands and OS/VS2 Supervisor Services and Macro Instructions.

The subparameters for HARDCOPY are positional and must be coded in the sequence shown in the macro 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.

address
: specifies the unit address of a device with at least output capability to be used as the hardcopy log device. The value specified is the unit address of one of the devices listed in Figure 25.

Devices	**Devices**
1052 Printer Keyboard model 7	3211 Printer
1403 Printer	3213 Printer
1443 Printer model N1	3215 Printer Keyboard
2740 Communication Terminal	3284 Printer model 1
3203 Printer model 4	3284 Printer model 2
3210 Console Printer Keyboard	3286 Printer model 1
	3286 Printer model 2

Figure 25. Devices that can be used as the hardcopy log device

The unit address specified must also be specified for the device in an IODEVICE macro and as the unit address of a console or part of a console in a CONSOLE macro. A graphics device cannot be specified as the hardcopy log device.

SYSLOG
: specifies that the data that is supposed to go to the hardcopy log will go to the system log, if a routecode is specified in the WTO or WTOR.

ALL
: specifies that all write-to-operator (WTO and WTOR) messages are to go to 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 MCONS and SECONS parameters of the CONSOLE macro. (Routing codes 1 and 2 are always assigned.)

For information on routing and descriptor codes, refer to *OS/VS Message Library: VS2 Routing and Descriptor Codes*.

CMDS
: specifies that operator and system commands, responses, and status displays (static and time-interval updated) are to go to the hardcopy log.

INCMDS
: specifies that operator and system commands and responses (but not status displays) are to go to the hardcopy log.

NOCMDS
: specifies that no operator or system commands or responses are to go to the hardcopy log.

STCMDS
: specifies that operator and system commands, responses, and status displays (except time-interval

updated status displays) are to go to the hardcopy log.

PRISUB={name|JES2|JES3}
specifies the name of the primary job entry subsystem. The name you specify cannot exceed four alphameric characters.

If JES3 is specified, the JES macro must be called.

SUBSYS={name|PRISUB}
specifies the name or names of secondary job entry subsystems. Each name cannot exceed four alphameric characters and up to fifteen names can be specified. If this parameter is not specified, the name specified in the PRISUB parameter is used. Do not specify the PRISUB name in the SUBSYS parameter.

Note: Secondary JOB ENTRY SUBSYSTEMS are not alternate JOB ENTRY SUBSYSTEMS for the primary JOB ENTRY SUBSYSTEM. The primary JOB ENTRY SUBSYSTEM must be started before INITIATORs will start successfully and useful work can be done. Refer to *OS/VS2 System Programming Library: Job Management* for an alternative method of adding secondary subsystems. This alternative method does not require a new system generation.

TAVR={200|556|800}
specifies the standard density for 7-track magnetic tape volumes used with automatic volume recognition (AVR).

EXAMPLE: This macro specifies the options for the job scheduler and master scheduler as follows:

- Twenty 130-byte records are to be set aside in SYS1.BRODCAST for broadcast messages.
- Data that would go to the hardcopy log is to go to the system log.
- All write-to-operator messages are to go to the hardcopy log.
- Operator and system ccommands, responses, and status displays are to go to the hardcopy log.
- An installation device preference table is to be created in the order of device performance (the default value).
- The name of the primary job entry subsystem is JES2 (the default value).
- The name of the secondary job entry subsystem is the same as the name of the primary job entry subsystem (the default value).
- The standard density for 7-track magnetic tape volumes with automatic volume recognition is 800 BPI.

```
SCHED   SCHEDULR   BCLMT=20,HARDCPY=(SYSLOG,ALL,CMDS)
```

Optional for:	Complete I/O Device
Not Applicable for:	Eligible Device Table

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

For each type 1, type 2, or type 6 SVC, a corresponding member should be specified in the DATASET macro that is being used to define SYS1.NUCLEUS. Each member may contain more than one SVC routine. For each type 3 or type 4 SVC, a corresponding member should be specified in the DATASET macro that is being used to define SYS1.LPALIB. Type 5 is used to reserve space for SVCs to be defined later. Each member may contain only one SVC routine. Refer to the discussion about the DATASET macro in this chapter for information about including user-written SVC routines in the system during system generation.

For an I/O device generation, this macro may be respecified with changes. If new user-written SVC routines not specified for the last complete generation are to be included, a separate job must be run to copy or link-edit the new user-written SVC routines into the receiving data sets.

For information about writing SVC routines and about the Authorized Program Facility (APF), see OS/VS2 System Programming Library: Supervisor. For an example of including your own SVC routines during system generation, refer to the section "Adding User-Written Routines to the System Control Program" in the chapter "Preparing for System Generation." For information about locking structures, refer to OS/VS2 System Programming Library: Supervisor.

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

operand

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

SVC-nnn -Tn[-(Ln,...)][-{FC01|FC00}][-NP]

where:

nnn

specifies the number of the SVC as a decimal number. You must assign unique numbers 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.

Tn

specifies the SVC type as 1, 2, 3, 4, 5, or 6.

Ln

specifies the value or values that will indicate to the first level interrupt handler which locks to obtain before entry to the SVC. The values that can be specified are:

1	LOCAL
2	CMS
3	SRM
4	SALLOC
5	DISPATCHER

For type 3 and type 4 SVC routines, SALLOC (L4) is invalid. If a type of lock is not specified, the only default is LOCAL (L1) for type 1 SVC routines.

Note: Ln cannot be specified for type 6 SVC routines. A type 6 SVC routine must acquire and release its own locks.

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

NP
specifies that the SVC runs nonpreemptible for I/O interrupts.

EXAMPLE: This macro identifies four unrestricted user-written SVCs. (Corresponding DATASET macros were specified for these SVCs.)

```
SVC   SVCTABLE   SVC-255-T1,                          X
                 SVC-254-T4,                          X
                 SVC-253-T4,                          X
                 SVC-252-T3
```

Optional for: Complete
I/O Device

Not Applicable for: Eligible Device Table

The TSO macro allows you to specify limits to certain functions of the TSO command system. If this macro is not specified, a full TSO command system is assumed and the default values given for the parameters are used. If CMDS=NO is specified in this macro, the TSO command system is still included but with limited command processing functions.

For an I/O device generation, the CMDS parameter must be specified either explicitly or by default to match the value specified (either explicitly or by default) in the last complete system generation. The LOGLINE and LOGTIME parameters may be respecified with or without change. If these parameters are not specified, the defaults are used.

For information about TSO, refer to OS/VS2 TSO Command Language Reference and to OS/VS2 TSO Terminal User's Guide.

[symbol]	**TSO**	**[CMDS=**{**NO**|**YES**}**]** **[LOGLINE=**{number|**10**}**]** **[LOGTIME=**{number|**300**}**]**

CMDS={**NO**|**YES**}**]**
specifies whether or not the full TSO command system is to be included. If NO is specified, the TSO command system is included but with limited TSO command processing functions.

LOGLINE={number|**10**}
is a number from 1 to 16,777,215 that specifies the number of lines that may be entered before an attempt to log on is automatically canceled.

LOGTIME={number|**300**}
is a number from 1 to 16,777,215 that specifies the number of seconds you may wait without a terminal response during a logon attempt.

EXAMPLE: This macro specifies that 600 seconds may elapse before LOGON issues a message and that 5 lines may be entered before a LOGON attempt is canceled.

```
  TSODF    TSO    LOGTIME=600,LOGLINE=5
```

UNITNAME

Required for: Complete
I/O Device
Eligible Device Table

The UNITNAME macro is used to specify a name for a collection of I/O devices. In order for a device to be referred to by other than its device type (for example, generic name, such as 2314), as specified in the UNIT parameter of an IODEVICE macro, the device must be specified in this macro. (See "Appendix A. Device Types" for a list of device types.) This macro is also used to specify those direct-access devices that can be used for virtual input/output (VIO) data sets.

This macro 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

If this system is ever to be used as a generating system for system generation, this macro should be used to assign the name SYSRDR to card readers.

In every system generation the system provides a unitname, SYSALLDA, to represent all direct access devices. This name is used by the system when allocating data sets and is the last name in the Eligible Device Table. You may not specify SYSALLDA as one of your unitnames.

You specify group names according to the following rules:

- The addresses specified for a device must also be specified in an IODEVICE macro.
- A maximum of 100 uniquely named groups can be specified.
- The maximum number of addresses that may be specified is 2056 minus the number of names.
- A maximum of 2055 addresses can be included in one group, regardless of the number of concatenations.
- A maximum of 255 characters can be used in the operand field of a UNITNAME macro. If more addresses must be listed than can fit in one operand, another UNITNAME macro can be coded using the same name. Those UNITNAME macros with the same name must be together in the group of macro instruction cards. If they are separated, the system assumes that the addresses on the first UNITNAME macro are the only addresses that belong to that UNITNAME.
- All UNITNAME macros having the same value in the NAME parameter must appear consecutively in your Stage I input deck.
- The UNITNAME macro should not be used to assign a name to a group of telecommunication devices. If this is done, when the user-assigned name is used in a DD statement, only the first unit specified in the UNITNAME is allocated.

For the Mass Storage System (MSS), UNITNAME macro instruction cards are generated as part of the Mass Storage Control Table Create run. The UNITNAME cards should be inserted into the deck used as input to Stage I. For more information on the Mass Storage Control Table Create run, see OS/VS Mass Storage Control Table Create.

For an I/O device generation, this macro may be used to name a new group of I/O devices or change the name of an existing group. If you want to include the same group of I/O devices, you must respecify the macro without change. If the catalog from the previous generation is used, and esoteric names are used to catalog data sets, the UNITNAME macros should be in the same sequence and contain the device classes that were in the previous generation. There should be no deletions, and any additions should be at the end.

For information on VIO, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

[symbol]	**UNITNAME**	NAME=name [UNIT=(address[,(address,n)]...)] [VIO={YES|NO}]

NAME=name

specifies the name to be given to a group of I/O devices. The name can be from one to eight characters. These characters can be alphameric, national (#, @, or $), or the two special characters, slash (/) and hyphen (-). Except for the devices you designate as VIO devices, the name you specify cannot be a device type (generic name) as listed in "Appendix A. Device Types." For VIO devices, the name you specify can be either a direct-access device type (such as 2305-1 or 2314) or an esoteric name (such as DISK).

UNIT=

specifies the addresses of a group of I/O devices that will be recognized by the name assigned. This parameter is not valid if VIO=YES is specified and the name specified in the NAME parameter is a generic name (such as 2314).

address

specifies the unit address of an I/O device that will be recognized by the name assigned. You can specify three hexadecimal digits, 000 through FFE.

(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 a 1- to 3-digit decimal number. If n is omitted, a value of 1 is assumed.

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, 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.

VIO={YES|NO}

specifies whether or not this group of devices will be VIO devices. If VIO=YES is specified for a group of devices having an esoteric name as the group name (SYSDA, for example), at least one device in the group must be a direct-access device. If VIO=YES is specified for a group of devices having a generic name as the group name (2314, for example), the name must be that of a direct-access device and the UNIT parameter must not be specified. For information on VIO, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

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

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

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

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

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

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

EXAMPLE: This macro specifies that the name SYSDA can be used to refer to devices at the specified addresses and that this group of devices can be used for VIO data sets.

```
UNITNAM   UNITNAME   NAME=SYSDA,VIO=YES,                     X
               UNIT=((110,2),(150,8),(213,5),                X
               (220,8),(250,8),(350,8),(410,8))
```

SELECTING AND DEFINING THE SYSTEM DATA SETS

Some or all of the system data sets can be defined either during system generation, by using the DATASET macro, or before system generation, by using JCL and/or the Access Method Services. This chapter contains information about each of the system data sets and about defining them.

"Defining the System Data Sets" tells how to allocate space for the system data sets and catalog them in the master catalog. In addition, this section gives examples of coding the parameters in the DATASET macro and the job control language, command statements, and parameters required to execute the Access Method Services.

The example in this section shows the use of JCL and the Access Method Services commands for defining the system data sets. This example does not show how the DATASET macro expands during Stage I processing. Instead, the example uses Access Method Services command options that you may want to specify. Before predefining your data sets, refer to OS/VS2 Acce⸗s Method Services.

"System Data Set Summary" presents an overview of the system data sets that may be helpful to refer to, particularly if you define them using JCL and/or the Access Method Services. Also in this section are individual discussions about each of the system data sets. The system data sets are arranged in alphabetic order by their qualified name.

DEFINING THE SYSTEM DATA SETS

Before components from the distribution libraries and user-defined data sets can be placed in the system, the master catalog must be built, space must be allocated for the system data sets on system volumes, and they must be cataloged in the master catalog.

USING THE DATASET MACRO TO DEFINE THE SYSTEM DATA SETS

The DATASET macro can be used to define the system data sets. If you use the DATASET macro, one macro must be specified for each data set to be defined during system generation. You need not, however, use the DATASET macro and, if you do, you need not use it exclusively. The information necessary to code the DATASET macro is in "Specifying the MVS System Control Program." (This chapter also contains information about using the DATASET macro to add your own routines to some of the system data sets.) Each of the system data sets is discussed individually later in this chapter. Refer to Figure 26 for an example of using the DATASET macro to define the system data sets.

The DATASET macros that you code with a SPACE parameter are used during system generation to allocate and define the system data sets. During Stage I, the macros are assembled and expanded into the job control language and commands required to execute the Access Method Services. During Stage II, the expansion is used to allocate the specified space on system volumes and catalog the system data sets in the master catalog. The first system data set that is created (if specified in a DATASET macro) is the master catalog. Then, space is allocated to the remaining system data sets and they are cataloged. Also, if you specified an index in the GENERATE macro other than SYS1, all of the system data sets are renamed during Stage II from the index you specified to SYS1.

```
BRODCAST DATASET  BRODCAST,VOL=(SG2001,3330),SPACE=(CYL,(5))
CMDLIB   DATASET  CMDLIB,VOL=(SG2001,3330),SPACE=(CYL,(10,5,100))
DCMLIB   DATASET  DCMLIB,VOL=(SG2001,3330),SPACE=(CYL,(1,,10))
HELP     DATASET  HELP,VOL=(SG2001,3330),SPACE=(CYL,(1,1,5))
IMAGELIB DATASET  IMAGELIB,VOL=(SG2001,3330),SPACE=(CYL,(5,1,30))
JES3LIB  DATASET  JES3LIB,VOL=(SG2001,3330),SPACE=(CYL,(15,2,75))
LINKLIB  DATASET  LINKLIB,VOL=(SG2001,3330),SPACE=(CYL,(15,3,300))
LPALIB   DATASET  LPALIB,VOL=(SG2001,3330),SPACE=(CYL,(20,3,300))
MACLIB   DATASET  MACLIB,VOL=(SG2001,3330),SPACE=(CYL,(13,3,50))
MANX     DATASET  MANX,VOL=(SG2001,3330),SPACE=(CYL,(2))
MANY     DATASET  MANY,VOL=(SG2001,3330),SPACE=(CYL,(2))
NUCLEUS  DATASET  NUCLEUS,VOL=(SG2001,3330),SPACE=(CYL,(15,,10))
PARMLIB  DATASET  PARMLIB,VOL=(SG2001,3330),SPACE=(CYL,(5,,25))
PROCLIB  DATASET  PROCLIB,VOL=(SG2001,3330),SPACE=(CYL,(3,5,20))
SAMPLIB  DATASET  SAMPLIB,VOL=(SG2001,3330),SPACE=(CYL,(1,1,10))
STGINDEX DATASET  STGINDEX,VOL=(SG2001,3330),SPACE=(CYL,(6))
SVCLIB   DATASET  SVCLIB,VOL=(SG2001,3330),SPACE=(CYL,(10,2,15))
TELCMLIB DATASET  TELCMLIB,VOL=(SG2001,3330),SPACE=(CYL,(8,3,70))
UADS     DATASET  UADS,VOL=(SG2001,3330),SPACE=(CYL,(8,1,20))
VSCATLG  DATASET  VSCATLG,VOL=(SG2001,3330),SPACE=(CYL,(50,5)),        X
               NAME=VSAMCAT
PAGE1    DATASET  PAGEDSN=SG2PAGE1,VOL=(SG2001,3330),SPACE=(CYL,(25))
PAGE2    DATASET  PAGEDSN=SG2PAGE2,VOL=(SG2001,3330),SPACE=(CYL,(25))
PAGE3    DATASET  PAGEDSN=SG2PAGE3,VOL=(SG2001,3330),SPACE=(CYL,(100))
SWAP1    DATASET  SWAPDSN=SG2SWAP1,VOL=(SG2001,3330),SPACE=(CYL,(40))
DUPLEX1  DATASET  DUPLEXDS,NAME=DUPLEX1,VOL=(SG2001,3330),             X
               SPACE=(CYL,(50))
```

Figure 26. Using the DATASET macro to define the system data sets

USING JCL AND THE ACCESS METHOD SERVICES TO DEFINE THE SYSTEM DATA SETS

Instead of using the DATASET macro, JCL and the Access Method Services can be used to define the system data sets. Generally, the only requirements for predefining the system data sets are:

- The system data sets must be defined before the Stage II part of system generation begins.
- The master catalog must be defined first. This can be accomplished by defining it in the first step of the job and defining the remaining system data sets in the second step.
- The unique name of the master catalog must be specified in a DATASET macro.
- You must specify a DATASET macro for each system data set that is predefined, except for those system data sets that are required to reside on the system residence volume or that are assumed to reside on it. By omitting the SPACE parameter in the DATASET macro, space will not be allocated.

Figure 27 shows how to specify the JCL to predefine nonVSAM system data sets. Figure 28 through Figure 31 are coding examples that (1) use the Access Method Services commands to define VSAM system data sets (master catalog, SMF data sets, SYS1.STGINDEX, page data sets, and swap data sets), and (2) use JCL and the Access Method Services DEFINE command to allocate space for and catalog the nonVSAM system data sets. Figure 28 through Figure 31 do not represent the expansion of the DATASET macro during Stage I. Instead, Figure 28 through Figure 31 use recommended options of the Access Method Services commands to define attributes of the data and index of the master catalog, SMF, SYS1.STGINDEX, page, and swap data sets. Complete details of these options are provided in OS/VS2 Access Method Services.

INPUT DECK

Statement			Comment
//name	JOB		Include any parameters required by your installation.
//name	EXEC	PGM=IDCAMS	Code this statement as shown.
//STEPCAT	DD	DSNAME=dsname,DISP=SHR	This statement identifies the name of the new master catalog in which the system, page and swap data sets are to be cataloged. Do not specify this statement in the step that defines the new master catalog (see Figure 28 through Figure 31).
//SYSPRINT	DD	SYSOUT=output class	This statement identifies the message output data set.

//ddname
//
//
//
//
//
	DD	DSNAME=dsname,
		VOLUME=(,RETAIN,SER=serial),
		UNIT=unit,DISP=(,KEEP),
		LABEL=EXPDT=99350,
		SPACE=(allocation),
		DCB=(dcb information)
			A DD statement is required for each nonVSAM system data set for which space is to be allocated. Refer to Figure 32 for DCB information.

//ddname	DD	UNIT=unit,VOL=SER=volser,
		DISP=OLD
			This statement identifies the device and volume to be used for the master catalog, VSAM system data sets, page and swap data sets. If this statement and the FILE parameter are omitted, space is allocated dynamically on the volumes specified in the VOLUMES parameter.

//SYSIN	DD	*	Column 1 must be blank. A hyphen (-) must be used to indicate continuation of a command. See Figure 28 through Figure 31 for an example of the commands and parameters.
The DEFINE function commands and parameters			
/*			

Figure 27. Predefining system data sets using job control language, command statements, and parameters

DEFINING THE MASTER CATALOG—STEP 1

This step (DEFINEMC) allocates the data space for the new master catalog and suballocates the actual catalog space. The size of the data space allocation (50 cylinders) is determined from the MASTERCATALOG parameter of the DEFINE command. The size of the suballocation for the catalog itself is determined from the DATA parameter. Additional details on estimating space allocation are provided in OS/VS2 Access Method Services.

The master catalog does not require all of the data space; consequently, the remaining space can be used in subsequent allocations of VSAM data sets on the same volume. Suballocations from this data space are obtained using the SUBALLOCATION parameter of the DEFINE command, which is the default.

After defining the data space and the new master catalog, the LISTCAT command is issued to print the names of the entries in the new catalog.

```
//DEFMCAT   JOB MSGLEVEL=1,MSGCLASS=A
//DEFINEMC  EXEC PGM=IDCAMS
//CATVOL    DD  VOL=SER=SG2001,UNIT=3330,DISP=OLD
//SYSPRINT  DD SYSOUT=A
//SYSIN DD *
  DEFINE  MASTERCATALOG (NAME (SYS1.A.NEW.MASTER.CATALOG)   -
                        FILE (CATVOL)                       -
                        VOLUME (SG2001)                     -
                        CYLINDERS (50 10))                  -
          DATA          (RECORDS  (3800 760))
  LISTCAT          CATALOG (SYS1.A.NEW.MASTER.CATALOG)
/*
```

Figure 28. Defining the master catalog using AMS—Step 1

CATALOGING THE NONVSAM DATA SETS—STEP 2

The step with step name NONVSAM creates catalog entries for each of the nonVSAM data sets represented by a DEFINE command. The entries are created in the new master catalog, which was created in the previous step.

The LISTCAT command prints the names of the nonVSAM entries in the new catalog.

```
//NONVSAM  EXEC PGM=IDCAMS
//STEPCAT  DD  DSN=SYS1.A.NEW.MASTER.CATALOG,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD *
 DEFINE NONVSAM (NAME(ALT1.BRODCAST) VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.CMDLIB)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.DCMLIB)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.DUMP00)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.HELP)     VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.IMAGELIB) VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.JES3LIB)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.LINKLIB)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.LPALIB)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.MACLIB)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.MANX)     VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.MANY)     VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.NUCLEUS)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.PARMLIB)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.PROCLIB)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.SAMPLIB)  VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.SVCLIB)   VOLUMES(SG2001) DEVICETYPES(3330))
 DEFINE NONVSAM (NAME(ALT1.TELCMLIB) VOLUMES(SG2001) DEVICEYTPES(3330))
 DEFINE NONVSAM (NAME(ALT1.UADS)     VOLUMES(SG2001) DEVICETYPES(3330))
 LISTCAT LEVEL (ALT1) CATALOG (SYS1.A.NEW.MASTER.CATALOG)
/*
```

Figure 29. Defining and Cataloging the nonVSAM system data sets using AMS—Step 2

ALLOCATING THE NONVSAM DATA SETS—STEP 3

The step named ALLOCATE allocates space for the nonVSAM data sets that were cataloged in the previous step. Since these data sets have already been cataloged, the catalog is not accessed or updated during this step.

```
//ALLOCATE EXEC PGM=IEFBR14
//BRODCAST DD DSN=ALT1.BRODCAST,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             UNIT=3330,SPACE=(CYL,(5),,CONTIG)
//*
//CMDLIB   DD DSN=ALT1.CMDLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//             UNIT=3330,SPACE=(CYL,(10,5,100))
//*
//DCMLIB   DD DSN=ALT1.DCMLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//             UNIT=3330,SPACE=(CYL,(1,,10),,CONTIG)
//*
//DUMP     DD DSN=ALT1.DUMP00,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,SPACE=(4096),(150),,CONTIG),
//             UNIT=3330
//*
//HELP     DD DSN=ALT1.HELP,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960),
//             UNIT=3330,SPACE=(CYL,(1,1,5))
//*
//IMAGELIB DD DSN=ALT1.IMAGELIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024),
//             UNIT=3330,SPACE=(CYL,(5,1,30))
//*
//JES3LIB  DD DSN=ALT1.JES3LIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             UNIT=3330,DCB=(RECFM=U,BLKSIZE=13030),
//             SPACE=(CYL,(15,2,75)),LABEL=EXPDT=99350
//*
//LINKLIB  DD DSN=ALT1.LINKLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//             UNIT=3330,SPACE=(CYL,(15,3,300))
//*
//LPALIB   DD DSN=ALT1.LPALIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//             UNIT=3330,SPACE=(CYL,(20,3,200))
//*
//MACLIB   DD DSN=ALT1.MACLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960),
//             UNIT=3330,SPACE=(CYL,(13,3,50))
//*
//MANX     DD DSN=ALT1.MANX,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,UNIT=3330,SPACE=(CYL,(2),,CONTIG)
//*
//MANY     DD DSN=ALT1.MANY,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,UNIT=3330,SPACE=(CYL,(2),,CONTIG)
//*
//NUCLEUS  DD DSN=ALT1.NUCLEUS,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//             LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//             UNIT=3330,SPACE=(CYL,(15,,10),,CONTIG)
//*
```

Figure 30 (Part 1 of 2). Allocating space for the nonVSAM system data sets using AMS—Step 3

```
//PARMLIB  DD DSN=ALT1.PARMLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//            LABEL=EXPDT=99350,DCB=(RECFM=F,LRECL=80),
//            UNIT=3330,SPACE=(CYL,(3,,50),,CONTIG)
//*
//PROCLIB DD DSN=ALT1.PROCLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//           LABEL=EXPDT=99350,DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960),
//           UNIT=3330,SPACE=(CYL,(3,5,20))
//*
//SAMPLIB  DD DSN=ALT1.SAMPLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//            LABEL=EXPDT=99350,DCB=(RECFM=F,LRECL=80,BLKSIZE=80),
//            UNIT=3330,SPACE=(CYL,(1,1,10))
//*
//SVCLIB   DD DSN=ALT1.SVCLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//            LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//            UNIT=3330,SPACE=(CYL,(10,2,15))
//*
//TELCMLIB DD DSN=ALT1.TELCMLIB,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//            LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=13030),
//            UNIT=3330,SPACE=(CYL,(8,3,70))
//*
//UADS     DD DSN=ALT1.UADS,DISP=(,KEEP),VOL=(,RETAIN,SER=SG2001),
//            DCB=(DSORG=PO,LRECL=80,RECFM=FB,BLKSIZE=800),
//            UNIT=3330,SPACE=(CYL,(8,3,70)),LABEL=EXPDT=99350
/*
```

Figure 30 (Part 2 of 2). Allocating space for the nonVSAM system data sets using AMS—Step 3

DEFINING THE VSAM DATA SETS—STEP 4

This step, named VSAMDEF, defines the VSAM system data sets—the storage index, the page data sets, and the swap data set(s). Since VSAM allocates the data space for a unique component when you define the data set, only nonunique data sets (that is, SYS1.STGINDEX) need a DEFINE SPACE. Page and swap data sets are defined with the UNIQUE attribute.

The first DEFINE CLUSTER command defines a VSAM key sequenced data set with the characteristics of the system data set SYS1.STGINDEX in the space allocated by the DEFINE SPACE command.

The second DEFINE CLUSTER command defines a VSAM entry sequenced data set with the characteristics of an SMF system data set in the space allocated by the DEFINE SPACE command.

Next, three DEFINE PAGESPACE commands are issued to create system paging data sets (PLPA, COMMON, and first LOCAL respectively). At least three page data sets must be defined. The first (PLPA) and second (COMMON) page data sets combined space must be large enough to hold the PLPA (pageable link pack area) and CSA (common system area) pages. Remaining page data sets, LOCAL, must be large enough to hold VIO pages and pageable private area pages. If no swap data sets are defined, the local page data sets will also be used to store the LSQA (local system queue area) areas of address spaces swapped out.

The fourth DEFINE PAGESPACE command in this example is issued to create a swap data set. Although swap data sets are also optional, they are recommended for reasons of performance. More than one swap data set can be defined. Enough space should be defined to hold LSQA areas of address spaces swapped out. Swap data sets cannot be defined on 2314 devices.

The last DEFINE PAGESPACE command is for an optional page data set, the duplex data set. IF a duplex data set is defined, there still must be at least three other page data sets defined. Since duplex processing is done only for PLPA and CSA pages, the duplex data set need never be larger than the combined space of the first (PLPA) and second (COMMON) page data sets defined.

See <u>OS/VS2 System Programming Library: Initialization and Tuning Guide</u> "ASM Initialization" to determi'ne the amount of space to allocate for each data set.

For simplification, the example defines all data sets on one volume.

Note: This is not recommended in the referenced document.

Finally, the LISTCAT command is issued to provide a list of the entries in the catalog. All of the attributes of each entry are printed.

```
//VSAMDEF  EXEC PGM=IDCAMS
//STEPCAT  DD   DSN=SYS1.A.NEW.MASTER.CATALOG,DISP=OLD
//VOLUME   DD   DISP=OLD,VOL=SER=SG2001,UNIT=3330
//SYSABEND DD   SYSOUT=A
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
  DEFINE SPACE             (VOLUME (SG2001)            -
                           FILE (VOLUME)               -
                           CYLINDERS (36))
  DEFINE CLUSTER           (NAME (ALT1.STGINDEX)       -
                           VOLUME (SG2001)             -
                           CYLINDERS (10)              -
                           KEYS (12 8)                 -
                           BUFFERSPACE (5120)          -
                           RECORDSIZE (2041 2041)      -
                           FILE (VOLUME)               -
                           REUSE)                      -
         DATA              (CONTROLINTERVALSIZE (2048))
  DEFINE CLUSTER           (NAME (SYS1.MANX)           -
                           VOLUME (SG2001)             -
                           NONINDEXED                  -
                           CYLINDERS (25)              -
                           REUSE                       -
                           RECORDSIZE (4086 32)        -
                           SPANNED                     -
                           CONTROLINTERVALSIZE (4)     -
                           SHAREOPTIONS (2))
  DEFINE PAGESPACE         (NAME (SYS1.PAGE1)          -
                           CYLINDERS (25)              -
                           FILE (VOLUME)               -
                           VOLUME (SG2001)             -
                           UNIQUE)
  DEFINE PAGESPACE         (NAME (SYS1.PAGE2)          -
                           CYLINDERS (25)              -
                           FILE (VOLUME)               -
                           VOLUME (SG2001)             -
                           UNIQUE)
  DEFINE PAGESPACE         (NAME (SYS1.PAGE3)          -
                           CYLINDERS (100)             -
                           FILE (VOLUME)               -
                           VOLUME (SG2001)             -
                           UNIQUE)
  DEFINE PAGESPACE         (NAME (SYS1.SWAP1)          -
                           CYLINDERS (40)              -
                           FILE (VOLUME)               -
                           VOLUME (SG2001)             -
                           UNIQUE SWAP)
  DEFINE PAGESPACE         (NAME (SYS1.DUPLEX)         -
                           CYLINDERS (50)              -
                           FILE (VOLUME)               -
                           VOLUME (SG2001)             -
                           UNIQUE)
  LISTCAT ALL

/*
```

Figure 31. Defining the VSAM system data sets using AMS—Step 4

SYSTEM DATA SET SUMMARY

Figure 32 lists the required and optional system data sets. This figure may be helpful to refer to when you are defining the system data sets. The values given for the DCB subparameters must be specified if you allocate space for the nonVSAM system data sets before Stage II using JCL. If you use the DATASET macro to define the nonVSAM system data sets, these values are the default values that are used. These values must not be specified if you are using the DATASET macro. See "Defining the

Page and Swap Data Sets" for information about the system paging spaces.

System Data Set	Type	Required or Optional Data Set	Sec. Space Allc.	DCB Parameters	Notes
Master Catalog	VSAM	required	yes	none	1,17
BRODCAST	direct	optional	no	none	14
CMDLIB	PDS	optional	yes	RECFM=U,BLKSIZE=7294	2,15
DCMLIB	PDS	optional	no	RECFM=U,BLKSIZE=7294	2,16
DUMPnn	seq.	optional	no	RECFM=F,BLKSIZE=4104	7,16
HELP	PDS	optional	yes	RECFM=FB,LRECL=80,BLKSIZE=7280	3,8,16
IMAGELIB	PDS	required	yes	RECFM=U,BLKSIZE=7294	16,21
INDMAC	PDS	optional	yes	RECFM=FB,LRECL=80,BLKSIZE=7280	3,9,20
JES3LIB	PDS	optional	yes	RECFM=U,BLKSIZE=7294	2,5
LINKLIB	PDS	required	yes	RECFM=U,BLKSIZE=7294	2,9
LOGREC	seq.	required	no	none	4,6
LPALIB	PDS	required	yes	RECFM=U,BLKSIZE=7294	2,9
MACLIB	PDS	required	yes	REDFM=FB,LRECL=80,BLKSIZE=7280	3,9
MANn	VSAM	optional	no	none	23
MANX	seq.	optional	no	none	11
MANY	seq.	optional	no	none	11
NUCLEUS	PDS	required	no	RECFM=U,BLKSIZE=7294	2,6,9
PARMLIB	PDS	required	no	RECFM=F,LRECL=80,BLKSIZE=80	
PROCLIB	PDS	required	yes	RECFM=FB,LRECL=80,BLKSIZE=7280	3,9
SAMPLIB	PDS	required	yes	RECFM=F,LRECL=80,BLKSIZE=80	
STGINDEX	VSAM	required	no	none	12,17,18
SVCLIB	PDS	required	yes	RECFM=U,BLKSIZE=7294	2,6,9
TCOMMAC	PDS	optional	yes	RECFM=FB,LRECL=80,BLKSIZE=7280	3,9,22
TELCMLIB	PDS	optional	yes	RECFM=U,BLKSIZE=7294	2,13,16
UADS	PDS	optional	yes	DSORG=PO,RECFM=F	10
VTAMLIB	PDS	optional	yes	RECFM=U,BLKSIZE=7294	2,19,20

Figure 32. Summary of the required and optional system data sets

Notes to Figure 32:

1. If the master catalog is predefined, a DATASET macro must also be specified.

2. Use BLKSIZE=7294 if the system data set resides on a 2314. Use BLKSIZE=14136 for a 2305-1, BLKSIZE=14660 for a 2305-2, BLKSIZE=13030 for a 3330 or 3330-1, BLKSIZE=8368 for a 3340, BLKSIZE=19069 for a 3350, and BLKSIZE=23476 for a 3380.

3. The value of BLKSIZE must be a multiple of 80 which is less than or equal to 7280 for a 2314, 14080 for a 2305-1, 14640 for a 2305-2, 12960 for a 3330 or 3330-1, 8320 for a 3340, 19040 for a 3350, or 23440 for a 3380.

4. The amount of space for SYS1.LOGREC is always calculated and allocated by the generating system on the system residence volume.

5. This system data set is required if JES3 is included in the system.

6. This system data set must be located on the system residence pack.

7. From one to ten DUMP<u>nn</u> (DUMP00—DUMP09) system data sets may be specified.

8. This system data set is required if the time-sharing HELP command is to be used.

9. For improved system efficiency, it is recommended that space be allocated on a cylinder boundary.

10. This system data set must be cataloged. Space is required only if terminal sessions are to be initiated.

11. This system data set must be cataloged. Space is required only if SMF or MF/1 recording is going to be done.

12. Space must be allocated in cylinders.

13. This system data set is required if BTAM and/or TCAM is included in the system.

14. This system data set must be cataloged. Space is required only if time-sharing messages are to be written.

15. This system data set must be cataloged and space must be allocated unless the TSO macro is specified to exclude the TSO command system (CMDS=NO).

16. If this system data set is predefined, a DATASET macro must also be specified. By omitting the SPACE parameter, space will not be allocated again.

17. It is strongly recommended that during the first system generation the master catalog, page data sets, and SYS1.STGINDEX be allocated space on a single volume. After the initial IPL, performance considerations may require a different configuration.

18. The STGINDEX data set must reside on a volume that is permanently mounted or reserved while it is in use.

19. This system data set is required if VTAM is included in the system.

20. This system data set is required if the industry subsystem support is to be included in the system.

21. With 3800 Enhancements, secondary extents can be specified for SYS1.IMAGELIB.

22. This system data set is required if ACF/TCAM is included in the system.

23. Valid only for System Extensions Release 2. Space is required only if recording is going to be done by SMF or other measurement facilities.

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

THE MASTER CATALOG

CONTENTS: The master catalog is a key-sequenced VSAM data set that contains data set and volume information required to locate data sets, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set, and to accumulate usage statistics for data sets. The master catalog is required.

REQUIREMENTS FOR SPECIFICATION

General Information: The master catalog must be on a direct-access volume, which can be the system residence volume. The master catalog must be assigned a unique name and the name must be specified in a DATASET macro—even if the master catalog is predefined.

Using DATASET: The NAME parameter must be specified. Space can only be allocated in units of tracks or cylinders.

Using the Access Method Services: The master catalog must be the first system data set defined. Also, you must specify a DATASET macro with the NAME parameter even if the master catalog is predefined.

Notes:

The master catalog must be password protected in order for the VSAM data sets cataloged in it to be password protected. The master catalog can be password protected by specifying the password and integrity parameters. It is recommended that you update the master catalog with the password *after* Stage II has been completed, to avoid requests for the password during Stage II.

The Access Method Services ALTER command that follows can be used, after the newly generated system has been IPLed, to update the new master catalog, thereby password protecting the catalog.

```
//ALTPSWD      JOB   ACCT123,PROGRAMMER,MSGLEVEL=(1,1)
//STEP1        EXEC  PGM=IDCAMS
//SYSPRINT     DD    SYSOUT=A
//SYSIN        DD    *
    ALTER -
         SYS1.A.NEW.MASTER.CATALOG -
         MASTERPW(master password) -
         CONTROLPW(control password)
/*
```

Some installations will want to utilize the master catalog from a previous production system. "Appendix G. Using the Master Catalog from the Old Production System" contains an example and a procedure for combining the system data sets from a newly generated system with the master catalog, SYS1.STGINDEX, and page data sets from an existing production system.

The master catalog should not be shared between systems.

ADDITIONAL INFORMATION: Information about using the Access Method Services to define the master catalog can be found in *OS/VS2 Access Method Services*. For information about the master catalog, refer to *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*.

CONTENTS: SYS1.BRODCAST is a direct data set that is required if time-sharing messages are to be written. This system data set contains two types of TSO messages:

- Notices—messages available for all users of a system
- Mail—messages available for specific users of a system

This system data set also contains a notice directory to facilitate the access of each type of message.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. It must be cataloged, but space is required only if time-sharing messages are to be written. Secondary space cannot be allocated.

Using JCL and the Access Method Services: DCB subparameters need not be specified.

The Access Method Services or JCL can be used for cataloging.

Notes:

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

This data set does not contain an expiration date.

This data set should not be shared between systems.

ADDITIONAL INFORMATION: For additional information about this system data set, refer to "Creating, Converting and Maintaining UADS and Broadcast Data Sets" in _OS/VS2 System Programming Library: TSO_.

SYS1.CMDLIB

CONTENTS: SYS1.CMDLIB is a partitioned data set that contains time-sharing command processor routines, service routines, and utility programs.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be cataloged. Space must be allocated unless the TSO macro is specified such that only a limited version of the TSO command system is included (CMDS=NO).

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

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.DCMLIB

CONTENTS: SYS1.DCMLIB is a partitioned data set that contains program function key (PFK) definitions for display consoles. It is required if a display console with the optional PFK feature is specified.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. Secondary space cannot be allocated.

If this system data set is used, a DATASET macro must be specified for it, even if it has been predefined. If it has been predefined, omit the SPACE parameter in DATASET.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

Note: This data set should not be shared between systems.

CONTENTS: The SYS1.DUMPnn system data sets (SYS1.DUMP00-SYS1.DUMP09) are sequential data sets that may contain system dumps used to record areas of virtual storage in case of system task failures. Each dump data set contains one dump. These system data sets are optional.

REQUIREMENTS FOR SPECIFICATION

General Information: From one to ten system data sets may be defined. Dump data sets may be all on tape, all on direct-access devices, or on a combination of devices. SYS1.DUMPnn system data sets that are to reside on direct-access devices can be defined before or during system generation. Dump data sets that are to reside on tape can only be defined when the system is initialized. Dump data sets can reside on the system residence volume.

Eligible device types are:

2400 series 9-track magnetic tape unit (or tape unit compatible with the 2400 series)

2305-1 fixed-head storage facility
2305-2 fixed-head storage facility
2314/2319 direct-access storage facility
3330 disk storage drive
3330-1 disk storage drive
3340/3344 disk storage drive
3350 direct access storage
3380 direct access storage

Secondary space cannot be allocated. Each dump data set must reside only on one volume. Direct-access dump data sets must reside on permanently resident volumes.

If any of these system data sets are predefined with an index other than SYS1, a DATASET macro must be specified in order to rename the data set to SYS1.DUMPnn. If any have been predefined, omit the SPACE parameter in DATASET.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=F,BLKSIZE=4104
```

The Access Method Services or JCL can be used for cataloging.

Note: These data sets should not be shared between systems.

ADDITIONAL INFORMATION: For further information about the dump data set(s), refer to OS/VS2 System Programming Library: Debugging Handbook.

For information about defining the dump data sets and identifying existing dump data sets to the system at system initialization, refer to the OS/VS2 System Programming Library: Initialization and Tuning Guide.

Refer to OS/VS2 System Programming Library: Service Aids for information about the AMDPRDMP service aid used to process the dumps.

SYS1.HELP

CONTENTS: SYS1.HELP is a partitioned data set that contains HELP information regarding the syntax, operands, and function of each time-sharing command. This system data set is required if the time-sharing HELP command is to be used.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. Secondary space can be allocated.

If this system data set is used, a DATASET macro must be specified for it, even if it has been predefined. If it has been predefined, omit the SPACE parameter in DATASET.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE={14080|14640|7280|12960|8320|19040|23440}
```

where:

BLKSIZE is a multiple of 80, which is less than or equal to:

```
14080 for a 2305 (Model 1)
14640 for a 2305 (Model 2)
 7280 for a 2314/2319
12960 for a 3330/3333 (Model 1 or 11)
 8320 for a 3340/3344
19040 for a 3350
23440 for a 3380
```

The Access Method Services or JCL can be used for cataloging.

ADDITIONAL INFORMATION: For information on this system data set, refer to <u>OS/VS2 TSO Command Language Reference</u>.

SYS1.IMAGELIB

CONTENTS: SYS1.IMAGELIB is a partitioned data set that contains the 1403, 3203-4, and 3211 universal character-set (UCS), 3203-4, 3211, and 3800 forms control buffer (FCB) modules, 3800 character arrangement table modules, 3800 graphic character modification modules, 3800 copy modification modules, Data Protection Images (DPI), and header records for the 3886 optical character reader. Since the UCSDFLT parameter of the DATAMGT macro defaults to ALL, SYS1.IMAGELIB is a required system data set. With 3800 Enhancements, SYS1.IMAGELIB can also contain library character set modules, and you can specify secondary allocation.

For information on adding either an IBM UCS (universal character set) or an IBM FCB (forms control buffer) to SYS1.IMAGELIB, refer to OS/VS2 System Programming Library: Data Management.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a permanently mounted direct-access volume, which can be the system residence volume. Secondary space can be allocated with 3800 Enhancements.

If this system data set is used, a DATASET macro must be specified for it, even if it has been predefined. If it has been predefined, omit the SPACE parameter in DATASET.

Using DATASET: User-written forms control buffer (FCB), universal character-set (UCS) images, and 3800 support modules can be included.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,
BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
BLKSIZE is:

14136 for a 2305 (Model 1)
14660 for a 2305 (Model 2)
 7294 for a 2314/2319
13030 for a 3330/3333 (Model 1 or 11)
 8368 for a 3340/3344
19069 for a 3350
23476 for a 3380
```

The Access Method Services or JCL can be used for cataloging.

Note: SYS1.IMAGELIB, when accessed through SVC 105, is automatically treated as an authorized library (that is, the bit is set in the DEB as if the library were entered in IEAAPFXX). Installations sensitive to security considerations are advised that SYS1.IMAGELIB should be protected to ensure that unauthorized programs cannot be made authorized by link editing into this library.

SYS1.INDMAC

CONTENTS: SYS1.INDMAC is a partitioned data set that contains the macro definitions for the industry subsystems. This data set is required if the industry subsystem support is included in the system.

REQUIREMENTS FOR SPECIFICATION

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

Secondary space can be allocated.

Using DATASET: If this system data set is used, a DATASET macro must be specified for it even if it has been predefined. If it has been predefined, omit the space parameter in the DATASET macro.

User-written macros can be included in SYS1.INDMAC during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE={14080|14640|7280|12960|8320|19040|23440}
```

where:

BLKSIZE is a multiple of 80, which is less than or equal to:

14080 for a 2305 (Model 1)
14640 for a 2305 (Model 2)
7280 for a 2314/2319
12960 for a 3330/3333 (Model 1 or 11)
8320 for a 3340/3344
19040 for a 3350
23440 for a 3380

The Access Method Services or JCL can be used for cataloging.

SYS1.JES3LIB

CONTENTS: SYS1.JES3LIB is a partitioned data set that contains all of the JES3 code except the subsystem interface modules that reside in SYS1.LPALIB. This system data set is required if JES3 is included in the system.

REQUIREMENTS FOR SPECIFICATION

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

Secondary space can be allocated.

Using DATASET: If this system data set is not specified in a DATASET macro, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
  RECFM=U,
  BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
  BLKSIZE is:

  14136 for a 2305 (Model 1)
  14660 for a 2305 (Model 2)
   7294 for a 2314/2319
  13030 for a 3330/3333 (Model 1 or 11)
   8368 for a 3340/3344
  19069 for a 3350
  23476 for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.LINKLIB

CONTENTS: SYS1.LINKLIB is a partitioned data set that contains programs and routines referred to by the XCTL, ATTACH, LINK, and LOAD macro instructions and nonresident system routines. This system data set also contains an assembler-language processor, a linkage editor, the utility programs and service aids. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

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

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

Using DATASET: If this system data set is not specified in a DATASET macro, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written routines, in load module form, can be included in SYS1.LINKLIB during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.LOGREC

CONTENTS: SYS1.LOGREC is a sequential data set that is used to contain statistical data about machine failures (CPU failures, I/O device errors, channel errors). It also contains records for program error recording, missing-interrupt information and dynamic device reconfiguration (DDR) routines. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must reside on the system residence volume. The system generation process defines SYS1.LOGREC, so you need not specify it.

Note: This data set should not be shared between systems.

ADDITIONAL INFORMATION: The size of SYS1.LOGREC can be increased or decreased after system generation by use of the IFCDIP00 program. Information on this program is in _OS/VS2 System Programming Library: SYS1.LOGREC Error Recording_.

SYS1.LPALIB

CONTENTS: SYS1.LPALIB is a partitioned data set that contains all of the modules that are loaded into the pageable link pack area (PLPA). This includes system routines, SVC routines, data management access methods, nonresident machine-check handler modules, authorization and accounting exit routines, logon mode tables, and some TSO modules. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. Because all of the modules in this system data set are brought into the LPA at system initialization, it may be placed on a demountable volume that may be removed after system initialization.

It is recommended that space be allocated on a cylinder boundary. Secondary space can be allocated.

Using DATASET: If this system data set is not specified in a DATASET macro, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written routines, in load module form, can be included in SYS1.LPALIB during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.MACLIB

CONTENTS: SYS1.MACLIB is a partitioned data set that contains the macro definitions for supervisor and data management macro instuctions. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

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

For improved system efficiency, it is recommended that space be allocated in cylinders. Secondary space can be allocated.

Using DATASET: If this system data set is not specified, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written system macros can be included in SYS1.MACLIB during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE={14080|14640|7280|12960|8320|19040|23440}
```

where:

BLKSIZE is a multiple of 80, which is less than or equal to:

14080 for a 2305 (Model 1)
14640 for a 2305 (Model 2)
7280 for a 2314/2319
12960 for a 3330/3333 (Model 1 or 11)
8320 for a 3340/3344
19040 for a 3350
23440 for a 3380

The Access Method Services or JCL can be used for cataloging.

SYS1.MANN (OS/VS2 MVS/EXTENSIONS RELEASE 2 (5740-XE1))

CONTENTS: The SYS1.MANn system data sets (SYS1.MANA through SYS1.MANZ and SYS1.MAN0 through SYS1.MAN9) are VSAM data sets that contain information collected by the system management facility (SMF) routines or other measurement facilities.

REQUIREMENTS FOR SPECIFICATION

General Information: These system data sets can be created at any time but must exist only if recording is to be done through SMF or other measurement facilities. They must reside on a permanently mounted volume, which can be the system residence volume. Secondary space cannot be allocated.

USING THE ACCESS METHOD SERVICES: The DCB parameter is not required.

The Access Method Services is used for cataloging.

Notes:

> These data sets have a protection date of 99350. The operator should reply "U" to message IEC107D when the system is initialized.
>
> These data sets should not be shared between systems.

ADDITIONAL INFORMATION: For information about SMF, refer to OS/VS2 System Programming Library: System Management Facilities (SMF).

You cannot specify SYS1.MANX and/or SYS1.MANY. Also if you allocate SYS1.MANX and/or SYS1.MANY using IDCAMS, you are notified that there are duplicate names in the catalog.

SYS1.MANX, SYS1.MANY (OTHER THAN OS/VS2 MVS/EXTENSIONS RELEASE 2 (5740-XE1))

CONTENTS: SYS1.MANX and SYS1.MANY are sequential data sets that contain information collected by the system management facility (SMF) routines. They may contain measurement information collected by the system activity measurement facility (MF/1), if MF/1 is supported by your system.

REQUIREMENTS FOR SPECIFICATION

General Information: These system data sets must be cataloged either before or during system generation, but space is required only if SMF or MF/1 recording is going to be done.

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

Space for both system data sets should be the same. Secondary space cannot be allocated.

Using JCL and the Access Method Services: The DCB parameter is not required.

The Access Method Services or JCL can be used for cataloging.

Notes:

These data sets have a protection date of 99350. The operator should reply "U" to message IEC107D when the system is initialized.

These data sets should not be shared between systems.

ADDITIONAL INFORMATION: For information about SMF, refer to OS/VS2 System Programming Library: System Management Facilities (SMF).

For information about MF/1, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

For OS/VS2 MVS/Extensions Release 2 (5740-XE1):: If you have System Extensions Release 2 (5740-XE1) installed on your system, you cannot specify SYS1.MANX and/or SYS1.MANY. Also, if you allocate SYS1.MANX and/or SYS1.MANY using IDCAMS, you are notified that there are duplicate names in the catalog.

SYS1.NUCLEUS

CONTENTS: SYS1.NUCLEUS is a partitioned data set that contains the resident portion of the control program (in member IEANUC0x), nucleus initialization modules, and a pointer to the master catalog (in member SYSCATLG). This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on the system residence volume. Secondary space cannot be allocated.

Using DATASET: A maximum of ten user-written load modules can be included in SYS1.NUCLEUS during system generation.

Using JCL and the Access Method Services: The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

Note: This data set should not be shared between systems.

ADDITIONAL INFORMATION: The CSECT, IEAPATCH, in member IEANUC01 will always be initialized to zeros after a complete system generation or an I/O device generation.

CONTENTS: SYS1.PARMLIB is a partitioned data set that contains IBM-supplied and installation-created lists of system parameter values. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. Secondary space cannot be allocated.

Using DATASET: If this system data set is not specified in a DATASET macro, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written system parameters can be included in SYS1.PARMLIB during system generation.

Using JCL and the Access Method Services: The following DCB subparameters must be specified:

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

The Access Method Services or JCL can be used for cataloging.

ADDITIONAL INFORMATION: For additional information, refer to OS/VS2 System Programming Library: Initialization and Tuning Guide.

SYS1.PROCLIB

CONTENTS: SYS1.PROCLIB is a partitioned data set that contains the cataloged procedures used to perform certain system functions. The cataloged procedures can be for system tasks or processing program tasks invoked by the operator or the programmer. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

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

For improved system efficiency, it is recommended that space be allocated on a cylinder boundary. Secondary space can be allocated.

Using DATASET: If this system data set is not specified, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written procedures can be included in SYS1.PROCLIB during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE={14080|14640|7280|12960|8320|19040|23440}
```

where:

BLKSIZE is a multiple of 80, which is less than or equal to:

```
14080 for a 2305 (Model 1)
14640 for a 2305 (Model 2)
 7280 for a 2314/2319
12960 for a 3330/3333 (Model 1 or 11)
 8320 for a 3340/3344
19040 for a 3350
23440 for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.SAMPLIB

CONTENTS: SYS1.SAMPLIB is a partitioned data set that contains the installation verification procedure (IVP), the independent utilities, and the IPL text. It also contains SMF sample exit routines. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. This system data set need not be cataloged. Secondary space can be allocated.

Using DATASET: If this system data set is not specified in a DATASET macro, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

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

The Access Method Services or JCL can be used for cataloging.

ADDITIONAL INFORMATION: For information on IVP, refer to "Testing the System Control Program" in this manual.

SYS1.STGINDEX

CONTENTS: SYS1.STGINDEX is a VSAM data set that contains auxiliary storage management records for virtual I/O (VIO) data sets that must be saved across job steps and between system initializations. This system data set is required for normal system operation, although it is possible to initialize the system (IPL) without it.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. This data set must reside on a volume that is permanently mounted or reserved while it is in use.

Using DATASET: Space must be allocated in cylinders.

Using the Access Method Services: If you predefine this system data set, it is recommended that space be allocated in cylinders.

Some of the parameters used to define this system data set require certain values. To predefine SYS1.STGINDEX, specify the following commands and parameters with appropriate values where indicated. The DEFINE SPACE command is not needed if SYS1.STGINDEX is to share space with another VSAM data set already defined.

```
[DEFINE SPACE-
 (VOLUME(volser)-
 CYLINDERS(larger than primary))]
DEFINE CLUSTER-
 (NAME(SYS1.STGINDEX)-
 VOLUME(volser)-
 CYLINDERS(primary)-
 KEYS(12 8)-
 BUFFERSPACE(5120) -
 RECORDSIZE(2041 2041)-
 REUSE))-
DATA-
 (CONTROLINTERVALSIZE(2048))-
INDEX-
 (CONTROLINTERVALSIZE(1024))
```

Note: This data set should not be shared between systems.

ADDITIONAL INFORMATION: The commands and parameters that are used to predefine SYS1.STGINDEX are discussed in OS/VS2 Access Method Services.

SYS1.SVCLIB

CONTENTS: SYS1.SVCLIB is a partitioned data set that contains some OLTEP and appendage modules. This system data set is required.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must reside on the system residence volume.

Space should be allocated in cylinders. Secondary space can be allocated.

Using DATASET: If this system data set is not specified, it is assumed to reside on the volume specified in the RESVOL parameter of the GENERATE macro (the system residence volume).

User-written routines, in load module form, can be included during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

Note: This data set should not be shared between systems.

SYS1.TCOMMAC

CONTENTS: SYS1.TCOMMAC is a partitioned data set that contains ACF/TCAM record API macros. It is required if ACF/TCAM is included in the system.

REQUIREMENTS FOR SPECIFICATION

General Information: This data set must be on a direct-access volume, which can be the system residence volume. This data set need not be cataloged.

Secondary space can be allocated.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE={14080|14640|7280|12960|8320|19040|23440}
```

where:

BLKSIZE is a multiple of 80, which is less than or equal to:

```
14080 for a 2305 (Model 1)
14640 for a 2305 (Model 2)
 7280 for a 2314/2319
12960 for a 3330/3333 (Model 1 or 11)
 8320 for a 3340
19040 for a 3350
23440 for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.TELCMLIB

CONTENTS: SYS1.TELCMLIB is a partitioned data set that contains telecommunications subroutines in load module form. It is required if BTAM and/or TCAM is included in the system.

REQUIREMENTS FOR SPECIFICATION

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

For improved system efficiency, it is recommended that space be allocated on a cylinder boundary. Secondary space can be allocated.

If this system data set is used, a DATASET macro must be specified for it, even if it has been predefined. If it has been predefined, omit the SPACE parameter in DATASET.

Using DATASET: User-written routines, in load module form, can be included during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

```
14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
 7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
 8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380
```

The Access Method Services or JCL can be used for cataloging.

SYS1.UADS

CONTENTS: SYS1.UADS is a partitioned data set that contains a list of terminal users who are authorized to use time-sharing and information about them.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be cataloged, but space is required only if terminal sessions are to be initiated. This system data set must be on a direct-access volume, which can be the system residence volume. Secondary space can be allocated.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
DSORG=PO
```

The LRECL and BLKSIZE parameters should be equal.

The Access Method Services or JCL can be used for cataloging.

Note: SYS1.UADS does not contain an expiration date. This data set should not be shared between systems.

ADDITIONAL INFORMATION: For information about this system data set, refer to "Creating, Converting and Maintaining UADS and Broadcast Data Sets" in OS/VS2 System Programming Library: TSO.

The SYS1.UADS supplied by IBM is allocated with LRECL=80 and BLKSIZE=800. IBMUSER is the only userid supplied, and this userid has the ability to update the SYS1.UADS data set with the entries for the TSO users your installation wants defined. As this SYS1.UADS does not have LRECL and BLKSIZE equal, you may wish to consider the SYS1.UADS copied during system generation as an intermediate UADS and create an additional UADS data set. After updating the latest UADS using ACCOUNT, the SYS1.UADS you copied during system generation should be deleted and the newly created UADS data set renamed SYS1.UADS.

SYS1.VTAMLIB

CONTENTS: SYS1.VTAMLIB is a partitioned data set that contains the VTAM load modules, installation-coded logon exit routines, authorization and accounting exit routines, USS definition tables, and the network solicitor. This system data set is required if VTAM is included in the system.

REQUIREMENTS FOR SPECIFICATION

General Information: This system data set must be on a direct-access volume, which can be the system residence volume. Secondary space can be allocated. This data set must be cataloged in the system catalog.

Using DATASET: User-written routines, in load module form, can be included in SYS1.VTAMLIB during system generation.

Using JCL and the Access Method Services: It is recommended that the following DCB subparameters be specified:

```
RECFM=U,BLKSIZE={14136|14660|7294|13030|8368|19069|23476}
```

where:

14136 is for a 2305 (Model 1)
14660 is for a 2305 (Model 2)
7294 is for a 2314/2319
13030 is for a 3330/3333 (Model 1 or 11)
8368 is for a 3340/3344
19069 is for a 3350
23476 is for a 3380

The Access Method Services or JCL can be used for cataloging.

Note: SYS1.VTAMLIB, as well as the user-defined Network Control Program load module library, must be specified in the APFLIB parameter of the CTRLPROG macro instruction.

DEFINING THE PAGE AND SWAP DATA SETS

This chapter contains general information about defining the page and swap data sets as well as information about using either the DATASET macro or the DEFINE function of the Access Method Services for defining the page and swap data sets.

Page or swap data sets are allocated from VSAM data spaces on direct-access storage devices. They contain the paged-out portions of address spaces, the common service area, and the data written to virtual I/O data sets.

Defining a page or swap data set consists of allocating space for it on a direct-access volume in a VSAM data space, cataloging it, formatting it, and including its name in the PAGE= or SWAP= portion of either the IEASYS00 or IEASYSxx system parameter list in SYS1.PARMLIB.

At least three page data sets must be defined before or during system generation. No swap data sets are required. However, it is recommended that swap data sets be supplied for maximum system performance. Up to twenty-five page data sets and up to ten swap data sets may be defined, and additional page or swap data sets may be defined after system generation. The page or swap data sets that are defined during system generation are automatically included in the IEASYS00 list in SYS1.PARMLIB. For those page or swap data sets defined before system generation, you must specify the name of the page or swap data set in a DATASET macro and then, during system generation, the name will be included in the IEASYS00 list. For those page or swap data sets defined after system generation, you must enter their names in either the IEASYS00 or IEASYSxx list yourself. (You can do this either by using the system utility IEBUPDTE or entering the name from the console.)

If page data sets from a previously generated system are to be reused, it is necessary to reuse the master catalog as well. See "Appendix G. Using the Master Catalog from the Old Production System."

The first named page data set in either the IEASYS00 or IEASYSxx list in SYS1.PARMLIB contains the pageable link pack area (PLPA). It is important, therefore, that the page data set intended for the PLPA be the first named page data set in SYS1.PARMLIB. For IEASYS00 this means that it appears on the first DATASET macro that contains the PAGEDSN parameter. For IEASYSxx the PAGE= portion of the member must have this data set listed first. The second named page data set in either IEASYS00 or IEASYSxx list in SYS1.PARMLIB will be used as a common data set system for common area pages.

For information about adding page or swap data set names to SYS1.PARMLIB and about system parameter lists, refer to the _OS/VS2 System Programming Library: Initialization and Tuning Guide_.

DEFINING PAGE OR SWAP DATA SETS USING THE DATASET MACRO

The DATASET macro can be used to define the page or swap data sets during system generation. If you choose to use the DATASET macro, one macro must be specified for each page or swap data set to be defined and you must include at least three page data sets. The parameters for the DATASET macro are discussed in "Specifying the MVS System Control Program."

When the DATASET macro is used to define a page or swap data set, its name is automatically included in the PAGE=, SWAP=, or DUPLEX= portion of the IEASYS00 list in SYS1.PARMLIB.

Even if you choose to predefine the page or swap data sets, you can use the DATASET macro to include the name of the page data sets in IEASYS00 as follows:

```
  Name     DATASET     PAGEDSN=dsname
  Name     DATASET     SWAPDSN=dsname
  Name     DATASET     DUPLEXDS,NAME=dsname
```

Absence of the SPACE parameter prevents space allocation but allows the page or swap data set name to be entered in the IEASYS00 list in SYS1.PARMLIB. When the SPACE parameter is not specified, the VOL parameter need not be specified. If it is specified, it will be ignored.

When using the DATASET macro to define page and swap data sets, space can only be allocated in cylinders and only primary extents can be specified.

Figure 26 gives an example of using the DATASET macro to define the page and swap data sets.

DEFINING PAGE OR SWAP DATA SETS USING THE ACCESS METHOD SERVICES

The page and swap data sets can be defined before system generation using the DEFINE command of the Access Method Services.

In allocating space using the Access Method Services, it is recommended that your allocation be in cylinders. Only primary extents may be specified for page or swap data sets.

Figure 28 shows the input deck for using the Access Method Services for defining page and swap data sets as well as system data sets.

PROTECTING PAGE AND SWAP DATA SETS

Page data sets should be password protected to prevent unauthorized use. During system generation, the page and swap data sets are defined without password protection. At the successful conclusion of the system generation process, you should update the master catalog to protect the data sets (that is, the space and its data component). The output from job 6 of a complete system generation includes a listing of the master catalog. The names of the page or swap space and its data component can be determined from this listing. Alternatively, the Access Method Services LISTCAT command can be issued to determine the page or swap space and data component names of the data sets. The ALTER commands that follow will alter the catalog entries for a page data set. OS/VS2 Access Method Services provides additional details on the format and function of the LISTCAT and ALTER commands.

```
//ALTPSWD      JOB  ACCT123,PROGRAMMER,MSGLEVEL=(1,1)
//STEP1        EXEC PGM=IDCAMS
//SYSPRINT     DD   SYSOUT=A
//SYSIN        DD   *
    ALTER -
         SYS1.PAGE01 -
         MASTERPW(master password) -
         CONTROLPW(control password) -
         UPDATEPW(update password) -
         READPW(read password) -
         AUTHORIZATION(authorization) -
         ATTEMPTS(0)
    ALTER -
         VSAMDSET.Tbbbbbbb.DFDyyddd.Taaaaaaa.Tbbbbbbb -
         MASTERPW(master password) -
         CONTROLPW(control password) -
         UPDATEPW(update password) -
         READPW(read password) -
         AUTHORIZATION(authorization) -
         ATTEMPTS(0)
/*
```

The name of the data component is generated by the system. It is always in the form VSAMDSET.Tbbbbbbb.DFDyyddd.Taaaaaaa.Tbbbbbbb, where yy is the year, ddd is the day, and aaaaaaa and bbbbbbb are the time of creation of the page or swap data set.

PREPARING FOR SYSTEM GENERATION

This chapter presents a general discussion about the procedures that need to be done to prepare for system generation. The following topics are discussed:

- Initializing direct-access volumes.
- Preparation for using the starter system as the generating system.
- Preparation for using an existing MVS system as the generating system.
- Adding your own routines to the new MVS system to be generated.

INITIALIZING DIRECT-ACCESS VOLUMES

Before an MVS system control program can be installed, the volumes that are to contain the starter system, the distribution libraries, and the MVS system control program to be generated 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) text must be written on the direct-access volume that is to become the system residence volume for the new system.

Volumes are initialized by executing either the IBCDASDI or the IEHDASDR utility programs. The IBCDASDI utility program is self-loading and its operation is independent of an operating system. The IEHDASDR utility program operates under the control of an operating system.

The IEHDASDR utility program is in the starter system in SYS1.LINKLIB or in your existing MVS system in SYS1.LINKLIB. The IBCDASDI independent utility program is in the first file of the first starter system tape and is used to initialize the volume(s) that are to contain the starter system. Once the starter system is restored from tape to the initialized volume(s) 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 new MVS system control program.

These utility programs and the control statements they require are discussed in more detail in OS/VS2 MVS Utilities.

PREPARING FOR SYSTEM GENERATION USING THE STARTER SYSTEM

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

This section discusses what you should 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 system initialization (IPL); any device not ready will automatically be taken offline. However, all teleprocessing control units such as the 2701, 2702, and 2703 should be disabled before the starter system is initialized. If a device that was not ready at system initialization is required during a job step, the operator should enter a VARY ONLINE command for that device.

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

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

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

Spool Volume

One of the starter system tapes contains the data set used for spooling, SYS1.HASPACE, which can be restored to a direct-access volume (VOL=SER=SPOOL0) for use during system generation. You may already have a spool data set. If you do, it can be used in conjunction with (not in place of) the starter system's spool data set. If it is used, the first five characters of the volume serial number must be SPOOL. If they are not, they must be changed. The LABEL function of the IEHDASDR utility can be used to change the volume serial number.

Volume Serial Number Assignments

Volume serial numbers must be assigned to the direct-access volumes that will contain the starter system and the distribution libraries.

For 3330, 3330-1, and 3340 Model 70 (or 3344) direct-access volumes, the volume serial numbers that are to be assigned are:

START1	for the starter system volume
SPOOL0	for the starter system spool and paging volume
DLIB01	for the distribution library volume

For 3340 Model 35 direct-access volumes, the volume serial numbers to be assigned are:

START1	for the starter system volume
SPOOL0	for the starter system spool volume
DLIB01	for the first distribution library volume
DLIB02	for the second distribution library volume

Note: The underscores indicate that these volume serial numbers were chosen arbitrarily. DLIB01 is the default value for the distribution library. If a second distribution library volume is required (for 2314 and 3340 Model 35 direct-access volumes), you must specify a volume serial number for the second volume.

PROCESSING THE STARTER SYSTEM AND DISTRIBUTION LIBRARY TAPES

Beginning with Release 3.8, the distribution libraries are distributed in SMP Release 4 relfile format, and SMP Release 4 must be used to load the libraries from the distribution tape onto a direct access device before performing a system generation. For detailed information on SMP Release 4, refer to OS/VS System Modification Program (SMP) System Programmers Guide. The starter system and distribution libraries are distributed on 1600 or 6250 BPI tape volumes. Along with the tape(s) containing the Release 3.8 distribution libraries, there is a distribution tape reel (DTR) provided which contains the procedures for allocating and cataloging the DLIBs on a direct access device, a copy of the SMP Release 4 load modules, and an SMP procedure for processing the distribution library tape(s). Before the starter system and distribution libraries can be used for system generation, certain procedures need to be followed to make them ready. This section describes a procedure that can be followed to prepare for system generation. This procedure is based on the minimum machine configuration requirements for system generation using the starter system as described in "Requirements for Generating an MVS System Control Program." The direct-access volumes in this illustration are 3330 direct-access volumes. This procedure can also be followed using 3330-1 or 3340 Model 70 volumes, changing the parameters in the examples that specify a device or unit address. If 3340 Model 35 volumes are used, the minimum requirements are slightly different (that is, as mentioned in the following procedure, when loading the distribution libraries to 3340 Model 35 direct-access devices, two volumes must be used, whereas only one of the other direct-access volumes is needed in such a procedure).

Note: Only one direct-access volume is required as the new system volume to perform system generation. However, before the new system control program can be initialized, one or more volumes may be required for the spool data set for the new system. Refer to the description of SYS1.HASPACE in OS/VS2 MVS System Programming Library: JES2 or System Programming Library: Network Job Entry Facility for JES2.

For illustrative purposes, the procedures that follow assume the devices and unit addresses listed below.

Note: The underscores in this and subsequent examples indicate that the volume serial number, device, unit address, or parameter was arbitrarily chosen.

I/O Device	Device Function	Address
3210	console keyboard	01F
3330 disk storage drive	starter system volume —START1	150
3330 disk storage drive	starter system volume —SPOOL0	151
3330 disk storage drive	distribution library volume—DLIB01	152
3330 disk storage drive	distribution library volume—DLIB02	153
3330 disk storage drive	new system volume —SYSRES	250
2540 card reader	system input	00C
2540 card punch	punched output	00D

I/O Device	Device Function	Address
1403 printer	printed output	00E
2400 or 3400 tape drive	tape drive for starter system tape(s)	182
2400 or 3400 tape drive	tape drive for distribution library tapes	183

Prepare to Initialize the Volumes That Will Contain the Starter System

1. Mount the direct-access volumes onto which the contents of the starter system tape is to be restored and the contents of the distribution library tape is to be copied.

2. Mount the starter system tape. The first file of the tape contains the IBCDASDI independent utility program and the IBCDMPRS independent utility program.

3. Load the IBCDASDI independent utility program from the starter system tape (START1 restore 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

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

```
START1  JOB
        MSG    TODEV=1403,TOADDR=00E
        DADEF  TODEV=3330,TOADDR=150,VOLID=SCRATCH,FLAGTEST=NO
        VLD    NEWVOLID=START1,OWNERID=DEPT38
        VTOCD  STRTADR=05,EXTENT=10
        END

SPOOL0  JOB
        MSG    TODEV=1403,TOADDR=00E
        DADEF  TODEV=3330,TOADDR=151,VOLID=SCRATCH,FLAGTEST=NO
        VLD    NEWVOLID=SPOOL0,OWNERID=DEPT38
        VTOCD  STRTADR=05,EXTENT=10
        END
```

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

2. Define the control statement input device by pressing the REQUEST key of the console keyboard. Message IBC105A 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 or unit address of your input device, enter the correct information.

3. When the START1 volume initialization is complete, message IBC163A END OF JOB is printed on the message output device, and the system continues to process the SPOOL0 job.

   If the 3066 console (used with the Model 165/168, System/370) is being used and the console typewriter is not available, then you must execute the independent utilities (IBCDASDI, IBCDMPRS, and ICAPRTBL) in the following manner:

Enter at storage location 0110 (hexadecimal): 1cuu for a 1442 Card Read Punch; 2cuu for a 2400 9-track tape unit, or 0cuu for a 2540 Card Read Punch, 2501 card reader, 3410 tape, or 3420 tape. Press the console INTERRUPT key.

For further information about these utility programs, see OS/VS2 MVS Utilities.

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

1. Load the IBCDMPRS independent utility program from the starter system tape (START1 restore 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 restore the contents of the starter system tape to disk:

```
RSTART1  JOB
         MSG   TODEV=1403,TOADDR=00E
         RESTORE FROMDEV=2400,FROMADDR=182,               X
               TODEV=3330,TOADDR=150,VOLID=START1
         END
```

2. Define the control statement input device by pressing the REQUEST key of the console keyboard. Message IBC105A 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 or unit address of your input device, enter the correct information.

3. When the contents of the starter system tape for START1 have been restored to disk, message IBC163A END OF JOB is printed on the message output device. Place the cards for job RSPOOL0 in the input device and repeat the above process starting at No. 1 to restore the SPOOL0 volume.

```
RSPOOL0  JOB
         MSG   TODEV=1403,TOADDR=00E
         RESTORE FROMDEV=2400,FROMADDR=182,               X
               TODEV=3330,TOADDR=151,VOLID=SPOOL0
         END
```

Start the Starter System

Making the starter system operative includes initial program loading (IPL), initializing the nucleus (NIP), readying the scheduler, and starting JES2. The following procedures can be used to make the starter system operative.

1. Set the load selector switches on the control panel to the unit address of the START1 volume that contains the starter system. Then press the LOAD key.

2. Signal EOB to message IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zzz.

3. Enter REPLY XX,COLD to JES2 parameters (for the first IPL after SPOOL0 has been restored). For each IPL following that, enter REPLY XX,U.

4. For message $HASP400 ENTER REQUESTS, press the REQUEST key and enter $S.

Initialize the Volume That is to Contain the Distribution Libraries

1. The volume to be initialized must first be varied offline. Then place the following IEHDASDR control statements in the input device. After IEHDASDR has been executed, the initialized volume must be varied online.

```
  //INIT    JOB ACCT123,PROGRAMMER,MSGLEVEL=(1,1)
  //DLIB01 EXEC PGM=IEHDASDR
  //SYSPRINT DD SYSOUT=A
  //SYSIN    DD *
     ANALYZE TODD=(152),VTOC=2,EXTENT=10,            X
             NEWVOLID=DLIB01,OWNERID=ID
/*
```

Load the Contents of the Distribution Tape to a Direct-Access Volume

1. Mount the DTR containing the cataloged procedures and the SMP load modules. Using IEBCOPY, load the cataloged procedures (member names DLBDELTE, DLBALLOC, and DLBPROC) into the starter system's PROCLIB, and load the SMP Release 4 load module (member names HMASMP and HMASMUXD) into an authorized library in the starter system.

2. If the DLIBs have already been cataloged from a previous system generation, issue the START DLBDELTE command to uncatalog them. Issue the START DLBALLOC command to allocate and catalog the DLIBs.

3. Mount the distribution tape containing the DLIBs, and execute SMP RECEIVE,ACCEPT,NOAPPLY (specifying as a joblib the authorized library that contains the SMP Release 4 load module) against file 1 of the distribution tape.

   Note: The SMP procedure (DLBPROC), which is provided in the DTR, processes all the component groups (system subsets) on the distribution tape in one SMP job. If you wish, the component groups may be processed individually.

   At the end of this step, the distribution libraries are loaded and ready for subsequent system generation processing.

Punch the Utility Programs and IPL Text

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

   Use the IEBPTPCH utility program to punch the IBCDMPRS, IBCDASDI, and ICAPRTBL independent utility programs and the IPL text (IEAIPL00). Place the following IEBPTPCH control cards in the input device.

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

Initialize the Volume That Will Contain the New MVS System

1. Mount the disk volume that is to contain the MVS system that is to be generated.

2. The volume to be initialized must be varied offline. Place the following IEHDASDR control statements in the input device. 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.

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

After you have selected and specified the system generation macro instructions, selected and specified the system data sets, and specified the control statements for initializing the required volumes, dumping the contents of the starter system tape to a direct-access volume, and copying the distribution library tape to a direct-access volume, you are ready to specify the job control language required to execute Stage I. Refer to the "Stage I Input" section of "System Control Program Installation" for this information. After you have specified the Stage I job control language, you are ready to begin the actual processing. Refer to the beginning of this section 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 MVS System Control Program" in this chapter before executing Stage I.

PREPARING FOR A SYSTEM GENERATION USING AN EXISTING MVS SYSTEM AS THE GENERATING SYSTEM

After your first system generation using the starter system, your existing MVS system control program can be used as the generating system for subsequent system generations. When you use an existing MVS system as the generating system, it is recommended you do the following:

- Use your existing system's master catalog, page data sets, and SYS1.STGINDEX.

- Separate your nonVSAM system data sets on volumes that do not contain the master catalog, page data sets, and SYS1.STGINDEX.

- Use a DATASET macro without a SPACE parameter for the master catalog, page data sets, and SYS1.STGINDEX.

This section lists the preparatory steps that must be done when an MVS system, other than the starter system, is used as the generating system. The steps for readying the distribution libraries are similar to those that must be performed when the starter system is used; the control statements for the utility programs and the operator commands are the same.

The procedures to make the distribution libraries operational are:

1. Initialize the volume that is to contain the distribution libraries, as described in Step 1 of the preceding section "Initialize the Volume that is to Contain the Distribution Libraries."

2. Mount the DTR and load the cataloged procedures (member names DLBDELTE, DLBALLOC, and DLBPROC) into SYS1.PROCLIB of your generating system and load the SMP Release 4 load modules (member name HMASMP and HMASMUXD) into an authorized library of your generating system.

   Note: If the generating system is a Release 3.8 level MVS system, the following step is not required. The cataloged procedures and SMP Release 4 have been included during the generation of the Release 3.8 system.

3. If the DLIBs have already been cataloged from a previous system generation, issue the START DLBDELTE command to uncatalog the distribution libraries. Issue the START DLBALLOC command to allocate and catalog the DLIBs.

4. Mount the distribution tape containing the DLIBs, and execute SMP as described in Step 3 of the preceding section "Load the Contents of the Distribution Tape to a Direct-access Volume." If the generating system is Release 3.7 or older, specify as a joblib the authorized library that contains the SMP Release 4 load module.

5. Punch the utility programs and the IPL text using the instructions provided in step 1 of the preceding section "Punch the Utility Programs and IPL Text."

6. Mount the volume that is to contain the new MVS system.

7. Initialize the volume that is to contain the new MVS system. The volume to be initialized must first be varied offline. Then place the following IEHDASDR control statements in the input device and enter a START RDR, <u>00C</u> command. After IEHDASDR has been executed, the initialized volume must be varied online.

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

Once you have completed the above procedures, you are ready to specify the job control language for Stage I and begin system generation. Refer to the "Stage I Input" section of "System Control Program Installation" for this information. If you plan to include your own routines in the new system during system generation, refer to "Adding User-Written Routines to the MVS System Control Program" in this chapter before executing Stage I.

ADDING USER-WRITTEN ROUTINES TO THE MVS SYSTEM CONTROL PROGRAM

During system generation, you can include your own macros, parameters, procedures, appendages, and routines in the system data sets.

The components to be included must be members of cataloged partitioned data sets. Except for some routines that are to go into SYS1.NUCLEUS, executable code must be in load module form. That is, each component must have been compiled and link-edited.

You specify the name of the partitioned data set and the names of the members to be included in the DATASET macro that defines the system data set in which the user-written components are to be included. During system generation, the user-written components are copied or link-edited into the designated system data set.

For SYS1.NUCLEUS, you may specify the name of the partitioned data set in the DATASET macro, and optionally, the names of members. If only the data set name is specified, all of its members are copied into SYS1.NUCLEUS as individual members. If member names are also given on the DATASET macro, each named member will be link-edited into the nucleus being generated (IEANUC01). In the first case, the members must be load modules; in the second case, they may be load modules or object modules.

If user-written SVC routines are to be included in SYS1.LPALIB or SYS1.NUCLEUS, an SVCTABLE macro must also be specified. This macro adds an entry to the SVC table that specifies the characteristics of each SVC routine.

The system data sets into which user-written components can be included are listed in Figure 11 in "Specifying the MVS System Control Program." Also, in this chapter, in the discussion about the DATASET macro, are the parameters you specify to include the user-written components.

Figure 33 is an example of adding a user-written routine to the system control program to be generated. In this example, IGC255, a type 1 user-written SVC routine, is to be included in SYS1.NUCLEUS.

In the first step, a 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 has been cataloged in the system catalog of the generating system.

To include the SVC routine in SYS1.NUCLEUS, the following parameters are specified in the DATASET macro, in addition to the parameters that define SYS1.NUCLEUS:

```
  DATASET  NUCLEUS,PDS=SYS1.USERLIB,MEMBERS=(IGC255)
```

Since this is an SVC routine, an SVCTABLE macro must be specified in the following format:

```
  SVCTABLE  SVC-255-T1
```

For a description of the control statements required by the assembler and linkage-editor, refer to the OS/VS-VM/370 Assembler Programmer's Guide.

```
//USERSVC   JOB  MSGLEVEL=1
//STEP1    EXEC  PGM=ASMBLR,PARM='NODECK,OBJECT'
//SYSPRINT   DD  SYSOUT=A
//SYSUT1     DD  UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSUT2     DD  UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSUT3     DD  UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSLIB     DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSGO      DD  DSN=&&LOADSET,UNIT=SYSDA,DISP=(,PASS),
//               SPACE=(CYL,(5,1,1)),DCB=BLKSIZE=400
//SYSIN      DD  *
IGC255       CSECT
        SOURCE STATEMENTS
             .
             .
             .
             END
/*
//STEP2    EXEC  PGM=AEWL,PARM='LIST,NCAL,XREF,RENT'
//SYSPRINT   DD  SYSOUT=A
//SYSUT1     DD  SPACE=(1024,(200,20)),UNIT=SYSDA
//SYSLMOD    DD  DSN=SYS1.USERLIB(IGC255),DISP=(,CATLG),
//               VOL=SER=PACK01,SPACE=(CYL,(2,1,10)),
//               DCB=(RECFM=U,BLKSIZE=7294),UNIT=SYSDA
//SYSLIN     DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
/*
```

Figure 33. Adding a user-written routine to the system

Figure 34 is another example of adding a user-written routine to the system control program to be generated. 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. These constants identify nucleus locations in SYSABEND, virtual-image, and stand-alone dumps.

During the first step, a CSECT is assembled and placed in a temporary data set, &LOADSET. The CSECT is link-edited in the second step and the resulting module becomes member IEAXYZ1 of SYS1.USER. SYS1.USER is a partitioned data set residing on volume PACK01 and cataloged in the generating system.

To include this routine in SYS1.NUCLEUS, the following parameters are specified, in the DATASET macro, in addition to the parameters that define SYS1.NUCLEUS:

```
  DATASET  NUCLEUS,PDS=SYS1.USER,MEMBERS=(IEAXYZ1)
```

```
//USER    JOB MSGLEVEL=1
//STEP1  EXEC PGM=ASMBLR,PARM='NODECK,OBJECT'
//SYSLIB   DD DSNAME=SYS1.MACLIB,DISP=OLD
//SYSUT1   DD UNIT=SYSSQ,SPACE=(1700,(400,50))
//SYSUT2   DD UNIT=SYSSQ,SPACE=(1700,(400,50))
//SYSUT3   DD UNIT=(SYSSQ,SEP=(SYSUT1,SYSUT2,SYSLIB)),
//            SPACE=(1700,(400,50))
//SYSPRINT DD SYSOUT=A
//SYSGO    DD DSNAME=&LOADSET,UNIT=SYSSQ,SPACE=(80,(200,50)),
//            DISP=(MOD,PASS)
//SYSIN    DD *
IEAXYZ1   CSECT
           DC C'XXXXXXXXXX-NUCLEUS ID CSECT-XXXXXXXXXX'
           DC C'SYSTEM CONTROL PROGRAM GENERATED 7/1/74'
           DC C'OWNER--DEPT. D58'
           DC C'NUCLEUS--01'
           DC C'XXXXXXXXXX-END ID CSECT-XXXXXXXXXX'
          END
/*
//STEP2  EXEC PGM=AEWL,PARM=(XREF,LIST,NCAL)
//SYSLIN   DD DSNAME=&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD  DD DSNAME=SYS1.USER(IEZXYZ1),UNIT=2314,DISP=(,CATLG),
//            VOLUME=(,RETAIN,SER=PACK01),SPACE=(1024,(50,20,5))
//SYSUT1   DD UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),
//            SPACE=(1024,(50,20,5))
//SYSPRINT DD SYSOUT=A
/*
```

Figure 34. 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 as output. The second section presents an explanation of Stage II and discusses Stage II input, Stage II processing, and the output from Stage II, which is the system control program. Also included are procedures that you must perform during Stage II.

STAGE I: PRODUCING THE JOB STREAM

This section discusses Stage I input, processing, and output.

STAGE I INPUT

The input deck required for Stage I consists of job control language and system generation macro instructions. The sequence of the cards in the deck is shown in Figure 35.

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

- A JOB statement with any parameters required by your installation. If the system generation process immediately follows the defining of system data sets using the Access Method Services, this card is not to be specified.
- An EXEC statement with PGM=ASMBLR.
- A SYSLIB DD statement that allocates the SYS1.AGENLIB and SYS1.AMODGEN macro libraries to this job.
- Three DD statements named SYSUT1, SYSUT2, and SYSUT3 that are used to allocate space to the three temporary data sets required by the assembler for Stage I. (Refer to Figure 36 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 class.
- A SYSIN DD * statement.
- The system generation macro instructions.
- An END card.
- A /* card.

Figure 36 lists the values to be specified in the SPACE parameter to allocate space for the SYSUT1, SYSUT2, and SYSUT3 temporary data sets. The allocations are listed according to the type of direct-access device on which they may reside and are given in primary and secondary cylinder allocations. 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 (refer to OS/VS2 MVS Utilities for the control statements for using IEHLIST).

```
//SYSGEN    JOB MSGLEVEL=1
//START   EXEC PGM=ASMBLR
//SYSLIB     DD DSNAME=SYS1.AGENLIB,DISP=SHR
//           DD DSNAME=SYS1.AMODGEN,DISP=SHR
//SYSUT1     DD UNIT=SYSDA,SPACE=(allocation)
//SYSUT2     DD UNIT=SYSDA,SPACE=(allocation)
//SYSUT3     DD UNIT=SYSDA,SPACE=(allocation)
//SYSPUNCH   DD UNIT=unitadr,LABEL=(,NL)
//SYSPRINT   DD SYSOUT=A
//SYSIN      DD *
       system generation macro instructions
           END
/*
```

Figure 35. Input deck for Stage I

Device Type	**Space Allocation**		
	SYSUT1	SYSUT2	SYSUT3
2305, 3330, 3330-1, 3350, 3380	3,2	5,2	8,2
2314, 3340/3344	15,2	10,2	16,2

Figure 36. Space allocation (in cylinders) for the temporary data sets

STAGE I EXECUTION

The volumes containing the starter system and the distribution libraries must be mounted.

The system generation macro instructions are executed as one assembler job. During Stage I, the macro instructions are assembled and analyzed for valid parameters and dependencies upon other macro instructions. If no errors are found, a job stream consisting of job control language and control statements is produced to be used during Stage II to select and process components from the distribution libraries and from your own data sets. If errors are found, error messages are written and the job stream is not produced unless the error is identified after the call to the inner checking macros called by the GENERATE macro. If there are errors and the job stream has been produced, the errors are identified by the last record having // in columns 1 and 2. If there are no errors, the last record is /*. Under all circumstances you must check the return code prior to using the Stage I output as Stage II input.

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

The use of the TIME parameter may be necessary when executing Stage I.

STAGE I OUTPUT

If invalid macro instructions are found during the execution of Stage I, error messages are printed (see Appendix F) and the job stream is not produced. If there are no errors, (return code 0) the job stream is produced as punched cards, 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 statement. 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.

STAGE II: PROCESSING THE JOB STREAM

When you have completed Stage I and produced a job stream, you are ready to begin Stage II. This section discusses Stage II input, processing, and output.

STAGE II INPUT

The input to Stage II consists of the job stream that is the output from Stage I. It may also consist of the job control language required to define the three object module utility data sets that are required for Stage II.

Defining the Three Utility Data Sets

The three object module utility data sets must be defined before Stage II begins.

The names you specify for these data sets must be the same as the general name specified in the OBJPDS parameter in the GENERATE macro, plus the numbers 01, 02, or 03 added to the general name to give each data set a unique name.

Figure 37 shows the job control language that can be used to define the utility data sets. Figure 38 shows the values to be specified in the SPACE parameter to define the object module utility data sets. The allocations are listed, in terms of the number of primary and secondary cylinder allocations and directory blocks required, according to the type of direct-access device on which they may reside. To determine if there is enough space available on a direct-access volume, list its VTOC before Stage II using the IEHLIST utility program (refer to <u>OS/VS2 MVS Utilities</u> for the control statements for using IEHLIST).

```
//OBJPDS   JOB MSGLEVEL=1
//START  EXEC PGM=IEFBR14
//OBJPDS1  DD DSNAME=SYS1.yOBJPDSy01,SPACE=(allocation),
//            DISP=(,CATLG),UNIT=unitadr,VOLUME=SER=volser
//OBJPDS2  DD DSNAME=SYS1.yOBJPDSy02,SPACE=(allocation),
//            DISP=(,CATLG),UNIT=unitadr,VOLUME=SER=volser
//OBJPDS3  DD DSNAME=SYS1.yOBJPDSy03,SPACE=(allocation),
//            DISP=(,CATLG),UNIT=unitadr,VOLUME=SER=volser
/*
```

Figure 37. Defining the object module utility data sets: The underlined values must agree with the general name specified in the GENERATE macro. To this name is appended the numbers 01, 02, and 03.

Device Type	Space Allocation (in cylinders)		
	OBJPDS01	OBJPDS02	OBJPDS03
2305, 3330, 3330-1, 3350, 3380	2,1,15	2,1,15	2,1,15
2314, 3340/3344	4,1,15	4,1,15	4,1,15

Figure 38. Space allocation for the object module utility data sets

The Job Stream

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, utility programs, and the Access Method Services.

The JOB Statement: The format of the Stage II JOB statements that are produced by the system generation process, except for job 6, is:

```
//SYSGENnn  JOB 1,'SYSTEM GENERATION',MSGLEVEL=1,
//          CLASS=&SGCTRLC(42),MSGCLASS=&SGCTRLC(41)
```

The *nn* represents the sequential identification numbers supplied by the system generation process to indicate the number of the job (1 through 6 for a complete system generation, 1 through 4 for an I/O device generation). If the default value (SYS1) is used for the INDEX parameter in the GENERATE macro, SYSGEN is used in the jobname field. If a value other than "SYS1" is specified as the index value, that value is substituted for "SYSGEN." The value specified in the JCLASS parameter in the GENERATE macro is substituted for the global &SGCTRLC(42) for the job class and the value specified in the OCLASS parameter in the GENERATE macro is substituted for the global &SGCTRLC(41) for the output class for system messages. For job 6, in addition to the parameters described above, the JOB statement has the TYPRUN=HOLD parameter to enable job 6 to be held until you release it. (Refer to "Initializing the New Master Catalog" in this chapter for this information.)

You may choose to use your own JOB statements by changing the jobname of 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/VS2 MVS 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 one through six 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 will produce unique jobnames. (For information on coding a JOB statement, refer to *OS/VS2 JCL*.)

Figure 39 is an example of creating a unique JOB statement. In this example, the jobname will be "SYS#10*nn*." The job class will be "C," and the output class for the system messages will be "C." The TYPRUN=HOLD parameter will be present on the last job card, as required when more than one initiator is used to process the several jobs.

```
Col
1 3     9
//JOBA    JOB (78700,J22),NAME,MSGLEVEL=(1,1)
//UPDTE  EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=SYS1.AGENLIB,DISP=OLD
//SYSUT2   DD DSN=SYS1.AGENLIB,DISP=OLD
//SYSIN    DD *
./       REPL NAME=JOBCARD,LIST=ALL
./     NUMBER NEW1=10000,INCR=10000
        MACRO
      JOBCARD
         COPY SGGBLPAK
  PUNCH '//SYS#10&SGCTRLA(3) JOB (78700,J22),NAME,    X
              MSGLEVEL=1,'
          AIF (&SGHOLDB).HOLD
.*  THIS CARD IS USED FOR JOBS WITHOUT WAIT
        PUNCH '// CLASS=C,MSGCLASS=C'
          AGO .MEND
.*  THIS CARD IS USED FOR JOBS WHICH MUST WAIT
.HOLD   PUNCH '// CLASS=C,MSGCLASS=C,TYPRUN=HOLD'
.MEND    MEND
./      ENDUP
/*
//
```

Figure 39. 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:

```
//namexx EXEC PGM=program[,parameter(s)]

or

//namexx EXEC procname[,parameter(s)]
```

The step name is either SG or an abbreviation of the name of the system data set being created. If the step being executed is called by the GENERATE macro, the step name is an abbreviation of the name of the system data set being created (for example, LNK refers to SYS1.LINKLIB). In job 5, where many system data sets are being created, the step name is MISC. If the step being executed is called by an internal macro, the step name is SG. Both types of step names contain a sequential number (xx) to indicate the order of the step being executed in the job (job steps execute sequentially). The number of job steps in a job may vary according to what you specified in the macro instructions.

The names of the programs that are executed during system generation are: IDCAMS, IEBCOPY, IEBGENER, IEBUPDTE, IEHLIST, IFCDIP00, and IFCEREP0. (IDCAMS is described in OS/VS2 Access Method Services. IFCDIP00 and IFCEREP0 are described in OS/VS2 System Programming Library: SYS1.LOGREC Error Recording. The remaining programs are described in OS/VS2 MVS Utilities.)

The names of the cataloged procedures that are executed are ASMS and LINKS for assemblies and link-edits, respectively.

STAGE II EXECUTION

Execution of a complete system generation requires a job stream of seven jobs, six jobs if the master catalog has been previously defined. For an I/O device generation, job stream execution occurs in five jobs. The sequence of execution is the same as for a complete system generation.

During execution of the job stream for a complete system generation, each system data set and page data set (if specified) is created within a single job. During the process of creating a system data set or page data set, some or all of the following occur:

- The system or page data set is defined (if a DATASET macro was specified).

- Selected modules are assembled.

- Selected modules are link-edited to construct members of the system data sets.

- Utility programs and service aids complete the construction of the system data sets.

Processing the Job Stream for a Complete System Generation

For a complete system generation, processing the job stream consists of the following seven jobs. The first job, job 0, is executed only if the master catalog has not been previously defined. If the DATASET macro is not used to define the system and/or page data sets, space would not be allocated for system or page data sets during Stage II.

Job 0: During job 0, the master catalog is defined. This job is included in the input stream only if the master catalog has not been defined previously.

Job 1: During job 1, SYS1.LPALIB is created as follows:

- SYS1.LPALIB is defined.

- Macros from SYS1.AMODGEN are assembled, based on your parameter specifications, into the object module utility data set designated as 01 (SYS1.OBJPDS01).

- Components from the distribution libraries and from user-defined data sets are copied into SYS1.LPALIB.

- Components from the distribution libraries and from the 01 utility data set are link-edited into SYS1.LPALIB.

- If an index other than SYS1 was specified in the GENERATE macro, the index is renamed to SYS1.

Job 2: During job 2, SYS1.LINKLIB is created as follows:

- SYS1.LINKLIB is defined.

- Macros from SYS1.AMODGEN are assembled, based on your parameter specifications, and placed in the object module utility data set designated as 02 (SYS1.OBJPDS02).

- Components from the distribution libraries and from user-defined data sets are copied into SYS1.LINKLIB.

- Components from the distribution libraries and from the 2 utility data set are link-edited into SYS1.LINKLIB.

- If an index other than SYS1 was specified in the GENERATE macro, the index is renamed to SYS1.

Job 3: During job 3, SYS1.SVCLIB is created as follows:

- SYS1.SVCLIB is defined.
- Components from the distribution libraries and from user-defined data sets are copied into SYS1.SVCLIB.
- If an index other than SYS1 was specified in the GENERATE macro, the index is renamed to SYS1.

Job 4: During job 4, SYS1.NUCLEUS and SYS1.DCMLIB are created as follows:

- SYS1.NUCLEUS is defined.
- SYS1.DCMLIB is defined.
- Components from the distribution libraries and from user-defined data sets are copied into SYS1.NUCLEUS.
- Components from the distribution libraries, user-defined data sets, and from the 03 utility data set are link-edited into SYS1.NUCLEUS.
- Components from the distribution libraries and from the 3 utility data set are link-edited into SYS1.DCMLIB.
- If an index other than SYS1 was specified in the GENERATE macro, the index of both system data sets is renamed to SYS1.

Job 5: During job 5, the remaining system data sets and the page data sets are created as follows:

- Macros from SYS1.AMODGEN are assembled, based on your parameter specifications, into the object module utility data set designated as 03 (SYS1.OBJPDS03).
- The remaining nonVSAM system data sets are defined (including the TSO system data sets, if they were specified, and SYS1.LOGREC).
- Components from the distribution libraries and from user-defined data sets are copied into specific system data sets.
- Components from the distribution libraries and from the object module utility data set designated as 03 (SYS1.OBJPDS03) are link-edited into specific system data sets.
- The IEBUPDTE utility program adds members to SYS1.PARMLIB. The members that are added were specified in the CTRLPROG, DATASET, and SCHEDULR macros.
- The page data sets and the SYS1.STGINDEX VSAM system data set are defined.
- If an index other than SYS1 was specified in the GENERATE macro, the index of the system data sets is renamed to SYS1.

Job 6: This job must be executed last and must not be executed until jobs 0 through 5 have successfully run to completion. During this job, the catalog that is created during system generation becomes the master catalog of the new system. During job 6, the following occurs:

- The VTOCs of all the volumes containing the MVS system control program are listed.
- The directories of the system data sets are listed.
- The master catalog is listed.

- The pointer in the master catalog of the generating system (the starter system or an existing system) that points to the new system's master catalog is deleted.

- The pointer to the new master catalog is initialized. The volume serial number of the new master catalog for the new system is placed in member SYSCATLG of SYS1.NUCLEUS in the new system, making the new catalog the master catalog.

If, after system generation, you would ever want to change the pointer in SYS1.NUCLEUS that points to the master catalog, you could do it by executing the steps that are in job 6. Refer to "Initializing the New Master Catalog" in this chapter for additional information about job 6.

Processing the Job Stream for an I/O Device Generation

For an I/O device generation, processing the job stream consists of the following four or five jobs.

Job 1: During job 1, SYS1.LPALIB is updated as follows:

- If an index other than SYS1 was specified in the GENERATE macro, the index of SYS1.LPALIB is renamed to that value.

- Macros from SYS1.AMODGEN are assembled, based on your new parameter specifications, into the object module utility data set designated as 01 (SYS1.OBJPDS01).

- Components from the distribution libraries are copied into SYS1.LPALIB.

- Components from the distribution libraries and from the 01 utility data set are link-edited into SYS1.LPALIB.

- If the index was renamed in the first step to other than SYS1, it is again renamed to SYS1.

Job 2: During job 2, SYS1.LINKLIB is updated as follows:

- If an index other than SYS1 was specified in the GENERATE macro, the index of SYS1.LINKLIB is renamed to that value.

- Components from the distribution libraries are copied into SYS1.LINKLIB.

- If the index was renamed in the first step to other than SYS1, it is again renamed to SYS1.

- If TCAM is not in the system and a 2740 has been specified as a console, the SYS1.SVCLIB job is performed.

- If an index other than SYS1 was specified in the GENERATE macro, the index of SYS1.SVCLIB is renamed to that value.

- Components from the distribution libraries are copied into SYS1.SVCLIB.

- If the index was renamed in the first step to other than SYS1, it is again renamed to SYS1.

Job 3: During job 3, a member in SYS1.NUCLEUS is created or updated, and, optionally, SYS1.DCMLIB is updated as follows:

- If an index other than SYS1 was specified in the GENERATE macro, the index of SYS1.NUCLEUS and SYS1.DCMLIB is renamed to that value.

- Macros from SYS1.AMODGEN are assembled, based on your parameter specifications, into the object module utility data set designated as 03 (SYS1.OBJPDS03).

- Components from the distribution libraries and from the 3 utility data set are link-edited into SYS1.NUCLEUS.

- Components from the distribution libraries and from the 3 utility data set are link-edited into SYS1.DCMLIB.

- If the index was renamed in the first step to other than SYS1, it is again renamed to SYS1.

Job 4: SYS1.LOGREC is redefined and SYS1.IMAGELIB is updated as follows:

- If an index other than SYS1 was specified in the GENERATE macro, the index of SYS1.IMAGELIB is renamed to that value.

- The IFCEREP0 service aid is used to scratch SYS1.LOGREC from the system residence volume.

- The IFCDIP00 service aid is used to reallocate it on the system residence volume.

- Components from the distribution libraries are link-edited into SYS1.IMAGELIB.

- If the index was renamed in the first step to other than SYS1, it is again renamed to SYS1.

Job 5: The directories of SYS1.IMAGELIB, SYS1.DCMLIB, SYS1.LINKLIB, SYS1.LPALIB and SYS1.NUCLEUS are listed.

INITIALIZING THE NEW MASTER CATALOG

In a complete system generation, the master catalog is initialized during job 6 of Stage II, as described in the previous section. The JOB card for job 6 contains the TYPRUN=HOLD parameter which keeps job 6 from executing until you release it. To release job 6, enter the following coammand:

```
  $A 'jobname'
```

where jobname is the name of the job, determined by the value specified in the INDEX parameter of the GENERATE Macro. For example, if the default value (SYS1) is used, the command would be: $A 'SYSGEN6'.

IMPORTANT: Do not release job 6 until jobs 0 through 5 have successfully completed. Job 6 initializes the pointer to the new master catalog and disconnects the generating system's master catalog by:

- Deleting the pointer in the generating system's master catalog that points to the new system's master catalog and

- Placing a pointer in SYS1.NUCLEUS in the new system that points to the new system's master catalog.

Some installations will want to utilize the master catalog from a previous production system. "Appendix G. Using the Master Catalog from the Old Production System" contains an example and a procedure for combining the system data sets from a newly generated system with the master catalog, SYS1.STGINDEX, and page data sets from an existing production system.

MULTIPROGRAMMING THE JOB STREAM

The design of Stage II allows the jobs in the job stream to be multiprogrammed.

After the job stream has been read in and enqueued, start from one to three initiators to begin execution. (It is not recommended to start more than three initiators for a single system generation.) The following test assumes three initiators.

It also assumes that the master catalog is being defined using a DATASET macro.

For a complete system generation, job 0 begins execution first. The first step of job 1 completes before the rest of the jobs begin executing. The first step defines the master catalog. Once the master catalog has been created, the remainder of job 1 and jobs 2 and 3 execute simultaneously. Job 3 will complete before jobs 1 and 2 have completed and job 4 will begin. Job 5 remains enqueued on the SYSPUNCH data set until job 4 has completed. When job 4 has completed, job 5 begins. Job 6 will not begin execution until you release it from the queue (see "Initializing the New Master Catalog" in this chapter).

For an I/O device generation, jobs 1, 2, and 3 will begin execution and job 4 will begin execution when one of the three jobs has completed. Job 5 must be released after the first 4 jobs complete (same as Job 6 in a complete system generation).

STAGE II OUTPUT

The output from Stage II is an MVS system control program. Program products are neither distributed nor installed 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, the utility programs, and the Access Method Services.

When you have completed system generation and have installed the JES2 job entry subsystem, you are ready to test the newly installed MVS system control program. It can be tested by using the installation verification procedure (IVP). For the information necessary to use IVP, refer to "Testing the System Control Program."

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 space for 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 F) printed on the SYSPRINT data set. If any errors are found during Stage I, the job stream is not produced (for exceptions, see "Stage I Execution").

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:

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

The most probable error during Stage II is faulty space allocation for a system data set. If this occurs, it is recommended that you reallocate space for that system data set, increasing its size, and reexecute that job starting at the beginning of the job. Note that if job 0 was successful (the new master catalog was defined) before stage II failed, you must reinitialize the target volume and delete the new master catalog name from the starter system's master catalog before you restart stage II from the beginning. The IEHDASDR utility can be used to reinitialize the volume that is to contain the new master catalog. The Access Method Services EXPORT command can be used as shown in the following example to delete the new master catalog's name from the starter system's catalog.

```
//DELCATNM     JOB ACCT123,PROGRAMMER,MSGLEVEL=(1,1)
//STEP1        EXEC PGM=IDCAMS
//SYSPRINT     DD  SYSOUT=A
//SYSIN        DD  *
    EXPORT -
         SYS1.A.NEW.MASTER.CATALOG -
         DISCONNECT
/*
```

If the system data set contains data, you can save the data and restart Stage II from the failing job step by doing the following:

1. Rename the system data set using IEHPROGM.

2. Reallocate space for the system data set using its original name.

3. Copy the data in the renamed data set into the reallocated system data set using IEBCOPY.

4. Delete the renamed data set using IEHPROGM.

5. Restart Stage II from the beginning of the job step that failed. The actual restarting can be accomplished by using one of the methods described in "Restart Techniques."

If the problem encountered is other than space allocation, such as component failures or machine malfunctions, follow the instructions printed out in the error messages or error codes and then refer to OS/VS Message Library: VS2 System Messages or OS/VS Message Library: VS2 System Codes.

For information on the utilities, refer to OS/VS2 MVS Utilities. For information on allocating space for system data sets, refer to "Selecting and Defining the System Data Sets" and to OS/VS2 Access Method Services.

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 job stream can be restarted from the job or job step that failed. This section discusses the techniques used for restarting the job stream after any other necessary operations have been performed.

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 beginning from that job step.

Punching the Job Stream

If the unit (SYSPUNCH) specified during Stage I for the job stream was not a card punch, and you would like the job stream on cards, the IEBEDIT utility can be used to punch all or some of the job stream. Figure 40 shows the statements required to punch the job stream using IEBEDIT.

When you use the IEBEDIT utility program to punch any part or all of the job stream, consider the following:

- The value of the UNIT parameter of the SYSUT1 DD statement is the unit address of the device on which the job stream resides. If the job stream is on disk, the LABEL parameter must specify a standard label.

- The value of the UNIT parameter of the SYSUT2 DD statement is the unit address of the card punch.

```
//PUNCH   JOB ACCOUNT123
//       EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD UNIT=unitadr, LABEL=(,NL),VOLUME=SER=volser,
//            DISP=(OLD,KEEP),DCB=(dcb information)
//SYSUT2   DD UNIT=unitadr
//SYSIN    DD *
         EDIT START=SYSGEN1,STEPNAME=XXX
         EDIT START=SYSGEN2,STEPNAME=XXX
         EDIT START=SYSGEN3,STEPNAME=XXX
         EDIT START=SYSGEN4,STEPNAME=XXX
         EDIT START=SYSGEN5,STEPNAME=XXX
         EDIT START=SYSGEN6,STEPNAME=XXX
/*
```

Figure 40. Punching the job stream

- You can specify one or more EDIT statements when executing IEBEDIT. You can also select to punch certain jobs or job steps. Refer to Figure 41 for the control statements for punching certain jobs or job steps.

Restarting from Tape or a Direct-Access Volume

The IEBEDIT utility can also be used to restart Stage II from any job or job step when the job stream is on tape or a direct-access volume.

The IEBEDIT creates a new job stream by editing and selectively copying the job stream provided as input. The IEBEDIT utility can copy an entire set of jobs, including JOB statements and associated job step statements, or selected job steps in a job. Figure 41 shows the control statements required by IEBEDIT.

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

Figure 41. Control statements for IEBEDIT when the job stream is on tape

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

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

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

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

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

- 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/VS2 MVS Utilities.

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 MVS system control program is operational. It also tests whether the newly created MVS system control program supports your machine configuration.

The jobs in IVP test only the MVS system control program. IVP does not test any program products, programs with service classification "C," or similar programs added after the MVS system control program has been generated. Any program of this type for which verification of 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 MVS machine configurations and requires only the minimum MVS machine configuration.

THE IVP JOB STREAM

The IVP job stream is contained in your MVS system control program in the partitioned data set SYS1.SAMPLIB. Its member name is IVPJOBS. The IVP job stream consists of jobs that:

- Provide device information, such as unit address, device type, device status, volume serial number, volume state, ordered output, and device bypass.
- Execute an assemble/link-edit/go to test the basic functions of the newly installed system control program.
- Provide a listing of SYS1.PARMLIB.
- Force an 806 ABEND and the resulting dump (see OS/VS Message Library: VS2 System Codes).

PROCEDURES FOR USING IVP

To verify that your MVS system control program is correctly installed, carry out the following steps:

- Upon successful completion of system generation and JES2 generation (no unaccounted-for errors are indicated in the Stage II listings), carry out the system initialization procedures.

- Issue a START RDR command to the IVP job stream (IVPJOBS). The following command can be used:

  ```
  START RDR.R,132,VOL=SER=SYSRES,LABEL=(,SL),
  DSNAME=SYS1.SAMPLIB(IVPJOBS)
  ```

- If 132 and SYSRES are not the unit address and volume serial number of the volume on which SYS1.SAMPLIB resides, enter the correct information.

- The JOB card on IVPJOBS is

  ```
  //IVPJOB JOB PS40230301,MSGLEVEL=1,MSGCLASS=A,REGION=256K
  ```

  If this JOB card does not meet the requirements for your installation, modify the JOB card in SYS1.SAMPLIB(IVPJOBS) before issuing the START RDR command.

- Review the I/O device information produced by the job stream to verify the machine configuration.

- Review the listing of SYS1.PARMLIB for completeness.

- Review the results of the assemble/link-edit/go for errors.

- Verify that an 806 ABEND occurred and that a dump was provided.

- Save the IVP output for future reference.

The completion of the above steps and normal end-of-job (EOJ) of the IVP job stream (excluding the 806 ABEND) constitutes successful verification of your new MVS system control program.

VERIFYING NEW DEVICE ALLOCATION TABLES

This chapter discusses the Eligible Device Table verification program, IEFEB400. Included in this chapter are:

- A definition of the verification program
- The JCL needed to execute IEFEB400
- Validity checking performed by the verification program
- Validity checking performed by other components

IEFEB400

The verification program:

- May be executed as part of Stage II or run at a later date.
- Executes in problem program state.
- Must be executed on the system on which the new device allocation tables are to be used.
- Matches device-type definitions in the new tables against those actually existing in the nucleus of the system on which the program is running. (The header of an Eligible Device Table contains the user-id and the date and time of the Stage II assembly to aid in identifying the version of the table being verified.)

SAMPLE JCL FOR THE VERIFICATION PROGRAM

The verification program requires two DD statements, SYSPRINT to define the output data set for messages, and EDT to define the partitioned data set for the device allocation tables.

```
//CHECK         JOB     ...
//CHECK1        EXEC    PGM=IEFEB400
//SYSPRINT      DD      SYSOUT=A
//EDT           DD      DSN=SYS1.MLPALIB,DISP=SHR
//SYSABEND      DD      SYSOUT=A
```

VALIDITY CHECKING BY THE VERIFICATION PROGRAM

Message IEF923I is issued to identify the tables being verified. The identification consists of the user-id and the date and time of the Stage II assembly. If all the device-type definitions match those in the existing nucleus, message IEF926I is issued. However, if either of the two errors shown below is detected, the following occurs:

- If a device type defined in the new tables does not match the corresponding UCB, message IEF924I is issued. A return code of 4 is set for condition code processing.
- If a device type defined in the new tables does not have a corresponding UCB referred to in the IOS look-up table, message IEF925I is issued. A return code of 4 is set for condition code processing.

ADDITIONAL VALIDITY CHECKING

The following possible errors are detected by components other than the verification program.

- If an OPEN error occurs, data management issues an appropriate error message and/or abends the task. A return code of 8 is set for condition code processing.

- If a PUT error occurs while writing messages to SYSPRINT, data management issues an appropriate error message and/or abends the task. A return code of 12 is set for condition code processing.

- If a LOAD error occurs, message IEA703I is issued and the task is abended.

Note: Do not use IEFEB400 to verify TP devices, since for TP devices only the device class needs to be the same for a match to occur. Therefore, the TP device type being verified could appear to have a match when in reality it was merely matched with a TP device of the same class.

SYSTEM GENERATION MACRO INSTRUCTIONS FOR A COMPLETE SYSTEM GENERATION

The following example illustrates a Stage I input deck for a complete system generation. The system data sets and page data sets are defined using the DATASET macro.

```
Col
1              16

CHAN0    CHANNEL    ADDRESS=0,TYPE=MULTIPLEXOR
CHAN1    CHANNEL    ADDRESS=1,TYPE=BLKMPXR
CHAN2    CHANNEL    ADDRESS=2,TYPE=BLKMPXR
CHAN3    CHANNEL    ADDRESS=3,TYPE=BLKMPXR
CHAN4    CHANNEL    ADDRESS=4,TYPE=BLKMPXR
CHAN5    CHANNEL    ADDRESS=5,TYPE=SELECTOR
CHAN6    CHANNEL    ADDRESS=6,TYPE=SELECTOR
***
*
*        CHANNEL  0 DEVICES
*
***
         IODEVICE  UNIT=DUMMY,ADDRESS=(000,2),OFFLINE=YES,                 X
               DEVTYPE=108008FF,ERRTAB=220
         IODEVICE  UNIT=3211,ADDRESS=(002,1)
         IODEVICE  UNIT=3540,ADDRESS=(003,2)
         IODEVICE  UNIT=3036,ADDRESS=006
         IODEVICE  UNIT=7443,ADDRESS=007
         IODEVICE  UNIT=3215,ADDRESS=009
         IODEVICE  UNIT=3505,ADDRESS=00A
         IODEVICE  UNIT=3525,ADDRESS=00B
         IODEVICE  UNIT=2540R,ADDRESS=00C,MODEL=1
         IODEVICE  UNIT=2540P,ADDRESS=00D,MODEL=1
         IODEVICE  UNIT=1403,ADDRESS=(00E,2),FEATURE=UNVCHSET,             X
               OFFLINE=YES,MODEL=N1
         IODEVICE  UNIT=1403,ADDRESS=010,MODEL=N1,                         X
               FEATURE=UNVCHSET
         IODEVICE  UNIT=2740,ADDRESS=011,ADAPTER=IBM1,TCU=2703,            X
               FEATURE=CHECKING,OFFLINE=YES
         IODEVICE  UNIT=3505,ADDRESS=012
         IODEVICE  UNIT=3525,ADDRESS=013
         IODEVICE  UNIT=3277,ADDRESS=014,MODEL=2,                          X
               FEATURE=(OCKY3277,KB78KEY,SELPEN)
         IODEVICE  UNIT=3213,ADDRESS=015
         IODEVICE  UNIT=3277,ADDRESS=016,MODEL=2,                          X
               FEATURE=(OCKY3277,KB78KEY,SELPEN)
         IODEVICE  UNIT=3213,ADDRESS=017
         IODEVICE  UNIT=3066,ADDRESS=(018,2)
         IODEVICE  UNIT=3215,ADDRESS=01F
         IODEVICE  UNIT=2741P,ADDRESS=(020,2),TCU=2703,                    X
               ADAPTER=IBM1,SETADDR=1
         IODEVICE  UNIT=2741P,ADDRESS=(084,2),TCU=2703,                    X
               ADAPTER=IBM1,SETADDR=1,FEATURE=AUTOANSR
         IODEVICE  UNIT=3705,ADDRESS=0FF,                                  X
               ADAPTER=CA1
```

Figure 42 (Part 1 of 5). Macro instructions for a complete system generation

```
Col
1             16

         CHANNEL 0 DEVICES (Cont'd)

         IODEVICE  UNIT=BSC3,ADDRESS=(0F2,9),TCU=2703,                 X
               ADAPTER=BSCA
         IODEVICE  UNIT=BSC1,ADDRESS=(0FB,3),TCU=2703,                 X
               ADAPTER=BSCA
         IODEVICE  UNIT=BSC2,ADDRESS=0FE,ADAPTER=BSCA,                 X
               TCU=2703
***
*        CHANNEL 1 DEVICES
*
***
         IODEVICE  UNIT=3330,ADDRESS=(110,2)
         IODEVICE  UNIT=2314,ADDRESS=(130,6)
         IODEVICE  UNIT=2305,ADDRESS=140,MODEL=2,OFFLINE=YES
         IODEVICE  UNIT=2305,ADDRESS=148,MODEL=2,OFFLINE=YES
         IODEVICE  UNIT=3330,ADDRESS=(150,8)
         IODEVICE  UNIT=2305,ADDRESS=1B0,MODEL=1
         IODEVICE  UNIT=2305,ADDRESS=1B8,MODEL=1
         IODEVICE  UNIT=2305,ADDRESS=1D0,MODEL=2,FEATURE=SHARED
         IODEVICE  UNIT=2305,ADDRESS=1D8,MODEL=2,FEATURE=SHARED
         IODEVICE  UNIT=2305,ADDRESS=1E0,MODEL=2
         IODEVICE  UNIT=2305,ADDRESS=1E8,MODEL=2
         IODEVICE  UNIT=2305,ADDRESS=1F0,MODEL=1,FEATURE=SHARED
         IODEVICE  UNIT=2305,ADDRESS=1F8,MODEL=1,FEATURE=SHARED
***
*
*        CHANNEL 2 DEVICES
***
         IODEVICE  UNIT=DUMMY,ADDRESS=(200,2),OFFLINE=YES,             X
               DEVTYPE=108008FF,ERRTAB=220
         IODEVICE  UNIT=2250,MODEL=1,ADDRESS=206,OFFLINE=YES,          X
               FEATURE=(ALKYB2250,PRGMKYBD,BUFFER8K,LIGHTPEN,CHARGNTR)
         IODEVICE  UNIT=3330,ADDRESS=(210,8)
         IODEVICE  UNIT=3330,ADDRESS=(220,8)
         IODEVICE  UNIT=2314,ADDRESS=(230,8)
         IODEVICE  UNIT=3330,ADDRESS=(250,8)
         IODEVICE  UNIT=2401,ADDRESS=(272,4),MODEL=5,                  X
               FEATURE=(9-TRACK,DUALDENS)
         IODEVICE  UNIT=3420,ADDRESS=(280,4),MODEL=8,                  X
               FEATURE=(9-TRACK,OPT1600),OFFLINE=YES
         IODEVICE  UNIT=3330,ADDRESS=(290,2),MODEL=11
         IODEVICE  UNIT=3340,ADDRESS=(2C0,2)
         IODEVICE  UNIT=2305,ADDRESS=2D0,MODEL=2
         IODEVICE  UNIT=2305,ADDRESS=2D8,MODEL=2
         IODEVICE  UNIT=2305,ADDRESS=2E0,MODEL=2
         IODEVICE  UNIT=2305,ADDRESS=2E8,MODEL=2
***
*
*        CHANNEL 3 DEVICES
*
***
         IODEVICE UNIT=3330,ADDRESS=(310,8),FEATURE=SHARED
         IODEVICE UNIT=2314,ADDRESS=(330,8)
         IODEVICE  UNIT=2314,ADDRESS=(340,8)
         IODEVICE  UNIT=3330,ADDRESS=(350,8)
         IODEVICE  UNIT=3420,ADDRESS=(380,5),MODEL=5,                  X
               FEATURE=(9-TRACK,DUALDENS),OFFLINE=YES
         IODEVICE  UNIT=3330,ADDRESS=(390,2),MODEL=11,OPTCHAN=4
         IODEVICE  UNIT=3340,ADDRESS=(3C0,2),OPTCHAN=4
         IODEVICE  ADDRESS=3F0,ADAPTER=CA2,UNIT=3705
         IODEVICE  ADDRESS=3FF,ADAPTER=CA2,UNIT=3705
```

Figure 42 (Part 2 of 5). Macro instructions for a complete system generation

```
Col
1              16

***
*
*        CHANNEL 4 DEVICES
*
***
         IODEVICE  UNIT=3330,ADDRESS=(410,8)
         IODEVICE  UNIT=2314,ADDRESS=(420,8)
         IODEVICE  UNIT=2314,ADDRESS=(430,8)
         IODEVICE  UNIT=2314,ADDRESS=(440,8)
         IODEVICE  UNIT=3330,ADDRESS=(450,8)
         IODEVICE  UNIT=2401,ADDRESS=(482,3),MODEL=5,                      X
               FEATURE=(9-TRACK,DUALDENS)
***
*
*        CHANNEL 5 DEVICES
*
***
         IODEVICE  UNIT=2314,ADDRESS=(530,8)
         IODEVICE  UNIT=2314,ADDRESS=(540,8)
         IODEVICE  UNIT=3420,ADDRESS=(580,8),MODEL=5,                      X
               FEATURE=(9-TRACK,DUALDENS),OFFLINE=YES
         IODEVICE  UNIT=2314,ADDRESS=(5A0,8),OPTCHAN=6,                    X
               FEATURE=SHARED
         IODEVICE  UNIT=2314,ADDRESS=(5B0,8),OPTCHAN=6,                    X
               FEATURE=SHARED
         IODEVICE  UNIT=2314,ADDRESS=(5E0,8)
***
*
*        CHANNEL 6 DEVICES
*
***
         IODEVICE  UNIT=3036,ADDRESS=606
         IODEVICE  UNIT=7443,ADDRESS=607
         IODEVICE  UNIT=2955,ADDRESS=608
         IODEVICE  UNIT=2314,ADDRESS=(630,8)
         IODEVICE  UNIT=2314,ADDRESS=(640,8)
         IODEVICE  UNIT=2401,ADDRESS=(670,6),MODEL=5,                      X
               FEATURE=(9-TRACK,DUALDENS),OFFLINE=YES
         IODEVICE  UNIT=3420,ADDRESS=(680,4),MODEL=5,                      X
               FEATURE=(9-TRACK,DUALDENS),OFFLINE=YES
         IODEVICE  UNIT=2401,ADDRESS=(690,6),MODEL=5,                      X
               FEATURE=(9-TRACK,DUALDENS),OFFLINE=YES
         IODEVICE  UNIT=2260,ADDRESS=(6C0,2),MODEL=1,                      X
               FEATURE=(ALKYB2260,LINEADDR),OFFLINE=YES
         IODEVICE  UNIT=2260,ADDRESS=(6D0,2),MODEL=1,                      X
               FEATURE=(ALKYB2260,LINEADDR),GCU=2848-3,OFFLINE=YES
         IODEVICE  UNIT=2250,ADDRESS=(6D2,2),MODEL=3,NUMSECT=16,           X
               FEATURE=(ALKYB2250,PRGMKYBD),PCU=1,EXPBFR=8192,OFFLINE=YES
         IODEVICE  UNIT=2314,ADDRESS=(6E0,8)
         IODEVICE  UNIT=3277,ADDRESS=(6F0,2),MODEL=2,OFFLINE=YES,          X
               FEATURE=(ASKY3277,KB78KEY,SELPEN,AUDALRM,DOCHAR)
         IODEVICE  UNIT=3284,ADDRESS=6F2,MODEL=2,FEATURE=DOCHAR,OFFLINE=YES
         IODEVICE  UNIT=3286,ADDRESS=6F3,MODEL=2,FEATURE=DOCHAR,OFFLINE=YES
         UNITNAME  NAME=SYSDA,VIO=YES,UNIT=((110,2),(150,8),(213,5),       X
               (220,8),(250,8),(310,8),(350,8),(410,8),(450,8),(1D0,16),   X
               (1E0,16),(2D0,8),(1B0,16),(1F0,16),(290,2),(2C0,2),         X
               (390,2),(3C0,2))
         UNITNAME  NAME=SYSDA,VIO=YES,UNIT=((130,6),(230,8),(330,8),       X
               (340,8),(430,8),(440,8),(530,8),(540,8),(630,8),(640,8),    X
               (5E0,8),(6E0,8),(5A0,8),(5B0,8),(420,8))
         UNITNAME  NAME=200,UNIT=200,VIO=NO
         UNITNAME  NAME=201,UNIT=201,VIO=NO
         UNITNAME  NAME=000,UNIT=000,VIO=NO
         UNITNAME  NAME=001,UNIT=001,VIO=NO
         UNITNAME  NAME=DISK1,VIO=NO,UNIT=(290,291,390,391)
         UNITNAME  NAME=DISK2,VIO=NO,UNIT=(2C0,2C1,3C0,3C1)
```

Figure 42 (Part 3 of 5). Macro instructions for a complete system generation

```
Col
1              16
         UNITNAME  NAME=DRUM1,VIO=NO,UNIT=((1B0,8),(1B8,8),(1F0,8),(1F8,8)
         UNITNAME  NAME=DRUM2,VIO=NO,UNIT=((1D0,8),(1D8,8),(1E0,8),        X
               (1E8,8),(2D0,8),(2D8,8),(2E0,8),(2E8,8),(140,8),(148,8))
         UNITNAME  NAME=SYSSQ,VIO=NO,UNIT=((150,8),(110,2),(220,8),        X
               (213,5),(250,8),(350,8),(310,8),(410,8),(450,8),(6E0,8),    X
               (540,8),(530,8),(630,8),(430,8),(440,8),(5E0,8),(640,8))
         UNITNAME  NAME=D2314,VIO=NO,UNIT=((430,8),(440,8),(530,8),        X
               (330,8),(340,8),(6E0,8),(130,6),(230,8),(420,8),(540,8),    X
               (5A0,8),(5B0,8),(640,8),(630,8),(5E0,8))
         UNITNAME  NAME=PTTPE,VIO=NO,UNIT=((580,2),(670,2),(680,2),(690,2))
         UNITNAME  NAME=PTDSK,VIO=NO,UNIT=(210,211,212)
         UNITNAME  NAME=PTPUN,VIO=NO,UNIT=(00B,013,00D)
         UNITNAME  NAME=PTRDR,VIO=NO,UNIT=(00A,012,00C)
         UNITNAME  NAME=PTPRT,VIO=NO,UNIT=(002,004,00E)
         UNITNAME  NAME=TAPE1,VIO=NO,UNIT=(280,281,282,283)
         UNITNAME  NAME=TAPE,VIO=NO,UNIT=((580,8),(670,6),(680,4),         X
               (690,6),(380,5),(272,4),(482,3))
CONS     CONSOLE   MCONS=006,ALTCONS=606,ROUTCDE=(1,2,10)
         CONSOLE   SECONS=606,ALTCONS=006,ROUTCDE=(1,2),                   X
               VALDCMD=(1,2,3)
         CONSOLE   SECONS=009,ALTCONS=01F,ROUTCDE=(1,2,3,4,5)
         CONSOLE   SECONS=01F,ALTCONS=009,ROUTCDE=(1,2),                   X
               VALDCMD=(1,2,3)
         CONSOLE   SECONS=016,ALTCONS=014,ROUTCDE=ALL,VALDCMD=(1,2,3)
         CONSOLE   SECONS=014,ALTCONS=016,ROUTCDE=ALL,                     X
               VALDCMD=(1,2,3)
         CONSOLE   SECONS=O-6D0,ROUTCDE=(9,10,11)
         CONSOLE   SECONS=6D3,ROUTCDE=(12),AREA=(10,12)
         CONSOLE   SECONS=O-6F2,ROUTCDE=(13,14,15)
         CONSOLE   SECONS=O-6F3,ROUTCDE=(11)
         CONSOLE   SECONS=O-6C1,USE=SD
         CONSOLE   SECONS=019,ALTCONS=018,ROUTCDE=ALL,                     X
               VALDCMD=(1,2,3)
         CONSOLE   SECONS=O-015,ROUTCDE=(1,2,3,4,7,8,10)
         CONSOLE   SECONS=O-017,ROUTCDE=ALL
         CONSOLE   SECONS=018,ALTCONS=019,ROUTCDE=ALL,                     X
               VALDCMD=(1,2,3)
         CONSOLE   SECONS=206,ROUTCDE=(3,4,5,6,7,8)
         CONSOLE   SECONS=(I-00C,O-00E),VALDCMD=(1,2),ROUTCDE=ALL
         CONSOLE   SECONS=(I-011,O-00F)
CTRL     CTRLPROG  OPTIONS=(DEVSTAT,RDE,RER,BLDL),SQA=1,REAL=4,            X
               TZ=(W,5),ASCII=INCLUDE,STORAGE=2048
SCH      SCHEDULR  BCLMT=20,HARDCPY=(SYSLOG,ALL,CMDS),                     X
               DEVPREF=(3330,2305-2,2305-1,3330-1,2314,2400,3400-5)
CHKPT    CKPTREST  ELIGIBLE=(20,100,101,102,103,110,120,140,160,4092)
DATAMAN  DATAMGT   ACSMETH=(BTAM,TCAM,ISAM),GRAPHICS=(PORRTNS,GSP)
TSO      TSO       LOGTIME=20000
EDIT     EDIT      DSTYPE=SYSTEST,BLOCK=800,FORMAT=FIXED,                  X
               FIXED=(80-80),CONVERT=CAPSONLY
BRODCAST DATASET   BRODCAST,VOL=(SG2001,3330),SPACE=(CYL,(5))
CMDLIB   DATASET   CMDLIB,VOL=(SG2001,3330),SPACE=(CYL,(10,5,100))
DCMLIB   DATASET   DCMLIB,VOL=(SG2001,3330),SPACE=(CYL,(1,,10))
HELP     DATASET   HELP,VOL=(SG2001,3330),SPACE=(CYL,(3,1,6))
```

Figure 42 (Part 4 of 5). Macro instructions for a complete system generation

```
Col
1              16

IMAGELIB DATASET   IMAGELIB,VOL=(SG2001,3330),SPACE=(CYL,(1,,10))
JES3LIB  DATASET   JES3LIB,VOL=(SG001,3330),SPACE=(CYL,(15,2,75))
LINKLIB  DATASET   LINKLIB,VOL=(SG2001,3330),SPACE=(CYL,(23,3,300))
LPALIB   DATASET   LPALIB,VOL=(SG2001,3330),SPACE=(CYL,(28,3,310))
MACLIB   DATASET   MACLIB,VOL=(SG2001,3330),SPACE=(CYL,(40,3,50))
MANX     DATASET   MANX,VOL=(SG2001,3330),SPACE=(CYL,(10))
MANY     DATASET   MANY,VOL=(SG2001,3330),SPACE=(CYL,(10))
NUCLEUS  DATASET   NUCLEUS,VOL=(SG2001,3330),SPACE=(CYL,(15,,10))
PARMLIB  DATASET   PARMLIB,VOL=(SG2001,3330),SPACE=(CYL,(5,,25))
PROCLIB  DATASET   PROCLIB,VOL=(SG2001,3330),SPACE=(CYL,(3,5,200))
STGINDEX DATASET   STGINDEX,VOL=(SG2001,3330),SPACE=(CYL,(6))
SVCLIB   DATASET   SVCLIB,VOL=(SG2001,3330),SPACE=(CYL,(10,5,150))
TELCMLIB DATASET   TELCMLIB,VOL=(SG2001,3330),SPACE=(CYL,(8,3,70))
UADS     DATASET   UADS,VOL=(SG2001,3330),SPACE=(CYL,(8,1,20))
VSCATLG  DATASET   VSCATLG,VOL=(SG2001,3330),SPACE=(CYL,(50,5)),           X
               NAME=SYS1.VSAMCAT
SAMPLIB  DATASET   SAMPLIB,VOL=(SG2001,3330),SPACE=(CYL,(8,1,10))
PAGE1    DATASET   PAGEDSN=SG2PAGE1,VOL=(SG2001,3330),SPACE=(CYL,(100))
PAGE2    DATASET   PAGEDSN=SG2PAGE2,VOL=(SG2001,3330),SPACE=(CYL,(50))
PAGE3    DATASET   PAGEDSN=SG2PAGE3,VOL=(SG2001,3330),SPACE=(CYL,(50))
AFF      AFFINITY  AFF-AFFPGM01-1,AFF-AFFPGM02-0,AFF-AFFPGM03-1,           X
               AFF-AFFPGM04-0,AFF-AFFPGM05-1
GEN      GENERATE  GENTYPE=ALL,OBJPDS=SYS1.OBJPDS,                         X
               RESVOL=(SG2001,3330),INDEX=SG2,JCLASS=K,OCLASS=K
         END
```

Figure 42 (Part 5 of 5). Macro instructions for a complete system 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.

Magnetic Tape Devices

Device Type	Description
2400	2400 9-track magnetic tape drive having an 800 bits-per-inch (density) capability when the dual-density feature is not installed or an 800 and 1600 bits-per-inch (density) capability when the dual-density feature is installed
2400-1	2400 magnetic tape drive with 7-track capability and without data conversion
2400-2	2400 magnetic tape drive with 7-track capability and data conversion
2400-3	2400 or 2420 9-track magnetic tape drive having only a 1600 bits-per-inch (density) capability
2400-4	2400 9-track magnetic tape drive having an 800 and 1600 bits-per-inch (density) capability
2495	2495 tape cartridge reader
3400-2	3420 magnetic tape drive having 7-track capability and data conversion
3400-3	3410 or 3420 9-track magnetic tape drive having 1600 bits-per-inch (density) capability
3400-4	3410 or 3420 9-track magnetic tape drive having 800 and 1600 bits-per-inch (density) capability
3400-5	3420 9-track magnetic tape drive having 6250 bits-per-inch (density) capability
3400-6	3420 9-track magnetic tape drive having 1600 and 6250 bits-per-inch (density) capability

Direct-Access Devices

Device Type	Description
2305-1	2305 fixed-head disk storage (Model 1)
2305-2	2305 fixed-head disk storage (Model 2)
2314	2314/2319 direct-access storage facility
3330	3330 disk storage drive Model 1 and Model 2 or 3333 disk storage and control Model 1
3330-1	3330 disk storage drive Model 11 or 3333 disk storage and control Model 11
3330V	virtual volume for MSS
3340	3340/3344 disk storage drive
3350	3350 direct-access storage

| 3375	3375 direct-access storage
| 3380	3380 direct-access storage

Unit Record Equipment and other Devices

Device Type	Description
1052	1052 printer keyboard (Model 7)
1053	1053 printer (Model 4)
1403	1403 printer or 1404 printer (continuous form only)
1443	1443 printer (Model N1)
2501	2501 card reader
2520	2520 card reader punch
2540-1	2540 card reader punch (read feed)
2540-2	2540 card reader punch (punch feed)
2671	2671 paper tape reader
3066	3066 system console—System/370 Model 165II and Model 168
3158	3158 display console keyboard—System/370 Model 158
3203-4	3203 Model 4 printer
3210	3210 console printer keyboard
3211	3211 printer
3213	3213 printer
3215	3215 console printer keyboard
3505	3505 card reader
3525	3525 card punch with read feature
3540	3540 diskette I/O unit
3800	3800 printing subsystem
3838	Array Processor
3848	Cryptographic Unit
7443	7443 Model 1 Service Record File

Control Units

Device Type	Description
3851	3851 mass storage control for MSS

Graphics Devices

Device Type	Description
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
3036	3036 display console
3277-1	3277 Model 1 display station, or 3278 Model 1 display station
3277-2	3277 Model 2 display station, or 3278 Model 2, 2A, 3, 4 display station, or 3279 Model 2A, 2B, 3A, 3B 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

Device Type	Description
1275	1275 optical reader sorter (available through World Trade branch offices only)
1287	1287 optical reader
1288	1288 optical reader
3886	3886 optical reader
3895	3895 Document Reader/Inscriber

Magnetic Character Reader

Device Type	Description
1419	1419 magnetic character reader
3890	3890 document processor

Audio Response Unit

Device Type	Description
7770	7770 audio response unit

Remote Analysis Unit

Device Type	Description
2955	2955 remote analysis unit

APPENDIX B. DESCRIPTION OF THE STARTER SYSTEM AND DISTRIBUTION TAPES

The starter system and the distribution libraries are distributed on tapes provided by IBM or on tapes that you provide.

The starter system is distributed in a dump/restore format for either 3330/3333 (Model 1 or 11), or 3340 Direct-Access Storage Devices.

The distribution libraries are in SMP Release 4 relfile format. For SMP installation, the relfile tape contains the required MVS system subsets; each includes a number of MVS components. File 1 contains the consolidated SMP control statements for all the subsets.

For Stage I and Stage II, the starter system must be restored to a direct-access volume, and the distribution libraries must be loaded to direct-access volumes via SMP RECEIVE,ACCEPT,NOAPPLY processing.

Figure 43 lists the distribution libraries that are loaded as a result of SMP processing.

SYS1.ACMDLIB
SYS1.AGENLIB
SYS1.AHELP
SYS1.ALPALIB
SYS1.AMACLIB
SYS1.AMODGEN
SYS1.AOSA0
SYS1.AOSA1
SYS1.AOSB0
SYS1.AOSB3
SYS1.AOSBN
SYS1.AOSCA
SYS1.AOSCD
SYS1.AOSCE
SYS1.AOSC2
SYS1.AOSC5
SYS1.AOSC6
SYS1.AOSD0
SYS1.AOSD7
SYS1.AOSD8
SYS1.AOSG0
SYS1.AOSH1
SYS1.AOSH3
SYS1.AOST3
SYS1.AOST4
SYS1.AOSU0
SYS1.AOS00
SYS1.AOS03
SYS1.AOS04
SYS1.AOS05
SYS1.AOS06
SYS1.AOS07
SYS1.AOS10
SYS1.AOS11
SYS1.AOS12
SYS1.AOS20
SYS1.AOS21
SYS1.AOS24
SYS1.AOS25
SYS1.AOS26
SYS1.AOS29
SYS1.AOS32
SYS1.APARMLIB
SYS1.APROCLIB
SYS1.ASAMPLIB
SYS1.ATCAMMAC
SYS1.ATSOMAC
SYS1.AUADS
SYS1.HASPSRC
SYS1.SMPACDS

Figure 43. The distribution libraries

APPENDIX C. INSTALLATION DEVICE PREFERENCE TABLE

The following is a list of device types that may be specified to create a device preference table during system generation (see the discussion about the SCHEDULR macro in the chapter "Specifying the New System Control Program"). If you do not create your own table, then this is the table, based on device performance, that is created for you

Description	Device Type
Direct-Access	2305-1
	2305-2
	3380
	3350
	3375
	3330-1
	3330
	3340
	2314
Tape Drives	2400
	2400-3
	2400-4
	3400-3
	3400-4
	2400-1
	2400-2
	3400-2
	3400-5
	3400-6
Printers	3800
	3211
	3203-4
	1403
	1443
Readers/Punches	2501
	3505
	3525
	2671
	2520
	2540
	2540-2
Printer	1053
Console Printer/Keyboard	3210
	3215
Console Printer	3213
Display Unit	2250-1
	2250-3
Console	1052
Display Station	3277-1
	3277-2
Display Console/Keyboard	3158
Display Console	3036
Printer	3284-1
	3284-2
	3286-1
	3286-2

Console	3066
Display Station	2260-1 2260-2
MICR	3890
OCR	3886
	1287 1288
Service Record File	7443
MICR	1419
Reader/Inscriber	3895
Tape Cartridge Reader	2495
Diskette I/O Unit	3540
Mass Storage Facility	3851
3330 Virtual Volume	3330V
Array Processor	3838
Optical Reader/Sorter	1275
Communications Control Unit	3704 3705 3791L
Cryptographic Unit	3848

APPENDIX D. FUNCTIONALLY EQUIVALENT I/O DEVICES

I/O Device	Equivalent to	Usage
2319	2314	The 2319 is a three-drive disk storage unit that attaches to any System/370. The 2319 drives are functionally equivalent to the 2314 drives. As with the 2314, a maximum of eight drives can be used. To use a 2319, specify UNIT=2314 in an IODEVICE macro. A 2319 can be specified as the unitname of the device. To refer to a 2319 as a 2319, specify UNIT=2314 in an IODEVICE macro and NAME=2319 in a UNITNAME macro.
2402	2401	The IBM 2402 Magnetic Tape Unit is functionally equivalent to the IBM 2401 Magnetic Tape Unit. To use a 2402, specify UNIT=2401 in an IODEVICE macro. A 2402 can be specified as the unitname of the device. To refer to a 2402 as a 2402, specify UNIT=2401 in an IODEVICE macro and NAME=2402 in a UNITNAME macro.
2403	2401	The IBM 2403 Magnetic Tape Unit is functionally equivalent to the IBM 2401 Magnetic Tape Unit. To use a 2403, specify UNIT=2401 in an IODEVICE macro. A 2403 can be specified as the unitname of the device. To refer to a 2403 as a 2403, specify UNIT=2401 in an IODEVICE macro and NAME=2403 in a UNITNAME macro.
3250	2250	The IBM 3250 System is compatible with and functionally equivalent to the IBM 2250 Model 3 attached to a 2840 Model 2. Specify UNIT=2250 in the IODEVICE macro. Refer to *An Introduction to the IBM 3250 Graphic Display System*.
3287 3288 3289	3284 3286	The IBM 3287, 3288, and 3289 are functionally equivalent to the IBM 3284 and 3286. In order to define the IBM 3287, 3288 or 3289, specify UNIT=3284 or UNIT=3286 in the IODEVICE macro.
3333 Model 1	3330 Model 1	The IBM 3333 Disk Storage and Control Model 1 is functionally equivalent to the IBM 3330 Disk Storage Drive Model 1. 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.
3333 Model 11	3330 Model 11	The IBM 3330 Disk Storage and the IBM 3333 Disk Storage and Control Model 11 are functionally equivalent. To use either of these devices, specify UNIT=3330 and MODEL=11 in an IODEVICE macro. For all other parameters requiring a specific device type, specify the devices as 3330-1.

I/O Device	Equivalent to	Usage
3344	3340	The IBM 3344 Direct Access Storage Device is functionally equivalent to the IBM 3340 mounted with a 3348 Model 70. To use a 3344, specify UNIT=3340 in an IODEVICE macro. For more information on compatibility, see <u>3350/3344 Installation and Conversion Guide.</u>
3767 Model 1,2	2740 Model 1	The IBM 3767 Model 1 and 2 Communication Terminal, and the IBM 2740 Model 1, are functionally equivalent. To use these devices, specify UNIT=2740 in an IODEVICE macro. For other parameters or for use as a console, see 2740 under the IODEVICE 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 found.

CSECT IEFEDTTB will not be assembled if any errors were found which caused the QUIT switch to be set.

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
```

If you are using the EDTGEN macro, you may specify DIAGNOS=YES, in addition to or instead of the above statements.

APPENDIX F. 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 44 shows the message code and format of system generation error messages. The messages follow.

IEIaaannn text

Explanation: The error indicated by the message text is a coding error in the system generation macro instruction aaa that you coded. The message serial number nnn identifies the message.

For the CHANNEL and IODEVICE macro instructions, the message text begins with either the name field of the macro instruction or, if the name field was omitted, the sequential identification number provided by the system.

Examples of error messages are:

```
  5,***IEICHA102 CHANNEL2-ADDRESS VALUE NOT SPECIFIED
  5,***IEICHA102 CHAN#2 ADDRESS VALUE NOT SPECIFIED
```

The first example illustrates a message for a CHANNEL macro instruction. "CHANNEL 2" is the name field of the macro instruction. The second example illustrates the same message, but in this case the name field of the macro instruction was omitted and "CHAN#2" was supplied by the macro instruction.

System Action: The assembler program did not produce a job stream in the SYSPUNCH data set. The program analyzed all remaining system generation macro instructions and printed any other required messages. Either message IEIGEN113 or IEIGEN116 was printed, followed by the message GENERATION TERMINATED. Then the system generation process was abnormally terminated.

Severity: 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: 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: 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: 7

User Response: None. This message follows message IEIGEN113 and/or IEIGEN116.

WARNING MESSAGES

Figure 44 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 CTRLPROG MACRO DEFAULTED
```

Explanation: The CTRLPROG macro instruction was not specified and the default options were taken.

Severity: 0

User Response: None.

0, * * * IEIGEN 933 CONSOLE MACRO, CONSOLE ADDRESS xxx DISPLAY AREAS DEFAULTED TO LENGTH 14

Explanation: The dimensions for the status displays for this console were not specified and the default value was taken.

Severity: 0

User Response: None.

INFORMATIVE MESSAGES

***, message text**

Explanation: This type of message documents the options selected for the new system through the system generation macro instructions. All options are described, whether the selection was explicit or implicit.

*****macro 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.

Message Code	Code
IEI	s,* * * IEIaaannn text

s = Severity code:

- 0 Warning message; the condition indicated may cause errors in the new system.
- 5 Error message; user error in coding of a system generation macro instruction.
- 7 Error message; message is produced by the GENERATE macro instruction.

aaa = An abbreviation of the system generation macro instruction at which the error was detected:

aaa	Macro Instruction
AFF	AFFINITY
CHA	CHANNEL
CKP	CKPTREST
CON	CONSOLE
CTR	CTRLPROG
DAT	DATAMGT
DTS	DATASET
EDI	EDIT
EDT	EDTGEN
GEN	GENERATE
IOD	IODEVICE
SCH	SCHEDULR
SVC	SVCTABLE
TSO	TSO
UNI	UNITNAME

nnn = Message serial number
text = Message text

Figure 44. System generation error and warning messages

APPENDIX G. USING THE MASTER CATALOG FROM THE OLD PRODUCTION SYSTEM

Some installations will want to substitute some of the data sets from their old production system for those in the newly generated system. This substitution can make it unnecessary to recatalog the data sets and procedures that are in the old system, but were not cataloged in the new system. The example that follows describes a procedure in six steps for reconfiguring your system, substituting the master catalog, page data sets, and SYS1.STGINDEX data set from the old production system for their equivalent members in the newly generated system. Before beginning a procedure such as this, you should execute the installation verification procedure (IVP) to ensure that the new system was correctly generated (see "Testing the System Control Program"). Figure 45 shows selected data sets from an old production system on four volumes being combined with selected data sets from a newly generated system on three volumes.

This example makes the assumption that none of the volumes on the old production system or on the newly generated system contain both VSAM and nonVSAM data sets. If VSAM and nonVSAM data sets reside on any volume, they should be reallocated, redefined, and recataloged on different volumes before the following procedures are begun.

STEP 1: The SYSCATLG member of SYS1.NUCLEUS of the newly generated system must be updated to contain the volume serial number of the volume that contains the old production system's master catalog, which resides on the 666666 volume. The following procedure deletes the SYSCATLG member (STEPA), then adds the new member which will contain the volume serial number of the volume on which the production master catalog for the old production system resides (STEPB).

```
//STEPA   EXEC PGM=IEHPROGM
//NEWRES DD   DISP=OLD,UNIT=3330,VOL=SER=111111
//SYSPRINT DD SYSOUT=A
//SYSIN   DD   *
   SCRATCH DSNAME=SYS1.NUCLEUS,VOL=3330=111111,MEMBER=SYSCATLG
/*

//STEPB   EXEC PGM=IEBGENER
//SYSIN   DD   DUMMY
//SYSUT2 DD   DSN=SYS1.NUCLEUS(SYSCATLG),DISP=(MOD,KEEP),
//            DCB=(BLKSIZE=13030),VOL=SER=111111,UNIT=3330
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD   *
666666     (See Note)
/*
```

Note: The data card for STEPB has the following format:

Column	**Content**
1–6	volume serial number for the volume containing the catalog
7	unused

8 device type code

device	code in hex	card punch
2305 (Model 1)	06	12-6-9
2305 (Model 2)	07	12-7-9
2314/2319	08	12-8-9
3330/3333(Model 1)	09	12-1-8-9
3330/3333(Model 11)	0D	12-5-8-9
3340	0A	12-2-8-9
3350	0B	12-3-8-9
3375	0C	12-4-8-9
3380	0E	12-6-8-9

Note: Some device codes are not printable characters. Please refer to OS/VS2 System Programming Library: Debugging Handbook UCBTYP field for a list of device type codes.

9–10 unused

11–54 master catalog name

55–80 unused

STEP 2: The volume that contains the master catalog, the page data sets, and the SYS1.STGINDEX data set of the newly generated system (the volume serial number is 555555) is no longer required. It can be demounted or varied off line, if desired.

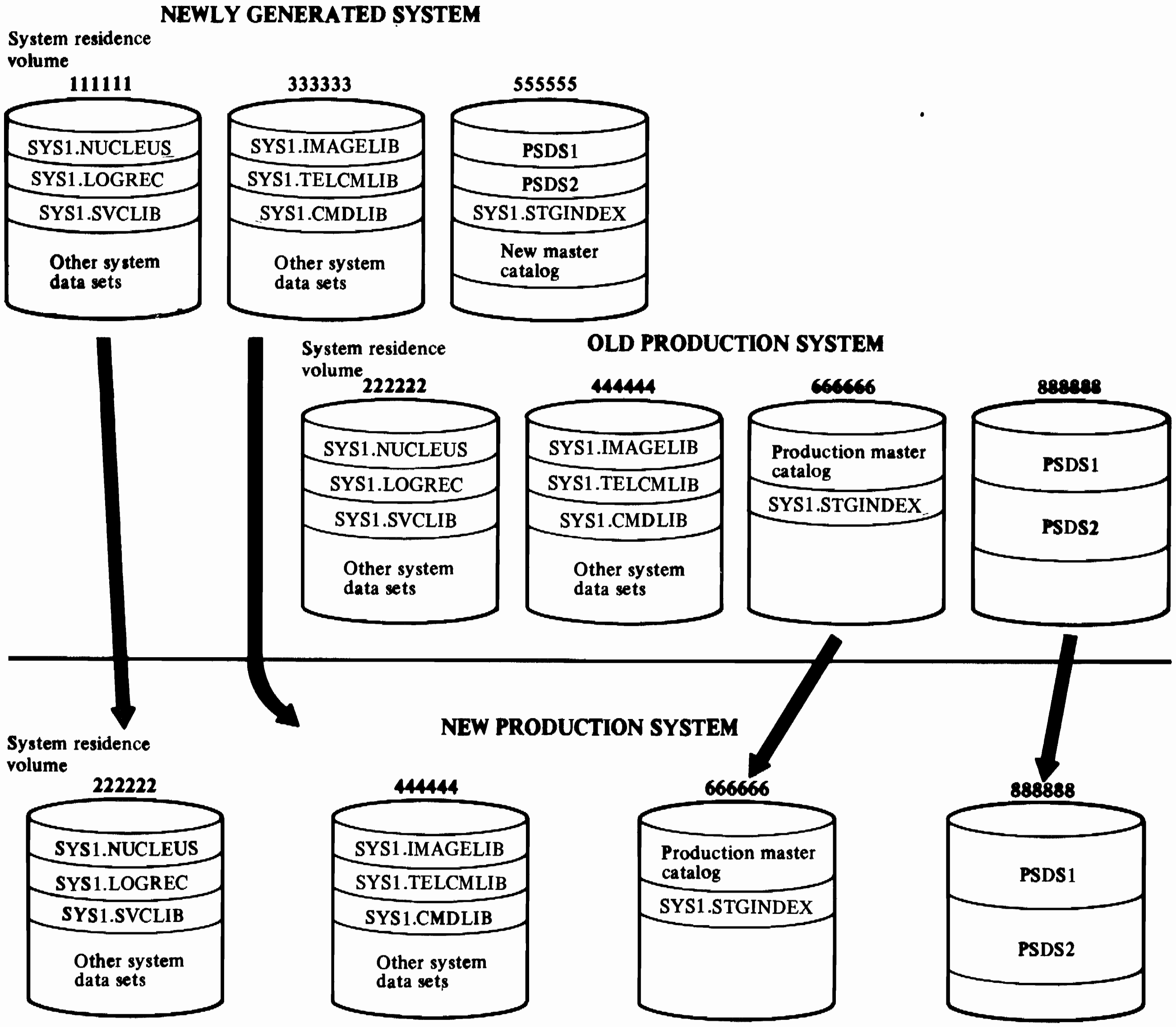

Figure 45. Reconfiguring the new production system

STEP 3: Vary volumes 111111 and 333333 off line. Change the volume serial numbers of these volumes, which contain the newly generated versions of SYS1.NUCLEUS, SYS1.LOGREG SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.TELCMLIB, SYS1.CMDLIB, etc. The volume serial numbers of 111111 and 333333 must be changed to the volume serial numbers of their counterparts on the old production system, 222222 and 444444. The following job control language could be used:

```
//VOLSER EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//SYSIN  DD   *
     LABEL  TODD=(unit addr),NEWVOLID=222222
     LABEL  TODD=(unit addr),NEWVOLID=444444
/*
```

STEP 4: Recatalog any new system libraries. In this example, none are shown; however, if, for example, SYS1.CMDLIB were on the system residence volume of the newly generated system, but on some volume other than the system residence volume in the old production system, it would have to be uncataloged and recataloged using the Access Method Services.

STEP 5: The old production system residence volume and the volume containing SYS1.IMAGELIB, etc., from the old production system (the volume serial numbers are 222222 and 444444) should now be removed from the system.

STEP 6: IPL the integrated system using the new configuration.

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.

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers American National Standards Institute, 1430 Broadway, New York, New York 10018. ANSI definitions are preceded by an asterisk.

ABEND: Abnormal end of task.

Access Method Services: A multifunction service program that defines VSAM data sets and allocates space for them, converts indexed sequential data sets to key-sequenced data sets with indexes, modifies data-set attributes in the catalog, reorganizes data sets, facilitates data portability between operating systems, creates back up copies of data sets and indexes, helps make inaccessible data sets accessible, and lists data-set records and catalog records.

AGENLIB: A distribution library that contains the macro definitions of the system generation macro instructions used during Stage I.

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

AMODGEN: A distribution library that contains the macro definitions of the system generation macro instructions used during Stage II assemblies.

APF: Authorized program facility.

***ASCII (American National Standard Code for:**Information Interchange, X3.4-1968) The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, communications systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

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.

Attached Processing Unit: That part of an Attached Processor System that acts as an additional instruction processor. It increases the instructional processing power without adding main storage, disk storage capacity, or communication channels.

Attached Processor System: A tightly-coupled instruction processing system consisting of a main processing unit combined with an attached processing unit (APU). The main processor provides instruction processing, input/output, and storage functions. The APU has similar instruction processing capability of its own, but shares the input/output and storage facilities of the main processor.

***baud:** A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one-half dot cycle per second in Morse code, one bit per second in a train of binary signals, and one 3-bit value per second in a train of signals each of which can assume one of eight different states.

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.

***bootstrap:** A technique or device designed to bring itself into a desired state by means of its own action, for example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device.

BTAM: Basic telecommunications access method.

burst mode: A means of transferring data as a continuous block to or from a particular I/O device on either the multiplexer or selector channel. All channel controls are monopolized for the duration of data transfer.

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.

***central processing unit:** A unit of a computer that includes the circuits controlling the interpretation and execution of instructions.

***channel:** (1) A path along which signals can be sent, for example, data channel, output channel. (2) The portion of a storage medium that is accessible to a given reading or writing station, for example, track, band. (3) In communication, a means of one-way transmission. Several channels may share common equipment. For example, in frequency multiplexing carrier systems, each channel uses a particular frequency band that is reserved for it.

Channel Set (CS): A collection of channels which can be concurrently addressed by one processor.

Channel Set Switching (CHS): A hardware facility which provides the control program with the ability to connect and disconnect channel sets. This facility allows all channels configured in a multiprocessor or attached processor environment to be accessed from any one processor to prevent disruption of I/O activity in the event of a processor failure.

character arrangement table: A module that establishes the link between the user's data and the printable graphic character. It contains a 256-byte translate table, and identifies up to four character sets and graphic character modification modules to be used.

CHS: Channel set switching.

cold start: The initialization procedure that causes an operating system to commence operation. Synonym for initial program load.

common service area: A part of the common area that contains data areas addressable by all address spaces, but protected during its use by the key of the requester.

complete system generation: The creation of an entirely new System Control Program.

composite console: A console consisting of two different physical devices which are considered as one unit. One device is used for input and the other for output such as a reader and printer.

concatenated data set: 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.

CPU: Central processing unit.

CRH: Channel reconfiguration hardware.

CS: Channel set.

CSA: Common service area.

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.

data space: A storage area defined in the volume table of contents of a direct-access volume for the exclusive use of VSAM to store data sets, indexes, and catalogs.

DCB: Data control block.

DD statement: Data definition statement.

ddname: Data definition name.

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

device type: The general name for a kind of device; for example, 2314, 3330. See also group name, unit address.

direct-access volume initialization: The use of the IBCDASDI or IEHDASDR utility programs for writing a home address, a volume label, and a volume table of contents on a direct-access volume, for checking for defective tracks and assigning alternate tracks for those that are defective, and for writing the IPL program on a new system volume.

display unit: A device which provides a visual representation of data.

distribution libraries: IBM-supplied partitioned data sets on tape containing single VS2 components or combinations of VS2 components that the user restores to disk for subsequent inclusion in a new system.

EBCDIC: Extended binary coded decimal interchange code.

EDT Eligible Device Table.: Used by device allocation.

execute statement (EXEC): A job control statement that marks the beginning of a job step and identifies the program to be executed or the cataloged or in-stream procedure to be used.

existing VS2 system: A system the user has that is used as a generating system to process Stage I and Stage II. It is also a system that the user has and wants to change to include new or additional I/O devices, additional program options, or updates to include data management and user-written routines.

FCB: Forms control buffer.

GMT: Greenwich Mean Time.

GPS: Graphic programming services.

graphic character modification module: A module, constructed, named and stored in SYS1.IMAGELIB. When named in a character arrangement table, the graphic character modification module will be called to substitute those characters in a character set identified by the character arrangement table and the translate table code associated with each graphic character modifier.

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.

GSP: Graphic subroutine package.

***half duplex:** In communications, pertaining to an alternate, one way at a time, independent transmission.

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

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

hard copy log: In systems with multiple console support or a graphic console, a permanent record of system activity.

IBCDASDI: A program that initializes direct-access volumes, and assigns alternate tracks on direct-access storage volumes.

IBCDMPRS: A program that dumps and restores data on direct-access 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.

IEBGENER: A program used to copy a sequential data set or a partitioned member, or to create a partitioned data set from a sequential or partitioned member used as input. IEBGENER can be used to expand an existing partitioned data set by creating partitioned members and merging them into the data set that is to be expanded.

IEBPTPCH: A program that prints or punches all, or selected portions, of a sequential data set. Records can be printed or punched to meet either standard specifications or user specifications.

IEBUPDTE: A program that incorporates source language modifications into sequential data sets or into partitioned data sets.

IEHDASDR: A program that prepares direct-access volumes for use and ensures that any permanent machine errors (i.e., defective tracks) that are found on a direct-access volume do not seriously degrade the performance of those volumes. Also, this program can dump the entire contents or portions of a direct-access volume onto a volume or volumes of the same direct-access device type, onto a magnetic tape volume or volumes, or onto a system output device.

IEHLIST: A program that can be used to list entries in a catalog, entries in the directory of one or more partitioned data sets, and entries in a volume table of contents.

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.

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.

I/O device generation: A type of system generation that can be performed against an existing system to add or delete I/O devices of channels, add universal character set support, change I/O device group names, or change console specifications.

IPL: Initial program loader.

IVP: Installation verification procedure.

JCL: Job control language.

JES2: A functional extension of the HASP II program that receives jobs into the system and processes all output data produced by the job.

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

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.

link pack area (LPA): An area of virtual storage containing reenterable routines that are loaded at IPL and can be used concurrently by all tasks in the system.

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

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

lock: A means of serialization used by supervisory control program routines.

lockout: In multiprocessing, a programming technique used to prevent access to critical data by both central processing units at the same time.

lock/unlock facility: A supervisor facility that controls the execution of instruction strings when a disabled page fault occurs.

LPA: Link pack area.

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 catalog: A key-sequenced VSAM data set that contains data set and volume information required to locate data sets, to allocate and deallocate storage space, to verify the authorization of a program of operator to gain access to a data set, and to accumulate usage statistics for data sets.

master console: In a system with multiple consoles, the basic console used for communication between the operator and the system.

master scheduler: A control program routine that responds to operator commands and initiates the requested action.

MF/1: System activity measurement facility.

MSS: Mass storage system.

multiple console support: An optional feature that permits selective message routine to up to 32 operator's consoles.

multiplexer 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.

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

mutually exclusive parameters: Parameters that cannot be coded on the same job control statement.

MVS: Multiple virtual storage.

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.

NIP: Nucleus initialization program.

nonresident program: A program that is loaded into real storage as needed and can be overlaid after completion.

NRZI: Non-return-to-zero-inverted recording.

nucleus initialization program (NIP): The program that initializes the resident control program; it allows the operator to request last minute changes to certain options specified during system generation.

***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.

***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.

output stream: Diagnostic messages and other output data issued by the operating system or the processing program on output devices especially activated for this purpose by the operator.

page: (1) A fixed-length block of instructions, data or both, that can be transferred between real and external storage. (2) To transfer instructions, data, or both between real storage and external page storage.

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

paging device: A direct-access storage device on which pages (and possibly other data) are stored.

page frame: A block of real storage that can contain a page.

pageable supervisor area: The area of virtual storage containing supervisor routines that can be paged into and out of real storage.

paging: The process of transferring pages between real and external page storage to assist in allocating real storage among concurrently executing program.

paging supervisor: A part of the supervisor that allocates and releases real storage space (page frames) for pages, and initiates page-in and page-out operations.

partitioned data set: A data set in direct-access storage that is divided into partitions, called members, each of which can contain a program or part of a program. Each partitioned data set contains a directory (or index) that the control program can use to locate a program in the library.

PID: Program Information Department.

PLPA: Pageable link pack area.

PWF: Power warning feature.

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.

resident program: Pertaining to a program that is permanently located in storage.

restricted device: A device that is not allocated by system allocation routines when allocation is requested using JCL or dynamic allocation (SVC 99).

RJE: Remote job entry.

RMT generation: Generation of remote work stations for remote job entry.

routing code: A code assigned to an operator message and used to route the message to the proper console.

RTAM: Remote terminal access method.

RTP: Remote terminal processor.

secondary console: In a system with multiple consoles, any console except the master console.

secondary storage: Data storage other than real or virtual storage; for example, storage on magnetic tape or direct-access devices.

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.

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

SQA: System queue area.

SRM: System resources manager.

Stage I: A single assembly of user-supplied macro instructions with output consisting of job control language statements and utility control statements for Stage II assemblies, link-edits, and copies.

Stage II: The execution of job control language statements and utility control statements from Stage I to assemble, link-edit, and copy selected modules, specified by the Stage I macro instructions, into the new system.

starter system: An IBM-supplied VS2 system consisting of a control program that supports any of the central processing units needed to perform a system generation, an assembler and linkage editor for Stage I and Stage II, and utility programs for a new system data set and volume initialization and Stage II processing. It is used when there is not an existing VS2 system with which to generate a new system.

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.

synchronous idle character: A communication control character used by a synchronous data transmission system in the absence of any other character to provide a signal from which synchronism may be achieved or retained between data terminal equipment. Abbreviated SYN.

***system:** An organized collection of men, machines, and methods required to accomplish a set of specific functions.

system control program: A group of programs that (1) accept and schedule jobs in a continuous flow (job management); (2) supervise, on a sequential or priority basis, each unit of work to be done (task management); (3) simplify storage, retrieval, and maintenance of data, regardless of the way it is organized and stored (data management).

system data set: A user-allocated data set on a new system volume.

system generation: Obtaining VS2 components on tape, ordering a starter system if there is no existing system, planning the program options for the new system, specifying macros to include the options in the new system including the options by macro execution, executing the macro expansion to generate the new system, and testing the new system.

system initialization: The process of preparing job management for processing, including such functions as locating, cataloging, and formatting system data sets. System initialization is performed by the master scheduler at IPL.

system log: Data sets on which the communication between problem programs, operators, and the system is recorded.

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

system queue area (SQA): A virtual area reserved for system-related control blocks and tables.

system residence volume: The volume that contains the IPL program, the volume index of the SYSCTLG system data set, and the system data sets SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LOGREC. The system residence volume must reside on the I/O device which is addressed when initial program loading is performed.

TCAM: Telecommunications access method.

TCU: Transmission control unit.

temporary data set: A data set that is created and deleted in the same job.

TOD clock: Time-of-day clock.

transmission control unit (TCU): An input/output control unit that addresses messages to and receives messages from remote terminals.

TSO: Time sharing option.

UCB: Unit control block.

UCS: Universal character set.

uniprocessing: Sequential execution of instructions by a CPU or independent use of a CPU in a multiprocessing system.

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.

VIO: Virtual I/O.

virtual address: An address that refers to virtual storage and must, therefore, be translated into a real storage address when it is used.

virtual I/O: A facility that pages data into and out of external page storage, although to the problem program, the data appears to be read from or written to direct-access storage devices.

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.

virtual storage access method (VSAM): An access method for direct of sequential processing of fixed and variable length record on direct-access devices.

volume table of contents (VTOC): A table on a direct-access volume that describes each data set on the volume.

VSAM: Virtual storage access method.

VTAM: Virtual telecommunications access method.

VTOC: Volume table of contents.

warm start: A restart that allows reuse of previously-initialized input and output work queues.

WCGM: Writable character generation module.

writable character generation module: A 64-position portion of the 3800's character generation storage that holds the scan elements of one character set. There are two WCGMs in the character generation storage of the basic 3800, and optional added storage provides for two more.

writer procedure: A cataloged procedure that controls the output stream writer.

INDEX

GC26-3792-8

OS/VS2 System Programming Library: System Generation Reference (File No. S370-34) Printed in U.S.A. GC26-3792-8

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

OS/VS2 System Programming Library:
System Generation Reference
GC26-3792-8

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

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

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ______________________

Previous TNL ___________________

Previous TNL ___________________

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Reader's Comment Form

Fold and tape

Please do not staple

Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and tape

Please do not staple

Fold and tape

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601