

GT26-3790-1
(Formerly GC26-3790-1)

Systems

**OS/VS System Generation
Introduction**

**VS1 Release 1
VS2 Release 1**

IBM

Second Edition (September 1972)

This edition replaces the previous edition (numbered GC26-3790-0) and makes that edition obsolete.

This edition applies both to Release 1 of OS/VS1 and to Release 1 of OS/VS2, and to all subsequent releases of either system unless otherwise indicated in new editions or technical newsletters. Any information pertaining to DSS is for planning purposes only until DSS is available.

Significant technical and editorial changes made in this edition are outlined in "Summary of Major Changes" following the list of figures. Each technical change is marked by a vertical line to the left of the change.

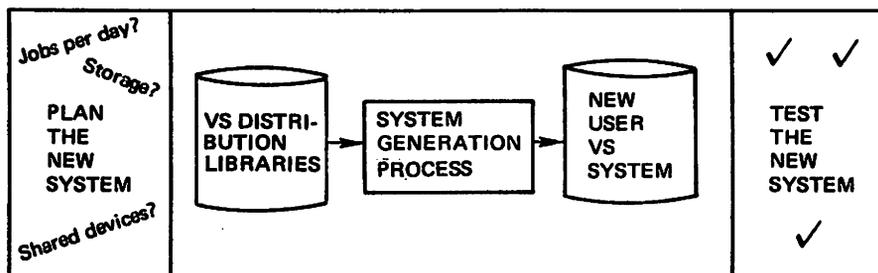
Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest *IBM SRL Newsletter*, GN20-0360, that amends *IBM System/360 and System/370 Bibliography*, GA22-6822, to learn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

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

Forms for readers' comments are provided at the back of this publication. If the forms have been removed, comments may be addressed to IBM Corporation, Programming Publications, Department D78, Monterey and Cottle Roads, San Jose, California 95114. All comments become the property of IBM.

PREFACE

System generation is a tailoring process. Modules are selected from IBM-distributed libraries and joined together forming a new system. The user defines the new system by means of macro specifications to meet his installation's machine and data processing needs. The execution of the macros results in modules being selected from the distribution libraries and combined into user-allocated data sets on new system volume(s). The system generation process, in its broadest sense, consists of defining what an installation's needs are, ordering distribution libraries from IBM, selecting modules from these libraries and placing them by means of macro specification and execution in a new system, and testing the new system.



The System Generation Process

This manual describes what the system generation process is. The book is for the system programmer who is to generate a new operating system with virtual storage (VS) for the first time and for the system planner or manager who needs to know the considerations and requirements for generating a new system. The manual describes what the system programmer needs to consider in selecting his new system, what he must do to select the modules to be in his new system, the requirements in generating a new system, and how a new system is generated. The purpose of this introduction is to provide background information for the more detailed *OS/VS1 System Generation Reference*, GC26-3791, and *OS/VS2 System Generation Reference*, GC26-3792.

PREREQUISITE PUBLICATIONS: The reader must be familiar with OS/VS systems, as described in:

IBM System/370 Summary, GA22-7001
OS/VS1 Planning and Use Guide, GC24-5090
OS/VS2 Planning and Use Guide, GC28-0600

REFERENCE PUBLICATIONS: In using this manual, the reader should refer to:

OS/VS1 Storage Estimates, GC24-5094
OS/VS2 Storage Estimates, GC28-0604
OS/VS JCL Reference, GC28-0618
Operator's Library: OS/VS1 Reference, GC38-0110
Operator's Library: OS/VS2 Reference, GC38-0210
OS/VS Service Aids, GC28-0633
OS/VS Utilities, GC35-0005
OS/VS Message Library: VS1 System Codes, GC38-1003
OS/VS Message Library: VS2 System Codes, GC38-1008

This book is divided into three major sections. The order of the information presented in each section of the manual indicates the order in which a system generation can be performed.

System Generation Description: System Generation, System Generation Requirements, System Control Program Installation, New System Verification, Types of System Generation.

System Generation Process: New System and Program Option Selection, New System Macro Specification, Module Selection and Placement in the New System, New System Volume Initialization, New System Data Set Initialization, Stage I Execution Requirements, Stage II Execution Requirements, Restart Procedures, Installation Verification Procedures, Program Product Installation.

New System Planning Chart: Chart to be used after reading this book for writing down what is to be in the new system.

A *glossary* is provided at the back of the book.

Any information pertaining to DSS is for planning purposes only until DSS is available.

CONTENTS

iii	Preface
vii	Summary of Major Changes
1	System Generation Description
3	System Generation
4	System Generation Requirements
4	Using an Existing VS Operating System
4	Ordering the VS Starter Operating System and/or the VS Modules
6	Preparing for a System Generation
7	System Control Program Installation
7	Stage I
8	Stage II
8	New System Verification
8	Types of System Generation
11	System Generation Process
13	New System and Program Option Selection
13	New System Macro Specification
16	Module Selection and Placement in the New System
16	New System Volume Initialization
18	New System Data Set Initialization
18	Selecting the New System Data Sets
19	Allocating the New System Data Sets
19	Stage I Execution Requirements
19	Stage II Execution Requirements
20	Restart Procedures
20	Stage I
20	Stage II
21	Installation Verification Procedures
21	Program Product Installation
23	New System Planning Chart
25	Glossary
27	Index

SUMMARY OF MAJOR CHANGES

The section on new system macro specifications now includes options that can be specified for VS1 or VS2 system generation.

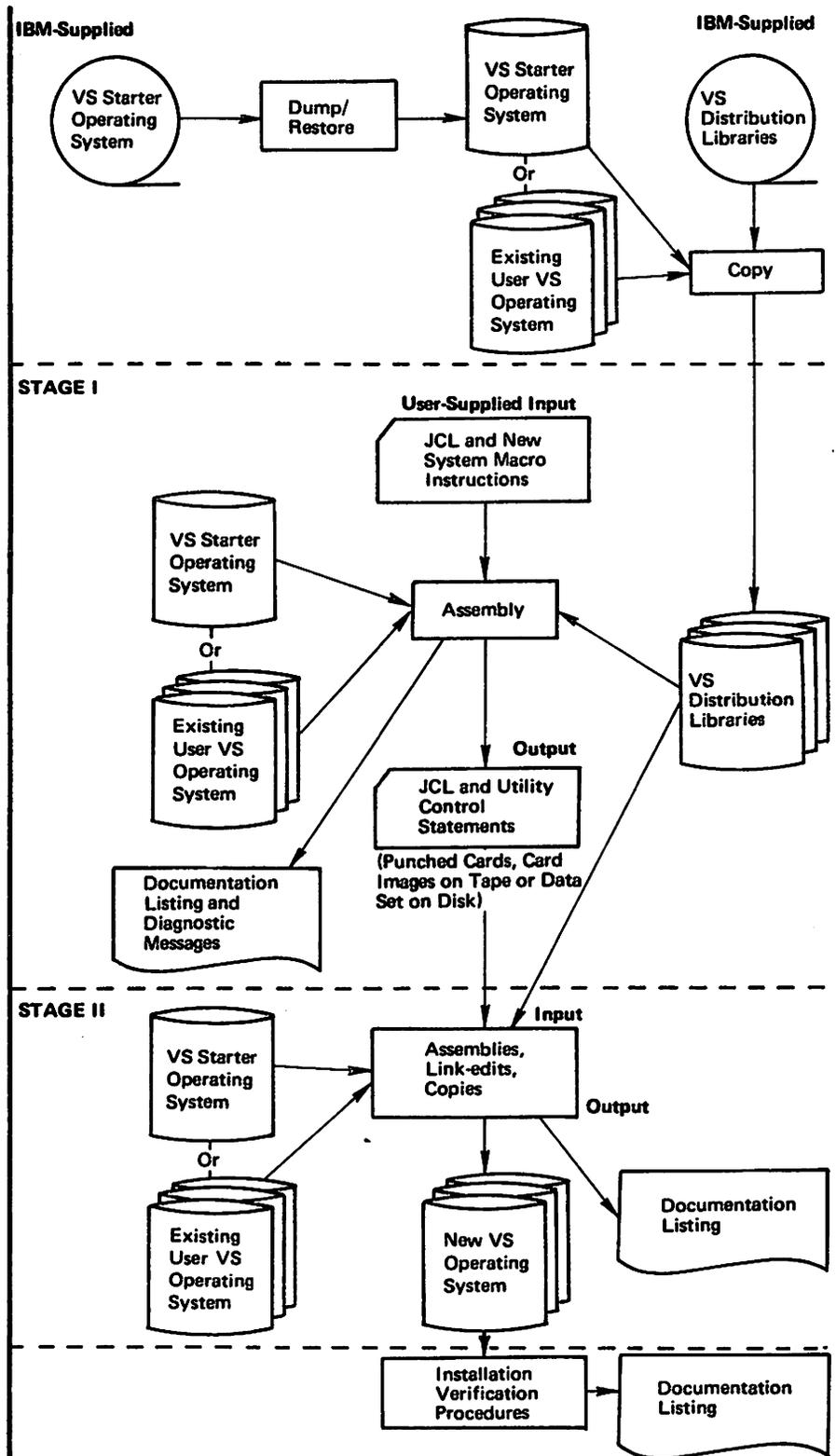
SYSTEM GENERATION DESCRIPTION

PLAN THE NEW SYSTEM
(Consider Installation's Needs and Options)

SELECT AND CODE MACRO INSTRUCTIONS FOR PLANNED OPTIONS

INITIALIZE DISK PACKS AND NEW SYSTEM VOLUME
(Determine Size and Placement of Data Sets)

TEST THE NEW SYSTEM



This section defines system generation. It describes what is needed to perform a system generation, how a system is generated, what must be done after a system is generated, and the types of generations that can be performed.

System Generation

System generation is the process of selecting modules from IBM-distributed libraries and tailoring them to create a new VS Operating System (operating system with virtual storage) for your installation.

To meet the needs of many installations, IBM distributes to its customers tape libraries that contain all of the modules that it produces in source, object, or load module form. Some of these modules are required for every VS Operating System and some are optional. Using the distribution libraries and another VS Operating System, you specify, through the use of macro instructions, the new VS Operating System to be generated.

Your new system is created when your macros are executed by the system generation process. This process relates options you desire to actual supporting code within the VS modules. And, it produces the job control language and utility control statements necessary to select the code from the VS distribution libraries and to build your new system. During the generation of the new VS Operating System, source modules are assembled according to the installation data you provide, object modules are linked together, and load modules are copied into the new system.

There are five major steps in generating a new VS Operating System. These are:

- *Planning*

To plan a new system, you must consider what your installation's needs are and from these needs determine the VS options that will, with the required, standard VS features, constitute your new system. (See "New System and Program Option Selection.")

- *Stage I*

Stage I is a single assembly of the macro instructions that you code. You select and code various macro instructions to specify the options you planned for your new system and the standard VS features that are required to be in the new system. (See "New System Macro Specification.") After the macros are coded and executed, they are assembled and expanded into job control language and utility control statements.

- *Initialization*

Before a new system can be generated, you must initialize direct-access volumes to contain the distribution libraries and/or the starter operating system. (See "Preparing for a System Generation.") You must also initialize the direct-access volume(s) onto which the new system will be generated and allocate space for the data sets on the new system volume(s). (See "New System Volume Initialization" and "New System Data Set Initialization.")

- *Stage II*

Stage II is the execution of the job control language and utility control statements from Stage I. During execution, the job control language and utility

control statements assemble, link-edit, and copy specific modules from the distribution libraries into the new VS Operating System. (See "System Control Program Installation.")

- *Testing the New System*

After your new system is generated, you should verify that it is operational. (See "Installation Verification Procedures.")

System Generation Requirements

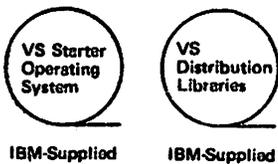
To perform a system generation, you must have IBM-supplied distribution libraries containing the VS modules, an existing VS Operating System, or a starter operating system from IBM.

Using an Existing VS Operating System

If an existing VS Operating System is used, it must contain the IEHDASDR utility to initialize direct-access volumes for the distribution libraries and the new system. (See "Preparing for a System Generation" and "New System Volume Initialization.") To execute Stage I only, the existing system must contain an assembler. To execute Stage II only, it must contain an assembler and a linkage editor, a copy utility (IEBCOPY), a catalog and index utility (IEHPROGM), a source language modification utility (IEBUPDTE), a utility for updating track address (IEHIOSUP) (VS1), a utility for reinitializing the data set containing machine error data (IFCDIP00), and a utility (IQADVM00) for page formatting the DSS load module (VS2) (Dynamic Support System, a debugging tool for the Program Systems Representative that provides extensive debugging facilities to aid in locating errors and determining the system state quickly).

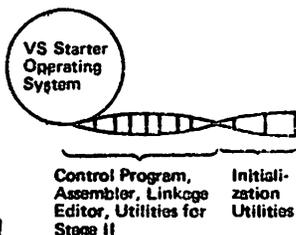
The assembler for Stage I and the assembler and linkage editor for Stage II must be compatible with the VS System Control Program previously generated.

Ordering the VS Starter Operating System and/or the VS Modules



If you have an existing VS Operating System with the program and machine components for generating a new VS Operating System, you need to order only the distribution libraries containing the VS modules. You obtain the distribution libraries by sending a tape to IBM. IBM copies the VS modules on the tape and returns it to you. If you do not have a VS Operating System, you send two tapes to IBM, one for the VS modules and one for a starter operating system. If you do not have any tapes or a VS system, you order two tapes from IBM, one containing the VS modules and the other containing the starter operating system.

The VS Starter Operating System



The starter operating system must be used for the first installation of a VS Operating System. For subsequent installations, either the starter operating system or your own VS Operating System can be used.

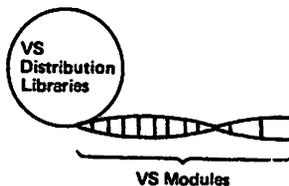
The starter operating system consists of:

- A control program that supports the central processing unit(s) and I/O devices you need to run the system generation jobs

- An assembler for Stage I and Stage II
- A linkage editor for Stage II
- The utilities used for data set and volume initialization
- The utilities for Stage II processing

Requirements, such as the amount of storage and the devices you need for the starter operating system, are detailed in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*. Those manuals also contain descriptions of the utilities used before Stage I and during Stage II.

The VS Modules

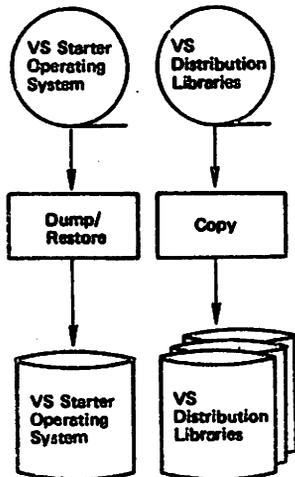


When you send in your own tape to IBM for the modules or order an IBM tape, the module(s) are put onto tape as unloaded partitioned data sets with the use of the IEBCOPY utility program. These partitioned data sets are called distribution libraries. The distribution libraries contain:

- Program options (VS features that can be included in your new system)
- Stand-alone utilities that operate in support of or without support of the VS System
- Routines that manipulate collections of data and system control data
- Routines that manipulate partitioned, sequential, or indexed sequential data sets
- Data management routines
- Error recovery routines
- Problem determination routines
- Job management routines
- Teleprocessing routines
- A parameter library (a library that contains a system parameter list used by the Nucleus Initialization Program (NIP), a PRESRES list, and a LNKST00 list described in *OS/VS1 Planning and Use Guide* and *OS/VS2 Planning and Use Guide*)
- A library of cataloged procedures to perform, for example, a compile/link-edit/go
- A library of macro definitions used during the Stage I assembly
- A library of macro definitions used during Stage II assemblies
- A library that contains installation verification procedures for testing the new system (see "Installation Verification Procedures")
- A system macro library (a library that contains macro definitions for the system macros used by the assembler)

Within each distribution library are one or more VS modules. Single modules or combinations of modules are selected from the distribution libraries and put into your new system during Stage II. The selection of modules is determined by the program options that you choose. These options are described in "New System Macro Specification."

Preparing for a System Generation



Before the generation of a new system, you must:

- Add any user-written routines to private libraries for inclusion in the new system
- Initialize disk packs for the starter operating system and the distribution libraries
- Restore the starter operating system from tape to a direct-access device of the type described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*
- Copy the VS distribution libraries from tape to any of the direct-access devices described in *OS/VS1 System Generation Reference* or *OS/VS2 System Generation Reference*

To add your own routines to the new system, they must be compiled, link-edited and placed as members in cataloged partitioned data sets so that they can be accessible during Stage II. (See *OS/VS1 System Generation Reference* or *OS/VS2 System Generation Reference*.) In Stage I, you specify, with various macro instructions, the name of the partitioned data set and the name of the members containing your routines to be in the new system.

The starter operating system and the distribution libraries must be on initialized disk packs for Stage I and Stage II execution. Initialization is the process of writing home addresses, a volume label, and a volume table of contents on a direct-access volume. It is also done to check for defective tracks and to assign alternate tracks for those that are defective. The IBCDASDI utility or the IEHDASDR utility initializes disk packs.

IBCDASDI must be used to initialize the disk pack for the starter operating system. After the pack is initialized, the starter operating system is put on the pack by IBCDMPRS (a dump/restore utility program). IBCDASDI and IBCDMPRS are on the starter operating system tape.

The disk packs for the distribution libraries can be initialized by the IEHDASDR utility. IEHDASDR is in your starter operating system or in your existing VS Operating System. If you are using the starter operating system, you must restore it to disk using IBCDMPRS and perform the IPL procedures (see *Operator's Library: OS/VS1 Reference* or *Operator's Library: OS/VS2 Reference* for a description of the IPL procedures) before IEHDASDR can be used. After the packs for the distribution libraries are initialized, the IEBCOPY utility must be used to copy the distribution libraries from tape to disk.

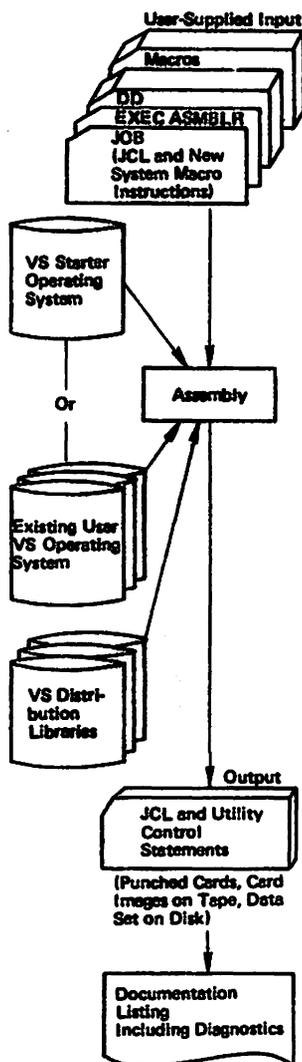
For a detailed description of the utilities mentioned above, see the *OS/VS Utilities* reference manual.

The VS distribution libraries may reside on direct-access devices of several different types, whereas the starter operating system is limited to specific types. The types of devices required for the starter operating system and the VS distribution libraries are described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*.

System Control Program Installation

Your new system is generated in two stages. Stage I is a single assembly of macro instructions that you code to describe your new system. The output from Stage I is a job stream and a diagnostic listing described in the following section. The job stream consists of job control language and utility control statements on punched cards, as card images on tape or in a data set on disk. Stage II is the execution of the job stream produced by Stage I. During this stage, source modules from the distribution libraries are assembled, object modules from the distribution libraries are link-edited, and/or load modules from the distribution libraries are copied into data sets that you allocate on the new system volume(s). (See "New System Data Set Initialization.")

Stage I



Input to Stage I consists of job control language statements and macro instructions that you code. The macros specify the machine configuration, the control program, the data management routines, and your own routines to be included in the new system. The macro parameters specify the VS options you want in your new system. (See "New System Macro Specification.") Some of the macros are required and some are optional, depending on your needs. These are also described in more detail in "New System Macro Specification."

For your macros to be assembled, your Stage I input deck must have an execute statement for the assembler and data definition statements for the Stage I macro definition library and for three temporary utility data sets that are required for Stage I. It must also have data definition statements describing the output, one for the data set to contain the job stream and one for a printer. The sequence of the input deck (cards or card images) to Stage I is as follows:

- Job Statement
- Execute Assembler Statement
- Data Definition Statements
- New System Macro Statements

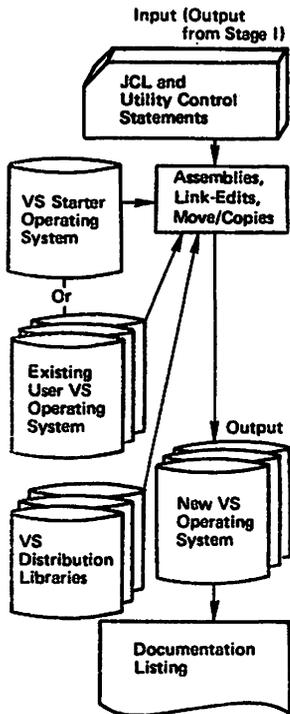
These are described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*.

During execution of your input, the macro instructions are assembled and analyzed for valid parameters and dependencies upon other macros. When invalid macro specifications are found, Stage I does not produce the job stream for Stage II. And, error messages are printed for each invalid macro. If there are no errors, the job stream is produced as punched cards or as card images on tape or in a data set on disk according to what you specified in your Stage I data definition statement. The Stage I output job stream contains several job statements, each followed by one or more execute statements. Each execute statement is followed by its associated data definition statements and other control statements required to be used by the assembler, the linkage editor, and the utility programs during Stage II.

The Stage I output job stream should be retained in a tape or direct-access data set in order to use the service aid program HMAPTFLE. This program generates the job control language necessary for applying a program temporary fix (PTF) from your distribution libraries to your new VS system; or it directly applies a PTF object module to your new VS system. (See the *OS/VS Service Aids* manual for a description of HMAPTFLE and its use.)

In addition to the job stream, Stage I also produces a documentation listing. It is a printout of the expansion of all the macros that you specify including punch statements that comprise input to Stage II.

Stage II



Input to Stage II is the output from Stage I. The processing of the Stage I output in Stage II causes multiple executions of the assembler, linkage editor, and system utilities. Single modules or combinations of modules are selected from the distribution libraries and placed into data sets that you allocate on the new system volume(s). See "Selecting the New System Data Sets."

Four temporary utility data sets described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference* are required for Stage II. Three of them are dedicated data sets (preallocated data sets described in *OS/VS1 Planning and Use Guide* and *OS/VS2 Planning and Use Guide*) in the procedure library. You must allocate the fourth data set before executing Stage II. This data set will contain object modules when Stage II is executed. During execution, required and selected modules are assembled and put into the object module utility data set that you have allocated; required and selected executable modules from the distribution libraries and object modules from the object module data set are then link-edited into the new system data sets; and executable modules from the distribution libraries are copied into the new system data sets.

The output from Stage II is your new VS Operating System. Stage II also produces a documentation listing. This is a printout of all the steps executed by the assembler, the linkage editor, and the utilities.

New System Verification

After your new VS Operating System has been generated, you should verify that the new system is operational and that it supports the new system machine configuration. (See "Installation Verification Procedures.")

Types of System Generation

There are three types of generation: complete, nucleus only, and I/O device only. The type of generation is specified in a macro called GENERATE. This macro produces the Stage II job stream. For an I/O device only or nucleus only generation, the job stream is applied against an existing system that is to be modified. For a complete generation, the Stage II job stream creates an entirely new system.

Following are descriptions of the types of system generation that can be performed:

- ***Complete***

This is the generation of a complete VS System Control Program. It is done when you are installing a new system for the first time or when you must create a new control program for an already existing VS Operating System. A complete generation is not required if the desired modifications can be achieved by a nucleus or I/O device generation.

- ***Nucleus only***

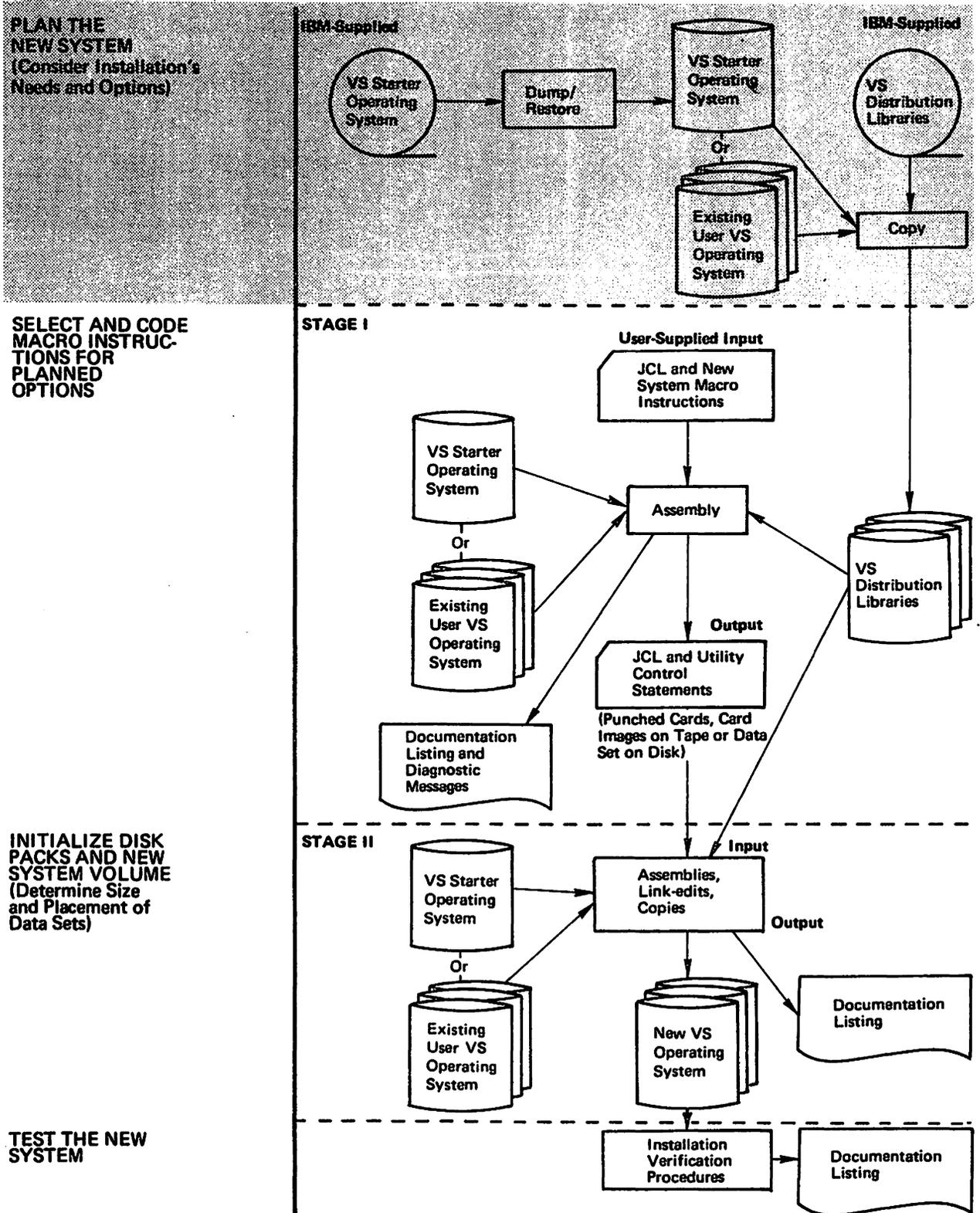
This is the generation of a new nucleus. The options and configuration specified must be compatible with the rest of the control program. It is done when only changes to the nucleus of the control program need to be made.

- ***I/O device only***

This is done when I/O devices and channels are to be added, deleted, or modified.

The type of generation determines which macro instructions you specify for your new system. For each type, some of the machine configuration macros and control program macros are required, some are optional, and some do not apply. The data management macros may be optional depending on the type of generation you select. (See "New System Macro Specification" for a description of some of these macros.) The selection of your new system according to the type of system generation is described in *OS/VS1 System Generation Reference* or *OS/VS2 System Generation Reference*.

SYSTEM GENERATION PROCESS



This section provides more detailed information about system generation. It describes some of the considerations you should think about when planning a new system, the selection and allocation of new system data sets, and some of the macros and how they are used to specify the options to be in your new system. It also explains how to correct errors that occur during Stage I or Stage II, what is needed for Stage I and Stage II execution, and how your new system is tested.

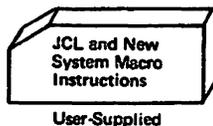
New System and Program Option Selection

The first step in generating a new system to satisfy the data-processing needs of your installation is to determine exactly what those needs are.

In analyzing your needs, you should consider such questions as: What is the total volume of data-processing work that must be done? What types of work must be performed? What are the relative priorities of the different types of work? What kind of assistance does each member of the installation need to do his job? What is the physical setup of the machines? How many procedures do you want in the procedure library? If you improve system performance, will the improvement use up storage space? Do you plan to extend the capabilities of the control program in the future?

After all your installation's needs have been considered and you have planned for the VS options and standard features described in *IBM System/370 System Summary*, and in either *OS/VS1 Planning and Use Guide* or *OS/VS2 Planning and Use Guide*, you should be able to select the macros and parameters that define the new system machine configuration and new operating system to meet your needs.

New System Macro Specification



You code specific macro instructions to describe your new system. The macros describe the machine configuration, the control program, the data management routines, and optionally your own routines. With each macro, you may code one or more parameters to specify the program options you have selected to be in your new system. Some of the macro instructions are described below to illustrate how they are used to specify a new VS Operating System. The selection of the macros to describe your new system depends on the type of system generation you are going to perform: complete, nucleus, or I/O device (see "Types of System Generation"). *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference* describe the macros in detail and how to select and code them.

- *Machine configuration*

The central processing unit macro, CENPROCS, specifies the central processing unit. The parameters specify the model number and the addition of a secondary model.

The channel macro, CHANNEL, specifies one or more channels. The parameters specify the channel address and type.

The I/O device macro, IODEVICE, specifies the I/O devices you want in your new system. The parameters specify their addresses, their types, their features, and their programming support.

The standard character set image macro, UCS, specifies the character set images for a 1403 printer with the universal character set (UCS) feature or for a 3211 printer. The parameters specify all the images or specific ones and default choices.

The unit name macro, UNITNAME, assigns a generic name to an I/O device or a group of I/O devices. The parameters specify the name of the group and the I/O device addresses.

Control program

The control program macro, CTRLPROG, specifies the control program options. The parameters specify such options as the number of I/O operations to be processed concurrently, the amount of time that a job has control of the central processing unit, a trace-table that stores information to be used for debugging and maintenance, and system queue area specifications.

The scheduler macro instruction, SCHEDULR, specifies the job scheduler and master scheduler. The parameters specify requirements and options such as unit addresses of the master and alternate consoles; volume mounting procedures; the use of a hardcopy log or system log for recording operator commands, system commands and responses, and write-to-operator (WTO and WTOR) messages; routing codes for the master console; routing codes for write-to-operator routines; automatic execution of the commands for starting a reader, a writer, or an initiator after IPL; the number of records in a logical track for the system job queue; the number of records in the system job queue reserved for the termination of jobs; the number of records in the system job queue reserved for each initiator started; the use of error volume analysis and the residence of volume error statistics records.

The checkpoint/restart macro, CKPTREST, specifies standard and user-written ABEND codes for which an automatic restart can occur. The parameters specify whether or not restart can occur for each ABEND code.

If you want multiple console support in your new system, the secondary console macro, SECONSLE, must be specified. Its parameters specify a secondary console, an alternate console, routing codes, and valid command codes that can be entered into a specified console.

The macro library macro, MACLIB, specifies certain macros to be excluded from the macro library in the new system. The parameters specify which macros are to be excluded.

The editor macro, EDITOR, specifies linkage editor options. The parameters specify size of virtual storage for the linkage editor and its buffer.

The loader macro instruction, **LOADER**, specifies the options to be included in the loader processing program. Its parameters specify the input data set that contains the loader program input, the library to be searched to resolve external references, the data set for the external reference map and diagnostic messages, and the amount of virtual storage required for the loader program buffers, tables, and problem program.

- *Data management routines*

The data management macro, **DATAMGT**, specifies optional access methods to be included in your new system. The parameters specify the methods.

The graphics macro, **GRAPHICS**, specifies graphic programming services. The parameters specify the inclusion of problem-oriented routines and the graphic subroutines.

- *User-written routines*

The link library macro, **LINKLIB**, specifies the inclusion of your own routines, in load module form, in the link library of your new system. The parameters specify the partitioned data set that contains your load modules and the members that are to be added.

The resident module macro, **RESMODS**, specifies the inclusion of your own routines, in object or load module form, in the system nucleus library. The parameters specify the partitioned data set that contains your modules and the members to be added to the system nucleus library.

Other user-written routines, such as **SVC** routines and I/O appendages, can also be included in the new system during system generation.

The **SVC** table macro, **SVCTABLE**, specifies entries to be in the **SVC** table. These entries are defined by the macro parameters. The parameters specify the number and type of the **SVC** routines, written by you, that are to be added to the new system.

- *New system data set allocation and cataloging*

The data set macro, **DATASET**, can be used to allocate space to the required and optional system data sets on the new system volume(s) and to catalog the data sets in the system catalog during system generation. The macro parameters specify the system data set to be allocated, the new system volume on which it will reside, and the amount of space to be allocated on the volume. Default values are used for the DCB information that is required for the data sets so this information is not specified. (System data sets can also have space allocated to them and they can be cataloged before system generation by using the **IEHPROGM** utility program. Refer to "New System Data Set Initialization" for this information.)

- *Generation*

The last macro to code, **GENERATE**, analyzes compatibilities between the various macros specified. If no errors are found, it produces the job control language and utility control statements for Stage II. Its parameters specify the volume name and device type for the new system.

When you code the macros, there are many things to be considered. One of these is dependency of one macro upon another as described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*. Another very important consideration is the storage needed for the specified options as described in *OS/VS1 Storage Estimates* and *OS/VS2 Storage Estimates*.

Module Selection and Placement in the New System

Specific modules from the distribution libraries are joined together during Stage II to provide the programming support for the new operating system as specified in Stage I. Modules are retrieved from the distribution libraries, assembled, link-edited or copied into the new system data sets according to the macro expansion produced by Stage I. The macro expansion, consisting of job control language and utility control statements, specifies the new system data sets into which the modules from the distribution libraries will be placed as shown in the illustration that follows.

Because you know which program options you want in the new system, you must allocate on the new system volume the data sets that will contain your program options before Stage II is executed. At this time, you must also allocate new system data sets to contain the modules that will make up the required, standard VS features. "New System Data Set Initialization" describes the selection and allocation of system data sets.

New System Volume Initialization

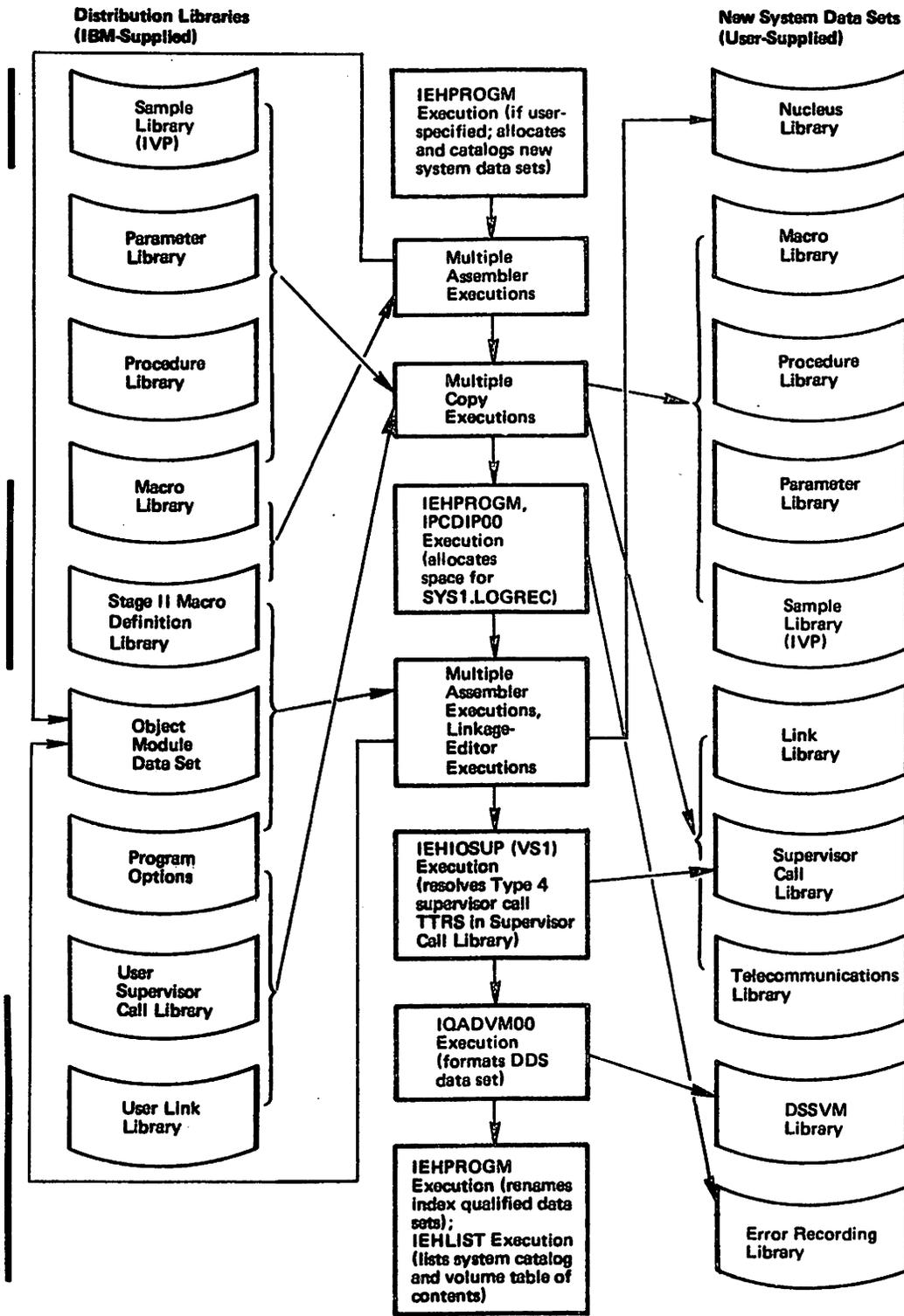
Before Stage II is executed, you must either mount the direct-access volumes containing the existing system to be modified for an I/O device or nucleus only generation or initialize direct-access volumes for a complete generation as described below.

If a complete system is being generated, the direct-access volume(s) for the new system must be checked for defective tracks and home addresses, and a volume label and volume table of contents must be written on the new system volume(s). This can be done by using either the IBCDASDI utility or the IEHDASDR utility.

The IBCDASDI utility is on the first file of the starter operating system tape; it is a stand-alone or independent utility, which means that it can be executed independently of an operating system. IBCDASDI can be used to initialize the volume that is to contain the starter operating system.

The IEHDASDR utility is in the starter operating system or in your existing system (if this is not your first system generation). IEHDASDR is a system utility, which means that it can only be executed under the control of an operating system. If you are using the starter operating system, you must restore it to disk and perform the IPL procedures before IEHDASDR can be used.

The initial program loader (IPL) program must be written on the new system volume. The IPL program is in the distribution libraries. You can use IEHDASDR from the starter operating system or IBCDASDI to write the IPL program on the new system volume. It is written on the volume by punching out the IPL program from the distribution libraries and inserting the cards with the IBCDASDI or IEHDASDR input deck. Or, you can retrieve the program directly from the starter operating system using IEHDASDR. *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference* describe the procedures for punching the IPL program into cards. The *OS/VS Utilities* reference manual describes the use of IBCDASDI or IEHDASDR for writing the IPL program on the new system volume and for initializing the disk packs.



→ Indicates the retrieval of modules from the distribution libraries and their placement in the new system

NOTE: This illustration does not show all of the distribution libraries, nor new system data sets. Its intent is to show how modules are selected and placed in the new system.

Module Selection and Placement in the New System

New System Data Set Initialization

New system data sets must be initialized before Stage II is executed. Initialization is the process of:

- Allocating space for system data sets on the new system volume
- Building the volume index of the system catalog
- Cataloging the data sets in the new system catalog

You can initialize the new system data sets by specifying the data set macro in Stage I. The data set macro, when executed, creates the job control language to allocate and catalog the data sets specified by its parameters. You can also initialize the new system data sets by executing your own job control language deck before Stage II is executed. If you provide your own deck, it should consist of:

- A job statement
- An execute statement for IEHPROGM (a utility that catalogs the data sets on the new system volume)
- A data definition statement for the message output data set
- Data definition statements for each new system data set and the system catalog data set
- A catalog control statement for each data set to be cataloged

Selecting the New System Data Sets

Some of the new system data sets are required in your new system and some are optional. The required data sets are basic to your new VS Operating System. These data sets are needed to contain load modules such as those that constitute catalog or installation verification procedures; nonresident Supervisor Call (SVC) routines; nonresident error handling or nonresident machine-check handler routines; access method routines; processors; macro definitions for system macro instructions; PRESRES, LINKST00 lists and a system parameter list used by the Nucleus Initialization program (NIP) (the types of lists are described in *OS/VS1 Planning and Use Guide* and *OS/VS2 Planning and Use Guide*); and statistical data about machine errors. A system catalog data set and a data set that the scheduler uses as a work area are also required data sets.

The optional data sets are those that contain certain program options. These become required when you have selected a particular program option to be in your new system. For example, if you select a teleprocessing access method, which is not required for every new system, you must allocate the telecommunications library on the new system volume. The telecommunications library is the only data set into which teleprocessing access methods can be put.

The complete group of required and optional new system data sets is described in *OS/VS1 System Generation Reference* and *OS/VS2 System Generation Reference*. They give each new system data set name, whether it is required or optional, its data definition statements, whether it should be cataloged or not, its type (sequential or partitioned), and its contents after the new system has been generated.

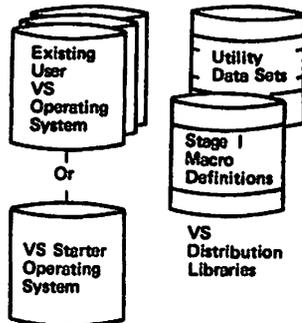
Allocating the New System Data Sets

The number of load modules put into a data set, according to the option you specify, defines the storage space that the data set will occupy. *OS/VS1 Storage Estimates* and *OS/VS2 Storage Estimates* explain how to calculate the amount of storage needed for the various program options.

When you have estimated the size of the new system data sets, you will have to decide if you want them on one volume or multiple volumes. They may have to be on multiple volumes because of their size or you may want them on multiple volumes for better system performance. (See *OS/VS1 Planning and Use Guide* or *OS/VS2 Planning and Use Guide* for performance considerations.)

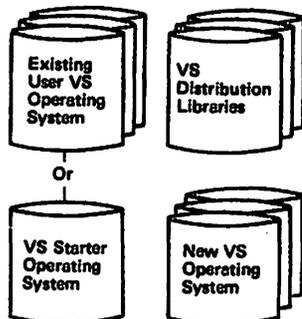
The new system data sets can be arranged on the new system volume(s) in different ways. The primary system residence volume contains the IPL program. It must also contain some of the required data sets described previously. If it is the only volume, it will have all the required data sets and those data sets needed for the program options you choose. When the new system is on one volume, frequently-used and closely-related data sets should be placed adjacent to one another. For best system performance, frequently-used data sets should be on a volume other than the primary system residence volume.

Stage I Execution Requirements



Your macro instructions are executed as one assembler job, which uses an existing VS Operating System or starter operating system described in "System Generation Requirements." In order for your macros to be executed, the volume containing the Stage I macro definition library and the volume containing the temporary utility data sets you defined for Stage I must be accessible during this stage. These utility data sets may be on any volume that has sufficient space for them.

Stage II Execution Requirements



The job control language and control statements from Stage I are executed as multiple jobs of assemblies, link-edits, and copies. Execution is done by an existing VS Operating System or starter operating system described in "System Generation Requirements." For Stage II, the volume(s) containing the Stage II macro definition library, all the distribution libraries, the dedicated utility data sets (preallocated data sets described in *OS/VS1 Planning and Use Guide* and *OS/VS2 Planning and Use Guide*), the object module data set you defined for Stage II, and the new system volume(s) must be accessible during this stage.

Restart Procedures

During execution of Stage I or Stage II, errors may occur requiring you to restart one or both of these stages. These errors and the procedures for restarting each stage are described below.

Stage I

The most common errors during Stage I are:

- Keypunching errors
- Contradictory or invalid specifications in the macro instructions

Keypunching errors are indicated by system generation error messages or assembler error messages. Invalid specifications are indicated by system generation error messages. If the system finds any errors, the job stream for Stage II is not produced.

When errors are found, the input deck described in "Stage I" must be corrected and the entire assembler job of your input deck must be reexecuted.

Stage II

The most common errors caused during Stage II are:

- Machine interruptions and noncontinuous machine time
- Faulty new system data set allocation
- Errors in the Stage I input deck that could not be detected during Stage I, such as allocation of the nucleus data set on a different residence volume than the one specified in the GENERATE macro instruction
- Procedural errors, such as invalid volume mounting

The input deck to be corrected for Stage II consists of several jobs each having several steps. Any step in the job stream can be reexecuted if all previous steps in that job have been properly executed.

Once a job or job step has been corrected, it is reexecuted in one of several ways. A single job that has only one step is reexecuted from the beginning of the step and a multiple-step job is reexecuted by your placing the job card in front of the step to be reexecuted. Other jobs, such as copying modules into a new system data set, may require that the data set be moved or copied to another volume in order to allocate more space on the new system volume if the data set was not large enough for the modules.

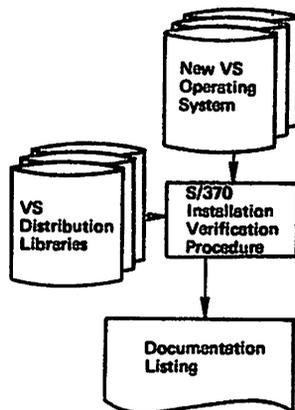
If any statements in the job stream are to be changed, the easiest way to change them is to have the job stream in cards. If the job stream is on tape, the IEBTPCH utility can be used to punch the job stream into cards. You can change statements on tape by using either the IEBUPDTE data set utility or the IEBGENER data set utility. (See the *OS/VS Utilities* reference manual for a description of how to use these utilities.)

Some of the errors during Stage II may not change any of the job stream statements. An example is the one mentioned previously. A data set used in a job step has to be reallocated, because it was too small for that step. In this event, another data set utility, IEBEDIT, can be used. IEBEDIT creates a new job stream onto another tape. This new job stream consists of jobs you want to restart. They can be:

- A job step and all the steps that follow it, or
- Specific job steps within a job, or
- All the job steps in the job

OS/VS1 System Generation Reference and *OS/VS2 System Generation Reference* describe the coding methods and use of the utilities needed to reexecute Stage I or Stage II. The *OS/VS Utilities* manual describes the utilities.

Installation Verification Procedures



Your new OS/VS Operating System must be tested to eliminate any possible malfunctions. You can test your new system by using a process called the Installation Verification Procedures (IVP).

The Installation Verification Procedures are in one of the distribution libraries. It contains jobs that run with all machine configurations. These jobs print out the procedure library, test the basic new system functions generated by executing an assemble/link-edit/go, force an 806 ABEND (see *OS/VS Message Library: VS1 System Codes* or *OS/VS Message Library: VS2 System Codes*), and issue a UNIT status command for I/O device addresses, type, status, volume serial number and state, ordered output, and device bypass. The job stream is used to test only the VS Operating System's System Control Program (SCP) that is generated and its subsequent updates. Program Products, programs with service classification "C," and similar programs to be added after a new system has been generated should be tested by their own Installation Verification Procedure or sample program.

Program Product Installation

The new operating system that is generated is a System Control Program. Program Products are neither distributed nor generated with the System Control Program. Manuals describing specific program products contain the procedures and requirements for installing them.

NEW SYSTEM PLANNING CHART

This chart is intended as an aid for considering the options to be in your new system.

LIST IN EACH SECTION BELOW THE FEATURES TO BE IN THE NEW SYSTEM.	
DISK — _____ RESIDENT VOLUME IDENTIFICATION — _____ (6-character maximum) LIBRARY VOLUME IDENTIFICATION — _____ (6-character maximum) CPU MODEL — _____ REAL STORAGE SIZE — _____	
CONTROL PROGRAM	
DATA MANAGEMENT	
JOB SCHEDULER TYPE — _____	DESIGN — _____

TASK SUPERVISOR RESIDENT FUNCTIONS

USER-WRITTEN ROUTINES

IF ANY SPECIAL I/O DEVICES MUST BE INCLUDED, LIST THEM HERE. ALSO LIST ANY OTHER NEEDS SUCH AS USER SVCs.

GLOSSARY

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the *IBM Data Processing Glossary*, GC20-1699.

control program: A program that is designed to schedule and supervise the performance of data processing work by a computing system.

complete generation: The creation of an entirely new VS Operating System's System Control Program or the modification of the existing VS Operating System's System Control Program.

dedicated data sets: Preallocated data sets belonging to the initiator that are used only temporarily during a job step.

distribution libraries: IBM-supplied partitioned data sets on tape containing single VS modules or combinations of VS modules that the user restores to disk for subsequent inclusion in a new system.

dynamic support system (DSS): An interactive debugging facility that allows authorized maintenance personnel to monitor and analyze events and alter data.

existing operating system: A system the user has that is used to process Stage I and Stage II. It is also a system that the user has and wants to change to include new or more I/O devices, more program options, or updates to data management and user-written routines.

initialization: The use of the IBCDASDI or the IEHDASDR utility 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 the new system volume.

***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.

installation verification procedures: Jobs within a partitioned data set in the distribution libraries that test if the new operating system is working correctly.

*American National Standard Definition

I/O device generation: The generation of a new system to add, delete, or modify I/O devices and channels. It is done when only changes to the I/O devices have to be made.

load module: A module that is the output of a linkage editor. Or, a program in a format suitable for loading into virtual storage for execution.

***module:** A program unit that is discrete and identifiable with respect to compiling, combining with other units and loading, for example, the input to, or output from, an assembler, compiler, linkage editor, or executive routine.

new system: An entirely new VS Operating System that has been generated. Or, an existing VS Operating System that is changed to include support for new or more I/O devices, more program options, or updates to data management and user-written routines.

new system data sets: User-allocated data sets on the new system volume into which load modules and executable modules from the distribution libraries are link-edited or copied in Stage II.

new system data set initialization: The process of allocating space for data sets on the new system volume, building the volume index of the system catalog, and cataloging the data sets in the new system catalog.

new system volume: The volume that contains user-allocated data sets into which modules or combinations of modules are placed during Stage II.

nucleus generation: The creation of a new nucleus only.

***object module:** A module that is the output of an assembler or compiler and is input to a linkage editor.

***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.

page: A fixed-length block of instructions and/or data.

pageable supervisor area: Virtual storage where supervisor routines are loaded only when required for execution.

parameter library: A library that contains a system parameter list used by the Nucleus Initialization Program (NIP), a PRESRES list, a LINKST00 list, and other lists described in *OS/VS1 Planning and Use Guide* and *OS/VS2 Planning and Use Guide*.

primary system residence volume: The new system volume that contains the initial program loader (IPL) program, the nucleus of the operating system, the highest level index of the catalog, and the SVC library.

program options: Features of the VS Operating System that the user may include in his system by coding specific macro parameters.

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.

source module: The source statements that constitute the input to a language translator for a particular translation.

stage I: A single assembly of user-coded macros 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 macros, into the new system.

starter operating system: An IBM-supplied VS Operating System consisting of a control program that supports any of the central processing units needed to generate a new system, a minimum machine configuration, an assembler and linkage editor for Stage I and Stage II, and utilities for new system data set and volume initialization. It is used when there is not an existing VS Operating System with which to generate a new system.

***system:** An organized collection of men, machines, and methods required to accomplish a set of specific functions.

system macro library: A library that contains macro definitions for the system macros used by the assembler.

system generation: The process of obtaining VS modules on tape, ordering a starter operating system if there is no VS Operating System, specifying macros to include VS features and options in the new system, including the VS features and options by macro execution, executing the macro expansion to generate the new system, and testing the new system.

***system residence volume:** The volume on which the nucleus of the operating system and the highest-level index of the catalog are located.

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.

*American National Standard Definition. IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard Vocabulary for Information Processing (Copyright © 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3.5 on Terminology and Glossary of American National Standards Committee X3.

INDEX

Indexes for reference manuals are consolidated in *OS/VS Master Index*, GC28-0602. For additional information about any subject listed below, refer to other publications listed for the same subject in the *Master Index*.

A

allocating data sets 16,18-19
applying temporary program fix 11

C

cataloging data sets 18
coding macro instructions 13-15
complete generation
 definition of 9
control program
 specification of 14
copy utility
 use of 5,6
copying distribution libraries 6

D

data management routines
 specification of 15
data set macro
 description of 15
 use of 18
data sets
 allocation of 16,18-19
 cataloging of 8
 dedicated 8,19
 new system 16-19
 object module 7,8
 reallocation of 21
dedicated data sets
 definition of 8
 use of 19
direct-access devices
 use of 6-7
disk packs
 initializing of 6,16
distribution libraries
 contents of 5
 copying of 5,6
 definition of 5
documentation listing
 installation verification 21
 Stage I 8
 Stage II 8

DSS (Dynamic Support System)
 data set 17
 description of 4
 dump/restore utility
 use of 6

E

errors
 Stage I 20
 Stage II 20-21
existing system
 definition of 4
 requirements for 4

G

generation
 types of 9

H

HMAPTFLE
 definition of 7

I

IBCDASDI
 use of 6,16
IBCDMPRS
 definition of 6
IEBCOPY
 use of 4,5,6
IEBEDIT
 use of 21
IEBGENER
 use of 20
IEBPTCH
 use of 20
IEBUPDTE
 definition of 4
 use of 20

IEHDASDR
definition of 4
use of 16

IEHIOSUP
definition of 4
use of 17

IEHPROGM
definition of 4,18
use of 17

IFCDIP00
definition of 4
use of 17

initialization
definition of 3

initializing
disk packs 6,16
new system data sets 18-19
new system volume 16

input
Stage I 7
Stage II 8

installation considerations 13

I/O device generation
definition of 9

IPL (Initial Program Loader)
location of 16,19
use of 6

IVP (Installation Verification Procedures)
description of 21

IQADVM00
definition of 4
used of 17

J

job stream
use of 8
restarting of 20-21

M

machine configuration
specification of 13-14

macro definition library
Stage I 5,19
Stage II 5,19

macro instructions
coding of 13-16
control program 14-15
data management 15
data set allocation 15
description of 13-16
execution of 7
for allocating space to system data sets 14
for cataloging system data sets 14
machine configuration 13
parameters 7,13
user-written routines 15

modules
definition of 5
ordering of 4
selection and placement of 16,17

multiple-step job
reexecution of 20

N

new system
considerations for 13
generation of 7-8
planning for 3
selection of 18
testing of 21

new system data sets
allocation of 19
optional 18
placement of 19
reallocation of 21
required 18
selection of 18
size of 19
testing of 21

new system volume
definition of 16

nucleus generation
definition of 9
use of 13

O

object module
data set 8
use of 8

optional new system data sets 18

ordering tapes for system generation 4

output
Stage I 7
Stage II 8

P

pageable supervisor area
definition of 15

parameter library
definition of 5

parameters
specifying of 13-15

partitioned data sets
cataloged 6
unloaded 5

planning considerations 13,3

preparing for system generation 6

primary system residence volume
definition 19

- program options
 - definition of 5,18
 - specifying of 7
 - storage space for 18,19
- program products
 - generation of 21
- program temporary fix
 - applying of 7

R

- reallocating data sets 21
- required new system data sets 18
- requirements
 - existing system 4
 - Stage I 19
 - Stage II 19
 - system generation 3-4
- restarting
 - job stream tape 20-21
 - Stage I 20
 - Stage II 20-21
- restoring the starter operating system 6

S

- selecting the new system 13-16
 - data sets 18
- service aid, IFCDIP00 4,17
- Stage I
 - definition of 3
 - documentation listing 7
 - errors 20
 - execution of 7
 - input 7
 - macro definition library 5,19
 - output 7-8
 - requirements 19
 - restart procedures 20
- Stage II
 - definition of 3-4
 - documentation listing 9
 - errors 20-21
 - execution of 8
 - input 8
 - macro definition library 5,19
 - output 8
 - requirements 19
 - restart procedures 20-21
- starter operating system
 - definition of 4-5
 - restoring of 6
- storage space
 - allocation of 18-19
- system control program
 - generation of 7-8,21

- system generation
 - considerations for 13
 - definition of 3-4
 - preparing for 6
 - program 3
 - requirements 4-6
 - types of 9,13
- system macro library
 - definition of 5
- system performance 19,13

T

- tape
 - job stream 7,8,20
 - starter operating system 4-5
 - VS module 4,5
- testing the new system 21
- types of system generation 9

U

- user-written routines
 - specification of 6,15
- utility data sets
 - use of 7,8,19
- utility programs
 - copy 4,5,6
 - dump/restore 6
 - HMAPTFLE 7
 - IBCDASDI 6,16
 - IBCDMPRS 6
 - IEBCOPY 4,5,6
 - IEBEDIT 21
 - IEBGENER 20
 - IEBPTCH 20
 - IEBUPDTE 4,20
 - IEHDASDR 4,6,16
 - IEHIOSUP 4,17
 - IEHPROGM 4,28,17
 - IQADVM00 4

V

- verification procedures 21
- virtual storage
 - specification of 14,15
- volume
 - initializing of 3,16
 - new system 15,16,19
 - primary system residence 19

VS modules
definiton of 5
selection of 3,16,17

VS Operating System (operating system with virtual storage)
generation of 3

READER'S COMMENT FORM

OS/VS System Generation Introduction

Order Number GC26-3790-1

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

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

Reply requested

Yes

No

Name _____

Job Title _____

Address _____

_____ Zip _____

No postage necessary if mailed in the U S A

YOUR COMMENTS, PLEASE . . .

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

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

fold

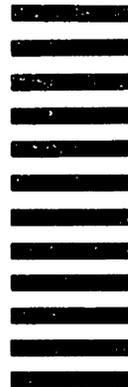
fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

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

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Monterey & Cottle Rds.
San Jose, California
95114



Attention: Programming Publications, Dept. D78

fold

fold



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