

## Systems Reference Library

### OS System Generation

#### Release 21

This publication provides the information required to generate the Operating System adapted to both the machine configuration and the data processing requirements of an installation. It can be used as a reference manual by the personnel responsible for planning an operating system and implementing its operation. It can also be used by the personnel responsible for carrying out the machine operations and those concerned with any restart procedures required.

This publication contains the information necessary to:

- Plan the Operating System (detailed storage requirements can be obtained from OS Storage Estimates, GC28-6551)
- Prepare for the generation of the Operating System
- Execute the system generation process
- Test the system that has been generated

In addition, this publication contains the following:

- Guidelines and procedures for restarting the system generation process
- A suggested method for testing the system that has been generated
- Detailed examples of various types of system generation
- A description of the starter operating system and a list of the operating procedures that it requires
- The types of system generation messages produced

**Information in this publication for TSO with the Model 65 Multiprocessing (M65MP) configuration is for planning purposes until that item is available.**



Eleventh Edition (February 1972)

This is a major revision of, and obsoletes, GC28-6554-10. The information that has been changed or added is described on the following pages.

This edition applies to Release 21 of the operating system and to all subsequent releases until otherwise indicated in new editions or technical newsletters. Changes to the information in this book may be made at any time; before using this publication in connection with the operation of IBM systems, consult the latest OS SRL Newsletter, GN20-0360, for the editions that are applicable and current.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Department D78, Monterey and Cottle Roads, San Jose, California 95114.

© Copyright International Business Machines Corporation 1966, 1967, 1968, 1969, 1970, 1971, 1972



System generation is the process by which an operating system is created or modified. The system that is being created or modified is tailored to the machine configuration and data processing requirements of an installation by the specifications you provide to the system generation program.

This publication provides you with the information necessary to:

- Prepare for the system generation process.
- Execute the system generation process.
- Test the results of the system generation.

The information in this manual is provided in the following sections:

"Introduction" describes the four types of generation that you can specify -- a complete generation, a nucleus generation, an I/O device generation, and a processor/library generation.

"Preparation for System Generation" describes the preparatory steps necessary for a system generation and provides examples to illustrate the coding required for each step; provides the requirements for the generating system and the steps necessary to prepare the system; includes considerations for using an MFT or MVT system as the generating system; describes the procedure for adding user-written functions to the new SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LINKLIB; the requirements and detailed operating procedures for the starter operating system package; and provides the requirements for the new or modified system and the steps required to initialize, allocate space on, and catalog any new direct-access volumes that will be used.

"The Job Stream" provides a diagram and brief description of the job stream that is the result from Stage I of the system generation program.

"Operating Considerations" briefly describes various situations that will probably be encountered by the console operator during the execution of the system generation program.

"Restart Procedures" describes the restart procedures that may be required during Stage I or Stage II of the system generation program and provides instructions and coding examples for recovering.

"Testing the New System" provides the operating procedures for the sample programs that are included in the starter operating system package.

"Examples" provides four examples, each of which illustrates the generation of a particular type of control program as well as showing how to do various types of generation; illustrates data set allocation and the coding required for the entire system generation process.

"Program Options Descriptions" provides a brief summary of each operating system option available.

"Data Sets" contains the detailed information about the required and optional data sets for the generating operating system and the new or modified operating system.

"System Generation Macro Instructions" provides detailed coding information for each system generation macro instruction followed by a table showing any cross-reference relationship between various macro instructions, and provides guidelines for carrying out the various types of system generation.

"Appendixes" provides the following:

A list of device type names provided on every operating system.

- The additional preparation required if more than 96 I/O devices are to be specified during system generation.
- The distribution methods are discussed.
- The messages produced during system generation.

#### REFERENCES

The following operating system publications are required to understand and select the operating system to be generated:

OS Introduction, GC28-6534

OS Storage Estimates, GC28-6551

OS Data Management for System Programmers, GC28-6550

OS MFT Guide, GC27-6939

OS MVT Guide, GC28-6720

OS Job Control Language Reference, GC28-6704

OS Loader and Linkage Editor, GC28-6538

OS Assembler Language, GC28-6514

Reference is made throughout this publication to several utility programs. The reader must be familiar with the requirements and facilities they provide. The utility programs are described in the OS publication:

OS Utilities, GC28-6586

The following publications are referred to in this manual and may be required to provide more detailed information about a particular topic.

OS Advanced Checkpoint/Restart, GC28-6708

OS ALGOL Language, GC28-6615

OS ALGOL Programmer's Guide, GC33-4000

OS American National Standard COBOL Programmer's Guide, GC28-6399

OS Assembler (F) Programmer's Guide, GC26-3756

OS CRJE System Programmer's Guide, GC30-2016

OS Data Management Services Guide, GC26-3746

OS Data Management Macro Instruction, GC26-3794

Emulating the IBM 1401, 1440, and 1460 on the IBM System/370 Models 135, 145 and 155 using OS/360, GC27-6945

Emulating the IBM 1410 and 7010 on the IBM System/370 Models 145 and 155 using OS/360, GC27-6946

Emulating the IBM 7074 on the IBM System/370 Models 155 and 165 using OS/360, GC27-6948

Emulating the IBM 7080 on the IBM System/370 Model 165 using OS/360, GC27-6952

Emulating the IBM 7094 on IBM Models 85 and 165 using OS/360, GC27-6951  
OS FORTRAN IV (G & H) Programmer's Guide, GC28-6817  
OS FORTRAN IV Library, GC28-6596  
OS FORTRAN IV Library: Mathematical and Service Subprograms, GC28-6818  
OS GPS for IBM 2250 Display Unit, GC27-6909  
OS GPS for IBM 2260 Display Station (Local Attachment), GC27-6912  
OS GPS for IBM 2280 and 2282 Display Units, GC27-6927  
OS GSP for FORTRAN IV, COBOL, and PL/I, GC27-6932  
Introduction to Main Storage Hierarchy Support for IBM 2361, Mod 1 and 2 under OS, GC27-6942  
OS Messages & Codes, GC28-6631  
OS MFT Supervisor Logic, GY27-7236  
OS MVT Supervisor Logic, GY28-6659  
OS OLTEP, GC28-6650  
OS Operator's Guide for Display Consoles, GC27-6949  
OS Operator's Procedures, GC28-6692  
OS Operator's Reference, GC28-6691  
OS PL/I (F) Programmer's Guide, GC28-6594  
OS PL/I (F) Subroutine Library, GC28-6590  
OS Principles of Operation, GA22-6821  
Guide to Reading OS System Dumps, GC28-6670  
OS RDE Guide, GC28-6747  
OS RJE, GC30-2006  
OS RPG Language, GC24-3337  
OS Service Aids, GC28-6719  
709/7090/7094 II Emulator Program for System/360 Model 85, GC27-6944  
OS Sort/Merge, GC28-6543  
OS Supervisor Services and Macro Instructions, GC28-6647  
OS System Control Blocks, GC28-6628  
OS SMF, GC28-6712  
OS Data Management for System Programmers, GC28-6550  
OS Tape Labels, GC28-6680  
OS TCAM, GC30-2020  
OS TSO Command Language Reference, GC28-6732  
OS TSO Guide, GC28-6698

OS TSO Guide to Writing a Terminal Monitor Program or a Command Processor,  
GC28-6764

Data Transmission for FORTRAN under S/360 OS and 1130 DMS, GC27-6937

# Contents

iii	Preface
xv	Summary of Changes for Release 21
1	Introduction
1	The System Generation Process
1	Stage I: Producing the Job Stream
1	Stage II: Processing the Job Stream
2	Types of Generation
5	Preparation for System Generation
7	Initializing Direct-Access Volumes
8	Initializing New System Data Sets
9	Input Deck for Initialization
9	DD Statements
11	CATLG Statements
11	Sample Data Set Initializations
14	Location of System Data Sets
15	Generating a New System onto One Volume Using Four 2311 Drives
17	Generating a New TSO System on One Volume Using Four 2311 Drives
18	Generating a New System on One Volume Using Two 2314 Drives
19	Generating a New System on One Volume Using Three 2314 Drives
19	Generating a New System on Two Volumes Using Four 2314 Drives
20	Generating a New System on One Volume Using Two 3330 Drives
20	Input Deck for System Generation
23	Including IEBUPDAT in the New System
23	Adding User-Written Functions
25	Considerations for MFT OR MVT (Including M65MP) Generating Systems
25	System Generation Using an Existing Operating System
26	System Generation as the Only Job (Method 1)
26	Cataloging Data Sets
28	Planning for SYS1.PROCLIB
28	Allocating and Cataloging the Data Sets for the New System
28	Allocating and Cataloging the Utility Data Sets
28	System Generation as One Job in the Job Stream (Method 2)
29	Cataloging the Required Data Sets
30	Obtaining the Catalog and Uncatalog Utility Programs
31	Allocating and Cataloging the Data Sets for the New System
32	Allocating and Cataloging the Utility Data Sets
32	Restoring the Data Set Names in the Starter Operating System
33	System Generation Using the Starter Operating System
33	The Starter Operating System Package
34	Starter System Requirements
35	Starter System Special Considerations
35	Multiple System Generation
35	The Universal Character Set Feature
35	Models 91 and 195
35	The M65MP System
35	The 2305, 2314, or 3330 Direct-Access Storage Facility
36	The Dual Density Feature and 9-Track Magnetic Tape Drives
36	Model 85
36	S/370 Model 165
40	Processing the Starter Package
42	Procedures for Processing the Starter Package
43	2311 Package -- Tape Distribution
48	2311 Package -- Disk Pack Distribution (Tape Backup)
52	2311 Package -- Disk Pack Distribution (Disk Backup)
56	2314 Package -- Tape Distribution
59	Deleting Libraries
60	3330 Package -- Tape Distribution
62	Deleting Libraries Using a 3330
63	The Job Stream -- Stage II
65	Non-TSO Generation
71	TSO Generation

77	Restart Procedures
79	Restarting Stage I
80	Restarting Stage II
81	Restart Techniques
81	Restarting From Cards
81	Punching the Job Stream
82	Restarting From Tape
83	Guidelines for Restarting Stage II (Non-TSO Generation)
84	Restarting During Assemblies
84	Restarting Any IEHPROGM Step
84	Restarting Link-Edit Steps
85	Restarting First IEBCOPY Step
85	Restarting All Other IEBCOPY Steps
86	Restarting IEHIOSUP, IFCDIP00, and IEHLIST
86	Guidelines for Restarting Stage II (TSO Generation)
86	Restarting During Assemblies
86	Restarting Any IEHPROGM Step
86	Restarting Link-Edit Steps
87	Restarting IEBCOPY Step
87	Restarting All Other IEBCOPY Steps
88	Restarting IEHIOSUP, IFCDIP00, and IEHLIST
88	Reallocating Data Sets
88	Reallocation of OBJPDS
88	Reallocation of SYSUT1, SYSUT2, SYSUT3, and SYSUT4
89	Reallocating the Same Space
91	Reallocating With More Space
94	SYS1.SYSJOBQE Considerations
95	Testing the New System
98	ALGOL Sample Program (IEXSAMP)
99	American National Standard COBOL, X3.23-1968 Sample Program (IKFSAMP)
100	Assembler F Sample Program (IEUESP)
101	COBOL E Sample Program (IEPSAMP)
102	Data Generator Sample Program (IEBDATGN)
103	Data Transmission and Conversion Subroutines for FORTRAN IV Sample Program (IKDSAMPL)
104	FORTRAN G and H Sample Program (IEYSP)
105	GRAPHICS Sample Programs (SAMP2250 and SAMP2260)
108	Graphic Subroutine Package for FORTRAN and PL/I Sample Program (GSPSAMP, and PL1SAMP)
111	IEHDASDR Sample Program (DRISAMP)
112	Loader Program Test Program
113	PL/I F Sample Program (IEMSP2)
114	RPG Sample Program (RPGSMPL)
116	Sample Account Routine (SAMACTRT)
117	Sort/Merge Sample Program (IERSP)
118	Update Analysis Program Sample Program (IHGSAMP)
120	User Label Sample Program (USERLABEL)
121	Examples
123	Example 1
123	Volumes Used for System Generation
123	Machine Configuration
126	Deck for Initializing New System Data Sets
127	Input Deck for Stage I
129	Backup of New System
130	Scratching Utility Data Sets
130	Obtaining SYS1.GENLIB, SYS1.MODGEN, and Component Libraries
131	Nucleus Generation
133	Example 2
133	Machine Configuration
136	Decks Required to Use the Starter Operating System
138	Initializing Volumes for New System
139	Volumes Used for System Generation
139	Initializing New System Data Sets
141	Input Deck for Stage I
144	Decks for System Residence on 2303
145	Input Deck for I/O Device Generation

149 Example 3

149 Volumes Used for System Generation

150 Deck for Initializing New System Data Sets

151 Input Deck for Stage I

153 Backup of New System

153 Scratching Utility Data Sets

155 Program Options Descriptions

157 Additional Pairs of Transient Areas

157 Alternate Path Retry (APR)

158 ATTACH Function

159 ATTACH Function Made Resident

160 Basic Direct Access Method (BDAM)

160 BLDL Table Made Resident

161 Channel-Check Handler (CCH)

162 Checkpoint/Restart

164 Consoles -- Alternate and Composite Console Options

165 Consoles -- Multiple Console Support (MCS)

168 Conversational Remote Job Entry (CRJE)

170 Decimal Simulation Option for Model 91

170 Direct-Access Volume Serial Number Verification

171 Dynamic Device Reconfiguration (DDR)

173 EXTRACT Function Made Resident

174 Graphic Programming Services

174 IDENTIFY Function Made Resident

175 Indexed Sequential Access Method (ISAM)

176 Job-Step Timing

176 Main Storage Hierarchy Support

177 PL/I F Shared Library Feature

178 Program Controlled Interrupt (PCI)

179 Reenterable Load Modules Made Resident

180 Remove Job Entry (RJE)

181 Resident Access Method Routines

182 Rollout/Rollin Option

182 Shared DASD

183 SPIE Routines Made Resident

184 Storage Protection Option

184 System Management Facilities (SMF)

186 Telecommunications Option

186 Timing Options

188 Time Sharing Option (TSO)

190 Time-Slicing

191 Trace Option

192 Transient SVC Table Made Resident

192 Type 3 and 4 SVC Routines Made Resident

193 User-Added SVC Routines

194 Validity Check Option

195 Volume Statistics

197 Data Sets

199 Required and Optional Data Sets for Generating Systems

200 SYS1.GENLIB

201 SYS1.MODGEN, SYS1.MODGEN2

202 SYS1.TSOGEN

203 Component Data Sets

207 System Data Sets

209 SYSCTLG

210 SYS1.ACCT

211 SYS1.ALGLIB

212 SYS1.ASRLIB

213 SYS1.BROADCAST

214 SYS1.CMDLIB

215 SYS1.COBLIB

216 SYS1.DCMLIB

217 SYS1.DUMP

218 SYS1.FORTLIB

219 SYS1.HELP

220 SYS1.IMAGELIB

221 SYS1.LINKLIB  
223 SYS1.LOGREC  
224 SYS1.MACLIB  
225 SYS1.MANX, SYS1.MANY  
226 SYS1.NUCLEUS  
227 SYS1.PARMLIB  
228 SYS1.PL1LIB  
229 SYS1.PROCLIB  
230 SYS1.ROLLOUT  
231 SYS1.SORTLIB  
232 SYS1.SVCLIB  
233 SYS1.SYSJOBQE  
234 SYS1.SYSVLOGX, SYS1.SYSVLOGY  
235 SYS1.TELCMLIB  
236 SYS1.UADS

237 System Generation Macro Instructions

239 Conventions

239 Coding Macro Instructions

240 Describing Macro Instructions

240 Macro Instructions Text Format

241 Macro Instruction Name

242 Macro Instruction Summary

245 ALGLIB  
247 ALGOL  
251 ASSEMBLR  
253 CENPROCS  
257 CHANNEL  
259 CHECKER  
261 CKPTREST  
263 CMDLIB  
265 COBLIB  
267 COBOL  
275 CTRLPROG  
281 DATAMGT  
283 DCMLIB  
285 EDIT  
291 EDITOR  
293 EMULATOR  
295 FORTLIB  
301 FORTRAN  
307 GENERATE  
317 GENTSO  
319 GJOBCTL  
325 GRAPHICS  
327 HELP  
329 IMAGELIB  
331 IOCONTRL  
337 IODEVICE  
355 LINKLIB  
357 LOADER  
361 MACLIB  
363 OUTPUT  
365 PARMLIB  
367 PARTITNS  
371 PL1  
381 PL1LIB  
389 PROCLIB  
391 PTOP  
393 RESMODS  
395 RPG  
397 SCHEDULR  
421 SECMODS  
425 SECONSLE  
433 SORTLIB  
435 SORTMERG  
439 SUPRVSOR  
449 SVCLIB



451	SVCTABLE
453	SYSUTILS
455	TELCMLIB
457	TSOPTION
459	UADS
461	UCS
463	UNITNAME
467	Cross-References Between the System Generation Macro Instructions
473	Appendixes
475	Appendix A: Device Types
479	Appendix B: Supporting Additional I/O Devices
479	Calculating the Storage Required by Unit Control Blocks
480	Calculating the Number of Devices for Groups
481	Redefining Two Members of SYS1.GENLIB
481	Inserting Two Control Statements
482	Reestablishing the Standard Definitions
483	Appendix C: Distribution Methods
489	Appendix D: System Generation Messages
493	Appendix E: Functionally Equivalent I/O Devices
495	Index



## Figures

- xx Figure 1. The System Generation Process
- 4 Figure 2. Types of System Generation
- 8 Figure 3. Initializing the System-Residence Volume With IBCDASDI
- 8 Figure 4. Initializing the System-Residence Volume With IEHDASDR
- 12 Figure 5. Initializing the System Data Sets -- One-Volume Residence on 2311
- 13 Figure 6. (Part 1 of 2) Initializing the System Data Sets -- Two-Volume Residence on a 2301 and 2311
- 16 Figure 7. Example of Generation on Four Drives
- 17 Figure 8. Example of Generation on Four 2311 Drives
- 18 Figure 9. Example of Generation on Two 2314 Drives
- 19 Figure 10. Example of Generation on Three 2314 Drives
- 19 Figure 11. Example of Generation on Four 2314 Drives
- 20 Figure 12. Example of Generation on Two 3330 Drives
- 21 Figure 13. (Part 1 of 2) Input Deck Organization for System Generation
- 22 Figure 14. Space Allocation (in Tracks) for Utility Data Sets
- 23 Figure 15. Input Deck for Including IEBUPDAT in the New System
- 24 Figure 16. Preparing a User-Written Load Module
- 27 Figure 17. Input Deck to List the Starter Operating System Catalog
- 30 Figure 18. Input Deck for Punching the Catalog and Uncatalog Utility Programs
- 31 Figure 19. Input Deck for Allocating and Cataloging System Data Sets With a User-supplied Qualifier of 'X'
- 37 Figure 20. Maximum System Configuration for the Starter Operating System
- 38 Figure 21. Minimum I/O Requirements
- 39 Figure 22. Device Types
- 39 Figure 23. Additional Group Names Supporting IBM-Supplied Cataloged Procedures
- 41 Figure 24. Processing the Starter Package
- 42 Figure 25. Sample Configuration
- 66 Figure 26. (Part 1 of 2) The Stage II Non-TSO Job Stream
- 69 Figure 27. (Part 1 of 2) Sample Steps in the Job Stream
- 72 Figure 28. (Part 1 of 2) The Stage II TSO Job Stream
- 75 Figure 29. (Part 1 of 2) Sample Steps in the Job Stream
- 79 Figure 30. Control Statements to Scratch and Uncatalog the Utility Data Sets
- 80 Figure 31. Restarting Stage I
- 81 Figure 32. Control Statements for IEBPTPCH
- 82 Figure 33. Punching the Job Stream
- 82 Figure 34. Control Statements for IEBEDIT
- 83 Figure 35. Input deck for IEBEDIT
- 88 Figure 36. Control Statements to Reallocate the Utility Data Set for OBJPDS
- 89 Figure 37. Control Statements to Recatalog the Utility Data Set When UTDISP=UNCATLG
- 89 Figure 38. Reallocate and Recatalog the Utility Data Sets when UTDISP=DELETE
- 91 Figure 39. Example of Reallocation for SYS1.SVCLIB and SYS1.LINKLIB
- 91 Figure 40. Reallocating on Same Space
- 92 Figure 41. Reallocation on Same Volume
- 93 Figure 42. Reallocate Data Sets With System Data
- 94 Figure 43. Control Statements to Change SYS1.SYSJOBQE
- 106 Figure 44. 2250 Displays
- 107 Figure 45. 2260 Displays
- 110 Figure 46. Display for GSP Sample Program
- 113 Figure 47. PL/I Generated Output
- 115 Figure 48. RPG Sample Program Printed Report
- 123 Figure 49. Example 1: Generating and New System Volumes
- 124 Figure 50. (Part 1 of 2) Example 1: 3330 Starter Machine Configuration
- 126 Figure 51. (Part 1 of 2) Example 1: Initializing New System Data Sets
- 127 Figure 52. (Part 1 of 3) Example 1: Stage I Input Deck
- 129 Figure 53. Example 1: Creating Backup of SYSTEM
- 130 Figure 54. Example 1: Scratching Utility Data Sets
- 130 Figure 55. Example 1: Cataloging SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and Component Libraries
- 131 Figure 56. Example 1: Volumes for Nucleus Generation
- 132 Figure 57. Example 1: Input Deck for Nucleus Generation
- 134 Figure 58. (Part 1 of 2) Example 2: Machine Configuration
- 136 Figure 59. Example 2: Initializing Volumes for DLIB01 and DLIB02
- 136 Figure 60. Example 2: Restoring DLIB01
- 137 Figure 61. Example 2: Restoring DLIB02

137 Figure 62. Example 2: Punching Members of SYS1.SAMPLIB  
137 Figure 63. Example 2: Listing Data on DLIB01 and DLIB02  
138 Figure 64. Example 2: Initializing New System Volumes  
139 Figure 65. Example 2: Generating and New System Volumes (2314)  
140 Figure 66. (Part 1 of 2) Example 2: Initializing New System Data Sets  
141 Figure 67. (Part 1 of 4) Example 2: Input Deck for Stage I  
144 Figure 68. Example 2: IBCDASDI Deck for 2303  
145 Figure 69. Example 2: Allocation on 2303  
146 Figure 70. (Part 1 of 3) Example 2: Input for I/O Device Generation  
149 Figure 71. Example 3: Generating and New System Volumes  
150 Figure 72. (Part 1 of 2) Example 3: Initializing New System Data Sets  
152 Figure 73. (Part 1 of 2) Example 3: Stage I Input Deck  
153 Figure 74. Example 3: Creating Backup of SYSTEM and LINVOL  
154 Figure 75. Example 3: Scratching Utility Data Sets  
204 Figure 76. (Part 1 of 3) Components and Generating System Data Sets  
208 Figure 77. System Data Sets  
241 Figure 78. Format of Macro Instruction Descriptions  
243 Figure 79. System Generation Macro Instructions  
288 Figure 80. Attribute Defaults for Operands Omitted on the EDIT Macro  
334 Figure 81. (Part 1 of 3) Keyword Values for IOCTRL Macro Instruction  
348 Figure 82. (Part 1 of 5) keyword Values for the IODEVICE Macro Instruction  
387 Figure 83. (Part 1 of 2) Summary of Coding Dependencies for the Shared-  
Library Feature  
417 Figure 84. (Part 1 of 4) Devices for Which Unit Addresses can be Specified in  
CONSOLE and ALTCONS Parameters of SCHEDULR  
424 Figure 85. SER Default Values for Secondary CPUs  
432 Figure 86. (Part 1 of 3) Devices for Which Unit Addresses can be Specified in  
CONSOLE and ALTCONS Parameters of SECONSLE  
447 Figure 87. SER Defaults for CPUs  
449 Figure 88. SUPRVSOR Macro Instruction Values for MFT, MVT, and M65MP  
468 Figure 89. (Part 1 of 6) Cross-References Between Macro Instructions  
484 Figure 90. (Part 1 of 3) Arrangement of Data Sets  
487 Figure 91. (Part 1 of 2) Arrangement of Data Sets for the 3330  
491 Figure 92. System Generation Error and Warning Messages

# Summary of Changes for Release 21

## NEW PROGRAMMING FEATURES

### 3420/3803 MAGNETIC TAPE SUBSYSTEM AND 3410 TAPE SUBSYSTEM

The 3420/3803 magnetic tape subsystem and 3410 tape subsystem are specified by use of the IODEVICE macro and the IOCONTRL macro. The devices, their features, and use are described in "IODEVICE," "IOCONTRL," "SORTMERG," "Dynamic Device Reconfiguration (DDR)," "SYS1.DUMP," and "Appendix A: Device Types."

### 3505 CARD READER AND 3525 CARD PUNCH WITH READ AND PRINT FEATURE

The 3505 is a card reader and the 3525 is a card punch with a read and punch feature. They are specified in the IODEVICE macro and illustrated in a maximum system configuration for the starter system in Figure 20. Their use is explained in Figure 20 "Minimum I/O Requirements," Figure 22 "Device Types," Figure 23 "Group Names Supporting IBM-Supplied Catalog Procedures" and Figures 84 and 86 (unit addresses that can be specified in CONSOLE and ALTCONS parameters of the SCHEDULR and SECONSLE macros). The 3505/3525 devices are also described in "Appendix A: Device Types."

### 2596 CARD READ PUNCH

The 2596 is functionally equivalent to the 1442 N1 card read punch. The method for specifying it is described in the IODEVICE macro. The device is described in "Appendix E: Functionally Equivalent I/O Devices."

### S/370, MODEL 135

Support for the S/370 Model 135 is described in the EMULATOR macro, the CTRLPRG macro and Figures 84 and 86 (unit address specified for CONSOLE and ALTCONS parameters of the SCHEDULR and SECONSLE macros). The machine check handler support for the S/370, Model 135 is described in "SYS1.SVCLIB." Channel check handler support for S/370, Model 135 is described in "Channel Check Handler (CCH)" and the SUPRVSOR macro. S/370 Model 135 features and storage size are specified in the CENPROCS macro. SER defaults for the Model 135 are listed in Figure 87.

### S/370, MODEL 195

The S/370, Model 205 is shown in Figure 20. "Maximum System Configuration for the Starter Operating System". It is specified in the CENPROC macro and it supports the channel-check handler option (see "Channel-Check Handler").

### RDE (Reliability Data Extractor)

RDE is a new option specified and described in the SUPRVSOR macro.

## STATUS DISPLAYS

The SECONSLE and SCHEDULR macros specify status displays for displays consoles. New parameters specifying size of the status displays, the modules that support them, features, and restrictions are given in these macros. The Multiple Console Support option describes the requirements for status displays. A new data set, "SYS1.DCMLIB," description has been added for the display control module group and a new macro description, "DCMLIB," has been added to explain how to specify the inclusion of the SYS1.DCMLIB data set into the system.

## GENERALIZED TRACE FACILITY (GTF)

Figure 76 shows from which generating system data sets GTF modules are taken and into which new system data sets the GTF modules are placed. The GTF requirements and restrictions are described in "Trace Option," "Timing Options," and the PARTITNS macro.

## SPECIFICATION CHANGES

### SERVICE AIDS

Service aids in Figure 76, "Components and Generating System Data Sets," and in the CTRLPROG and MACLIB macro has been changed from an option to a requirement.

### EXISTING OPERATING SYSTEM

The Release 21 level of linkage editor, IEHIOSUP, and IFCDIP00 must be used when a prerelease-21-level operating system is to be used to generate a new system. See "System Generation Using an Existing Operating System."

### SORTMERG SIZE PARAMETER

The values for S in the SIZE parameter formula have been changed and the description expanded.

### RESIDENT BSAM MODULES

RESIDENT=ACSMETH in the SUPRVSOR Macro: BSAM modules must be resident to provide dumps of main storage during abnormal termination and for the system management facilities in MFT. Descriptions for making these modules resident have been added to this subparameter description. Another requirement description for specifying RESIDENT=ACSMETH has also been added.

OPTIONS=LOG in the SCHEDULR macro: A description of the BSAM modules that must be resident when this subparameter is specified in an MFT system has been added.

ACCTRTN=SMF in the SCHEDULR Macro: A description of the BSAM modules that must be resident when this subparameter is specified and system management facilities is a data set on a direct-access device or a recording on tape has been added.

2803 AND 2804

A feature for the 2813 Model 3, described in Figure 80 "Keyword Values for IOCTRL Macro Instruction", has been changed. The 2804 Model 3 description in this figure has also been changed.

ACSMETH= IN THE DATAMGT MACRO

The requirement for this subparameter has been deleted.

ALTERNATE PATH RETRY (APR), DYNAMIC DEVICE RECONFIGURATION (DDR), AND CHANNEL CHECK HANDLER (CCH) OPTIONS

A description about the VARY PATH function of alternate path retry has been added to the option description, a description of an operator requirement for permanent I/O error detected by the DDR routines has been added to the DDR option description, and a note about a restriction if neither APR, DDR, nor CCH is chosen has been added to the APR and DDR option description.

SYS1.LOGREC

A description about reallocating this data with IFCDIP00 has been added to this data set description.

2250, 2305, or 3330

Specification descriptions and restrictions for the 2250, 2305, or 3330 have been added to the IODEVICE macro description.

3270 Display System

Specification descriptions for the 3270 display system has been added to the IODEVICE macro and IOCONTROL macro descriptions.

## MISCELLANEOUS TECHNICAL CHANGES

M65MP: Information in this publication for TSO with the Model 65 Multiprocessing (M65MP) configurations is for planning purposes only until that item is supported.

SYS1.DUMP: The DD statement in this data set description as to when the DD statement is used and what happens if SYS1.DUMP is not allocated but is cataloged has been changed.

SYS1.MODGEN: SYS1.MODGEN in the starter operating system has been expanded into SYS1.MODGEN and SYS1.MODGEN2.

DASDI, DUMPREST, AND RECOVERP: The description for specifying DASDI, DUMPREST, and RECOVERP in a PUNCH statement in "Testing the New System" has been expanded.

Expiration Date with CSECT Identification Records: In "Input Deck for Initialization," an explanation about the high value of the expiration date when the CSECT identification record data is listed for programs link-edited during system generation has been added.

COBOL F: All information pertaining to COBOL F has been deleted.

PCP: All information pertaining to PCP has been deleted.

EDIT Macro: The TSO EDIT macro has been rewritten and expanded.

## MISCELLANEOUS PUBLICATION CHANGES

SRL References: The titles of the books referenced in this manual have been changed.



**Introduction** \_\_\_\_\_→

**Preparation for System Generation** \_\_\_\_\_→

**The Job Stream — Stage II** \_\_\_\_\_→

**Restart Procedures** \_\_\_\_\_→

**Testing the New System** \_\_\_\_\_→

**Examples** \_\_\_\_\_→

**Program Options Descriptions** \_\_\_\_\_→

**Data Sets** \_\_\_\_\_→

**System Generation Macro Instructions** \_\_\_\_\_→

**Appendix A: Device Types** \_\_\_\_\_→

**Appendix B: Supporting Additional I/O Devices** \_\_\_\_\_→

**Appendix C: Distribution Methods** \_\_\_\_\_→

**Appendix D: System Generation Messages** \_\_\_\_\_→

**Appendix E: Functionally Equivalent I/O Devices** \_\_\_\_\_→

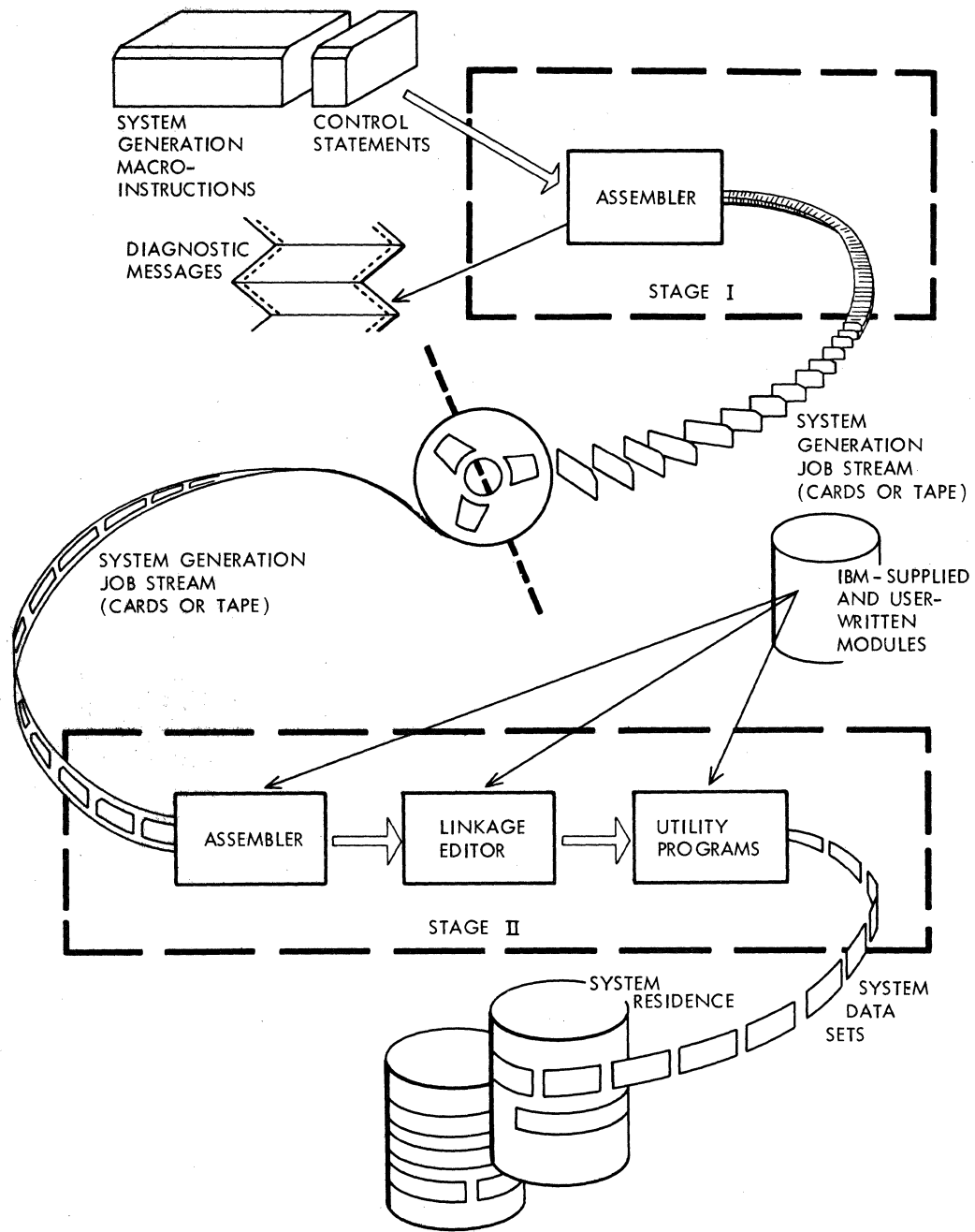


Figure 1. The System Generation Process

The Operating System is composed of modules that can be united in a variety of combinations to meet the given requirements of a particular installation. You select the programming options that meet your data processing requirements and that conform to the processing, storage, and input/output facilities of your machine configuration. System generation is the process of interpreting your selection and combining the operating system modules into the system data sets that form your installation's new operating system.

The new operating system is composed of the standard features incorporated in every operating system, those optional features selected from the distributed modules, and any additional features you provide. System generation also provides facilities for adding other features to the operating system after it has been generated.

The system generation process and the types of system generation are discussed in the following sections.

## The System Generation Process

System generation is a process that generates an operating system adapted to both the machine configuration and the data processing requirements of an installation. You specify the desired operating system through system generation macro instructions. During the system generation process, several operating system programs are used to build a new operating system according to the specifications in the macro instructions. These programs are executed under the control of an existing operating system (also called generating operating system or simply generating system throughout this publication).

An operating system is generated in two stages (see Figure 1). During Stage I, the macro instructions that describe the installation's machine configuration and the programming options desired are analyzed and used to generate a job stream. In Stage II, this job stream is processed to generate the libraries of modules that form the new operating system. These libraries contain modules supplied by IBM and, optionally, modules that you supply.

### STAGE I: PRODUCING THE JOB STREAM

Stage I consists of two phases. During the first phase, all the macro instructions are analyzed for errors. Error messages are written for each error found. If errors are not found in any of the macro instructions, a job stream is produced during the second phase. If errors are found, however, the second phase is bypassed and the job stream is not produced.

### STAGE II: PROCESSING THE JOB STREAM

During Stage II, the job stream is processed by the assembler, by the linkage editor, and by utilities. The following functions are performed:

- Selected modules are assembled.

- The linkage editor combines the modules selected for inclusion in the resident portion of the control program (nucleus).
- The linkage editor processes those selected modules to construct members of the new operating system libraries.
- Utility programs complete the construction and initialization of the libraries selected for the new operating system.

The generated operating system is then ready for use.

#### TYPES OF GENERATION

During each system generation process, you can specify one of four types of generation:

- Complete operating system generation
- Nucleus generation
- Processor/library generation
- I/O device generation

In the first type, you specify the generation of an operating system consisting of either a control program only, or a control program, language processors, and their associated libraries. The control program specified can be multiprogramming with a fixed number of tasks (MFT), multiprogramming with a variable number of tasks (MVT), or a version of MVT that controls multiprocessing and uses two model 65s (M65MP). Primary data management routines and system utilities are always provided with this type of generation. These standard features and the optional features specified are adapted to the installation's machine configuration during the generation process.

A complete operating system generation should always be performed whenever changes are to be made to the system generation options for the control program or when, in some cases, the installation's machine configuration is to be modified. A complete operating system generation may not be needed if only changes to the nucleus of the control program are to be made. In this case, a new nucleus can be added to the operating system through a nucleus generation. A complete operating system generation may not be needed if only processor(s) are to be added to the system. A processor/library generation will accomplish this. A complete operating system may not be needed if the installation's machine configuration is to be modified. The reconfiguration can be done by an I/O device generation.

In a nucleus generation, the user specifies that a nucleus is to be added to the operating system. The new nucleus must be of the same type (MFT, MVT, or M65MP) as that specified during the original complete operating system generation. That is, each nucleus of a given operating system must support the same control program, because the resident portion of the control program (nucleus) must be compatible with the nonresident portion. (SYS1.NUCLEUS is the only system data set affected by a nucleus generation.)

In a processor/library generation, you specify that processors and/or their associated libraries are to be added to the operating system. During the preparation for a processor/library generation, you must allocate space for, and, if desired, catalog any new system data set to be added to the operating system. If any of the existing system data sets are to be modified, you must have allocated sufficient space to those data sets when they were initially generated or, where permitted, provided for multiple extents in those data sets.

In an I/O device generation, you can add, delete, or change I/O devices, control units, or channels. You can add universal character set (UCS) support for the 1403 printer, change group names, change console specifications, add support for sharing direct-access devices, or change the address of any 2250 display units that are used for the graphic job processor (GJP) operations and/or 1130 subsystems to be used for the satellite graphic job processor (SGJP) operations. You cannot add the programming support for additional access methods, telecommunications, or graphics. (SYS1.NUCLEUS, SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.LOGREC are the data sets affected by an I/O Device.)

During a nucleus generation or a processor/library generation, the generating system may also be the system being modified, that is, the generating system can add a new nucleus or processors and libraries to itself. However, for an I/O device generation, the system being modified cannot be the generating system. To facilitate the description of system generation, the terms new operating system or new system are used throughout this publication to mean either the operating system generated in a complete operating system generation, or the operating system modified by a nucleus, processor/library, or I/O device generation.

Figure 2 illustrates the four types of system generation. A complete operating system generation is performed first to produce the installation's new operating system. A processor/library generation is used to add the PL/I processor and SYS1.PL1LIB to the new operating system. A nucleus generation adds a second nucleus to SYS1.NUCLEUS. An I/O device generation adds devices, control units, and channels to the system.

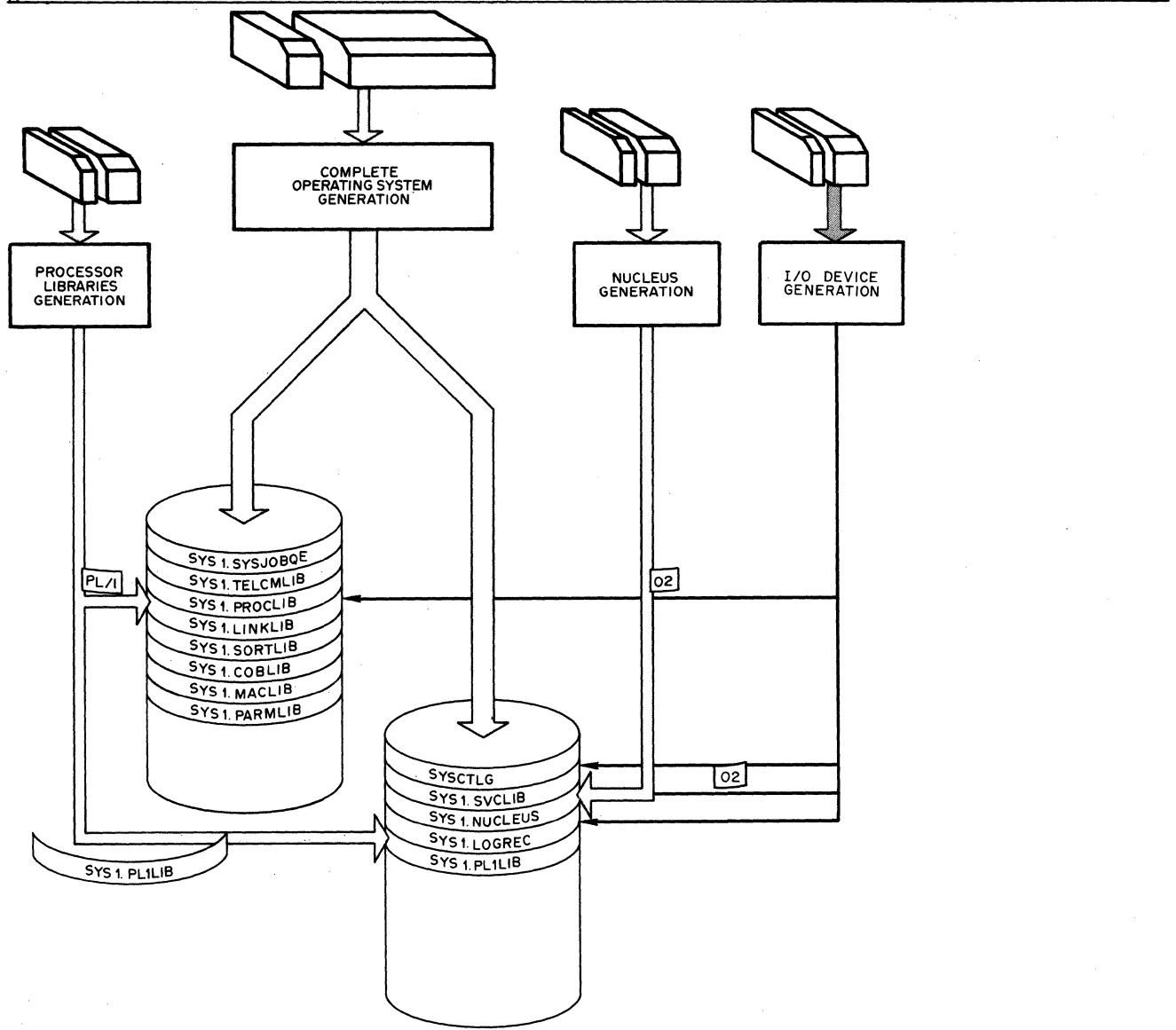


Figure 2. Types of System Generation

# Preparation for System Generation

Describes the preparatory steps necessary for a system generation and provides examples to illustrate the coding required for each step; provides the requirements for the generating system and the steps necessary to prepare the system; includes considerations for using an MFT or MVT system as the generating system; describes the procedure for adding the user-written functions to the new SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LINKLIB; the requirements and detailed operating procedures for the starter operating system packages and provides the requirements for the new or modified system and the steps required to initialize, allocate space on, and catalog any new direct-access volumes that will be used.





Before an operating system can be generated, a new system-residence volume (that volume on which the nucleus library, the SVC library, the IPL program, the SYS1.LOGREC and SYS1.ASRLIB data sets, and the volume index of the catalog are located), and any other direct-access volumes required, must be initialized. The highest level of the system catalog must be built on the new system-residence volume. Space must be allocated for the appropriate system data sets in the new operating system, and the appropriate data sets must be cataloged in the new system catalog.

Volume initialization is performed by either the IEHDASDR system utility program or the IBCDASDI independent utility program. The IEHPROGM system utility program is used to build the volume index of the system catalog, to allocate space for system data sets, and to catalog data sets. This system utility can be executed at any time after the system-residence volume is initialized.

The following paragraphs describe the initialization of the system-residence volume and other required direct-access volumes, the initialization of the system data sets, and considerations on allocating space for these data sets. Detailed descriptions of the utility programs and of the control statements they require are in OS Utilities.

## Initializing Direct-Access Volumes

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

The initialization functions are accomplished by either the IBCDASDI or the IEHDASDR utility program. IBCDASDI is self-loading and operates independently of the operating system. IEHDASDR operates under control of a Release 16 or later operating system. These programs and the control statements they require are described in OS Utilities.

In addition to performing the initialization functions, these utility programs check for defective tracks and, if any are found, assign alternate tracks and issue appropriate messages. Alternate tracks are not accepted for track 0 of the system-residence volume. (Track 0 is required for the IPL program.) If track 0 is found to be defective, another volume must be initialized for system residence.

The IPL program (IEAIPL00) is distributed with the starter system. The section "System Generation Using the Starter Operating System" describes the procedure used to punch IPL program cards for insertion into the IBCDASDI or IEHDASDR input deck whenever a system-residence volume is initialized. (IEHDASDR can also retrieve the IPL program directly from SYS1.SAMPLIB, as described in OS Utilities.)

The system-residence volume must reside on a 2301, 2303, 2305, 2311, 2314, or 3330. If it resides on a 2301 or 2314, the start address of its VTOC cannot be 0. If it resides on a 2303 or 2311, the start address of its VTOC cannot be 0 or 1.

Figure 3 is an example of the code for an input deck for initializing a system-residence volume using the IBCDASDI independent utility program. In this example, the volume to be initialized resides on an IBM 2311 Disk Storage Drive. The volume serial number to be written in the label is 111111. The volume table of contents (VTOC) starts at track 2 and is 8 tracks long.

**Note:** If the volume is being initialized for the first time, the parameter FLAGTEST=NO must be included in the DADEF statement (for 2311 and 2314 volumes only).

```

VOLINIT JOB          -SYSTEM-RESIDENCE VOLUME INITIALIZATION-
MSG      TODEV=1403,TOADDR=00E          -MESSAGE OUTPUT-
DADEF    TODEV=2311,TOADDR=191,IPL=YES,  -VOLUME           X
        VOLID=SCRATCH                   DEFINITION-
VLD      NEWVOLID=111111,OWNERID=DEPT89  -VOL LABEL DEF-
VTOCD    STRTADR=2,EXTENT=8             -VTOC DEFINITION-
IPLTXT

.
.   IPL PROGRAM (IEAIPLOO CARDS)
.
END

```

Figure 3. Initializing the System-Residence Volume With IBCDASDI

The IEHDASDR system utility program can be used to perform the initialization shown in Figure 3. Figure 4 shows the job control language and utility control statements required by IEHDASDR.

**Note:** If the volume is being initialized for the first time:

- The device must be varied off-line before the IEHDASDR step is executed.
- The DD statement that defines the volume must be omitted.
- The value of the TODD keyword of the ANALYZE statement must be the address of the device.
- The parameter FLAGTEST=NO must be included in the ANALYZE statement (for 2311 and 2314 volumes only).

```

//INIT    JOB MSGLEVEL=1          -IEHDASDR INPUT DECK-
//STEP    EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//VOL1    DD UNIT=(2311,,DEFER),DISP=OLD,          X
//        VOLUME=(PRIVATE,,SER=2311AA)
//SYSIN    DD *
        ANALYZE TODD=VOL1,VTOC=2,EXTENT=8,NEWVOLID=111111,      X
        IPLDD=SYSIN,OWNERID=BGS
        IPLTXT
.
.   IPL PROGRAM (IEAIPLOO CARDS)
.
END
/*

```

Figure 4. Initializing the System-Residence Volume With IEHDASDR

## Initializing New System Data Sets

The initialization of new system data sets is the process of allocating space to the system data sets, building the volume index of the system catalog, and cataloging system data sets in the system catalog. The contents of the system libraries are placed in the allocated space during system generation. The contents of the other data sets are placed in the allocated space during job execution in the generated operating system.

The volume index of the system catalog is built on the new system-residence volume by the IEHPROGM system utility program. This index contains the addresses of the system data sets that will form the new operating system. These data sets can be cataloged by the same utility program.

Space should be allocated for these data sets (except for SYS1.LOGREC, SYS1.ROLLOUT, SYS1.ASRLIB and, sometimes, SYS1.DUMP) by DD statements included in the input to this system utility program. The amount of space to be allocated to each system data set is shown in OS Storage Estimates.

## Input Deck for Initialization

The system data sets are allocated space and cataloged by the IEHPROGM utility program. Detailed descriptions of the control cards and functions of IEHPROGM can be found in the publication OS Utilities. The following text only describes the use of IEHPROGM for initializing the system data sets.

The input deck for IEHPROGM must contain the following:

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

```
//ddname DD DSNNAME=dsname,VOLUME=(,RETAIN,SER=serial), X
// UNIT=unit,LABEL=EXPDT=99350,SPACE=(allocation), X
// DISP(,KEEP),DCB=(see Figure 77 in the "Data Set" X
// section)
```

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

```
CATLG DSNNAME=dsname,CVOL=unit=serial,VOL=unit=serial
```

The DD and CATLG statements and examples of allocation are discussed in the following sections. For more information on the coding of parameters, refer to OS Job Control Language Reference.

### DD Statements

The DD statements in the input deck for initializing the system data sets have the following parameters:

ddname  
name of the DD statement.

**DSNAME=dsname**

name of the system data set. During the preparation for processor and libraries generation only those optional system data sets that are to be added to the system must be cataloged and allocated space.

**VOLUME=(,RETAIN,SER=serial)**

serial number of the direct-access volume on which the system data set is to reside.

**UNIT=unit**

the name of the direct-access device that can be allocated to the system data set. It is recommended that the value given to UNIT be a unit name. (Unit names are listed in Appendix A.)

**LABEL=EXPDT=99350**

the expiration date for all data sets chosen to prevent accidental deletion.

The data set protection provided by the expiration date requires additional action by the operator. If the current date is set in the generating system during system generation, the operator is required to override this current date each time a protected data set is opened.

Another, more convenient method can be used. For the system generation job (or any time the data sets are to be modified), the current date may be set at a higher value than the expiration date specified for the protected data sets. This may be done by the operator from the console, or by a card in the job stream. In either case, the current date should be reset in the generating system immediately after the completion of the system generation process. (See the "Operating Considerations" section.)

Note that when the current date is set to a high value, the Linkage Editor produces CSECT identification records containing this high value. Thus, when CSECT identification record data is later listed for the programs link-edited during system generation, it appears as if these programs were link-edited on a high value. For a complete description of CSECT identification records, see the publication OS Loader and Linkage editor.

**SPACE**

the amount of auxiliary storage to be allocated to the system data set can be obtained from OS Storage Estimates. The directory quantities for the system libraries are also shown in that publication.

Only the index of SYS1. must reside on the catalog of the system-residence volume. SYS1.NUCLEUS and SYS1.SVCLIB must be allocated space entirely on the system-residence volume. SYS1.SVCLIB may not occupy more than 1023 tracks on the system-residence volume. (The size of SYS1.LOGREC is determined during system generation and is allocated on the system-residence volume.) The maximum space that can be allocated to the remaining system data sets is one volume, except for SYS1.SYSJOBQE, which may not occupy more than 1215 tracks on a 2314 and 745 tracks on a 3330. Alternative track assignment is accepted for the system data sets. To achieve maximum efficiency in the new system, alternate tracks should not be used for SYS1.SVCLIB, SYS1.LINKLIB, and SYS1.SYSJOBQE.

**DISP=(,KEEP)**

this parameter must be coded as shown. DISP=(,CATLG) cannot be used because the data set would be cataloged in the generating system rather than in the catalog of the new system-residence volume.

**DCB**

certain system data sets require a DCB parameter. See "Data Sets."

## CATLG Statements

The CATLG statements in the input deck for initializing system data sets have the following parameters:

DSNAME=dsname  
name of the system data set to be cataloged.

CVOL=unit=serial  
specifies the unit name and serial number of the new system-residence volume. (The values must be the same as those specified in the DD statement for SYS1.NUCLEUS.)

VOL=unit=serial  
the unit name and serial number of the volume on which the system data set resides. These values must be the same as those specified in the corresponding DD statement for the system data set.

## Sample Data Set Initializations

This section contains two examples of initialization of system data sets. In the first example, all the system data sets required by a new system reside on one volume. In the second example, the system data sets of another new system are arranged on two volumes.

The numbers chosen for space allocation in these two examples are for illustrative purposes only. Space requirements for the system data sets are determined by several factors, especially the type of device used and the characteristics of the system to be generated. For example, the processors chosen for the new system affect the size of SYS1.LINKLIB. Exact-auxiliary storage requirements on various types of direct-access devices can be obtained from OS Storage Estimates.

Figure 5 is an example of an input deck for building the system catalog and for allocating space to the system data sets on one volume. It is assumed that the system-residence volume was initialized as shown in Figure 3 or Figure 4. The new system requires SYS1.MACLIB, SYS1.SVCLIB, and SYS1.LINKLIB. The optional data sets SYS1.SORTLIB, SYS1.COBLIB, SYS1.FORTLIB, and SYS1.DUMP are to be cataloged in the new system. The unit for the new system-residence volume is a 2311. The serial number of the system-residence volume is 11111.

Figure 6 is another example of an input deck for building the system catalog and for allocating space to the system data sets. The system data sets are to reside on two volumes. The unit for the system-residence volume is a 2301 and its serial number is AAA111. The unit for the second volume is a 2311 and its serial number is AAA112. It is assumed that both volumes were previously initialized. The new system will include all the optional system data sets.

The data sets SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.PARMLIB, SYS1.PROCLIB, and SYS1.LINKLIB are to be cataloged in the new system. All system data sets except SYS1.LINKLIB, SYS1.MACLIB, SYS1.IMAGELIB, SYS1.PARMLIB, SYS1.PROCLIB, SYS1.SYSVLOGX, and SYS1.SYSVLOGY reside on the system-residence volume. SYS1.LINKLIB, SYS1.MACLIB, SYS1.PARMLIB, SYS1.IMAGELIB, SYS1.PROCLIB, SYS1.SYSVLOGX, and SYS1.SYSVLOGY reside on the second volume (AAA112). SYS1.ROLLOUT is cataloged on AAA112.

```

//SYSGEN JOB MSGLEVEL=1 -ALLOCATE ON ONE VOLUME-
//STEP0 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//JOBQE DD DSNNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(120),,CONTIG) X
//SVCLIB DD DSNNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(15,1,75)), X
// LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024) X
//LINKLIB DD DSNNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(97,5,100)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625) X
//CATALOG DD DSNNAME=SYSCTLG,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(10,1)), X
// LABEL=EXPDT=99350 X
//PROCLIB DD DSNNAME=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(30,10,9)), X
// LABEL=EXPDT=99350,DCB=(RECFM=FB,BLKSIZE=3360) X
//PARMLIB DD DSNNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(5,,1),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80) X
//SORTLIB DD DSNNAME=SYS1.SORTLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(60,2,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625) X
//COBLIB DD DSNNAME=SYS1.COBLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(35,2,30)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625) X
//FORTLIB DD DSNNAME=SYS1.FORTLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(50,2,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625) X
//MACLIB DD DSNNAME=SYS1.MACLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(460,30,25)), X
// LABEL=EXPDT=99350,DCB=(RECFM=FB,BLKSIZE=3360,LRECL=80) X
//NUCLEUS DD DSNNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(30,,1),,CONTIG), X
// LABEL=EXPDT=99350 X
//SYSIN DD * -INPUT FOR CATALOGING SYSTEM DATA SETS-
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.LINKLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.MACLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.PROCLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.SORTLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.COBLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.FORTLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.SVCLIB
CATLG CVOL=2311=111111,VOL=2311=111111,DSNAME=SYS1.DUMP
/*

```

Figure 5. Initializing the System Data Sets -- One-Volume Residence on a 2311

```

//SYSGEN   JOB MSGLEVEL=1           -ALLOCATE ON TWO VOLUMES-
//ONE      EXEC PGM=IEHPRGM
//SYSPRINT DD SYSOUT=A
//JOBQE    DD DSNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,DISP=(,KEEP),SPACE=(TRK,(30),,CONTIG)
//SVCLIB   DD DSNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,DISP=(,KEEP),SPACE=(TRK,(40,10,75)), X
//          LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//CATALOG  DD DSNAME=SYS1.SYCTLG,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(2,1)), X
//          DISP=(,KEEP)
//SORTLIB  DD DSNAME=SYS1.SORTLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(15,2,40)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//COBLIB   DD DSNAME=SYS1.COBLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(10,1,30)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//FORTLIB  DD DSNAME=SYS1.FORTLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(12,2,40)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//PL1LIB   DD DSNAME=SYS1.PL1LIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(20,10,65)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//ALGLIB   DD DSNAME=SYS1.ALGLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(10,2,15)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//TELCLIB  DD DSNAME=SYS1.TELCLIB,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(10,1,10)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=20483)
//NUCLEUS  DD DSNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=AAA111), X
//          UNIT=2301,SPACE=(TRK,(10,,1),,CONTIG), X
//          DISP=(,KEEP),LABEL=EXPDT=99350
//LINKLIB  DD DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(125,10,100)), X
//          LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//MACLIB   DD DSNAME=SYS1.MACLIB,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,LABEL=EXPDT=99350,SPACE=(TRK,(440,50,25)), X
//          DISP=(,KEEP),DCB=(RECFM=FB,BLKSIZE=3360,LRECL=80)
//PROCLIB  DD DSNAME=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2301,LABEL=EXPDT=99350,SPACE=(TRK,(30,10,9)), X
//          DISP=(,KEEP),DCB=(RECFM=F,BLKSIZE=80)
//PARMLIB  DD DSNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(5,,1),,CONTIG), X
//          LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//IMAGELIB DD DSNAME=SYS1.IMAGELIB,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,DISP=(,KEEP), X
//          SPACE=(TRK,(5,,3),,CONTIG), X
//          LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024)
//ACCT     DD DSNAME=SYS1.ACCT,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,LABEL=EXPDT=99350,SPACE=(TRK,(5)), X
//          DISP=(,KEEP),DCB=(RECFM=U,BLKSIZE=3625)
//MVTLOGX  DD DSNAME=SYS1.SYSVLOGX,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,SPACE=(148,(100),,CONTIG),DISP=(,KEEP), X
//          LABEL=EXPDT=99350,DCB=(RECFM=VB,BLKSIZE=3625)
//MVTLOGY  DD DSNAME=SYS1.SYSVLOGY,VOLUME=(,RETAIN,SER=AAA112), X
//          UNIT=2311,SPACE=(148,(100),,CONTIG),DISP=(,KEEP), X
//          LABEL=EXPDT=99350,DCB=(RECFM=VB,BLKSIZE=3625)

```

Figure 6 (Part 1 of 2). Initializing the System Data Sets --  
Two-Volume Residence on a 2301 and 2311

```

//SYSIN      DD * -INPUT FOR CATALOGING SYSTEM DATA SETS-
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.PROCLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.SORTLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.COBLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.FORTLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.PL1LIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.ALGLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.TELCMLIB
CATLG       CVOL=2301=AAA111,VOL=2301=AAA111,DSNAME=SYS1.SVCLIB
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.LINKLIB
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.MACLIB
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.PARMLIB
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.SYSVLOGX
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.IMAGELIB
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.SYSVLOGY
CATLG       CVOL=2301=AAA111,VOL=2311=AAA112,DSNAME=SYS1.ROLLOUT

```

Figure 6 (Part 2 of 2). Initializing the System Data Sets --  
Two-Volume Residence on a 2301 and 2311

#### LOCATION OF SYSTEM DATA SETS

Guidelines provided in this section show various ways to allocate space to the data sets required by the new system, and to the four utility data sets required during system generation. Two of the utility data sets must reside on direct-access volumes; the other two may reside on either magnetic tape or direct-access volumes. The location of the new system data sets is determined by:

- The size of the system data sets, as given in the OS Storage Estimates.
- The size of the utility data sets required for system generation, as given in the "Input Deck for System Generation" section.
- The machine configuration, in particular the number and type of devices available in the generating system.
- The use of the generating system data sets, new system data sets, and utility data sets during system generation. The generating system data sets and utility data sets are described in the "Data Sets" section.

The following rules may be used to allocate the system data sets on the available volumes:

- During Stage I and the assembly steps of Stage II of system generation, the volumes that contain the utility data sets for system generation must be mounted at the same time.
- After the assembly steps of Stage II, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and two of the utility data sets (those that need not reside on direct-access volumes) are no longer needed and may be removed and the new system pack mounted. SYS1.MACLIB, SYS1.TSOMAC, and SYS1.TCAMMAC of the generating system may also be removed at this point if the MACLIB macro instruction is not used during system generation. If MACLIB is used, the volume that is to contain the new SYS1.MACLIB must be mounted. (See the description of the MACLIB macro instruction in the section on "System Generation Macro Instructions.")



- After the IEBCOPY step of Stage II (produced if the MACLIB macro instruction is used), SYS1.MACLIB, SYS1.TSOMAC, and SYS1.TCAMMAC of the generating system may be removed. If the volume that contains the new SYS1.MACLIB is not to contain any other new system data sets, it may also be removed.
- The volumes that contain component libraries, the new system data sets, and the volumes that contain your modules to be added to the new SYS1.NUCLEUS must be mounted before the Stage II link-edit steps are executed.
- Before the IEBCOPY step is produced, if the SVCLIB or LINKLIB, or both, macro instructions are used, the volumes that contain your modules to be added to the new SYS1.SVCLIB or SYS1.LINKLIB, or both, must be mounted. The volumes that contain the new SYS1.SVCLIB or SYS1.LINKLIB, or both, must remain mounted. The volumes that contain the component libraries and the remaining new system data sets may be removed.

If there are enough drives available, all generating system data sets, new system data sets, and system generation utility data sets should be mounted at the same time. If there are not enough drives, the volumes that contain data sets not required for a given step must be demounted and the volumes that contain the data sets required for that step must be mounted. The scheduler indicates to the operator which volume should be mounted.

The new system data sets can be arranged in different ways on one or more volumes. For best performance, it is desirable to place the system data sets on more than one volume. Whenever possible, frequently used data sets should be located on a volume other than the system-residence volume. System data sets on the same volume should be arranged according to the interaction between them; the more interaction, the closer they should be placed.

For example, for best performance, SYS1.LINKLIB should be placed on a volume other than the system-residence volume. However, if it must be on the system-residence volume, it should be adjacent to SYS1.SVCLIB, because SYS1.LINKLIB and SYS1.SVCLIB are frequently used data sets and are closely related. In this same example, space for SYS1.NUCLEUS should be allocated last, because this data set is referred to only by the IPL program.

If there are not enough drives available to achieve the desired distribution of new system data sets, the IEBCOPY utility program can be used to distribute the data sets after system generation. If SYS1.SVCLIB is changed, replaced, or moved after system generation, the IEHIOSUP utility program must be executed. If the MACLIB macro instruction was not used during system generation, the IEBCOPY utility program can be used to copy SYS1.MACLIB from the generating system to the new system.

The following pages contain examples of allocation on systems with three and four direct-access devices. In all these examples, the five utility data sets required for system generation are called SYS1.SYSUT1, SYS1.SYSUT2, SYS1.SYSUT3, SYS1.SYSUT4, and SYS1.OBJMOD. SYS1.SYSUT1 and SYS1.SYSUT2 may reside on magnetic tape.

#### Generating a New System onto One Volume Using Four 2311 Drives

Figure 7 shows the distribution of data sets using four 2311 disk storage drives and six volumes. The generating system data sets reside on DLIB01, DLIB02, DLIB03, DLIB04, and DLIB05. All the new operating system data sets reside on TARGET. The scheduler, when required, requests the demounting and mounting of the proper volumes.

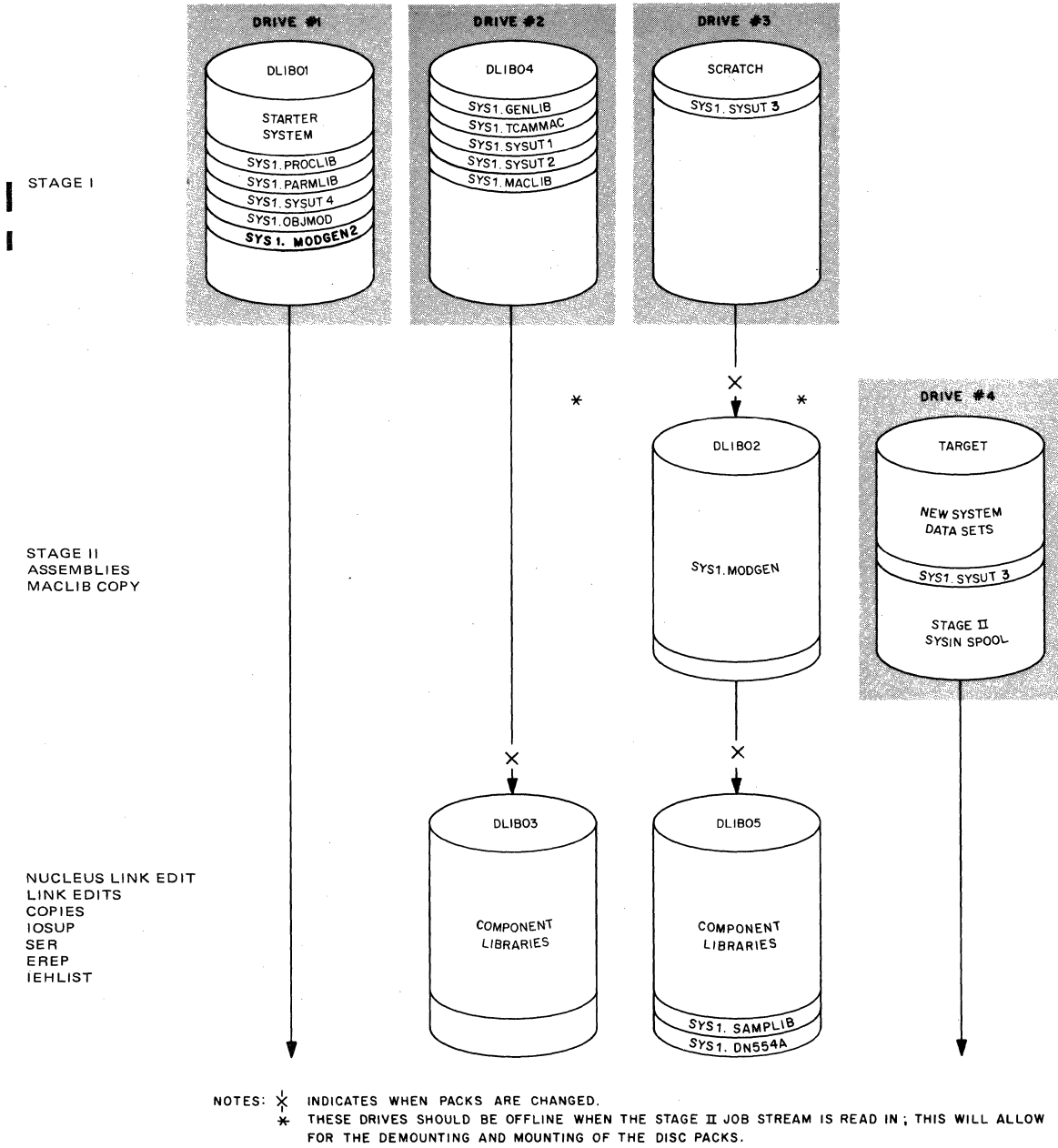
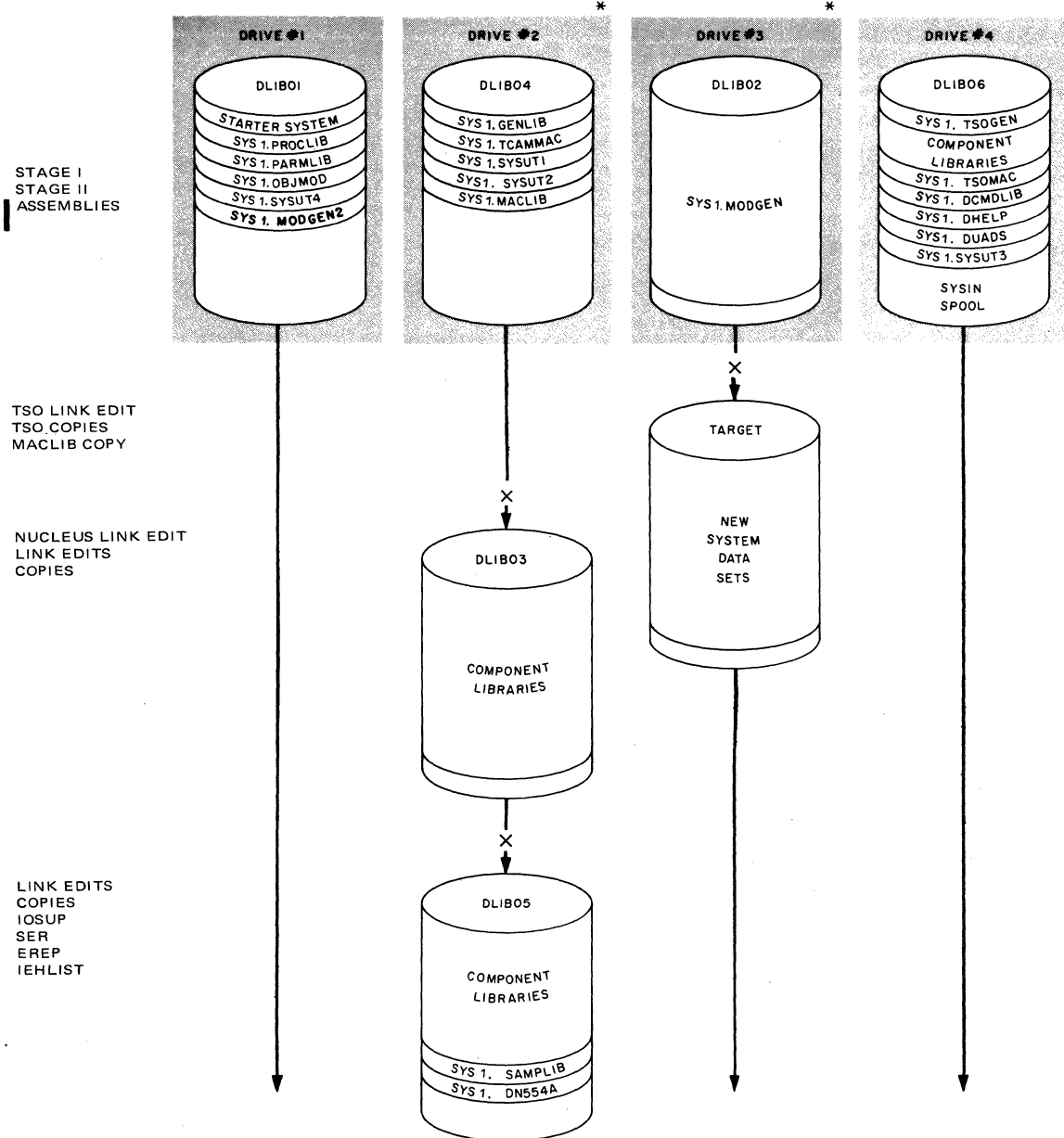


Figure 7. Example of Generation on Four Drives

Generating a New TSO System on One Volume Using Four 2311 Drives

Figure 8 shows the distribution of data sets using four 2311 disk storage drives and seven volumes. All the new system data sets reside on TARGET. The generating system data sets reside on DLIB01, DLIB02, DLIB03, DLIB04, DLIB05, and DLIB06. The scheduler, when required, requests the demounting and mounting of the proper volumes.



NOTES: X INDICATES WHEN PACKS ARE CHANGED.  
\* THESE DRIVES SHOULD BE OFFLINE WHEN THE STAGE II JOB STREAM IS READ IN; THIS WILL ALLOW FOR THE DEMOUNTING AND MOUNTING OF THE DISC PACKS.

Figure 8. Example of Generation on Four 2311 Drives

Generating a New System On One Volume Using Two 2314 Drives

Figure 9 shows the distribution of data sets using two 2314 direct-access drives. All new system data sets are located on TARGET. The generating system data sets are located on DLIB01 and DLIB02.

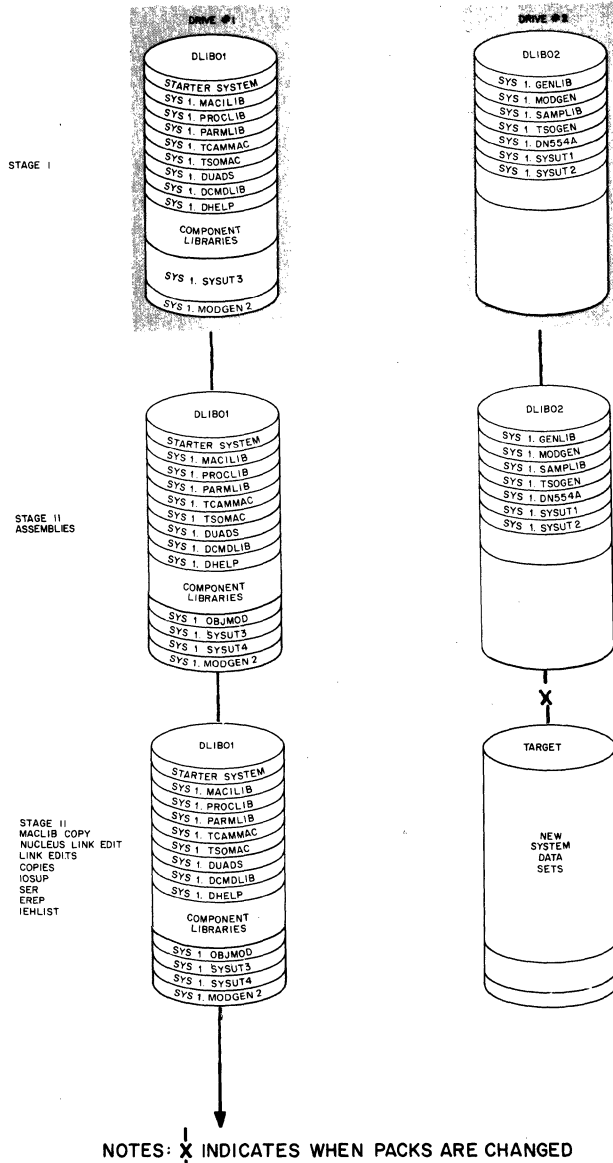


Figure 9. Example of Generation on Two 2314 Drives

Notes: SYSIN spooling should be on DLIB02 and TARGET. To force SYSIN spooling to DLIB02 and TARGET, these packs must have the lowest available addresses. The output data sets must be written directly to the printer using the direct SYSOUT writer (DSO). (See OS Job Control Language Reference for a description of DSO).

Stage I: Output must be punched cards and the utility data sets must be temporarily allocated on DLIB01 and DLIB02. (See Figure 14 for Stage I space estimates.)

Stage II Assemblies: Stage II should be separated into two job streams. The assemblies should be in the first job stream and the copies, link-edits, and utilities should be in the second job stream. Allocate and catalog the utility data sets before reading in the assembly job stream. (See Figure 14 for Stage II space estimates.)

Stage II Link Edits, Copies, and System Utilities: Allocate the new system data sets after replacing DLIB02 with TARGET. Read in the second job stream.

Generating a New System on One Volume Using Three 2314 Drives

Figure 10 shows the distribution of data sets using three 2314 direct-access drives. All new system data sets are located on TARGET. The generating system data sets are located on DLIB01 and DLIB02.

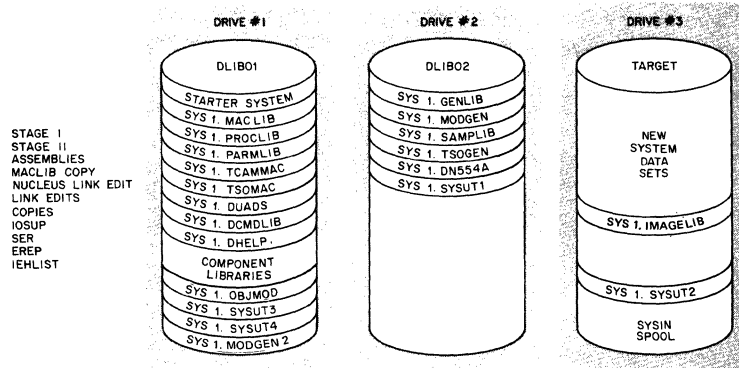


Figure 10. Example of Generation on Three 2314 Drives

Generating a New System on Two Volumes Using Four 2314 Drives

Figure 11 shows the distribution of data sets using four 2314 direct-access drives. All new system data sets are located on TARGET1 and TARGET2. The generating system data sets are located on DLIB01 and DLIB02.

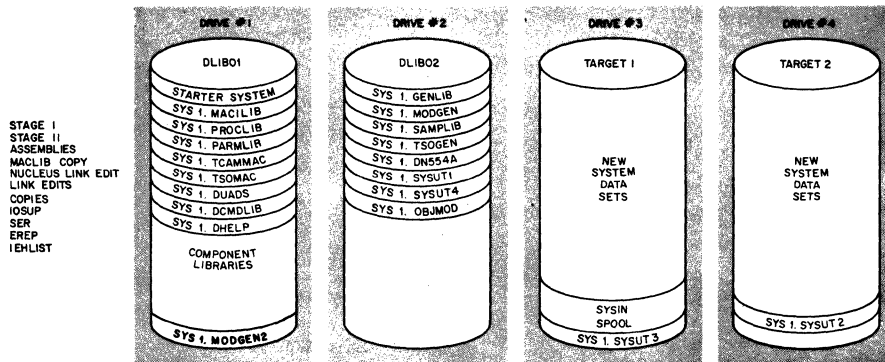


Figure 11. Example of Generation on Four 2314 Drives

## Generating a New System on One Volume Using Two 3330 Drives

Figure 12 shows the distribution of data sets using two 3330 direct-access drives. All new system data sets are located on TARGET. The generating system data sets are located on DLIB01.

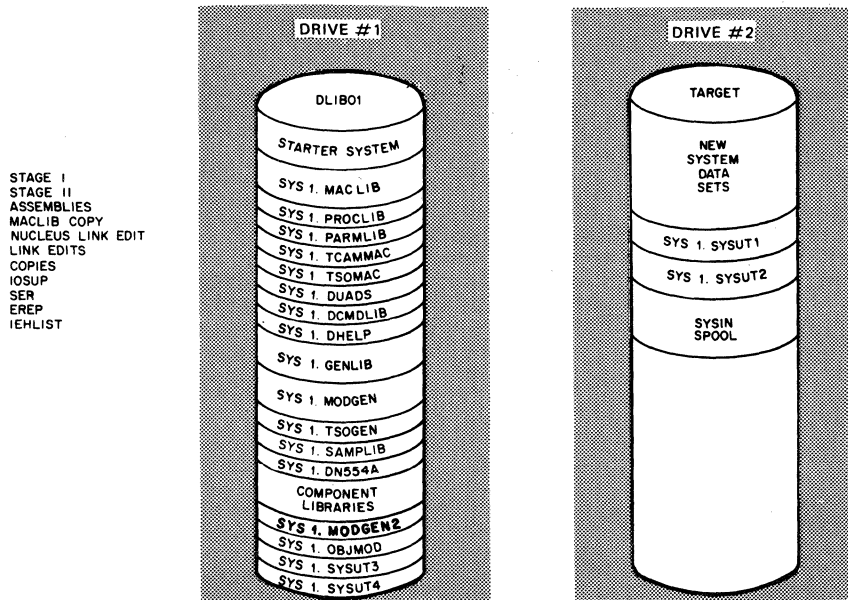


Figure 12. Example of Generation on Two 3330 Drives

## Input Deck For System Generation

The input deck required for Stage I of system generation consists of job control language statements and system generation macro instructions. The sequence of the deck and the job control language statements are shown in Figure 13. This figure represents a four-drive system generation. For other generations, only the underlined values need vary. (In Figure 13, the utility data sets are allocated on 2311 volumes whose serial numbers are DLIB01, DLIB04, and TARGET, and unit 182 if a 9-track magnetic tape is available.) It is recommended that the values given to the UNIT keywords of the DD statements be device types. (See Appendix A.)

The first statement of the deck is an EXEC statement indicating that the generation process immediately follows the catalog building step described in the previous section. Alternatively, system generation can be defined as an independent job.

The three DD statements named SYSUT1, SYSUT2, and SYSUT3 in Step 1 of the example in Figure 13 are used to allocate space to the three utility data sets required for Stage I of the system generation process. The five DD statements SYSUT1, SYSUT2, SYSUT3, SYSUT4, and OBJPDS shown in Step 2 are used to allocate space to the five utility data sets required in Stage II of the system generation process.

These data sets are cataloged as SYS1.name in the generating system, where the value of name cannot exceed eight alphanumeric characters, the first of which must be alphabetic. These names (SYS1.name) must also be specified as the value of the corresponding keywords (OBJPDS, UT1SDS, UT2SDS, UT3SDS and UT4SDS)

in the GENERATE macro instruction. The data set defined by the OBJPDS DD statement must be a partitioned data set. The other four data sets must be sequential data sets, of which the two specified by the SYSUT3 and SYSUT4 DD statement must reside on a direct-access volume.

---

```

//STEP1 EXEC PGM=ASMBLR -STAGE 1 INPUT DECK-
//SYSLIB DD DSN=SYS1.GENLIB,DISP=OLD
//SYSUT1 DD DSN=SYS1.UT1,VOLUME=(,RETAIN,SER=DLIB04), X
// UNIT=2311,SPACE=(TRK,(350,10)),DISP=(NEW,DELETE)
//SYSUT2 DD DSN=SYS1.UT2,VOLUME=(,RETAIN,SER=DLIB01), X
// UNIT=2311,SPACE=(TRK,(240,10)),DISP=(NEW,DELETE)
//SYSUT3 DD DSN=SYS1.UT3,VOLUME=(,RETAIN,SER=DLIB04), X
// UNIT=2311,SPACE=(TRK,(400,10)),DISP=(NEW,DELETE)
//SYSPUNCH DD UNIT=182,LABEL=(,NL)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

. System Generation Macro Instructions

/*
//STEP2 EXEC PGM=IEHPRGM -ALLOCATE WORK FILES FOR STAGE 2-
//SYSUT1 DD DSN=SYS1.UT1,VOLUME=(,RETAIN,SER=DLIB04) X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(50,10))
//SYSUT2 DD DSN=SYS1.UT2,VOLUME=(,RETAIN,SER=TARGET), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(50,10))

//SYSUT3 DD DSN=SYS1.UT3,VOLUME=(,RETAIN,SER=(DLIB01), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(150,10))
//SYSUT4 DD DSN=SYS1.UT4,VOLUME=(,RETAIN,SER=(DLIB01), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(5,10))
//OBJPDS DD DSN=SYS1.OBJMOD,VOLUME=(,RETAIN,SER=DLIB01), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(60,10,12))
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
// START RDR,182 (Optional statement -- see text)

```

**Note:** Underlined values represent variables. All other values must be coded as shown. The continuation characters are in column 72.

Figure 13. Input Deck Organization for System Generation

Figure 14 shows the values that may be used for the SPACE keyword in the utility data set's DD statements according to the type of direct-access device on which they may reside. The values shown in the table are for a minimum system configuration. To determine if there is enough space available in the direct-access volume, list its volume table of contents (VTOC) before Stage I using the IEHLIST utility program.

If magnetic tape drives are available, the data sets specified by the SYSUT1 and SYSUT2 DD statements can be assigned to 9-track magnetic tape. If only one utility data set is to reside on magnetic tape, the one specified by the SYSUT1 DD statement should be chosen. If the data sets defined by the SYSUT1 or SYSUT2 DD statements reside on unlabeled magnetic tape, LABEL=(,NL) must be specified in the corresponding DD statement, and NL must be specified in the UT1SDS or UT2SDS keyword of the GENERATE macro instruction.

Device Type	DD Statement				
	SYSUT1	SYSUT2	SYSUT3	SYSUT4	OBJPDS
<u>Stage I</u>					
2311	350,10	240,10	400,10		
2301	65,2	45,2	80,2		
2303	260,10	180,10	300,10		
2305	150,5	100,5	160,5		
2314	240,5	160,5	280,5		
3330	150,5	100,5	160,5		
<u>Stage II</u>					
2311	50,10	50,10	150,10	5,10	60,10,12
2301	10,2	10,2	30,2	1,5	15,2,12
2303	35,10	35,10	120,10	3,5	60,5,12
2305	20,5	20,5	70,5	2,5	30,5,12
2314	50,5	50,5	100,5	3,5	45,5,12
3330	20,5	20,5	70,5	2,5	30,5,12

Figure 14. Space Allocation (in Tracks) for Utility Data Sets

The DD statement named SYSPUNCH defines the data set that is to contain the job stream produced during Stage I of system generation. If any error messages (see Appendix D) occur during system generation, the job stream is not produced. After a successful completion of Stage I, the job stream produced becomes the input to Stage II. (The Stage I output should be saved after system generation for maintenance purposes.)

If the device defined by the SYSPUNCH DD statement is a card punch, the operator is required to place the cards in an input device and to issue a START RDR command for that device. However, operator intervention can be eliminated by making the output (SYSPUNCH) of Stage I become the input to Stage II. If the value given to UNIT in the SYSPUNCH DD statement is the unit address of a magnetic tape drive, this can be accomplished by inserting a // START RDR,xxx statement after the /\* card of the input deck (See Figure 13). In this statement, xxx is the unit address given to UNIT in the SYSPUNCH DD statement.

If there were any errors during Stage I, an end-of-file condition will be detected and the reader will be closed.

Notice that 7-track tapes can be used for SYSPUNCH. DEN=0 and TRTCH=C are the defaults for the system; these should be specified in the DCB parameter of the DD statement. The "Operating Considerations" section indicates some of the conditions that require operator intervention, and shows sample console listings from system generation processes.

**Note:** The input deck described is the input to Stage I of the system generation process. During Stage I, a job stream that serves as the input for Stage II is produced. After a satisfactory completion of Stage I, the beginning of Stage II is a logical restart point. If any errors that require restarting



from the beginning of Stage I are made, the data sets specified by the OBJPDS, SYSUT1, SYSUT2, and SYSUT3 DD statements may have to be scratched and uncataloged before restarting. For more information, refer to the "Restart Procedures" section.

## Including IEBUPDAT in the New System

IBM supplies two system utility programs that can be used to update sequential or partitioned data sets: IEBUPDAT and IEBUPDTE. Only IEBUPDTE is incorporated in SYS1.LINKLIB of the new operating system as a result of the system generation process. Because only IEBUPDAT can handle alphabetic characters in the high-order position of the sequence field and flagged output, it may be necessary to include this utility program in your new system.

To include IEBUPDAT in the new system, execute a separate job that will link-edit IEBUPDAT from SYS1.UT506 into SYS1.LINKLIB of the new operating system as soon as system generation is complete. At this time, SYS1.UT506 must still be mounted and cataloged. You must allocate additional space for this data set when SYS1.LINKLIB is initialized for the new system. (See the section on "Initializing System Data Sets.")

Figure 15 shows the required coding. You must provide the data for two of the parameters in the SYSLMOD DD statement: the volume name and unit type for SYS1.LINKLIB of the new operating system.

---

```
//LKED      JOB   OO,IEBUPDAT,MSGLEVEL=1
//STEP1     EXEC  PGM=IEWL,PARM='REUS,LIST,LET,XREF,DC,NCAL'
//SYSUT1    DD   UNIT=SYSDA,SPACE=(TRK,(10,10))
//SYSLMOD   DD   DSNAME=SYS1.LINKLIB,VOLUME=SER=LINKVOL,UNIT=2311,      X
//          DD   DISP=OLD
//UT506     DD   DSNAME=SYS1.UT506,DISP=OLD
//SYSPRINT  DD   SYSOUT=A
//SYSLIN    DD   *
             INCLUDE UT506(IEBUPDAT)
             NAME    IEBUPDAT(R)
/*
```

Note: Underlined values represent variables. All other values must be coded as shown. The continuation character is in column 72.

---

Figure 15. Input Deck for Including IEBUPDAT in the New System

## Adding User-written Functions

The system generation process provides facilities for adding your own functions to the new SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LINKLIB. These user-written functions must be load modules residing in a cataloged partitioned data set in the generating system; that is, each function must be compiled, link-edited, and placed in a cataloged partitioned data set before system generation. (Each load module must be a member of the data set.) The name of the partitioned data set must be of the form SYS1.name. The name cannot exceed eight alphanumeric characters, the first of which must be alphabetic. The name of the partitioned data set and of the members that contain your functions are specified with system generation macro instructions.

The RESMODS macro instruction specifies load modules, such as types 1 and 2 SVC routines, to be added to SYS1.NUCLEUS during the first link-edit step in Stage II.

The LINKLIB macro instruction specifies load modules, such as accounting routines, to be added to SYS1.LINKLIB during an IEBCOPY step in Stage II. These load modules become members of SYS1.LINKLIB.

The SVCLIB macro instruction specifies load modules, such as types 3 and 4 SVC routines and nonstandard label routines, to be added to SYS1.SVCLIB during an IEBCOPY step in Stage II. These load modules become members of SYS1.SVCLIB.

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

The RESMODS, LINKLIB, SVCLIB, and SVCTABLE macro instructions are described in the "System Generation Macro Instructions" section. Accounting routines, SVCs, and nonstandard label routines are described in OS MVT Guide and OS MFT Guide.

In the example shown in Figure 16, a user-written function is added to the new operating system. In this example, a CSECT is to be added to SYS1.NUCLEUS. This CSECT consists of a series of constants that describe the nucleus to be generated. During the first step, the CSECT is assembled and placed in a temporary data set (DSNAME=&LOADSET). The CSECT is link-edited during the second step and the resulting module becomes member ID of the SYS1.USER data set. SYS1.USER is a partitioned data set residing on volume 333333; it is cataloged in the generating system. During system generation, the CSECT will be included in the nucleus by the following macro instruction:

---

```
RESMODS    PDS=SYS1.USER, MEMBERS=ID
```

---

For a more detailed description of the control statements required by the assembler and the linkage editor, refer to OS Assembler (F) Programmer's Guide.

---

```
//USER      JOB MSGLEVEL=1
//STEP1     EXEC PGM=ASMBLR, PARM='NODECK,LOAD,LIST,NOTEST,NOXREF,NORENT'
//SYSLIB    DD  DSNAME=SYS1.MACLIB, DISP=OLD
//SYSUT1    DD  UNIT=SYSSQ, SPACE=(1700,(400,50))
//SYSUT2    DD  UNIT=SYSSQ, SPACE=(1700,(400,50))
//SYSUT3    DD  UNIT=(SYSSQ, SEP=(SYSUT1, SYSUT2, SYSLIB)),           X
//          SPACE=(1700,(400,50))
//SYSPRINT  DD  SYSOUT=A
//SYSGO     DD  DSNAME=&LOADSET, UNIT=SYSSQ, SPACE=(80,(200,50)),     X
//          DISP=(MOD,PASS)
//SYSIN     DD  *
ID          CSECT
DC  C'XXXXXXXXXX-NUCLEUS ID CSECT-XXXXXXXXXX'
DC  C'OPERATING SYSTEM GENERATED--5/20/67'
DC  C'OWNER--DEPT. D58'
DC  C'SUPPORTS--MFT-ALL ACCESS METHODS'
DC  C'NUCLEUS--01'
DC  C'XXXXXXXXXX-END ID CSECT-XXXXXXXXXX'
END

/*
//STEP2     EXEC PGM=IEWL, PARM=(XREF,LIST,NCAL)
//SYSLIN    DD  DSNAME=&LOADSET, DISP=(OLD,DELETE)
//SYSLMOD   DD  DSNAME=SYS1.USER(ID), UNIT=2311, DISP=(,CATLG),       X
//          VOLUME=(,RETAIN,SER=333333),                             X
//          SPACE=(1024,(50,20,5))
//SYSUT1    DD  UNIT=(SYSDA, SEP=(SYSLIN, SYSLMOD)),                 X
//          SPACE=(1024,(50,20,5))
//SYSPRINT  DD  SYSOUT=A
//
//
```

---

Figure 16. Preparing a User-Written Load Module

## Considerations for MFT or MVT (Including M65MP) Generating Systems

The following items must be considered for an MFT or MVT (including M65MP) generating system:

1. A system generation needs about 2500 tracks on 2311s for Stage II output data sets (SYSRINT). The required space may be on more than one volume. The type and number of volumes that can be used must be specified with the SYOUTDV keyword of the GENERATE macro instruction. All output data sets must stay mounted throughout Stage II. For MFT or MVT generating systems of Release 19 or later, the direct system output (DSO) procedure can be used to place the output directly on the printer or a magnetic tape.
2. It is recommended that all volumes used specifically for system generation be private. This can be accomplished by using the PRESRES volume characteristics list of SYS1.PARMLIB of the generating system. (For information about the PRESRES list, see OS MFT Guide and OS MVT Guide.)
3. The SYS1.SYSJOBQE of the MFT or MVT (including M65MP) generating system should be large enough for system generation. An average-sized system generation needs about 300 tracks on a 2311.
4. The SYS1.SYSJOBQE of the MFT or MVT (including M65MP) generating system must have about 960 176-byte records allocated for the initiator.
5. A JOB statement providing the job priority and region size should be included in the input deck and JOBSTMT=SUPPLIED should be specified in the GENERATE macro instruction.
6. For creation of an operating system, the partition or region size required for Stage I depends on the global pack used. The suggested sizes are:
  - 54K if SGGBLPAK is used. This pack is the one usually used in the starter operating system.
  - 56K if SGPAK248 is used. This pack is used if more than 96 but not more than 248 devices are defined.
  - 58K if SGPAK768 is used. This pack is used if more than 248 but not more than 768 devices are defined.
  - 85K if SGPAK768 and 3330 direct-access devices are used.

The requirements for Stage II vary according to the types of components in the generating system.

Note: For more information about the global packs, refer to Appendix B of this publication.

## System Generation Using an Existing Operating System

Your Operating System can be used as a generation system if it meets certain requirements. The data set requirements can be found in the "Data Sets" section and MFT and MVT considerations in the section, "Considerations for MFT or MVT (including M65MP) Generating Systems." When you use your own operating system you can:

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

When you use the first method, you can perform some modifications on your operating system while it is being used as a generating system. When you use the second method, you can execute the system generation as just another job in the job stream. You cannot use the second method, however, to perform modifications on your operating system while it is being used as a generating system.

Notes:

When using an existing operating system prior to Release 21, the following procedures must be followed for Stage II to ensure that the release 21 level of linkage editor, IEHIOSUP, and IFCDIP00 are used to generate the new system.

- The volume containing SYS1.LINKLIB in the starter operating system must be mounted during Stage II.
- Supply your own job card for Stage II.
- Supply a JOBLIB card. The JOBLIB card points to SYS1.LINKLIB in the Release 21 starter operating system.
- Use JOBSTMT=SUPPLIED in the GENERATE macro.

The job card and the JOBLIB card must immediately precede the GENERATE macro as shown below.

```
                REPRO
//JOB CARD JOB   (ACCT123,Dept), PROGRAMMER,MSGLEVEL=(1,0)
                REPRO
//JOB LIB  DD    DSN=SYS1.LINKLIB,VOL=SER=DLIB01,UNIT=xxxx,      X
                DISP=SHR
                GENERATE JOBSTMT=SUPPLIED
```

where:

xxxx is the unit on which the starter operating system resides, such as a 2314.

- Copy the module named IGG019C8 from SYS1.SVCLIB in the starter operating system to SYS1SVCLIB in the generating system.

SYSTEM GENERATION AS THE ONLY JOB (METHOD 1)

The preparatory steps for this method are:

- Catalog the required data sets.
- Plan for SYS1PROCLIB for the new operating system.
- Allocate and catalog the data sets for the new operating system.
- Allocate and catalog the utility data sets.
- Obtain the catalog utility program (CTLG2311, CTLG2314, or CTLG3330) and the uncatalog utility program (UNCT2311, UNCT2314, or UNCT3330).

These preparatory steps are described in the following sections.

Cataloging Data Sets

Cataloging SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and the component libraries: You must catalog SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and each required component library in your operating system catalog. Figure 76 contains the list of component library names in the starter operating system package.

Although you can catalog SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and selected component libraries by using the IEHPROGM utility program, an easier method is to use one of three utility programs -- CTLG2311, CTLG2314, or CTLG3330 -- and catalog all the data sets. The utility program selected depends on whether the starter operating system package being used for the generation is for the 2311, 2314, or the 3330. Then when the system generation process is finished, you can use either the UNCT2311, UNCT2314, or the UNCT3330 utility program to eliminate all the catalog entries that are no longer needed in the generating system.

Because each catalog entry must contain the address of the appropriate starter operating system volume, you must list the catalog of the starter system using the IEHLIST utility program. The control statements for IEHLIST that will list the catalog of the 2311 starter operating system are shown in Figure 17. If you use the 2314 or 3330 starter operating system, change the UNIT keyword parameter in the DD statement to UNIT=2314 or UNIT=3330 and the VOL keyword parameter in the LISTCTLG statement to VOL=2314=DLIB01, or VOL=3330=DLIB01.

---

```

//LISTCTLG JOB ACCT, PROGRAMMER,MSGLEVEL=1
//STEPS EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//SYSRS DD UNIT=2311,DISP=(OLD,KEEP),VOLUME=SER=DLIB01
//SYSIN DD *
LISTCTLG VOL=2311=DLIB01
/*

```

---

Figure 17. Input Deck to List the Starter Operating System Catalog

Obtaining and Cataloging SYS1.MACLIB, SYS1.PARMLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.PL1LIB, SYS1.SORTLIB, SYS1.TSOMAC, SYS1.TCAMMAC, SYS1.DUADS, SYS1.DHELP, and SYS1.DCMDLIB: You must have a SYS1.PARMLIB in your new or modified system. If you want the PARMLIB from your operating system, then no special action is necessary. If you want the PARMLIB from the starter operating system, you must first uncatalog the PARMLIB in your system catalog and then catalog the PARMLIB from the starter operating system.

SYS1.PARMLIB contains the RAM system parameter list (see "Required and Optional Data Sets for Generating Systems"). If the 3330 or 2305 is being used when generating a new operating system with an existing system, access methods IGG019C0, IGG019C4, IGG019EK, IGG019FN, and IGG019FP must be added to the RAM list. This can be done by using PARMLIB from the starter operating system as described above. After the above access method modules have been added to the RAM list, use the IPL procedure, step 16, "3330 Package -Tape Distribution."

You must also have cataloged in your operating system a SYS1.MACLIB, SYS1.TSOMAC, and SYS1.TCAMMAC that are of the same release level as SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and the component libraries being used for the generation. If you want SYS1.COBLIB (for COBOL E), SYS1.PL1LIB, SYS1.SORTLIB, SYS1.HELP, SYS1.UADS, and/or SYS1.CMDLIB in the new or modified system, you must also catalog them in your operating system; they must also be of the same release level as SYS1.GENLIB, SYS1.MODGEN, and the component libraries being used.

If your operating system has SYS1.MACLIB, SYS1.COBLIB (for COBOL E), SYS1.PL1LIB, and/or SYS1.SORTLIB of a release level other than the level of the starter operating system volumes being used, then you must uncatalog those data sets before the substitute library or libraries are cataloged.

If you want SYS1.FORTLIB in the new or modified system, it is not necessary to locate a FORTLIB of a particular release level. If your operating system has a FORTLIB, no action is necessary. If your system does not have a FORTLIB and you want one in the new or modified system, define it as a null data set and catalog it in your operating system catalog.

Note 1: If your operating system does not have the required MACLIB or any of the other data sets that are needed (COBLIB, PL1LIB, or SORTLIB), obtain them from any system that is the correct release level. Use the IEHPROGM utility program to catalog them in your operating system.

Note 2: If there are not enough drives available in your machine configuration to hold all the generating system data sets, the new system data sets, and the various additional data sets required, then determine if there is any space available on any of your system volumes. If there is enough space to hold one or more of the data sets, then use the IEBCOPY utility program to copy the data sets onto the volume or volumes. If this is not possible, then you will have to demount and mount volumes during the system generation process. (For examples of data set allocation, see Figures 7 through 11 in this section.)

### Planning for SYS1.PROCLIB

If you want the SYS1.PROCLIB from the starter operating system, you must copy it into the new system after the system generation process is complete. If you specify the system generation macro instruction PROCLIB, the SYS1.PROCLIB from your operating system will be included in the new or modified system, not the one from the starter operating system.

### Allocating and Cataloging the Data Sets for the New System

You must prepare the appropriate DD and CATLG statements for all the system data sets for the new or modified system that require them. Use the IEHPROGM utility program to allocate and catalog the data sets. (See the examples provided in "Input Deck for Initialization" and in "Location of System Data Sets" in this section.)

The names for these system data sets must have the qualifier SYS1 preceding their simple names. You can specify INDEX=SYS1 in the GENERATE macro instruction, but because this form is the default option, you don't have to code it.

### Allocating and Cataloging the Utility Data Sets

You must allocate space for and catalog five utility data sets in your operating system by means of JCL statements in your system generation input deck. (Detailed information about specifying these data sets is found in "Input Deck for System Generation" in this section.)

Four of these data sets must be sequential data sets and are used by the assembler, the linkage editor, and the utility programs during system generation. The fifth data set must be a partitioned data set and is used for the storage of object modules assembled during system generation. Two of the sequential data sets and the partitioned data set must reside on a direct-access volume.

### SYSTEM GENERATION AS ONE JOB IN THE JOB STREAM (METHOD 2)

This method of generating a system is used when you include the system generation job with other jobs in the job stream. The system generation program is executed in a multiprogramming system, permitting you to have more than one system generation job in the same job stream. The preparatory steps for this method are:

- Catalog the required data sets.
- Rename SYS1.MACLIB, SYS1.PARMLIB, and SYS1.PROCLIB in the starter operating system and catalog them by the new names. Also rename SYS1.COBLIB, SYS1.FORTLIB, SYS1.PL1LIB, and/or SYS1.SORTLIB and catalog them by the new names if you want them in the new system. Catalog SYS1.DHELP, SYS1.DUADS, and SYS1.DCMDLIB if you want them; you do not have to rename them.
- Name selected data sets in the new or modified operating system with the qualifier used in the INDEX keyword parameter of the GENERATE macro instruction.
- Obtain the catalog utility program (CTLG2311, CTLG2314, or CTLG3330) and the uncatalog utility program (UNCT2311, UNCT2314 or UNCT3330).
- Allocate and catalog the data sets for the new system.
- Allocate and catalog the utility data sets.
- Restore the data set names in the starter operating system catalog.

Note 1: When the system generation job is just one job in the job stream, none of the system data sets in your operating system have to be uncataloged for that particular system generation job. In addition, the data set SYS1.PROCLIB in the starter operating system can be included in the new system during the system generation process. You cannot use this method, however, if you are modifying your own operating system.

Note 2: When you have more than one system generation in the same job stream, be sure to:

- Specify a different qualifier in the INDEX keyword parameter for each GENERATE macro instruction.
- Specify five utility data sets for each generation in the job stream. Specify a unique name for each utility data set, because there cannot be any duplicate data set names in the job stream.

### Cataloging the Required Data Sets

Cataloging SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and the Component Libraries: You must catalog SYS1.GENLIB, SYS1.MODGEN, SYS1.TSOGEN, and the component libraries required for a particular system generation in your operating system catalog.

Although you can catalog SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and selected component libraries by using the IEHPROGM utility program, an easier method is to use one of three utility programs -- CTLG2311, CTLG2314, or CTLG3330 -- and catalog all the data sets. The utility program selected depends on whether the starter operating system package being used for the generation is for the 2311, 2314 or the 3330. Then when the system generation process is finished, you can use either the UNCT2311, UNCT2314, or the UNCT3330 utility program to eliminate all the catalog entries that are no longer needed in the generating system.

In addition to cataloging the data sets mentioned above, the utility programs perform another required function -- they rename (CTLG2311, CTLG2314, CTLG3330) and restore (UNCT2311, UNCT2314, or UNCT3330) the names of seven data sets in the starter operating system catalog. (See the next topic.) The method for obtaining these utility programs is described later.

Renaming SYS1.MACLIB, SYS1.PARMLIB, SYS1.PROCLIB, SYS1.FORTLIB, SYS1.PL1LIB, SYS1.COBLIB, and/or SYS1.SORTLIB: You must rename the system data sets (SYS1.MACLIB, SYS1.PARMLIB, and SYS1.PROCLIB) in the starter operating system and catalog them in the generating system catalog as SYS1.DMACLIB, SYS1.DPARMLIB, and SYS1.DPROCLIB for the duration of the system generation process.

If you want SYS1.COBLIB, SYS1.FORTLIB, SYS1.PL1LIB, and/or SYS1.SORTLIB in the new operating system, then you must rename these data sets in the starter operating system and catalog them in the generating system catalog as SYS1.DCOBLIB, SYS1.DFORTLIB, SYS1.DPL1LIB, and/or SYS1.DSORTLIB.

The easiest method is to use either the CTLG2311, CTLG2314, or the CTLG3330 utility program to rename all seven data sets and do the cataloging described in the preceding topic before beginning the system generation process. Then when the system generation process is finished you can restore all seven names using either the UNCT2311, UNCT2314, or the UNCT3330 utility program. The method for obtaining these utility programs is described in the next section.

SYS1.FORTLIB With User-written Modules: If the SYS1.FORTLIB in your operating system contains your own modules that you want in the new operating system, then you must place them in SYS1.DFORTLIB. Use the IEBCOPY utility program to transfer the contents of the entire data set or to transfer just your own modules. This copying can be done after either one of the utilities, CTLG2311, CTLG2314, or CTLG3330, has been run.

MACLIB, PARMLIB, and PROCLIB in the Starter Operating System: The SYS1.MACLIB, SYS1.PARMLIB, and SYS1.PROCLIB in the starter operating system are the ones that are incorporated into the new operating system. If you want your MACLIB, PARMLIB, and/or PROCLIB in the new operating system, use the IEBCOPY utility program to include them in the new system after system generation is over.

#### Obtaining the Catalog and Uncatalog Utility Programs

All six utility programs to catalog and uncatalog are in the starter operating system package as members of SYS1.SAMPLIB. They are in card format and you can punch them out by using the IEBPTPCH utility program.

Figure 18 shows the control statements required to punch the utility programs for the 2311 starter operating system package from SYS1.SAMPLIB. The underlined value must agree with the device type of a punch unit in your installation and, therefore, may be different from the value shown. If you want the utility programs for the 2314 or 3330 starter operating system package, just substitute 2314 or 3330 wherever 2311 appears in Figure 18.

---

```
//PUNCH      JOB      MSGLEVEL=1
//           EXEC      PGM=IEBPTPCH
//SYSUT1     DD        DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),          X
//           DD        UNIT=2311,VOLUME=SER=DLIB03
//SYSUT2     DD        UNIT=2540-2
//SYSPRINT   DD        SYSOUT=A
//SYSIN      DD        *
              PUNCH     TYPORG=PO,MAXNAME=2
              MEMBER    NAME=CTLG2311
              MEMBER    NAME=UNCT2311
/*
```

---

Figure 18. Input Deck for Punching the Catalog and Uncatalog Utility Programs



## Allocating and Cataloging the Data Sets for the New System

You must prepare the appropriate DD and CATLG statements for all the system data sets that require them. Use the IEHPROGM utility program to allocate and catalog the data sets. (See the examples provided in "Input Deck for Initialization" and in "Location of System Data Sets" in this section.)

If any of the following seventeen data sets are selected for the new system, you must define them in DD statements with the user-supplied qualifier that has been specified in the INDEX keyword parameter of the GENERATE macro instruction: ALGLIB, CMDLIB, COBLIB, DCMLIB, FORTLIB, HELP, IMAGELIB, LINKLIB, MACLIB, NUCLEUS, PARMLIB, PL1LIB, PROCLIB, SORTLIB, SVCLIB, TELCLIB, UADS. All other data sets you select are defined with SYS1 as the qualifier. For example, if INDEX=TEMP is specified, then the procedure library (PROCLIB) is defined in the DD statement as DSNAME=TEMP.PROCLIB, but the system job queue (SYSJOBQE) would be specified as DSNAME=SYS1.SYSJOBQE.

The use of your qualifier permits a data set to have a temporary, but unique, identification for the duration of the system generation process. At the end of the system generation process, the qualifier you supply is replaced by SYS1 by the system generation program.

You write all the catalog statements in the usual way -- using SYS1 as the qualifier for each entry in the catalog. (The qualifier you supply is only used in the DD statements -- never in the CATLG statements.) When the system generation process has been completed, the catalog and the system data sets for the new system are ready to be used.

Figure 19 contains an example of an input deck for allocating and cataloging the system data sets for a new system that will be located on one volume. A user-supplied qualifier of 'X' has been assumed. The INDEX keyword parameter in the GENERATE macro instruction will be coded as INDEX=X.

---

```
//ALLOC      JOB ACCNT,PROGRAMMER,MSGLEVEL=1  -ONE VOLUME-
//STEP       EXEC PGM=IEHPROGM
//SYSPRINT   DD  SYSOUT=A
//JOBQE      DD  DSNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=111111),      X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(250),,CONTIG)
//SVCLIB     DD  DSNAME=X.SVCLIB,VOLUME=(,RETAIN,SER=111111),          X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(175,,100)),          X
//           LABEL=EXPDT=98365,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//LINKLIB    DD  DSNAME=X.LINKLIB,VOLUME=(,RETAIN,SER=111111),        X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(890,50,100)),        X
//           LABEL=EXPDT=98365,DCB=(RECFM=U,BLKSIZE=3625)
//CATALOG    DD  DSNAME=SYSCTLG,VOLUME=(,RETAIN,SER=111111),          X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(3,10)),              X
//           LABEL=EXPDT=98365
//PROCLIB    DD  DSNAME=X.PROCLIB,VOLUME=(,RETAIN,SER=111111),        X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(35,5,7)),            X
//           LABEL=EXPDT=98365,DCB=(RECFM=FB,BLKSIZE=3360)
//PARMLIB    DD  DSNAME=X.PARMLIB,VOLUME=(,RETAIN,SER=111111),        X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(3,,1)),              X
//           LABEL=EXPDT=98365,DCB=(RECFM=F,BLKSIZE=80)
//IMAGELIB   DD  DSNAME=X.IMAGELIB,VOLUME=(,RETAIN,SER=111111),        X
//           UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(3,,2)),              X
//           LABEL=EXPDT=98365,DCB=(RECFM=U,BLKSIZE=1024)
```

---

Figure 19 (Part 1 of 2). Input Deck for Allocating and Cataloging System Data Sets With a User-Supplied Qualifier of 'X'

```

//FORTLIB DD DSNAME=X.FORTLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(45,5,30)), X
// LABEL=EXPDT=98365,DCB=(RECFM=U,BLKSIZE=3625)
//SORTLIB DD DSNAME=X.SORTLIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(75,10,45)), X
// LABEL=EXPDT=98365,DCB=(RECFM=F,BLKSIZE=3625)
//PL1LIB DD DSNAME=X.PL1LIB,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(100,10,100)), X
// LABEL=EXPDT=98365,DCB=(RECFM=U,BLKSIZE=3625)
//NUCLEUS DD DSNAME=X.NUCLEUS,VOLUME=(,RETAIN,SER=111111), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(30,,1)), X
// LABEL=EXPDT=98365
//SYSIN DD *
CATLG DSNAME=SYS1.FORTLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.LINKLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.PARMLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.IMAGELIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.PL1LIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.PROCLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.SORTLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.SVCLIB,VOL=2311=111111,CVOL=2311=111111
CATLG DSNAME=SYS1.SYSJOBQE,VOL=2311=111111,CVOL=2311=111111
/*

```

Figure 19 (Part 2 of 2). Input Deck for Allocating and Cataloging System Data Sets With a User-Supplied Qualifier of 'X'

### Allocating and Cataloging the Utility Data Sets

You must also allocate space for and catalog five utility data sets in your operating system by means of JCL statements in your system generation input deck. (Detailed information about specifying these data sets is found in "Input Deck for System Generation" in this section.)

Four of these data sets must be sequential data sets and are used by the assembler, the linkage editor, and the utility programs during system generation. The fifth data set must be a partitioned data set and is used for the storage of object modules assembled during system generation. Two of the sequential data sets and the partitioned data set must reside on a direct-access volume.

### Restoring the Data Set Names in the Starter Operating System

After the system generation process has been completed, you should restore the names of the renamed data sets in the starter operating system. Also, uncatalog all the catalog entries in the generating system catalog that are no longer required. You can use either of the three utility programs (UNCT2311, UNCT2314, or UNCT3330) to do this, depending on which starter operating system package you used.

## System Generation Using the Starter Operating System

IBM provides an MFT starter operating system that can be used for the first system generation. The starter operating system must be initialized and made operational before it can be used as a generating system. The starter operating system and the procedures required to initialize it are described in this section.

### THE STARTER OPERATING SYSTEM PACKAGE

The starter operating system package consists of an operating system and a set of libraries. This operating system includes:

- Control program
- Data set utilities
- System utilities
- Assembler F
- Linkage Editor F

In addition to the operating system, the starter operating system package includes the following libraries:

- SYS1.SAMPLIB (sample library) -- The members of the sample library are the sample programs used to test operating system components, and are usually kept in card decks by the installation. They are described in the "Testing the New System" section. Also included are:

The independent utility programs IBCDASDI, IBCDMPRS, IBCRVRP, and ICAPRTBL.

The IPL program IEAIPL00.

An example of how to write an accounting routine (SAMACTRT).

The system management facilities (SMF) sample programs, which, although in the sample library, are documented in OS SMF.

CTLG2311, CTLG2314, and CTLG3330: When your operating system is being used as the generating system and a user-supplied qualifier is specified in the INDEX keyword parameter of the GENERATE macro instruction, this utility program can be used to prepare the 2311, 2314, or 3330 distribution of the starter operating system for the system generation process.

SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the component libraries are cataloged in the generating system. Seven libraries (COBLIB, FORTLIB, MACLIB, PARMLIB, PL1LIB, PROCLIB, and SORTLIB) are cataloged in the generating system and renamed to SYS1.DCOBLIB, SYS1.DFORTLIB, SYS1.DMACLIB, SYS1.DPARMLIB, SYS1.DPL1LIB, SYS1.DPROCLIB, and SYS1.DSORTLIB.

UNCT2311, UNCT2314, and UNCT3330: These utility programs should be used after the system generation process is over if the CTLG2311, CTLG2314, or CTLG3330 utility programs were used. This utility program uncatalogs SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the component libraries from the starter operating system. It also uncatalogs and restores the names of the seven system libraries that were renamed.

- Component libraries.
- SYS1.GENLIB (Stage I generation library) -- This library has a blocking factor of 42.

- SYS1.MODGEN and SYS1.MODGEN2 (Stage II generation libraries) -- Each library has a blocking factor of 42.
- SYS1.PARMLIB (parameter library) -- This library is unblocked.
- SYS1.PROCLIB (cataloged procedures library) -- This library is unblocked.
- SYS1.MACLIB (macro library) -- This library has a blocking factor of 42.
- SYS1.COBLIB (COBOL library).
- SYS1.FORTLIB (FORTRAN library) -- This library is distributed without members.
- SYS1.PL1LIB (PL/I library).
- SYS1.SORTLIB (sort library).
- SYS1.TSOMAC (TSO macro library) -- This library has a blocking factor of 45.
- SYS1.TCAMMAC (TCAM macro library) -- This library has a blocking factor of 45.
- SYS1.TSOGEN (Stage II generation library) -- This library has a blocking factor of 45.
- SYS1.DUADS (TSO user attribute data set).
- SYS1.DHELP (TSO help data set).
- SYS1.DCMDLIB (TSO command library).
- SYS1.DN554A (service aids library) -- This library has a blocking factor of 45. It contains an independent service aid program called IMCJQDMP. Information on the retrieval and use of IMCJQDMP is provided in OS Service Aids.

#### STARTER SYSTEM REQUIREMENTS

The starter operating system contains a Release 21 level MFT control program that requires 128K bytes of main storage. Figure 20 shows the maximum machine configuration supported by the starter system. Any subset of this configuration can be used if it meets the minimum requirements shown in Figure 21. The device types that can be used are listed in Figure 22 and the group names that support the IBM-supplied cataloged procedures are provided in Figure 23.

All the devices to be used must be ready before IPL; any device not ready will automatically be taken off line. If a device that was not ready at IPL is required during a job step, the operator should enter a VARY ONLINE command for that device.

When using the starter operating system, it is permissible to have devices other than those shown in Figure 20 attached to the system. However, an interruption must not be issued from any of these additional devices while the starter operating system is running. For example, the operator must not make any one of these devices ready.

## STARTER SYSTEM SPECIAL CONSIDERATIONS

Special considerations that apply only to the starter system are described as a guide.

### Multiple System Generations

- When generating multiple systems, the MFT starter system may require a larger system queue area (SQA) than the generated 5K bytes. The SQA may be specified at IPL time, as explained in OS Operator's Reference manual. OS Storage Estimates manual explains how to determine the required SQA value, which must then be rounded to a doubleword boundary.
- A larger SYS1.SYSJOBQE data set may also be required. How to change the size of this data set is explained in the "Restart Procedures" section of this manual under "SYS1.SYSJOBQE Considerations."

### The Universal Character Set Feature

If a system generation is performed using a 1403 printer with the universal character set feature, the BPS UCS utility program (360-UT-048) must be executed before system generation in order to load the read/write storage units. When the UCS program is executed, NO-FOLDING and BLOCK-DATA-CHECK must be specified. This program is described in the IBM System/360 Basic Programming Support: Universal Character Set Utility Program Operating Guide.

### Models 91 and 195

If a system generation is done using a S/360 Model 91 or 195 central processing unit, the switch on the system control panel must be in the INHIBIT OVERLAP position.

### The M65MP System

The starter operating system cannot operate on an M65MP configuration with unit record devices as input/output, but it can operate with tape input/output. If the starter operating system is used on another configuration, it can generate an operating system for the M65MP system.

### The 2305, 2314, or 3330 Direct-Access Storage

The starter operating system will not work correctly if there is a two-channel switch on the 2305, 2314, or 3330 and one of the channels is not disabled.

If 3330 devices are used, a single partition of at least 85K is used.

## 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 operating system will assume a default value of 800 BPI. This applies whether the tape volumes mounted on the drives have standard labels or no labels.

If, however, you desire to use 1600 BPI for the 9-track tape drives with or without the dual density feature, then you must specify the density in the DCB parameter of the DD statements for all the data sets that will reside on these drives. This applies if the tape volumes mounted on the drives have standard labels or no labels.

### Notes:

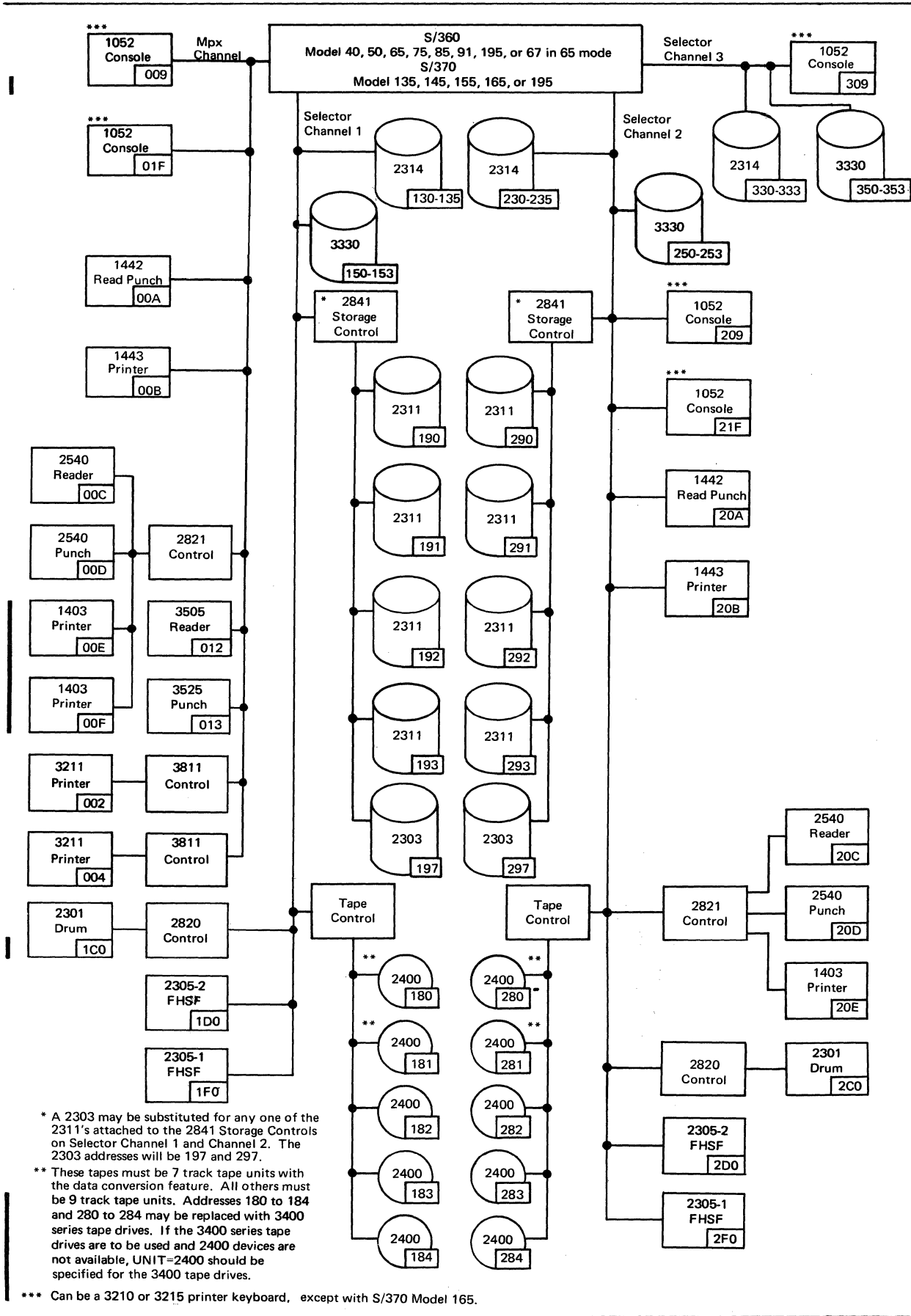
- This density specification for 1600 BPI will not be passed on from one step of a job to the next.
- 1600 BPI cannot be used for SYSOUT.

### Model 85

If a system generation is done using a Model 85 central processing unit, a 1052 console with an address of 009, 01F, 209, 21F, or 309 must be attached to the system. The soft machine-check interrupt switch on the system control panel must be in the INHIBIT position.

### S/370 Model 165

If a system generation is done using a Model 165 central processing unit, a 1052 console with an address of 009, 01F, 209, 21F, or 309 must be attached to the system.



\* A 2303 may be substituted for any one of the 2311's attached to the 2841 Storage Controls on Selector Channel 1 and Channel 2. The 2303 addresses will be 197 and 297.

\*\* These tapes must be 7 track tape units with the data conversion feature. All others must be 9 track tape units. Addresses 180 to 184 and 280 to 284 may be replaced with 3400 series tape drives. If the 3400 series tape drives are to be used and 2400 devices are not available, UNIT=2400 should be specified for the 3400 tape drives.

\*\*\* Can be a 3210 or 3215 printer keyboard, except with S/370 Model 165.

Figure 20. Maximum System Configuration for the Starter Operating System Preparation for System Generation 37

Minimum Requirement	Function	Choose from the following				
		Device	Device Address			
			MPX Channel	Selector Channel 1	Selector Channel 2	Selector Channel 3
1	System Console	1052, 3210, or 3215	009, 01F		209, 21F	309
1	System Input	2540 Reader	00C		20C	
		3505 Reader	012			
		3525 Card Punch	013			
		1442 Read Punch <sup>1</sup>	00A		20A	
		2400 (7 Tr-DC)		180, 181	280, 281	
1	Punch Output	2400 (9-Track)		182, 183, 184	282, 283, 284	
		2540 Punch	00D		20D	
		1442 Read Punch <sup>1</sup>	00A		20A	
		3525 Card Punch	013			
		2400 (7 Tr-DC)		180, 181	280, 281	
1	Print Output	2400 (9-Track)		182, 183, 184	282, 283, 284	
		1443	00B		20B	
		1403	00E, 00F		20E	
		3211	002, 004			
		2400 (7 Tr-DC)		180, 181	280, 281	
1	DLIB01	2311		190, 191, 192, 193	290, 291, 292, 293	
		2314		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353
1	DLIB02	2311		190, 191, 192, 193	290, 291, 292, 293	
		2314		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333
1	DLIB03	2311		190, 191, 192, 193	290, 291, 292, 293	
1	DLIB04	2311		190, 191, 192, 193	290, 291, 292, 293	
1	DLIB05	2311		190, 191, 192, 193	290, 291, 292, 293	
1	DLIB06 <sup>2</sup>	2311		190, 191, 192, 193	290, 291, 292, 293	
1	New System	2311		190, 191, 192, 193	290, 291, 292, 293	
		2301		1C0	2C0	
		2303		197	297	
		2314		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333
		2305-1		1F0	2F0	
		2305-2		1D0	2D0	
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353
2 <sup>3</sup>	System generation utility data sets	2311		190, 191, 192, 193	290, 291, 292, 293	
		2301		1C0	2C0	
		2303		197	297	
		2314		130, 131, 132, 133, 134, 135	230, 231, 232, 233, 234, 235	330, 331, 332, 333
		2305-1		1F0	2F0	
		2305-2		1D0	2D0	
		3330		150, 151, 152, 153	250, 251, 252, 253	350, 351, 352, 353
		2400 (9-Track)		182, 183, 184	282, 283, 284	

<sup>1</sup> A single 1442 may serve as either system input or punch output, but not both simultaneously.

<sup>2</sup> For TSO systems.

<sup>3</sup> The system generation utility data sets do not require additional direct access devices if sufficient space is available on the volumes that contain the starter system and the new system. (see "Input Deck for System Generation" in the section "Preparation for System Generation".)

Figure 21. Minimum I/O Requirements



---

**Magnetic Tape Drives**

<u>Device Type</u>	<u>Description</u>
2400	2400 series 9-track magnetic tape drive
2400-2	2400 series magnetic tape drive with seven track compatibility and data conversion
3400	3400 series 9-track magnetic tape device
3400-2	3400 series magnetic tape drive with seven track capability and data conversion

**Direct-Access Devices**

<u>Device Type</u>	<u>Description</u>
2311	2311 disk storage drive
2301	2301 drum storage
2303	2303 drum storage
2305-1	2305 fixed-head disk storage Model 1
2305-2	2305 fixed-head disk storage Model 2
2314	2314 disk storage facility
3330	3330 disk storage facility

**Unit Record Equipment**

<u>Device Type</u>	<u>Description</u>
1052	1052 printer keyboard
1403	1403 printer
1442	1442 serial reader punch
1443	1443 printer
2540	2540 reader punch (read feed)
2540-2	2540 reader punch (punch feed)
3210	3210 console printer keyboard
3211	3211 printer
3215	3215 console printer keyboard
3505	3505 card reader
3525	3525 card punch with read and print feature

---

Figure 22. Device Types

---

**Name Function**

SYSSQ	Sequential access on devices at any of the following addresses: 182, 183, 184, 282, 283, 284, 190, 191, 192, 193, 290, 291, 292, 293 (any 9-track tape or 2311 disk storage drive).
SYSDA	Direct access on devices at any of the following addresses: 190, 191, 192, 193, 290, 291, 292, 293, (any 2311); 1C0, 2C0, (any 2301); 197, 297, (any 2303); 1D0, 1F0, 2D0, 2F0, (any 2305); 130, 131, 132, 133, 134, 135, 230, 231, 232, 233, 234, 235, 330, 331, 332, 333, (any 2314); 150, 151, 152, 153, 250, 251, 252, 253, 350, 351, 352, 353, (any 3330).
SYSCP	A 2540 card punch at address 00D or 20D, or 1442 at address 00A or 20A, or 3525 at address 013.

---

Figure 23. Additional Group Names Supporting IBM-Supplied Cataloged Procedures

## PROCESSING THE STARTER PACKAGE

Before the starter operating system package can be used for system generation, it must be initialized and prepared for use. If the package is on tape, preparation consists of:

1. Restoring the system to disk (the tape then becomes a backup copy of the system).
2. Punching the independent utility programs and the sample programs from SYS1.SAMPLIB for later use.
3. Listing the data describing the system.

If the starter operating system package is on disk, preparation consists of:

1. Punching the independent utility programs and the sample programs from SYS1.SAMPLIB for later use.
2. Creating a backup copy of the system on disk or tape.
3. Listing the data describing the system.

Processing of the tape or disk distribution of the starter operating system package is depicted in Figure 24. Detailed processing instructions are contained in the procedure section that follows.

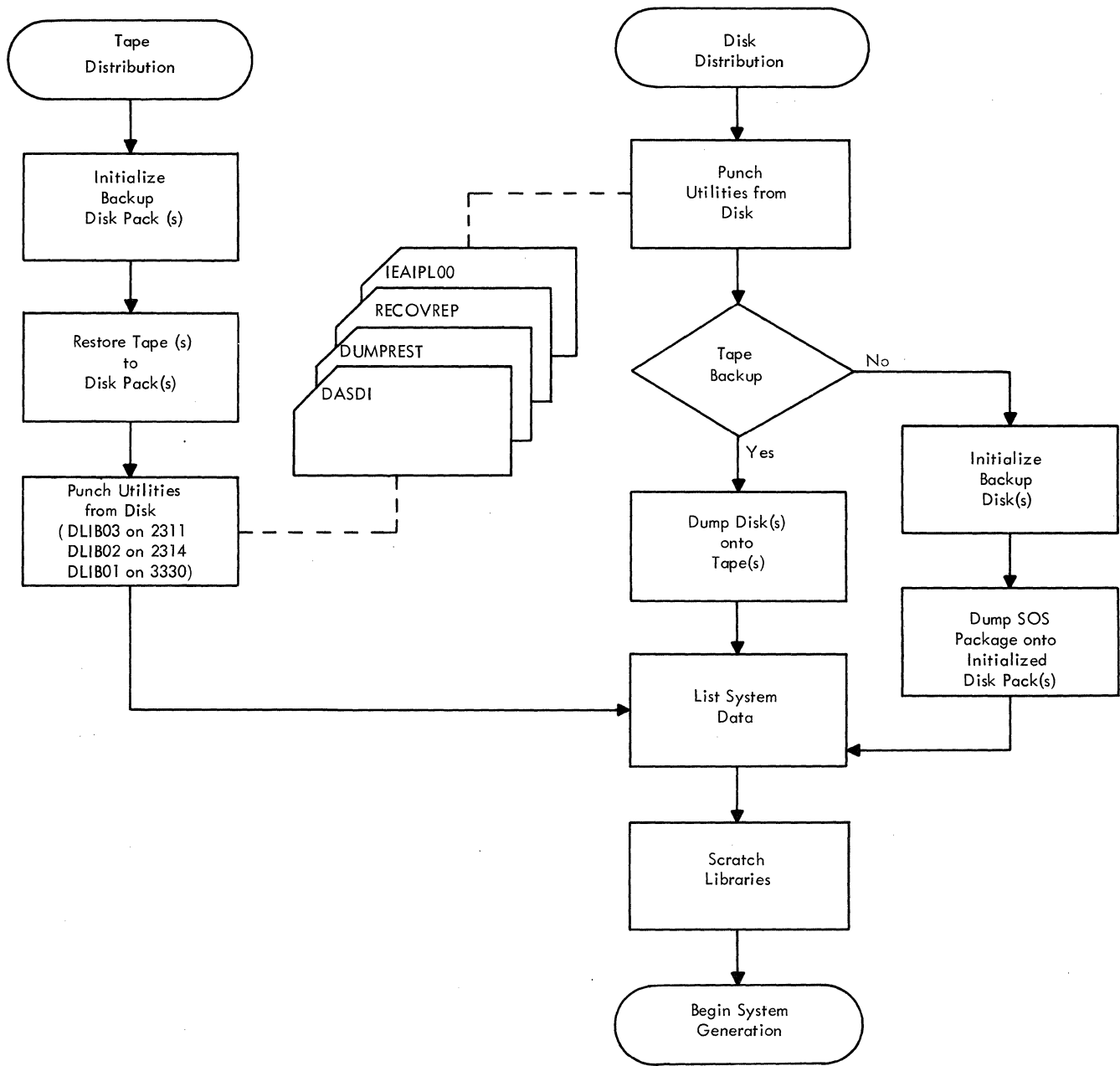


Figure 24. Processing the Starter Package

## PROCEDURES FOR PROCESSING THE STARTER PACKAGE

This section contains detailed procedures for processing the various types of starter operating system packages and for deleting libraries from the starter operating system. Procedures are given for:

- Complete 2311 package -- tape distribution
- Complete 2311 package -- disk pack distribution with tape backup
- Complete 2311 package -- disk pack distribution with disk pack backup
- Complete 2314 package -- tape distribution
- Complete 3330 package -- tape distribution

In using these procedures, it may be helpful to refer to Appendix C. This appendix describes the distribution methods and illustrates the arrangement of the starter package data sets on disk packs and tapes.

The procedures include examples of control decks. In these examples, the underlined fields are those that may require modification for different installations. Further explanation of the field requirements is contained in OS Utilities. Remember to tailor the control decks for the distribution you receive: a complete 2311 package uses volume serial numbers DLIB01, DLIB02, DLIB03, DLIB04, and DLIB05; the volume serial number DLIB06 must be ordered with the 2311 package if TSO is to be generated; a 2314 package uses volume serial numbers DLIB01 and DLIB02; and a 3330 package uses volume serial number DLIB01.

For illustrative purposes, the given procedures assume the set of devices and device addresses listed in Figure 25.

---

Device Function	Input/Output Device	Address
Printer Keyboard	1052	01F
System Residence and System Data Sets		
• DLIB01	2311 (2314 or 3330)	190 (130 or 150)
• DLIB02	2311 (or 2314)	191 (or 131)
• DLIB03	2311	192
• DLIB04	2311	193
• DLIB05	2311	290
• DLIB06	2311	291
System Input	2540 reader	00C
Punch Output	2540 punch	00D
Print Output	1403 printer	00E
Tape Drives <sup>1</sup>	2400 tape	180 <sup>2</sup>
	2400 tape	181 <sup>2</sup>
	2400 tape	182 <sup>2</sup>

<sup>1</sup>Tape units are required only if the starter package is received on tape or if system backup is created on tape.

<sup>2</sup>Tape units located at addresses 180 and 181 are 7-track tape units with the data conversion feature. The tape unit at 182 is a 9-track tape unit.

---

Figure 25. Sample Configuration

## 2311 Package -- Tape Distribution

The procedure for processing the tape distribution of the 2311 starter operating system package is as follows:

---

### Initialize Disk

---

1. Mount the disk pack onto which DLIB01 is to be restored.
2. Mount tape #1 (DLIB01) of starter package.
3. Load IBCDASDI program from tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
4. Place the following control deck in the input device to initialize the disk:

```
JOB
MSG      TODIV=1403,TOADDR=00E
DADEF    TODIV=2311,TOADDR=190,VOLID=SCRATCH,FLAGTEST=NO
VLD      NEWVOLID=111111,OWNERID=DEPT38
VTOCD    STRTADR=50,EXTENT=10
END
```

In the DADEF statement, the FLAGTEST=NO parameter must be used only when the disk is initialized for the first time.

Note: The VTOCD statement creates a temporary VTOC.

5. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu where, xxxx is the device type, c is the channel address, an uu is the unit address. The device type can be 1442, 2400, or 2540.
6. When the disk initialization is complete, the message END OF JOB is printed on the message output device, and the program enters the wait state.

---

### Restore Tape to Disk

---

7. Load the IBCDMPRS program from tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.

8. Place the following deck in the input device to restore the contents of the tape to the disk:

```
JOB
MSG      TODDEV=1403,TOADDR=00E
RESTORE  FROMDEV=2400,FROMADDR=180,TODEV=2311,      X
          TOADDR=190,VOLID=111111
```

END

when restoring is complete, the serial number of the disk has been changed from 111111 to DLIB01.

9. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
10. When the restoring is complete, the message END OF JOB is printed on the message output device, and the program enters the wait state.

---

#### Initialize and Restore Second Disk

---

11. The unit to be initialized must be in an offline status. Carry out the IPL procedure (the steps denoted with an \*\*) from the disk restored in Steps 1 through 10 with the following control deck in the input device:

```
// VARY 191,OFFLINE
//DLIB02  JOB  ACCT123,PROGRAMMER,MSGLEVEL=1
//INIT02  EXEC PGM=IEHDASDR,PARM='N=1'
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD  *
          ANALYZE TODD=191,VTOC=00004,EXTENT=00010,FLAGTEST=NO, X
          NEWVOLID=DLIB02

/*
// VARY 191,ONLINE
//REST02  EXEC PGM=IEHDASDR,PARM='N=1'
//SYSPRINT DD  SYSOUT=A
//FROM02  DD  UNIT=180,DISP=OLD,LABEL=(4,NL),      X
//          VOLUME=SER=REST02
//DLIB02  DD  UNIT=191,DISP=OLD,VOLUME=SER=DLIB02
//SYSIN   DD  *
          RESTORE FROMDD=FROM02,TODD=DLIB02,CPYVOLID=YES, X
          PURGE=YES

/*
```

---

#### Initialize and Restore Third, Fourth, Fifth, and Sixth Disks

---

12. Repeat Step 11 to initialize and restore DLIB03 and DLIB04. If the distribution was on three 800-BPI tapes, the first half of DLIB03 is on the first tape. The second half of DLIB03 is on the second tape. The file number subparameter of the LABEL=(n,NL) is 6 for the first half of DLIB03 on Tape #1. The file number subparameters on the second tape are 2 for DLIB03, and 4 for DLIB04. The file subparameters on the third tape are 2 for DLIB05 and 4 for DLIB06. DLIB06 is required only for a TSO generation. Repeat Step 11 to initialize and restore DLIB05 and DLIB06.

If the distribution was on two 1600-BPI tapes, the file count subparameter on the first tape will be 6 for DLIB03 and 8 for DLIB04. The file count subparameter on the second tape will be 2 for DLIB05 and 4 for DLIB06.

---

## Punch Utility and Sample Programs

---

13. Completion of the preceding steps provides operable disk packs with backup tapes. Proceed to punch the independent utility programs and any sample programs desired. Be sure all necessary volumes are mounted and make ready all devices to be used.
14. The following control deck should be placed in the input device:

```
//JOB1      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//          EXEC PGM=IEBPTPCH
//SYSUT1    DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP)
//SYSUT2    DD  UNIT=2540-2
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
            PUNCH  TYPORG=PO,MAXNAME=4
            MEMBER  NAME=IBCDMPRS
            MEMBER  NAME=IBCDASDI
            MEMBER  NAME=IEAIPLOO
            MEMBER  NAME=IBCRCVRP
/*
```

A MEMBER card should be added to the above control deck for each sample program desired for later use, and for SAMACTRT provided the MAXNAME field in the PUNCH control card is adjusted to show the revised number of member cards in the deck. The member name card (first card) should be removed from each member deck punched. (Refer to the "Testing the New System" section of this publication for names of the sample programs.)

- 15\*\* Set the LOAD UNIT switches on the control panel to the channel, control unit, and device of the system-residence volume (DLIB01). Then press the LOAD key.
- 16\*\* Reply 00,'RAM=,' to the SPECIFY SYSTEM PARAMETERS message to comply with 128K minimum configuration unless the 3330 or 2305 devices are being used, then reply 00,'RAM=01'. Reply no to the request for a SYS1.DUMP data set. Signal EOB to the SPECIFY SYSTEM PARAMETERS message. Reply no to the change partitions message. Wait for the READY message and for the WAIT light to be turned on. Then enter a SET command specifying the date, as follows:
  - a. Press the REQUEST key.
  - b. Wait for the READ light to go on.
  - c. Type SET DATE=yy.ddd,Q=(,F) to specify date.
  - d. Signal EOB.
  - e. Reply U to the SPECIFY JOB QUEUE PARAMETERS message and wait for the WAIT light to go on.

17\*\* Commands must be entered to start a SYSOUT writer, a reader, and an initiator. It is recommended that the direct SYSOUT writer be used. The commands are:

```
MN JOB NAMES
START DSO.P1,00E
START RDR.S,00C
START INIT.ALL
```

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

18\*\* When the job is completed, a READER ENDED message will be printed, followed by a JOB ENDED message. The system will then enter the wait state. See the OS Operator's Reference for a discussion of stopping the system for MFT.

---

### List System Catalog

---

19. In order to determine the location of the component libraries, place the control deck shown below in the input device.

Then enter a START RDR.S,00C command.

```
//LISTCTLG JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//SYSRS DD UNIT=2311,DISP=(OLD,KEEP), X
// VOLUME=(,RETAIN,SER=DLIB01)
//SYSIN DD *
LISTCTLG
/*
```

Note: You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding punch operation was skipped.

---

### List System Data

---

20. To list the data describing the system, place the proper control deck shown below in the input device.

Then enter a START RDR.S,00C command.



Control deck for users of the complete 2311 package:

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB01
//DD2      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB02
//DD3      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB03
//DD4      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB04
//DD5      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB05
//DD6      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB06 (if TSO)
//SYSIN    DD *
LISTVTOC  DUMP
LISTVTOC  DUMP,VOL=2311=DLIB02
LISTVTOC  DUMP,VOL=2311=DLIB03
LISTVTOC  DUMP,VOL=2311=DLIB04
LISTVTOC  DUMP,VOL=2311=DLIB05
LISTVTOC  DUMP,VOL=2311=DLIB06 (if TSO)
LISTPDS   DSNAME=(SYS1.PROCLIB                      X
           [,Optional PDSs on System Residence])
LISTPDS   VOL=2311=DLIB02,DSNAME=SYS1.MODGEN
LISTPDS   VOL=2311=DLIB01,DSNAME=SYS1.MODGEN2
LISTPDS   VOL=2311=DLIB03,DSNAME=(component library) X
           [,component library...])
LISTPDS   VOL=2311=DLIB04,DSNAME=SYS1.GENLIB
LISTPDS   VOL=2311=DLIB05,DSNAME=(component libraries)
LISTPDS   VOL=2311=DLIB06,DSNAME=(component library X
           [,component library...]) (if TSO)
/*
```

Specify the component libraries desired in the proper LISTPDS statement. In the list deck, any of the partitioned data sets shown in Figure 90 may be specified in the LISTPDS statement.

Note: You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding punch operation was skipped.

## 2311 Package -- Disk Pack Distribution (Tape Backup)

The procedure for processing the disk pack distribution of the 2311 starter operating system package, with tape backup, is as follows:

---

### Punch Utility and Sample Programs

---

1. The disk packs are operable as received. Punch the independent utility programs (needed to create a tape backup copy of the disks) and any sample programs desired. Be sure all necessary volumes are mounted and make ready all devices to be used.
2. The following control deck should be placed in the input device:

```
//JOB1      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//          EXEC PGM=IEBPTPCH
//SYSUT1    DD DSN=SYS1.SAMPLIB,DISP=(OLD,KEEP)
//SYSUT2    DD UNIT=2540-2
//SYSPRINT DD SYSOUT=A
//SYSIN     DD *
            PUNCH TYPORG=PO,MAXNAME=4
            MEMBER NAME=IBCDMPRS
            MEMBER NAME=IBCDASDI
            MEMBER NAME=IEAIPLOO
            MEMBER NAME=IBCRCVRP
/*
```

A MEMBER card should be added to the above control deck for each sample program desired for later use, and for SAMACTRT, provided the MAXNAME field in the PUNCH control card is adjusted to show the revised number of member cards in the deck. The member name card (first card) should be removed from each member deck punched. (Refer to the "Testing the New System" section of this publication for names of the sample programs.)

3. Set the LOAD UNIT switches on the control panel to the channel, control unit, and device of the system-residence volume (DLIB01). Then press the LOAD key.
- 4\*\* Reply no to the request for a SYS1.DUMP data set. Signal EOB to the SPECIFY SYSTEM PARAMETERS message. Reply no to the change partitions message. Wait for the READY message and for the WAIT light to be turned on. Then enter a SET command specifying the date, as follows:
  - a. Press the REQUEST key.
  - b. Wait for the READ light to go on.
  - c. Type SET DATE=yy.ddd to specify date.
  - d. Signal EOB.
  - e. Reply U to the SPECIFY JOB QUEUE PARAMETERS message and wait for the WAIT light to go on.

5\*\* Commands must be entered to start a SYSOUT writer, a reader, and an initiator. It is recommended that the direct SYSOUT writer be used. The commands are:

```
MN JOB NAMES
START DSO.P1,00E
START RDR.S,00C
START INIT.ALL
```

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

6\*\* When the job is completed, a READER ENDED message will be printed, followed by a JOB ENDED message. The system will then enter the wait state. See the OS Operator's Reference for a discussion of stopping the system for MFT.

---

#### Dump Disk to First Tape

---

7. Place the IBCDMPRS program (punched from SYS1.SAMPLIB in Step 2) in the input device, followed by the control deck shown below.

```
JOB
MSG      TODDEV=1403,TOADDR=00E
DUMP     FROMDEV=2311,FROMADDR=190,                X
          TODDEV=2400,TOADDR=181
END
```

8. Mount the tape that is to contain the backup copy of the first disk pack (DLIB01).

9. Load the IBCDMPRS program by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.

10. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.

11. After the dump to tape is completed, an END OF JOB message will be printed and the program will enter the wait state. The tape should be removed and stored in the tape library.

---

#### Dump Second Disk to Second Tape

---

12. Repeat Steps 7 through 11 to dump the contents of disk pack DLIB02.

---

#### Dump Third Disk to Third Tape

---

13. Repeat Steps 7 through 11 to dump the contents of disk pack DLIB03.

---

Dump Fourth Disk to Fourth Tape

---

14. Repeat Steps 7 through 11 to dump the contents of disk pack DLIB04.

---

Dump Fifth Disk to Fifth Tape

---

15. Repeat Steps 7 through 11 to dump the contents of disk pack DLIB05.

---

Dump Sixth Disk to Sixth Tape

---

16. Repeat Steps 7 through 11 to dump contents of disk pack DLIB06. (Required only for TSO systems.)

---

List System Catalog

---

17. In order to determine the location of the component libraries, place the control deck shown below in the input device:

Then enter a START RDR.S,00C command.

```
//LISTCTLG JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//SYSRS DD UNIT=2311,DISP=(OLD,KEEP), X
// VOLUME=(,RETAIN,SER=DLIB01)
//SYSIN DD *
LISTCTLG
/*
```

Note: You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding dump operation was performed.

---

---

List System Data

18. To list the data describing the system, load the operating system, and place the proper control deck shown below in the input device:

Then enter a START RDR.S,00C command.

Control deck for users of the complete 2311 package:

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1     EXEC PGM=IEHLIST
//SYSPRINT  DD  SYSOUT=A
//DD1       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB01
//DD2       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB02
//DD3       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB03
//DD4       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB04
//DD5       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB05
//DD6       DD  UNIT=2311,DISP=OLD,VOLUME=SER=DLIB06 (if TSO)
//SYSIN     DD  *
             LISTVTOC DUMP
             LISTVTOC DUMP,VOL=2311=DLIB02
             LISTVTOC DUMP,VOL=2311=DLIB03
             LISTVTOC DUMP,VOL=2311=DLIB04
             LISTVTOC DUMP,VOL=2311=DLIB05
             LISTVTOC DUMP,VOL=2311=DLIB06 (if TSO)
             LISTPDS  DSNAME=(SYS1.PROCLIB                               X
             [,Optional PDSs on System Residence ])
             LISTPDS  VOL=2311=DLIB02,                                  X
             DSNAME=SYS1.MODGEN
             LISTPDS  VOL=2311=DLIB01,DSNAME=SYS1.MODGEN2
             LISTPDS  VOL=2311=DLIB03,DSNAME=(component library        X
             [,component library... ])
             LISTPDS  VOL=2311=DLIB04,DSNAME=SYS1.GENLIB
             LISTPDS  VOL=2311=DLIB05,DSNAME=(component libraries)
             LISTPDS  VOL=2311=DLIB06,DSNAME=(component library        X
             [,component library... ]) (if TSO)
/*
```

Specify the component libraries desired in the proper LISTPDS statement. In the LIST deck, any of the partitioned data sets shown in Figure 90 may be specified in the LISTPDS statement.

Note: You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding dump operation was performed.

---

## 2311 Package -- Disk Pack Distribution (Disk Backup)

The procedure for processing the disk pack distribution of the 2311 starter operating system package, with disk backup, is as follows:

---

### Punch Utility and Sample Programs

1. The disk packs are operable as received. Punch the independent utility programs (needed to create a disk backup copy of the disks) and any sample programs desired. Be sure all necessary volumes are mounted and make ready all devices to be used.
2. The following control deck should be placed in the input device:

```
//JOB1      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//          EXEC PGM=IEBPTPCH
//SYSUT1    DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP)
//SYSUT2    DD  UNIT=2540-2
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
            PUNCH  TYPORG=PO,MAXNAME=4
            MEMBER  NAME=IBCDMPRS
            MEMBER  NAME=IBCDASDI
            MEMBER  NAME=IEAIPLOO
            MEMBER  NAME=IBCRCVRP
/*
```

A MEMBER card should be added to the above control deck for each sample program desired for later use, and for SAMACTRT, provided the MAXNAME field in the PUNCH control card is adjusted to show the revised number of member cards in the deck. The member name card (first card) should be removed from each member deck punched. (Refer to the "Testing the New System" section of this publication for names of the sample programs.)

- 3\*\* Set the LOAD UNIT switches on the control panel to the channel, control unit, and device of the system-residence volume (DLIB01). Then press the LOAD key.
- 4\*\* Reply no to the request for a SYS1.DUMP data set. Signal EOB to the SPECIFY SYSTEM PARAMETERS message. Reply no to the change partitions message. Wait for the READY message and for the WAIT light to be turned on. Then enter a SET command specifying the date, as follows:
  - a. Press REQUEST key.
  - b. Wait for READ light to go on.
  - c. Type SET DATE=yy.ddd,Q=(,F) to specify date.
  - d. Signal EOB.
  - e. Reply U to the SPECIFY JOB QUEUE PARAMETERS message and wait for the WAIT light to go on.

---

5\*\* Commands must be entered to start a SYSOUT writer, a reader, and an initiator. It is recommended that the direct SYSOUT writer be used. The commands are:

```
MN JOB NAMES
START DSO.P1,00E
START RDR.S,00C
START INIT.ALL
```

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

6\*\* When the job is completed, a READER ENDED message will be printed, followed by a JOB ENDED message. The system will then enter the wait state. See the OS Operator's Reference for a discussion of stopping the system for MFT.

---

### Initialize Backup Disk

---

7. Place the IBCDASDI program (punched from SYS1.SAMPLIB in Step 2) in the input device, followed by the control deck shown below:

```
JOB
MSG      TODEV=1403,TOADDR=00E
DADEF    TODEV=2311,TOADDR=190,VOLID=SCRATCH,FLAGTEST=NO
VLD      NEWVOLID=111111,OWNERID=DEPT38
VTOCD    STRTADR=50,EXTENT=10
END
```

In the DADEF statement, the FLAGTEST=NO parameter must be used only when the disk is initialized for the first time.

8. Mount the disk pack that is to receive the backup copy of the first starter disk pack (DLIB01).
  9. Load the IBCDASDI program by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
  10. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
  11. When the disk initialization is completed, the message END OF JOB is printed on the message output device, and the program enters the wait state.
-

---

## Restore Backup Disk

---

- Place the IBCDMPRS program (punched from SYS1.SAMPLIB in Step 2) in the input device, followed by the control deck shown below.

```
JOB
MSG      TODDEV=1403,TOADDR=00E
DUMP     FROMDEV=2311,FROMADDR=190,TODEV=2311,          X
          TOADDR=191,VOLID=111111
END
```

When restoring is completed, the serial number of the disk has been changed from 111111 to DLIB01.

- Load the IBCDMPRS program by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
- Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
- When the dumping is completed, the message END OF JOB is printed on the message output device, and the program enters the wait state.

---

## Initialize and Restore Second Backup Disk

---

- Repeat Steps 7 through 15 to create a backup copy of the second disk pack (DLIB02).

---

## Initialize and Restore Third Backup Disk

---

- Repeat Steps 7 through 15 to create a backup copy of the third disk pack (DLIB03).

---

## Initialize and Restore Fourth Backup Disk

---

- Repeat Steps 7 through 15 to create a backup copy of the fourth disk pack (DLIB04).

---

## Initialize and Restore Fifth Backup Disk

---

- Repeat Steps 7 through 15 to create a backup copy of the fifth disk pack (DLIB05).

---

## Initialize and Restore Sixth Backup Disk

---

- Repeat Steps 7 through 15 to create a backup copy of the sixth disk pack (DLIB06). (Required for TSO systems only.)



---

## List System Catalog

21. To determine the location of the component libraries, place the control deck shown below in the input device.

Then enter a START RDR.S,00C command.

```
//LISTCTLG JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM IEHLIST
//SYSPRINT DD SYSOUT=A
//SYSRS DD UNIT=2311,DISP (OLD,KEEP), X
// VOLUME=(,RETAIN,SER=DLIB01)
//SYSIN DD *
LISTCTLG
/*
```

**Note:** You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding initialization and restore operations were performed.

---

## List System Data

22. To list the data describing the system, load the operating system and place the control deck shown below in the input device.

Then enter a START RDR.S,00C command. Control deck for users of the complete 2311 package:

```
//JOB2 JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB01
//DD2 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB02
//DD3 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB03
//DD4 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB04
//DD5 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB05
//DD6 DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB06 (if TSO)
//SYSIN DD *
LISTVTOC DUMP
LISTVTOC DUMP,VOL=2311=DLIB02
LISTVTOC DUMP,VOL=2311=DLIB03
LISTVTOC DUMP,VOL=2311=DLIB04
LISTVTOC DUMP,VOL=2311=DLIB05
LISTVTOC DUMP,VOL=2311=DLIB06 (if TSO)
LISTPDS DSNAME=(SYS1.PROCLIB X
[,Optional PDSs on System Residence])
LISTPDS VOL=2311=DLIB02,DSNAME=SYS1.MODGEN
LISTPDS VOL=2311=DLIB01,DSNAME=SYS1.MODGEN2
LISTPDS VOL=2311=DLIB03,DSNAME=(component library X
[,component library...])
LISTPDS VOL=2311=DLIB04,DSNAME=SYS1.GENLIB
LISTPDS VOL=2311=DLIB05,DSNAME=(component libraries)
LISTPDS VOL=2311=DLIB06,DSNAME=(component library X
[,component library...]) (if TSO)
/*
```

Specify the component libraries desired in the proper LISTPDS statement. In the LIST deck, any of the partitioned data sets shown in Figure 90 may be specified in the LISTPDS statement.

**Note:** You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding initialization and restore operations were performed.

## 2314 Package -- Tape Distribution

The procedure for processing the tape distribution of the 2314 starter operating system package is as follows:

---

### Initialize Disk

1. Mount the disk packs onto which DLIB01 and DLIB02 will be restored.
2. Mount the tape distribution (DLIB01 and DLIB02) of the starter package. This is one 1600-BPI tape or two 800-BPI tapes.
3. Load the IBCDASDI program from the tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
4. Place the following control deck in the input device to initialize the disk:

```
JOB
MSG      TODEV=1403, TOADDR=00E
DADEF    TODEV=2314, TOADDR=130, VOLID=SCRATCH, FLAGTEST=NO
VLD      NEWVOLID=DLIB01, OWNERID=DEPT38
VTOCD    STRTADR=50, EXTENT=10
END
```

```
JOB
MSG      TODEV=1403, TOADDR=00E
DADEF    TODEV=2314, TOADDR=131, VOLID=SCRATCH, FLAGTEST=NO
VLD      NEWVOLID=DLIB02, OWNERID=DEPT38
VTOCD    STRTADR=50, EXTENT=10
END
```

In the DADEF statement, the FLAGTEST=NO parameter must be used when the disk is initialized for the first time only.

5. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
6. When the disk initialization is complete, the message END OF JOB is printed on the message output device, and the program enters the wait state.

---

### Restore Tape to Disk

7. Load the IBCDMPRS program from tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
8. Place the following control deck in the input device to restore the contents of the tape to the disk:

```
JOB
MSG      TODEV=1403, TOADDR=00E
RESTORE  FROMDEV=2400, FROMADDR=180, TODEV=2314,           X
          TOADDR=130, VOLID=DLIB01
END
```

- 
9. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
  10. When the restoring is completed, the message END OF JOB is printed on the message output device, and the program enters the wait state.

The DLIB01 tape should be removed and stored in the tape library for backup purposes.

---

#### Restore Second Tape to Disk

11. (Do this step if you have two 800-BPI tapes. If you have a 1600-BPI tape, proceed to Step 12.) After pressing the LOAD key to skip over the IBCDASDI program, repeat Steps 7 through 10 to restore the contents of the tape to the disk. When this step is completed, there will be an operable disk pack with backup tape.
12. You can now carry out the IPL procedure (the steps denoted with an \*\*) with the DLIB01 volume and use it with the following deck to restore the DLIB02 volume.

```

//DLIB02 JOB MSGLEVEL=1
//      EXEC PGM=IEHDASDR,PARM='N=1'
//SYSPRINT DD SYSOUT=A
//FROM DD UNIT=182,DISP=OLD,LABEL=(4,NL), X
//      VOLUME=SER=REST02
//TO DD UNIT=131,DISP=OLD,VOLUME=SER=DLIB02
//SYSIN DD *
        RESTORE FROMDD=FROM,TODD=TO,CPYVOLID=YES, X
        PURGE=YES
/*

```

If a 1600-BPI tape is used, the DLIB02 volume will be on the first tape. The file sequence number will be 4.

---

#### Punch Utility and Sample Programs

13. Completion of the preceding steps provides an operable disk pack with a backup tape. Punch the independent utility programs and any sample programs desired. Be sure the necessary volumes are mounted and make ready all devices to be used.
14. The following control deck should be placed in the input device.

```

//JOB1 JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//      EXEC PGM=IEBPTPCH
//SYSUT1 DD DSN=SYS1.SAMPLIB,DISP=(OLD,KEEP)
//SYSUT2 DD UNIT=2540-2
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        PUNCH TYPORG=PO,MAXNAME=4
        MEMBER NAME=IBCDMPRS
        MEMBER NAME=IBCDASDI
        MEMBER NAME=IEAIPLOO
        MEMBER NAME=IBCRVPRP
/*

```

---

---

A MEMBER card should be added to the above control deck for the sample program desired for later use, and for SAMACTRT, provided the MAXNAME field in the PUNCH control card is adjusted to show the revised number of member cards in the deck. The member name card (first card) should be removed from each member deck punched. (Refer to the "Testing the New System" section of this publication for names of the sample programs.)

15\*\* Set the LOAD UNIT switches on the control panel to the channel, control unit, and device of the system-residence volume (DLIB01). Then press the LOAD key.

16\*\* Reply 00,'RAM=,' to the SPECIFY SYSTEM PARAMETERS message to comply with 128K minimum configuration unless the 3330 or 2305 devices are being used, then reply 00,'RAM=01'. Reply no to the request for a SYS1.DUMP data set. Reply no to the change partitions message. Wait for the READY message and for the WAIT light to be turned on. Then enter a SET command specifying the date, as follows:

- a. Press the REQUEST key.
- b. Wait for the READ light to go on.
- c. Type SET DATE=yy.ddd, Q=(F) to specify date.
- d. Signal EOB.
- e. Reply U to the SPECIFY JOB QUEUE PARAMETERS message and wait for the WAIT light to go on.

17\*\* Commands must be entered to start a SYSOUT writer, a reader, and an initiator. It is recommended that the direct SYSOUT writer be used. The commands are:

MN JOB NAMES  
START DSO.P1,00E  
START RDR.S,00C  
START INIT.ALL

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

18\*\* When the job is complete, a READER ENDED message will be printed, followed by a JOB ENDED message. The system will then enter the wait state. See the OS Operator's Reference for a discussion of stopping the system for MFT.

---

---

## List System Data

19. To list the data describing the system, place the control deck shown below in the input device.

Then enter a START RDR.S,00C command.

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1      DD UNIT=2314,DISP=OLD,VOLUME=SER=DLIB01
//DD2      DD UNIT=2314,DISP=OLD,VOLUME=SER=DLIB02
//SYSIN    DD *
           LISTCTLG
           LISTVTOC DUMP,VOL=2314=DLIB01
           LISTVTOC VOL=2314=DLIB02,DUMP
           LISTPDS  DSNAME=(SYS1.PROCLIB                      X
                    [,Optional PDSs on System Residence])
           LISTPDS  DSNAME=(SYS1.MODGEN,SYS1.GENLIB,SYS1.SAMPLIB)
/*
```

If JOB2 is run immediately after JOB1 (in Step 12), eliminate the JOB2 card. In the LIST deck, any of the partitioned data sets shown in Figure 90 may be specified in the LISTPDS statement.

**Note:** You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding punch operation was skipped.

---

## Deleting Libraries

To make additional direct-access storage available for the system generation process, SYS1.SAMPLIB can be deleted from DLIB01 before starting the system generation process. This may be accomplished by loading the operating system and using the control deck shown in the input device.

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP2 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1      DD UNIT=2311,DISP=OLD,VOLUME=SER=DLIB03
//SYSIN    DD *
           SCRATCH DSNAME=SYS1.SAMPLIB,VOL=2311=DLIB03,PURGE
/*
```

## 3330 Package -- Tape Distribution

The procedure for processing the tape distribution of the 3330 starter operating system package is as follows:

---

### Initialize Disk

1. Mount the disk pack onto which DLIB01 will be restored.
2. Mount the tape distribution (DLIB01) of the starter package. This is one 1600-BPI tape or two 800-BPI tapes.
3. Load the IBCDASDI program from the tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
4. Place the following control deck in the input device to initialize the disk:

```
JOB
MSG      TODEV=1403, TOADDR=00E
DADEF    TODEV=3330, TOADDR=150, VOLID=SCRATCH, FLAGTEST=NO
VLD      NEWVOLID=DLIB01, OWNERID=DEPT38
VTOCD    STRTADR=50, EXTENT=10
END
```

In the DADEF statement, the FLAGTEST=NO parameter must be used when the disk is initialized for the first time only.

5. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.
6. When the disk initialization is complete, the message END OF JOB is printed on the message output device, and the program enters the wait state.

---

### Restore Tape to Disk

7. Load the IBCDMPRS program from tape by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the wait state is entered and the hexadecimal value FFFF is displayed in the console lights.
8. Place the following control deck in the input device to restore the contents of the tape to the disk:

```
JOB
MSG      TODEV=1403, TOADDR=00E
RESTORE  FROMDEV=2400, FROMADDR=180, TODEV=3330,           X
          TOADDR=150, VOLID=DLIB01
END
```

9. Define the control statement input device by pressing the REQUEST key of the printer keyboard. The message DEFINE INPUT DEVICE will be printed. Enter the message INPUT=xxxx cuu, where xxxx is the device type, c is the channel address, and uu is the unit address. The device type can be 1442, 2400, or 2540.

- 
10. When the restoring is complete, the message END OF JOB is printed on the message output device, and the program enters the wait state.

The DLIB01 tape should be removed and stored in the tape library for backup purposes.

---

#### Restore Second Tape to Disk

---

11. (Do this step if you have two 800-BPI tapes. If you have a 1600-BPI tape, proceed to step 12.) After pressing the LOAD key to skip over the IBCDASDI program repeat steps 7 through 10 to restore the contents of the tape to the disk. When this step is completed, there will be an operable disk pack with backup tape.
12. You can now carry out the IPL procedure (the steps denoted with an \*\*) with the DLIB01 volume.
- 

#### Punch Utility and Sample Programs

---

13. Completion of the preceding steps provides an operable disk pack with a backup tape. Punch the independent utility programs and any sample programs desired. Be sure the necessary volumes are mounted and make ready all devices to be used.
14. The following control deck should be placed in the input device.

```
//JOB1      JOB  ACCT123,PROGRAMMER,MSGLEVEL=1
//          EXEC  PGM=IEBPTPCH
//SYSUT1    DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP)
//SYSUT2    DD  UNIT=2540-2
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
              PUNCH  TYPORG=PO,MAXNAME=4
              MEMBER  NAME=IBCDMPRS
              MEMBER  NAME=IBCDASDI
              MEMBER  NAME=IEAIPLOO
              MEMBER  NAME=IBCRVVRP
/*
```

A MEMBER card should be added to the above control deck for the sample program desired for later use, and for SAMACTRT, provided the MAXNAME field in the PUNCH control card is adjusted to show the revised number of member cards in the deck. The member name card (first card) should be removed from each member deck punched. (Refer to the "Testing the New System" section of this publication for names of the sample programs.)

- 15\*\* Set the LOAD UNIT switches on the control panel to the channel, control unit, and device of the system-residence volume (DLIB01). Then press the LOAD key.
- 16\*\* Reply 00,'RAM=01' to the SPECIFY SYSTEM PARAMETERS message which makes access methods IGG019C0, IGG019C4, IGG019EK, IGG019FN, and IGG019FP resident.

Reply no to the request for a SYS1.DUMP data set. Reply no to the change partitions message. Wait for the READY message and for the WAIT light to be turned on. Then enter a SET command specifying the date, as follows:

- a. Press the REQUEST key.
  - b. Wait for the READ light to go on.
  - c. Type SET DATE=yy.ddd,Q=(,F) to specify date.
  - d. Signal EOB.
  - e. Reply U to the SPECIFY JOB QUEUE PARAMETERS message and wait for the WAIT light to go on.
-

---

17\*\* Commands must be entered to start a SYSOUT writer, a reader, and an initiator. It is recommended that the direct SYSOUT writer be used. The commands are:

```
MN JOB NAMES
START DSO.P1,00E
START RDR.S,00C
START INIT.ALL
```

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

18\*\* When the job is complete, a READER ENDED message will be printed, followed by a JOB ENDED message. The system will then enter the wait state. See the OS Operator's Reference for a discussion of stopping the system for MFT.

---

### List System Data

---

19. To list the data describing the system, place the control deck shown below in the input device.

Then enter a START RDR.S,00C command.

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP1     EXEC PGM=IEHLIST
//SYSPRINT  DD  SYSOUT=A
//DD1       DD  UNIT=3330,DISP=OLD,VOLUME=SER=DLIB01
//SYSIN     DD  *
             LISTCTLG
             LISTVTOC DUMP,VOL=3330=DLIB01
             LISTPDS  DSNAME=(SYS1.PROCLIB                      X
             [,Optional PDSS on System Residence])
             LISTPDS  DSNAME=(SYS1.MODGEN,SYS1.GENLIB,SYS1.SAMPLIB)
/*
```

If JOB2 is run immediately after JOB1 (in Step 12), eliminate the JOB2 card. In the LIST deck, any of the partitioned data sets shown in Figure 90 may be specified in the LISTPDS statement.

Note: You must carry out the IPL procedure (the steps denoted with an \*\*) if the preceding punch operation was skipped.

---

### Deleting Libraries Using a 3330

To make additional direct access storage available for the system generation process, SYS1.SAMPLIB can be deleted from DLIB01 before starting the system generation process. This may be accomplished by loading the operating system and using the control deck shown in the input device.

```
//JOB2      JOB ACCT123,PROGRAMMER,MSGLEVEL=1
//STEP2     EXEC PGM=IEHPRGM
//SYSPRINT  DD  SYSOUT=A
//DD1       DD  UNIT=3330,DISP=OLD,VOLUME=SER=DLIB01
//SYSIN     DD  *
             SCRATCH  DSNAME=SYS1.SAMPLIB,VOL=3330=DLIB01,PURGE
/*
```



## The Job Stream — Stage II

This section discusses the non-TSO and TSO job streams that result from Stage I of the system generation program. Both parts contain a brief description of the job stream, a diagram illustrating the job stream and the data sets affected, and a sample listing of the job control language statements.



## Non-TSO Generation

If no error messages are printed during Stage I, the job stream is produced on the SYSPUNCH data set. The job stream contains one JOB statement followed by many EXEC statements (see Figure 26). Each EXEC statement is followed by its associated DD statements and other data required to execute the assembler, linkage editor, and utility programs during Stage II.

The steps are numbered in Figure 26 for easy reference to the text.

The format of the JOB statement is:

```
//SYSGEN JOB 1,'SYSTEM GENERATION'
```

Note: This JOB statement is produced if JOBSTMT=SUPPLIED is not specified in the GENERATE macro instruction. If SUPPLIED is specified, your JOB statement is produced.

The format of the EXEC statement is:

```
//SGXX EXEC PGM=program[,COND=condition][,PARM=value]
```

where:

SGXX

is the step name. XX represents sequential identification numbers supplied by the system generation process. The step name is printed in the IEF236I allocation message while the step is being initiated. For example, the message

```
IEF236I ALLOC. FOR SYSGEN SG7
```

indicates that the seventh step is being executed.

PGM

indicates the name of the program being executed. The names are ASMBLR, IFCEREPO, IEWL, IEBCOPY, IEHIOSUP, IFCDIP00, IEHPROGM, and IEHLIST. The programs are executed in the order shown in Figure 19, the assembler (ASMBLR) and the linkage editor (IEWL) are executed several times. The IEWL step ⑦ builds the new SYS1.NUCLEUS. Six IEHPROGM steps may be executed; the ones that are executed depend on specifications in the GENERATE macro instruction.

COND

enables all the job steps to test whether the previous step was successfully completed. If the previous step was unsuccessful, the remaining steps are bypassed and Stage II is terminated.

PARM

is supplied for job steps that require PARM information.

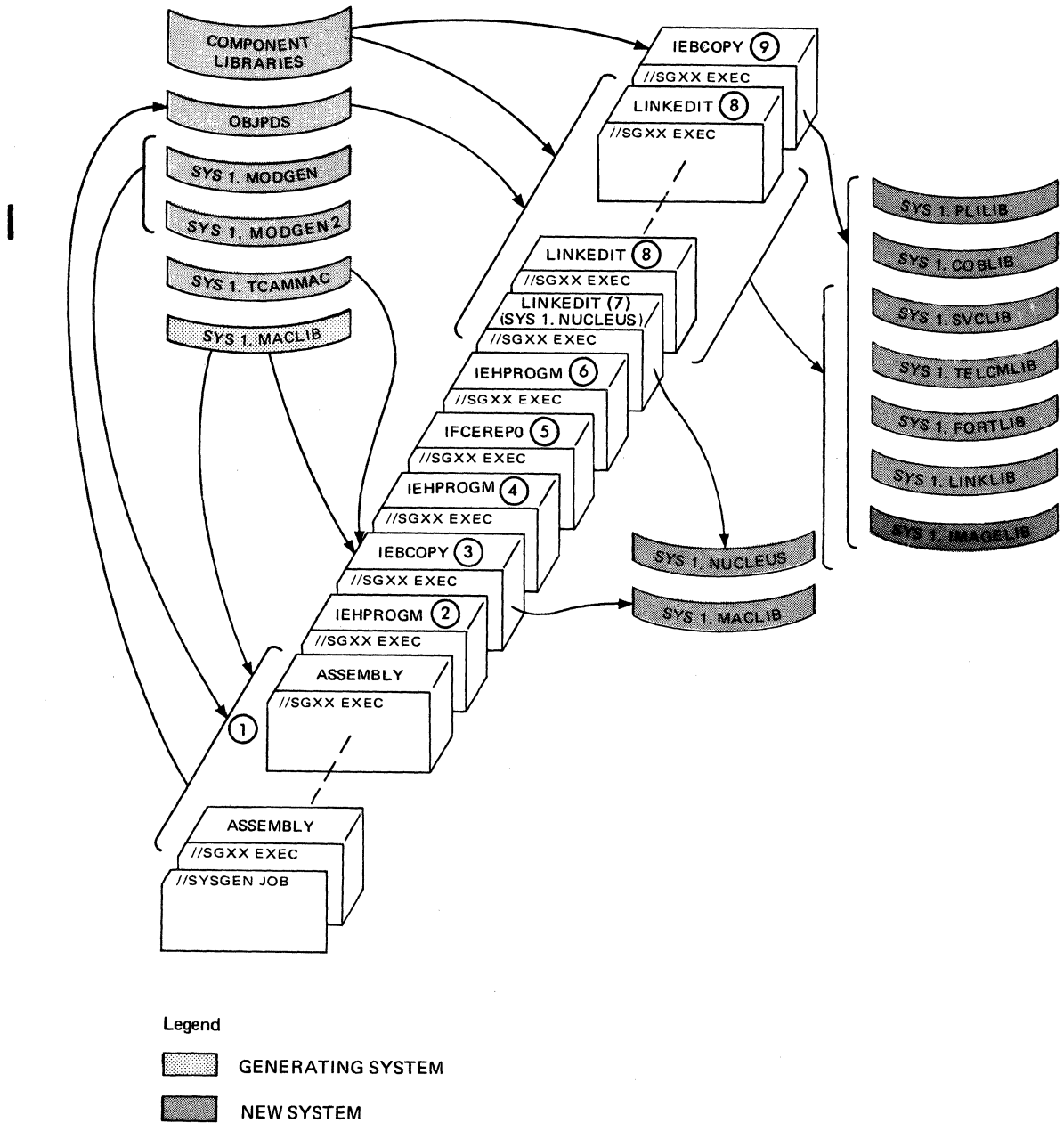


Figure 26 (Part 1 of 2). The Stage II Non-TSO Job Stream.

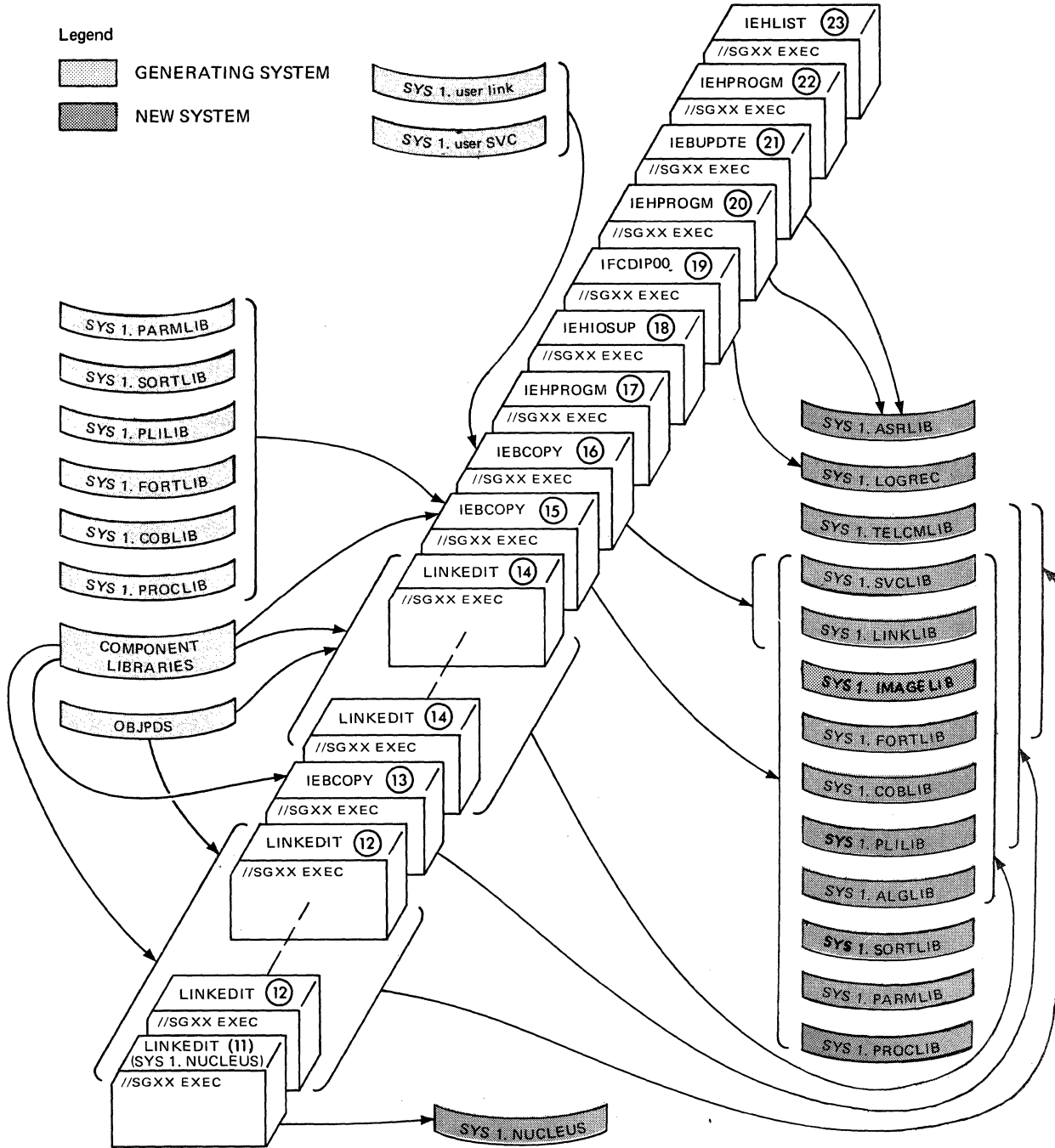


Figure 26 (Part 2 of 2). The Stage II Non-TSO Job Stream

During the ASMBLR steps ①, selected modules are assembled and stored in the utility data set defined by the OBJPDS DD statement in Stage I.

IEBCOPY utility program step ③ copies SYS1.MACLIB and SYS1.TCAMMAC (if TCAM is specified) to the new system. This IEBCOPY step is produced only if the MACLIB macro instruction was specified in Stage I. IEHPROGM utility program step ④ renames the new MACLIB data set to SYS1.MACLIB. This step is produced only if the MACLIB macro instruction is specified and your own qualifier was coded in the INDEX parameter of the GENERATE macro instruction.

The modules assembled during the ASMBLR steps ① and other modules from SYS1.component libraries are processed during the IEWL steps ⑦ ⑧ ⑩ to form load modules. These load modules are placed in the new SYS1.NUCLEUS, SYS1.SVCLIB, SYS1.LINKLIB, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.TELCMLIB, and SYS1.IMAGELIB. Only one IEWL step, step ⑦, is produced during a nucleus generation.

If UTDISP=UNCATLG or DELETE is specified in the GENERATE macro instruction, two IEHPROGM steps are executed. The first one, step ②, uncatalogs or deletes the data sets defined by UT1SDS and UT2SDS parameters of the GENERATE macro instruction. The second one, step ⑬, uncatalogs or deletes the data sets defined by the UT3SDS and UT4SDS keyword parameters of the GENERATE macro instruction. If UTDISP=KEEP is specified in the GENERATE macro instruction, the IEHPROGM steps are not produced.

During an I/O generation, if LOGREC=PRINT is specified in the GENERATE macro instruction, IFCEREPO (EREP) step ⑤ will print SYS1.LOGREC. IEHPROGM step ⑥ scratches SYS1.LOGREC and, if MCH is specified, uncatalogs and scratches SYS1.ASRLIB.

IEBCOPY step ⑨ copies load modules to the new SYS1.SVCLIB, SYS1.PL1LIB, SYS1.COBLIB, SYS1.LINKLIB, SYS1.IMAGELIB, SYS1.FORTLIB, and SYS1.TELCMLIB. This step is not produced during a nucleus generation or a processor generation.

IEBCOPY step ⑪ copies load modules to the new SYS1.PARMLIB, SYS1.PL1LIB, SYS1.FORTLIB, SYS1.COBLIB, SYS1.SORTLIB, SYS1.ALGLIB, SYS1.PROCLIB, SYS1.SVCLIB, and SYS1.LINKLIB. This step is produced only if GENTYPE=ALL or PROCESSOR and the appropriate processor macro instructions were specified during Stage I.

IEBCOPY step ⑫ copies user-written load modules to SYS1.SVCLIB and to SYS1.LINKLIB from the data sets specified in the SVCLIB and LINKLIB macro instructions. This step is produced only if the SVCLIB or LINKLIB macro instruction is used during Stage I.

IEHIOSUP utility program ⑭ builds the XCTL tables for type 4 SVCs in SYS1.SVCLIB. The IFCDIPO0 utility program ⑮ initializes SYS1.LOGREC and allocates space for SYS1.ASRLIB. The IEHIOSUP and IFCDIPO0 steps are produced for complete operating system and I/O device generations.

IEHPROGM step ⑯ catalogs SYS1.ASRLIB and SYS1.LOGREC. This step is produced only if SER=MCH is specified in the SUPRVSOR macro instruction during Stage I.

IEBUPDTE utility program ⑰ generates a 40-byte record on SYS1.ASRLIB. This step is produced only if MODEL=85 is specified in the CENPROCS macro instruction during Stage I.

IEHPROGM step ⑱ renames any of the following new system data sets if you defined them with qualifiers: ALGLIB, ASRLIB, COBLIB, FORTLIB, IMAGELIB, LINKLIB, LOGREC, NUCLEUS, PARMLIB, PL1LIB, PROCLIB, SORTLIB, SVCLIB, or TELCMLIB. This step is produced if a user-supplied qualifier was coded in the INDEX parameter of the GENERATE macro instruction.

IEHLIST utility program (19) lists the catalog of the new system-residence volume and any other data specified with the DIRDATA parameter of the GENERATE macro instruction. This step is not produced during a nucleus generation.

Figure 27 shows sample job control language statements for each type of step in the job stream. The values selected for the parameters result from the specifications in the system generation macro instructions. In Figure 27, the underlined macro instruction keywords are used to show where the values indicated by those keywords are placed. These keywords are from the GENERATE macro instruction unless otherwise indicated by a comment. Remarks (preceded by ...) do not appear in the statements.

#### Assembler

```
//SGXX EXEC PGM=ASMBLR,COND=(4,LT)
//SYSLIB DD DSNAME=SYS1.MODGEN,DISP=(SHR,PASS)
// DD DSNAME=SYS1.MODGEN2,DISP=(SHR,PASS)
// DD DSNAME=SYS1.MACLIB,DISP=SHR,VOLUME=(,RETAIN)
//SYSUT1 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT1SDS),DSNAME=UT1SDS
//SYSUT2 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT2SDS),DSNAME=UT2SDS
//SYSUT3 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT3SDS
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB, X
// LRECL=121,BLKSIZE=3509),SYSOUT=A
//SYSPUNCH DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=OBJPDS(member)
//SYSIN DD *
PRINT ON,NODATA
```

#### IEHPROGM

```
//SGXX EXEC PGM=IEHPROGM
//LOGREC DD DISP=OLD,UNIT=RESNAME,VOLUME=(,RETAIN,SER=RESVOL)
//SYSPRINT DD SYSOUT=A,SPACE=(121,(500,1000),RLSE), X
// DCB=(RECFM=FB,LREC=121,BLKSIZE=121)
//SYSIN DD *
```

#### IEBCOPY

```
//SGXX EXEC PGM=IEBCOPY,PARM='POWER=1',COND=(8,LT)
//SYSUT3 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT3SDS
//SYSUT4 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT4SDS
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB, X
// LRECL=121,BLKSIZE=121),SYSOUT=A
//complib DD DISP=SHR,DSNAME=component library,VOLUME=(,RETAIN)
//SVCLIB DD DISP=OLD,VOLUME=(,RETAIN,SER=RESVOL),UNIT=RESNAME, X
// DSNAME=index.SVCLIB
//LINKLIB DD DISP=OLD,VOLUME=(,RETAIN,SER=LINKVOL),UNIT=LINKNAME, X
// DSNAME=index.LINKLIB
//SYSIN DD *
```

#### IFCEREPO

```
//SGXX EXEC PGM=IFCEREPO
//STEPLIB DD DSNAME=SYS1.LINKLIB,UNIT=LNKNAME,DISP=SHR, X
// VOLUME=(,RETAIN,SER=LNKVOL)
//SERLOG DD DSNAME=SYS1.LOGREC,UNIT=RESNAME,DISP=SHR, X
// VOLUME=(,RETAIN,SER=RESVOL)
//ERFPPT DD SYSOUT=A,SPACE=(121,(500,1000),RLSE), X
// DCB=(RECFM=FB,LRECL=121,BLKSIZE=121)
```

Figure 27 (Part 1 of 2). Sample Steps in the Job Stream

---

Linkage Editor

```
//SGXX EXEC PGM=IEWL,PARM='NCAL,XREF,LIST,LET',COND=(8,LT)
//SYSUT1 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT3SDS
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB, X
//
//          LRECL=121,BLKSIZE=121),SYSOUT=A
//SYSLMOD DD DISP=OLD,UNIT=RESNAME,VOLUME=SER=RESVOL, X
//          DSNAME=index.name(member)
//complib DD DISP=SHR,DSNAME=component library,VOLUME=(,RETAIN)
//SYSPUNCH DD DISP=OLD,VOLUME=(,RETAIN),DCB=(,RECFM=F,BLKSIZE=80), X
//          DSNAME=OBJPDS
//RESLIB DD DISP=SHR,VOLUME=(,RETAIN),DSNAME=PDS ...RESMODS macro
//SYSLIN DD *
```

IEHIOSUP

```
//SGXX EXEC PGM=IEHIOSUP
//SYSUT1 DD DSNAME=index.SVCLIB,DISP=(OLD,KEEP), X
//          UNIT=RESNAME,VOLUME=SER=RESVOL
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB, X
//          LRECL=121,BLKSIZE=121)SYSOUT=A
```

IFCDIP00

```
//SGXX EXEC PGM=IFCDIP00,PARM=value
//SERERDS DD DSNAME=index.LOGREC,UNIT=RESNAME,DISP=(NEW,KEEP), X
//          VOLUME=(,RETAIN,SER=RESVOL),SPACE=(allocation)
```

IEHLIST

```
//SGXX EXEC PGM=IEHLIST
//LINK DD DISP=OLD,VOLUME(,RETAIN,SER=LNKVOL),UNIT=LNKNAME
//SYSRES DD DISP=OLD,VOLUME=(,RETAIN,SER=RESVOL),UNIT=RESNAME
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB, X
//          LRECL=121,BLKSIZE=121),SYSOUT=A
//SYSIN DD *
```

---

Figure 27 (Part 2 of 2). Sample Steps in the Job Stream



## TSO Generation

If no error messages are printed during Stage I, the job stream is produced on the SYSPUNCH data set. The job stream contains one JOB statement followed by many EXEC statements (see Figure 28). Each EXEC statement is followed by its associated DD statements and other data required to execute the assembler, linkage editor, and utility programs during Stage II.

The steps are numbered in Figure 28 for easy reference to the text.

The format of the JOB statement is:

```
//SYSGEN JOB 1,'SYSTEM GENERATION'
```

Note: This JOB statement is produced if JOBSTMT=SUPPLIED is not specified in the GENTSO macro instruction. If SUPPLIED is specified, your JOB statement is produced.

The format of the EXEC statement is:

```
//SGXX EXEC PGM=program[ ,COND=condition ][ ,PARM=value ]
```

where:

**SGXX**

is the step name. XX represents sequential identification numbers supplied by the system generation process. The step name is printed in the IEF236I allocation message while the step is being initiated. For example, the message

```
IEF236I ALLOC. FOR SYSGEN SG7
```

indicates that the seventh step is being executed.

**PGM**

indicates the name of the program being executed. The names are ASMBLR, IFCEREPO, IEWL, IEBCOPY, IEHIOSUP, IFCDIP00, IEHPROGM, and IEHLIST. The programs are executed in the order shown in Figure 28. The assembler (ASMBLR) and the linkage editor (IEWL) are executed several times. IEWL step ① builds the new SYS1.NUCLEUS. Seven IEHPROGM steps may be executed; the ones executed depend on specifications in the GENTSO macro instruction.

**COND**

enables all the job steps to test whether the previous step was successfully completed. If the previous step was unsuccessful, the remaining steps are bypassed and Stage II is terminated.

**PARM**

is supplied for job steps that require PARM information.

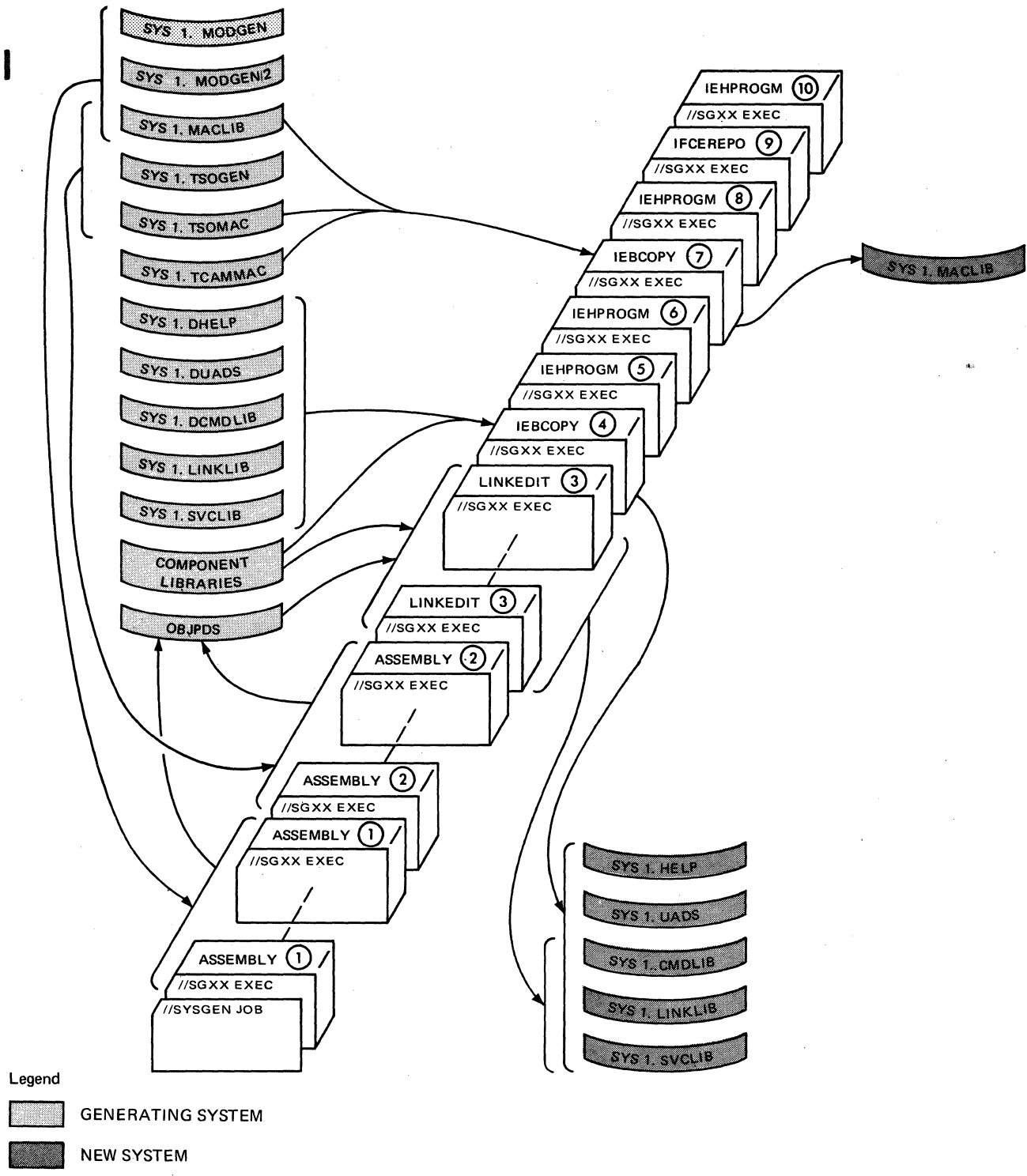


Figure 28 (Part 1 of 2). The Stage II TSO Job Stream

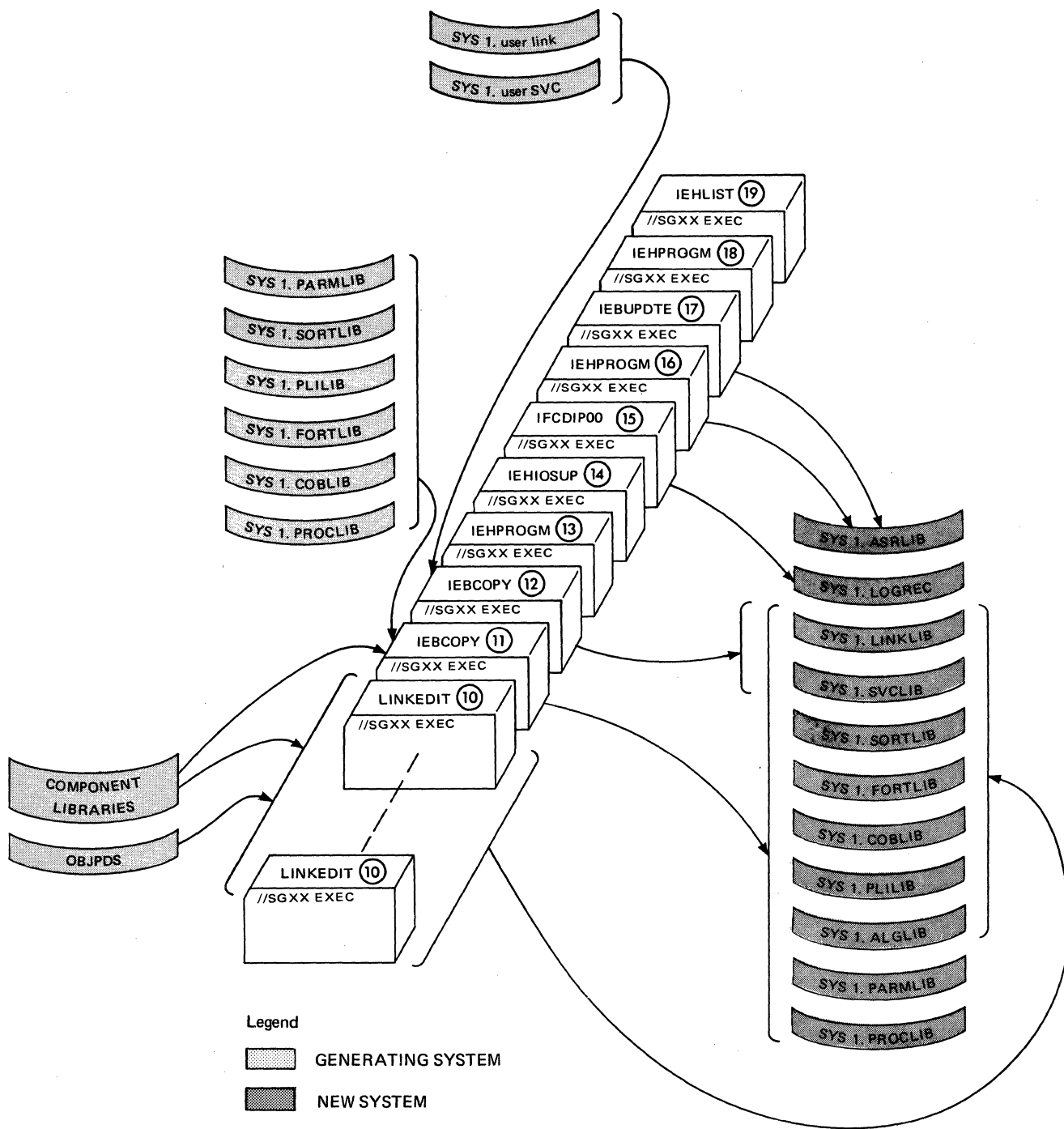


Figure 28 (Part 2 of 2). The Stage II TSO Job Stream

During the ASMBLR steps ① and ②, selected modules are assembled and stored in the utility data set defined by the OBJPDS DD statement in Stage I.

The IEBCOPY utility program step ④ copies modules to SYS1.HELP, SYS1.CMDLIB, SYS1.UADS, SYS1.LINKLIB, and SYS1.SVCLIB. IEHPROGM utility program step ⑤ renames SYS1.HELP and SYS1.UADS.

IEBCOPY step ⑦ copies SYS1.MACLIB, SYS1.TCAMMAC, and SYS1.TSOMAC to the new system. This IEBCOPY step is produced only if the MACLIB macro instruction was specified in Stage I. IEHPROGM step ⑧ renames the new MACLIB data set to SYS1.MACLIB. This step is produced only if the MACLIB macro instruction is specified and your own qualifier was coded in the INDEX parameter of the GENTSO macro instruction.

The modules assembled during the ASMBLR steps ② and other modules from SYS1 component libraries are processed during IEWL steps ③ to form load modules. These load modules are placed in the new SYS1.CMDLIB, SYS1.SVCLIB, and SYS1.LINKLIB.

The modules assembled during the ASMBLR step ① and other modules from SYS1 component libraries are processed during the IEWL steps ⑪ ⑫ ⑭ to form load modules. These load modules are placed in the new SYS1.NUCLEUS, SYS1.SVCLIB, SYS1.LINKLIB, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.SORTLIB, and SYS1.TELCMLIB. Only one IEWL step, step ⑪, is produced during a nucleus generation.

If UTDISP=UNCATLG or DELETE is specified in the GENTSO macro instruction, two IEHPROGM steps are executed. The first one, step ⑥, uncatalogs or deletes the data sets defined by UT1SDS and UT2SDS parameters of the GENTSO macro instruction. The second one, step ⑰, uncatalogs or deletes the data sets defined by the UT3SDS and UT4SDS keyword parameters of the GENTSO macro instruction. If UTDISP=KEEP is specified in the GENTSO macro instruction, these IEHPROGM steps are not produced.

During an I/O generation, if LOGREC=PRINT is specified in the GENTSO macro instruction, IFCEREPO (EREP) step ⑨ will print SYS1.LOGREC. IEHPROGM step ⑩ scratches SYS1.LOGREC and, if MCH is specified, uncatalogs and scratches SYS1.ASRLIB.

IEBCOPY step ⑬ copies load modules to the new SYS1.SVCLIB, SYS1.PL1LIB, SYS1.COBLIB, SYS1.LINKLIB, SYS1.FORTLIB, and SYS1.TELCMLIB. This IEBCOPY step is not produced during a nucleus generation or a processor generation.

IEBCOPY step ⑮ copies load modules to the new SYS1.PARMLIB, SYS1.PL1LIB, SYS1.FORTLIB, SYS1.COBLIB, SYS1.SORTLIB, SYS1.ALGLIB, SYS1.PROCLIB, SYS1.SVCLIB, and SYS1.LINKLIB. This step is produced only if GENTYPE=ALL or PROCESSOR and the appropriate processor macro instructions were specified during Stage I.

IEBCOPY step ⑯ copies user-written load modules to SYS1.SVCLIB and to SYS1.LINKLIB from the data sets specified in the SVCLIB and LINKLIB macro instructions. This step is produced only if the SVCLIB or LINKLIB macro instruction is used during Stage I.

IEHIOSUP utility program ⑱ builds the XCTL tables for type 4 SVCs in SYS1.SVCLIB. When running IEHIOSUP against a TSO system, the EXEC card must include PARM=TSO. The IFCDIP00 utility program ⑲ initializes SYS1.LOGREC and allocates space for complete operating system and I/O device IFCDIP00 steps are produced for complete operating system and I/O device generations.

IEHPROGM step ⑳ catalogs SYS1.ASRLIB and SYS1.LOGREC. This step is produced only if SER=MCH is specified in the SUPRVSOR macro instruction during Stage I.

IEBUPDTE utility program ㉑ generates a 40-byte record on SYS1.ASRLIB. This step is produced only if MODEL=85 is specified in the CENPROCS macro instruction during Stage I.

IEHPROGM step (22) renames any of the following new system data sets that were defined with user-supplied qualifiers: AIGLIB, ASRLIB, COBLIB, FORTLIB, LINKLIB, LOGREC, NUCLEUS, PARMLIB, PL1LIB, PROCLIB, SORTLIB, SVCLIB, or TELCMLIB. This step is produced if you coded your own qualifier in the INDEX parameter of the GENTSO macro instruction.

IEHLIST utility program (23) lists the catalog of the new system-residence volume and any other data specified with the DIRDATA parameter of the GENTSO macro instruction. This step is not produced during a nucleus generation.

Figure 29 shows sample job control language statements for each type of step in the job stream. The values selected for the parameters result from the specifications in the system generation macro instructions. In Figure 29, the underlined macro instruction keywords are used to show where the values indicated by those keywords are placed. These keywords are from the GENTSO macro instruction unless otherwise indicated by a comment. Remarks (preceded by ...) do not appear in the statements.

#### Assembler

```
//SGXX EXEC PGM=ASMBLR,COND=(4,LT)
//SYSLIB DD DSNAME=SYS1.MODGEN,DISP=(SHR,PASS)
//      DD DSNAME=SYS1.MODGEN2,DISP=(SHR,PASS)
//      DD DSNAME=SYS1.MACLIB,DISP=SHR,VOLUME=(,RETAIN)
//SYSUT1 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT1SDS),DSNAME=UT1SDS
//SYSUT2 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT2SDS),DSNAME=UT2SDS
//SYSUT3 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT3SDS
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),DCB=(,RECFM=FB,          X
//          LRECL=121,BLKSIZE=3509),SYSOUT=A
//SYSPUNCH DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=OBJPDS(member)
//SYSIN DD *
PRINT ON,NODATA
```

#### Assembler

```
//SGXX EXEC PGM=ASMBLR,COND=(4,LT)
//SYSLIB DD DSNAME=SYS1.TSOGEN,DISP=(SHR,PASS)
//      DD DSNAME=SYS1.MODGEN2,DISP=SHR,VOLUME=(,RETAIN)
//      DD DSNAME=SYS1.MODGEN,DISP=SHR,VOLUME=(,RETAIN)
//      DD DSNAME=SYS1.TSOMAC,DISP=SHR,VOLUME=(,RETAIN)
//      DD DSNAME=SYS1.MACLIB,DISP=SHR,VOLUME=(,RETAIN)
//SYSUT1 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT1SDS),DSNAME=UT1SDS
//SYSUT2 DD DISP=OLD,VOLUME=(,RETAIN),LABEL=(,UT2SDS),DSNAME=UT2SDS
//SYSUT3 DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=UT3SDS
//SYSPRINT DD SPACE=(121,(500,1000),RLSE),SCB=(,RECFM=FB,          X
//          LRECL=121,BLKSIZE=3509),SYSOUT=A
//SYSPUNCH DD DISP=OLD,VOLUME=(,RETAIN),DSNAME=OBJPDS(member)
//SYSIN DD *
PRINT ON,NODATA
```

#### IEHPROGM

```
//SGXX EXEC PGM=IEHPROGM
//LOGREC DD DISP=OLD,UNIT=RESNAME,VOLUME=(,RETAIN,SER=RESVOL)
//SYSPRINT DD SYSOUT=A,SPACE=(121,(500,1000),RLSE),          X
//          DCB=(RECFM=FB,LREC=121,BLKSIZE=121)
//SYSIN DD *
```

Figure 29 (Part 1 of 2). Sample Steps in the Job Stream

## IEBCOPY

```

//SGXX EXEC PGM=IEBCOPY, PARM='POWER=1', COND=(8,LT)
//SYSUT3 DD DISP=OLD, VOLUME=(, RETAIN), DSNAME=UT3SDS
//SYSUT4 DD DISP=OLD, VOLUME=(, RETAIN), DSNAME=UT4SDS
//SYSPRINT DD SPACE=(121, (500, 1000), RLSE), DCB=(, RECFM=FB,      X
//      LRECL=121, BLKSIZE=121), SYSOUT=A
//complib DD DISP=SHR, DSNAME=component library, VOLUME=(, RETAIN)
//SVCLIB DD DISP=OLD, VOLUME=(, RETAIN, SER=RESVOL), UNIT=RESNAME,    X
//      DSNAME=index.SVCLIB
//LINKLIB DD DISP=OLD, VOLUME=(, RETAIN, SER=LINKVOL), UNIT=LINKNAME,  X
//      DSNAME=index.LINKLIB
//SYSIN DD *

```

## IFCEREPO

```

//SGXX EXEC PGM=IFCEREPO
//STEPLIB DD DSNAME=SYS1.LINKLIB, UNIT=LNKNAME, DISP=SHR,            X
//      VOLUME=(, RETAIN, SER=LNKVOL)
//SERLOG DD DSNAME=SYS1.LOGREC, UNIT=RESNAME, DISP=SHR,             X
//      VOLUME=(, RETAIN, SER=RESVOL)
//ERFPPT DD SYSOUT=A, SPACE=(121, (500, 1000), RLSE),              X
//      DCB=(RECFM=FB, LRECL=121, BLKSIZE=121)

```

## Linkage Editor

```

//SGXX EXEC PGM=IEWL, PARM='NCAL, XREF, LIST, LET', COND=(8,LT)
//SYSUT1 DD DISP=OLD, VOLUME=(, RETAIN), DSNAME=UT3SDS
//SYSPRINT DD SPACE=(121, (500, 1000), RLSE), DCB=(, RECFM=FB,      X
//      LRECL=121, BLKSIZE=121), SYSOUT=A
//SYSLMOD DD DISP=OLD, UNIT=RESNAME, VOLUME=SER=RESVOL,             X
//      DSNAME=index.name(member)
//complib DD DISP=SHR, DSNAME=component library, VOLUME=(, RETAIN)
//SYSPUNCH DD DISP=OLD, VOLUME=(, RETAIN), DCB=(, RECFM=F, BLKSIZE=80), X
//      DSNAME=OBJPDS
//RESLIB DD DISP=SHR, VOLUME=(, RETAIN), DSNAME=PDS ... RESMODS macro
//SYSLIN DD *

```

## IEHIOSUP

```

//SGXX EXEC PGM=IEHIOSUP, PARM=TSO
//SYSUT1 DD DSNAME=index.SVCLIB, DISP=(OLD, KEEP),                  X
//      UNIT=RESNAME, VOLUME=SER=RESVOL
//SYSPRINT DD SPACE=(121, (500, 1000), RLSE), DCB=(, RECFM=FB,      X
//      LRECL=121, BLKSIZE=121), SYSOUT=A

```

## IFCDIP00

```

//SGXX EXEC PGM=IFCDIP00, PARM=value
//SERERDS DD DSNAME=index.LOGREC, UNIT=RESNAME, DISP=(NEW, KEEP),    X
//      VOLUME=(, RETAIN, SER=RESVOL), SPACE=(allocation)

```

## IEHLIST

```

//SGXX EXEC PGM=IEHLIST
//LINK DD DISP=OLD, VOLUME=(, RETAIN, SER=LNKVOL), UNIT=LNKNAME
//SYSRES DD DISP=OLD, VOLUME=(, RETAIN, SER=RESVOL), UNIT=RESNAME
//SYSPRINT DD SPACE=(121, (500, 1000), RLSE), DCB=(, RECFM=FB,      X
//      LRECL=121, BLKSIZE=121), SYSOUT=A
//SYSIN DD *

```

Figure 29 (Part 2 of 2). Sample Steps in the Job Stream

## Restart Procedures

This section contains detailed information about restarting the system generation program and describes recovery procedures. The Stage I text discusses the most common causes of error, restart procedures, and an example of an input deck for restarting Stage I.

The Stage II text discusses the most common causes of error, restart techniques, guidelines for restarting a non-TSO Stage II, guidelines for restarting a TSO Stage II, and reallocating data sets. Examples of the input decks are included.





The system generation process may come to an unsatisfactory completion due to errors during Stage I or Stage II. The procedures to restart Stage I and Stage II are described below.

## Restarting Stage I

The most common causes of error during Stage I are:

- Faulty allocation of the utility data sets for system generation. For example, if there is not enough space on the volume for the data set, an error occurs.
- Keypunching errors in the input deck.
- Contradictory or invalid specifications in the system generation macro instructions.

Faulty allocation of the utility data sets usually causes an abnormal-end-of-task (ABEND) termination. 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 D) printed in the SYSPRINT data set. If any system generation errors are found during Stage I, the job stream is not produced.

Stage I consists of only one job step (execution of the assembler) and it can be restarted only from the beginning. To restart Stage I, follow these procedures:

1. Correct the input deck for system generation.
2. Scratch and uncatalog the utility data sets (specified by the OBJPDS, SYSUT1, SYSUT2, and SYSUT3 DD statements in the input deck).

The utility data sets can be scratched and uncataloged if you insert the statements shown in Figure 30 in the input deck for system generation. These statements must precede the EXEC PGM=ASMBLR statement of the input deck, as shown in Figure 31.

---

```
//SCRATCH EXEC PGM=IEHPROGM -RESTART DECK-  
//SYSPRINT DD SYSOUT=A  
//OBJPDS DD DSN=SYS1.name,DISP=(OLD,DELETE)  
//SYSUT1 DD DSN=SYS1.name,DISP=(OLD,DELETE)  
//SYSUT2 DD DSN=SYS1.name,DISP=(OLD,DELETE)  
//SYSUT3 DD DSN=SYS1.name,DISP=(OLD,DELETE)  
//SYSIN DD DUMMY
```

---

Figure 30. Control Statements to Scratch and Uncatalog the Utility Data Sets

In the preceding statements, the values given to the DSN parameters must be the same as the values given to DSN in the corresponding DD statements in the input deck for Stage I. These statements represent a dummy execution of the IEHPROGM utility program. The utility data sets are scratched and uncataloged through the job control language or through the SCRATCH and UNCATLG statements of IEHPROGM as described in OS Utilities.

Figure 31 shows an input deck for restarting Stage I. It is assumed that all errors in the input deck have been corrected. The four utility data sets are named SYS1.OBJECT, SYS1.ONE, SYS1.TWO, and SYS1.THREE. SYS1.ONE resides on

an unlabeled 9-track magnetic tape volume whose serial number is 000167. SYS1.OBJECT, SYS1.TWO, and SYS1.THREE reside on 2311 volumes whose serial numbers are SYSTEM, GENVOL, and MODVOL, respectively. The job stream is to be punched by the device designated by the SYSCP unit name.

```

//SYSGEN   JOB MSGLEVEL=1   -SYSTEM GENERATION-
//SCRATCH  EXEC PGM=IEHPRGM  -RESTART DECK-
//SYSPRINT DD  SYSOUT=A
//OBJPDS   DD  DSNAME=SYS1.OBJECT,DISP=(OLD,DELETE)
//SYSUT1   DD  DSNAME=SYS1.ONE,DISP=(OLD,DELETE)
//SYSUT2   DD  DSNAME=SYS1.TWO,DISP=(OLD,DELETE)
//SYSUT3   DD  DSNAME=SYS1.THREE,DISP=(OLD,DELETE)
//SYSIN    DD  DUMMY
//STEP1    EXEC PGM=ASMBLR    -STAGE I INPUT DECK-
//SYSLIB   DD  DSNAME=SYS1.GENLIB,DISP=OLD
//OBJPDS   DD  DSNAME=SYS1.OBJECT,VOLUME=(,RETAIN,SER=SYSTEM),      X
//          DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(40,20,8))
//SYSUT1   DD  DSNAME=SYS1.ONE,VOLUME=(RETAIN,SER=000167),          X
//          DISP=(,CATLG),UNIT=2400,LABEL=(,NL)
//SYSUT2   DD  DSNAME=SYS1.TWO,VOLUME=(,RETAIN,SER=GENVOL),        X
//          DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(240,20))
//SYSUT3   DD  DSNAME=SYS1.THREE,VOLUME=(,RETAIN,SER=MODVOL),      X
//          DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(300,20))
//DUMMY    DD  VOLUME=(,RETAIN,REF=*.SYSUT3),SPACE=(TRK,(80))
//SYSPUNCH DD  UNIT=SYSCP
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
.
.
.   System generation macro instructions
.
.
END
/*
//

```

Figure 31. Restarting Stage I

## Restarting Stage II

The most common causes of error during Stage II are:

- Machine interruptions and noncontinuous machine time.
- Faulty space allocation of the system data sets during the preparation for system generation, especially the primary allocation and the directory size.
- Errors in the input deck that cannot be detected during Stage I. For example, if SYS1.NUCLEUS was allocated space on volume 111111 during the preparation of system generation, and if RESVOL=A11111 is specified in the GENERATE macro instruction, an error occurs.
- The job stream is too large for the generating SYS1.SYSJOBQE.
- Procedural errors, such as volume mounting problems.

Stage II consists of the execution of the job stream produced during Stage I. This job stream is one job that has several job steps. Restarting can be accomplished by starting at the beginning of any step in the job stream, providing all previous steps have been properly executed. Before restarting,

some other operations may be required depending upon the reason for restarting and the number of steps already executed. The following topics describe the job stream, discuss restart techniques, give guidelines for restarting Stage II, describe the reallocation of data sets, and give guidelines for changing the size of SYS1.SYSJOBQE.

## RESTART TECHNIQUES

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

### Restarting From Cards

If the job stream is on cards, a job step can be restarted by placement of a JOB card ahead of the step's EXEC card. A START RDR command must then be issued for the card reader.

### Punching the Job Stream

If the unit (SYSPUNCH) specified for the job stream during Stage I was not a card punch, the IEBPTPCH utility program can be used to punch the job stream. Figure 32 shows the statements required for IEBPTPCH.

---

```
//PUNCH      JOB          -PUNCH JOB STREAM-
//          EXEC PGM=IEBPTPCH
//SYSUT1     DD  UNIT=xxx,LABEL=(,NL),VOLUME=SER=serial,DISP=OLD
//SYSUT2     DD  (Parameters designating a card punch)
//SYSPRINT   DD  SYSOUT=A
//SYSIN      DD  *
              PUNCH  TYPORG=PS,MAXFLDS=1
              RECORD  FIELD=(80)
/*
```

---

Figure 32. Control Statements for IEBPTPCH

- The value of the UNIT parameter of the SYSUT1 DD statement is the specific unit address of the magnetic tape drive on which the job stream tape resides. Unless the job stream tape had been demounted, the value of this UNIT parameter is the same as the value of the UNIT parameter of the SYSPUNCH DD statement in the input deck for Stage I.
- The value of the VOLUME parameter of the SYSUT1 DD statement is either any external serial number you have assigned to the job stream tape reel or any arbitrary name chosen by the programmer. The system will issue a MOUNT command for the specified volume on the magnetic tape drive indicated with the UNIT parameter. (The system has no way of checking that the correct volume has been mounted because the job stream tape is unlabeled.)

- Sequence numbers can be specified for the punched cards using the CDSEQ and CDINCR parameters in the PUNCH utility statement.

The assembly listing produced (SYSPRINT) at the end of Stage I contains a series of PUNCH statements. The operands of these PUNCH statements are the cards of the job stream. If a separate listing of the job stream is desired, the IEBPTPCH utility program can be used to print the job stream.

Figure 33 shows a series of statements that can be used to punch the job stream on any 2540 card read punch. The SYSPUNCH DD statement in the Stage I input deck was: //SYSPUNCH DD UNIT=182,LABEL=(,NL)

---

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

---

Figure 33. Punching the Job Stream

#### Restarting From Tape

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

IEBEDIT can be used to create a new job stream. The new job stream contains one of the following arrangements:

- The job step you specify and all the steps that follow it, or
- Only those job steps you specify, or
- All job steps except those you specify.

---

```
//RESTART  JOB                -RESTART STAGE II-
//          EXEC PGM=IEBEDIT
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  UNIT=xxx,LABEL=(,NL),VOLUME=SER=serial,DISP=OLD
//SYSUT2   DD  UNIT=xxx,LABEL=(,NL)
//SYSIN    DD  *
           EDIT   START=SYSGEN,STEPNAME=SGXX[,NOPRINT]
or EDIT    START=SYSGEN,TYPE=INCLUDE,                               X
           STEPNAME=(SGXX[,SGXX]...)[,NOPRINT]
or EDIT    START=SYSGEN,TYPE=EXCLUDE,                               X
           STEPNAME=(SGXX[,SGXX]...)[,NOPRINT]
/*
```

---

Figure 34. Control Statements for IEBEDIT

- The value of the UNIT parameter of the SYSUT1 DD statement is the specific unit address of the magnetic tape drive on which the job stream tape resides. Unless the job stream tape had been demounted, the value of this UNIT parameter is the same as the value of the UNIT parameter of the SYSPUNCH DD statement in the input deck for Stage I.

- The value of the VOLUME parameter of the SYSUT1 DD statement is either any external serial number assigned to the job stream tape reel or any arbitrary name chosen by the programmer. The system will issue a MOUNT command for the specified volume on the magnetic tape drive indicated with the UNIT parameter. (The system has no way of checking that the correct volume has been mounted because the job stream tape is unlabeled.)
- The value of the UNIT parameter of the SYSUT2 DD statement is the specific unit address of a magnetic tape drive.
- Only one EDIT statement must be used.
- If the TYPE parameter is omitted, STEPNAME specifies the first job step to be placed in the new job stream. All the steps that follow the job step specified are also placed in the new job stream.
- If TYPE=INCLUDE or TYPE=EXCLUDE is specified, STEPNAME specifies the job steps to be included or excluded, respectively, from the new job stream. Individual job steps and sequences of job steps can be specified for inclusion or exclusion. For example:

```
STEPNAME=(SG20,SG32-SG37,SG50)
```

indicates that steps SG20, SG32 through SG37, and SG50 are to be included or excluded from the operation.

- NOPRINT must be included if a listing of the new job stream is not desired. After the new job stream is created, a START RDR command must be issued for the magnetic tape drive designated by the SYSUT2 DD statement.

Refer to OS Utilities, for a more detailed description of IEBEDIT.

Figure 35 shows an IEBEDIT input deck for restarting Stage II. The job stream resides on unit 182. The new job stream will reside on unit 282. All job steps after and including SG7 are to be restarted. A listing of the new job stream is not desired. The START RDR command to start the new job stream is included with the IEBEDIT deck. The appropriate operations described in "Guidelines for Restarting Stage II" must have been performed.

---

```
//RESTART JOB          -RESTART STAGE II-
//          EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD UNIT=182,LABEL=(,NL),VOLUME=SER=JOBSTM,DISP=OLD
//SYSUT2   DD UNIT=282,LABEL=(,NL)
//SYSIN    DD *
//          EDIT      START=SYSGEN,STEPNAME=SG7,NOPRINT
/*
//          START     RDR,282
```

---

Figure 35. Input deck for IEBEDIT

#### GUIDELINES FOR RESTARTING STAGE II (NON-TSO GENERATION)

The following sections give guidelines for restarting during Stage II of a non-TSO generation. Restarting may require the scratching and reallocation

of data sets. When this is necessary, reference is made to the section "Reallocating Data Sets" for the procedure to be followed. After these operations have been performed, the actual restarting of Stage II can be accomplished by one of the methods described in "Restart Techniques."

The numbered steps in this section refer to steps in Figure 26.

**Note:** If you have specified a qualifier in the INDEX parameter of the GENERATE macro instruction and errors have occurred before the two IEHPROGM rename steps ④ and ⑱, you should check to be sure that these rename steps have not been executed. If a library has been renamed with the qualifier SYS1 and it is necessary to restart a step that refers to that library, the IEHPROGM utility program can be used to rename the library with the qualifier you supplied.

#### Restarting During Assemblies

To restart at the beginning of any assembly, place the JOB card in front of the EXEC card of that assembly or use the IEBEDIT program. If more than one ASMBLR step has been executed, and if it is desired to restart at the beginning of Stage II (SG1 step), the data set defined by the OBJPDS DD statement must be reallocated (see "Reallocating Data Sets").

All assemblies must be satisfactorily executed before the link-edit (IEWL) steps can be executed.

#### Restarting Any IEHPROGM Step

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

The ASMBLR, IEWL, IEBCOPY, IEHIOSUP, and IFCDIP00 steps must be executed without error before the IEHPROGM step ⑱, the rename IEHPROGM step, is executed.

#### Restarting Link-Edit Steps

IEWL step ⑦ builds the new SYS1.NUCLEUS. Subsequent IEWL steps, ⑧ and ⑩ produce load modules for the new SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.LINKLIB, and, if specified for the new system, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.SORTLIB, and SYS1.TELCMLIB.

If IEWL step ⑦ was executed, SYS1.NUCLEUS must be reallocated before the step is restarted (see "Reallocating Data Sets").

To restart at the beginning of IEWL step ⑧, any of the following libraries that contain system data must be reallocated: SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.SORTLIB, and SYS1.TELCMLIB. If more than one IEWL step was executed, and if you want to restart with IEWL step ⑦, SYS1.NUCLEUS must also be reallocated.

If reallocation is not performed, the space already used in the new system libraries is not available, and therefore there may not be enough space available for these data sets. This is particularly important for those system data sets that cannot have a secondary allocation. If more than sufficient space was allocated to a data set during the preparation for system generation, the IEWL step that failed can be restarted without reallocation.

It is recommended that the reallocations indicated above be performed when you restart from the beginning of the IEWL steps ⑦ or ⑧. To restart from the beginning of Stage II, the data sets defined by the OBJPDS, SYSUT1, and SYSUT2 DD statements must also be reallocated.

During an I/O generation, no reallocation is needed to restart a link-edit step unless a step failed because there was not enough space in a system data set. If there was not enough space, move the contents of the old library to the new library. Copy the data set to a scratch volume using the IEBCOPY utility program. Use IEHPROGM to scratch and uncatalog the space for the data set on its original volume. Allocate and catalog the required space for the data set on its original volume and use IEBCOPY to copy the data set from the scratch volume to its original volume. Then restart Stage II at the beginning of the link-edit step that failed.

#### Restarting First IEBCOPY Step

This IEBCOPY step ⑦ copies SYS1.MACLIB to the new system. To restart at the beginning of this step, the new SYS1.MACLIB must be reallocated (see "Reallocating Data Sets").

#### Restarting All Other IEBCOPY Steps

IEBCOPY step ⑨ copies members to the new SYS1.SVCLIB, SYS1.COBLIB, SYS1.IMAGELIB, SYS1.PL1LIB, SYS1.LINKLIB, SYS1.FORTLIB, and SYS1.TELCLIB. No reallocation is needed to restart this step unless the step failed because there was not enough space in a system data set. Only the system data set that did not have enough space must be reallocated (see "Reallocating Data Sets"), because any member that existed in the new system data set will not be moved into it again when the step is restarted.

If the data set that did not have enough space was SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.FORTLIB, or SYS1.TELCLIB, one of the two following procedures must be performed:

- Follow the procedures for restarting with IEWL step ⑧ as described in "Restarting Link Edit Steps" in the "Guidelines for Restarting Stage II (Non-TSO Generation)" section, or
- Copy the data set to a scratch volume using the IEBCOPY utility program. Scratch and uncatalog the space for the data set on its original volume with IEHPROGM. Allocate and catalog the required space for the data set on its original volume. Use IEBCOPY to copy the data set from the scratch volume to its original volume. Restart Stage II at the beginning of IEBCOPY step ⑨.

IEBCOPY step ⑪ copies members to SYS1.SVCLIB, SYS1.FORTLIB, SYS1.COBLIB, SYS1.ALGLIB, SYS1.PL1LIB, SYS1.LINKLIB, SYS1.PROCLIB, SYS1.SORTLIB, and SYS1.PARMLIB. The procedures to restart IEBCOPY step ⑨ also apply to this step.

IEBCOPY step ⑫ copies your members to the new SYS1.SVCLIB or SYS1.LINKLIB. The procedures to restart IEBCOPY step ⑨ also apply to this step.

Restarting IEHIOSUP, IFCDIP00, and IEHLIST

The system data sets do not have to be reallocated when IEHIOSUP step (14) and IEHLIST step (19) are restarted. Before restarting IFCDIP00 step (15), the DISP parameter of the DD statement for SYS1.LOGREC or SYS1.ASRLIB must be changed to DISP=(OLD,KEEP), or SYS1.LOGREC and SYS1.ASRLIB must be scratched.

The ASMBLR, IEWL, and IEBCOPY steps must be executed without error before IEHIOSUP step (14) is executed.

#### GUIDELINES FOR RESTARTING STAGE II (TSO GENERATION)

The following sections give guidelines for restarting during Stage II of a TSO generation. Restarting may require the scratching and reallocation of data sets. When this is necessary, reference is made to the section "Reallocating Data Sets" for the procedure to be followed. After these operations have been performed, the actual restarting of Stage II can be accomplished by one of the methods described in "Restart Techniques." The numbered steps in this section refer to steps in Figure 28.

Note: If you have specified a qualifier in the INDEX parameter of the GENTSO macro instruction and errors have occurred before the two IEHPROGM rename steps (8) and (22), you should check to be sure that these rename steps have not been executed. If a library has been renamed with the qualifier SYS1 and it is necessary to restart a step that refers to that library, the IEHPROGM utility program can be used to rename the library with the qualifier you supplied.

#### Restarting During Assemblies

To restart at the beginning of any assembly, place the JOB card in front of the EXEC card of that assembly or use the IEEDIT program. If more than one ASMBLR step has been executed, and if you want to restart at the beginning of Stage II (SG1 step), the data set defined by the OBJPDS DD statement must be reallocated (see "Reallocating Data Sets").

All assemblies must be satisfactorily executed before the link-edit (IEWL) steps can be executed.

#### Restarting Any IEHPROGM Step

To restart, place the JOB card in front of the EXEC card of the step, or use the IEEDIT program if the job stream is on tape.

The ASMBLR, IEWL, IEBCOPY, IEHIOSUP, and IFCDIP00 steps must be executed without error before IEHPROGM step (22) (the rename IEHPROGM step) is executed.

#### Restarting Link-Edit Steps

IEWL steps 3 produce load modules for the new SYS1.SVCLIB, SYS1.LINKLIB, and SYS1.CMDLIB. IEWL step (11) builds the new SYS1.NUCLEUS. Subsequent IEWL steps (12) and (14) produce load modules for the new SYS1.SVCLIB, SYS1.LINKLIB, and, if specified for the new system, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.SORTLIB, SYS1.IMAGELIB, and SYS1.TELCMLIB.



If IEWL step 11 was executed, SYS1.NUCLEUS must be reallocated (see "Reallocating Data Sets") before restarting the step.

To restart at the beginning of IEWL step 3, any of the following libraries that contain system data must be reallocated: SYS1.SVCLIB, SYS1.IMAGELIB, SYS1.LINKLIB, SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.SORTLIB, SYS1.CMDLIB, SYS1.NUCLEUS, and SYS1.TELCMLIB.

If reallocation is not performed, the space already used in the new system libraries is not available, and therefore there may not be enough space available for these data sets. This is particularly important for those system data sets that cannot have a secondary allocation. If more than sufficient space was allocated to a data set during the preparation for system generation, the IEWL step that failed can be restarted without reallocation.

It is recommended that the reallocations indicated above be performed when restarting from the beginning of IEWL step 3. To restart from the beginning of Stage II, the data sets defined by the OBJPDS, SYSUT1, and SYSUT2 DD statements must also be reallocated.

During an I/O generation, no reallocation is needed to restart a link-edit step, unless a step failed because there was not enough space in a system data set. If there was not enough space, move the contents of the old library to the new library. Copy the data set to a scratch volume using the IEBCOPY utility program. Use IEHPROGM to scratch and uncatalog the space for the data set on its original volume. Allocate and catalog the required space for the data set on its original volume and use IEBCOPY to copy the data set from the scratch volume to its original volume. Then restart Stage II at the beginning of the link-edit step that failed.

#### Restarting IEBCOPY Step

This IEBCOPY step 7 copies SYS1.MACLIB, SYS1.TSOMAC, and SYS1.TCAMMAC to the new system. To restart at the beginning of this step, the new SYS1.MACLIB must be reallocated (see "Reallocating Data Sets").

#### Restarting All Other IEBCOPY Steps

IEBCOPY step 4 copies load modules to the new SYS1.SVCLIB, SYS1.LINKLIB, SYS1.UADS, SYS1.CMDLIB, and SYS1.HELP. No reallocation is needed to restart this step, unless the step failed because there was not enough space in a system data set. Only the system data set that did not have enough space must be reallocated (see "Reallocating Data Sets"), because any member that existed in the new system data set will not be moved into it again when the step is restarted.

If the data set that did not have enough space was SYS1.SVCLIB, SYS1.LINKLIB, SYS1.UADS, SYS1.CMDLIB, or SYS1.HELP, one of the two following procedures must be performed:

- Follow the procedures for restarting with IEWL step 3 as described in "Restarting Link Edit Steps" in the "Guidelines for Restarting Stage II (TSO Generation)" section.
- Copy the data set to a scratch volume using the IEBCOPY utility program. Scratch and uncatalog the space for the data set on its original volume with IEHPROGM. Allocate and catalog the required space for the data set on its original volume. Use IEBCOPY to copy the data set from the scratch volume to its original volume. Restart Stage II at the beginning of IEBCOPY step 13.

IEBCOPY step 13 copies members to the new SYS1.SVCLIB, SYS1.LINKLIB, SYS1.FORTLIB, SYS1.COBLIB, SYS1.PL1LIB, SYS1.TELCMLIB, and SYS1.IMAGELIB.

IEBCOPY step 15 copies members to SYS1.SVCLIB, SYS1.FORTLIB, SYS1.COBLIB, SYS1.ALGLIB, SYS1.PL1LIB, SYS1.LINKLIB, SYS1.PROCLIB, SYS1.SORTLIB, and SYS1.PARMLIB. The procedures to restart IEBCOPY step 4 also apply to this step.

IEBCOPY step 16 copies members to the new SYS1.SVCLIB or SYS1.LINKLIB. The procedures to restart IEBCOPY step 4 also apply to this step.

#### Restarting IEHIOSUP, IFCDIP00, and IEHLIST

The system data sets do not have to be reallocated when the IEHIOSUP step 18 and IEHLIST step 23 are restarted. Before the IFCDIP00 step 19 is restarted, the DISP parameter of the DD statement for SYS1.LOGREC or SYS1.ASRLIB must be changed to DISP=(OLD, KEEP), or SYS1.LOGREC and SYS1.ASRLIB must be scratched.

The ASMBLR, IEWL, and IEBCOPY steps must be executed without error before the IEHIOSUP step 18 is executed.

#### REALLOCATING DATA SETS

Reallocating data sets includes the following operations:

- Uncataloging and scratching the space allocated to the data set during the preparation for system generation.
- Allocating new space to the data set and cataloging it.

The following sections discuss the reallocation of the utility data sets defined during Stage I, reallocating new system data sets using the same amount of space, and reallocating system data sets using more space.

The reallocations described in this section may alter the order in which the new system data sets were originally allocated. If this order is important, the IEBCOPY utility program can be used after system generation to rearrange the new system data sets.

#### Reallocation of OBJPDS

Use the following statements to reallocate the utility data set specified by the OBJPDS DD statement during Stage I. These statements are coded when more than one assembler step is executed and the generation process is to be restarted from the beginning.

---

```
//OBJPDS    JOB
//STEP1    EXEC    PGM=IEHPROGM -SCRATCH OBJPDS-
//OBJPDS    DD      DSNAME=SYS1.name,DISP=(OLD,DELETE)
//SYSPRINT  DD      SYSOUT=A
//SYSIN     DD      DUMMY
//STEP2    EXEC    PGM=IEHPROGM -REALLOCATE OBJPDS-
//OBJPDS    DD      (Parameters for OBJPDS in Stage I input deck)
//SYSPRINT  DD      SYSOUT=A
//SYSIN     DD      DUMMY
//
```

---

Figure 36. Control Statements to Reallocate the Utility Data Set for OBJPDS

In the preceding statements:

- The DSNNAME parameter of the first OBJPDS DD statement must contain the name given to the utility data set in the input deck to Stage I.
- The second OBJPDS DD statement must contain the same parameters that it contains in the Stage I input deck.

#### Reallocation of SYSUT1, SYSUT2, SYSUT3, and SYSUT4

The operations to be performed depend on whether UTDISP=UNCATLG or UTDISP=DELETE was specified in the GENERATE macro instruction.

Use the following statements to recatalog the utility data sets when UTDISP=UNCATLG was specified:

---

```
//CATLG      JOB          ~UTDISP=UNCATLG~
//STEP       EXEC      PGM=IEHPROGM
//SYSPRINT   DD        SYSOUT=A
//SYSIN      DD        *
              CATLG     DSNNAME=SYS1.name,VOL=unit=serial
/*
```

---

Figure 37. Control Statements to Recatalog the Utility Data Set When UTDISP=UNCATLG

In the preceding statements:

- Use one CATLG statement for each utility data set to be recataloged.
- The DSNNAME parameter must contain the name given to the utility data set in the input deck to Stage I.
- The unit and serial number in the VOL parameter must correspond to those specified for the utility data set in the input deck to Stage I.

Use the following DD statements to allocate space and recatalog the utility data sets when UTDISP=DELETE was specified:

---

```
//ALLOC      JOB          -UTDISP=DELETE-
//STEP       EXEC      PGM=IEHPROGM
//SYSPRINT   DD        SYSOUT=A
//SYSUT1     DD        (Parameters for SYSUT1 in Stage I input deck)
//SYSUT2     DD        (Parameters for SYSUT2 in Stage I input deck)
//SYSUT3     DD        (Parameters for SYSUT3 in Stage I input deck)
//SYSUT4     DD        (Parameters for SYSUT4 in Stage I input deck)
//SYSIN      DD        DUMMY
//
```

---

Figure 38. Reallocate and Recatalog the Utility Data Sets When UTDISP=DELETE

In the preceding statements:

- Omit the SYSUT1 and SYSUT2 DD statements if only the data sets defined by the SYSUT3 and SYSUT4 DD statements in Stage I are to be reallocated.
- Omit the SYSUT3 DD statement if only the data sets defined by the SYSUT1 and SYSUT2 DD statements in Stage I are to be reallocated.

- The SYSUT1, SYSUT2, SYSUT3, and SYSUT4 DD statements must contain the same parameters that they contain in the Stage I input deck.

### Reallocating the Same Space

The input deck for scratching and reallocating space to the new system data sets must contain the following statements. The statements must be in the order shown.

1. JOB statement.
2. EXEC statement with the PGM=IEHPROGM parameter.
3. SYSPRINT DD statement defining the system output unit.
4. DD statement defining the unit and serial number of the generating system-residence volume:  

```
//GENRES DD UNIT=unit,VOLUME=SER=serial,DISP=OLD
```
5. DD statement defining any other permanent volume on which data sets to be reallocated reside:  

```
//ddname DD UNIT=unit,VOLUME=SER=serial,DISP=OLD
```
6. DD statement for each type of removable device on which data sets to be reallocated reside:  

```
//ddname DD UNIT=(unit,,DEFER),VOLUME=PRIVATE,DISP=OLD
```
7. DD \* statement (SYSIN).
8. SCRATCH and an UNCATLG statement for each new system data set to be reallocated. The SCRATCH and UNCATLG statements must have the following format:  

```
SCRATCH   DSNAME=dsname,VOL=device=serial,PURGE
UNCATLG   DSNAME=dsname,CVOL=device=serial
```

where CVOL designates the new system-residence volume.
9. /\* statement.
10. EXEC statement with the PGM=IEHPROGM parameter.
11. GENRES DD statement (described above).
12. DD statement for each permanent device (described above).
13. DD statement for each type of removable device (described above).
14. SYSPRINT DD statement defining the system output unit.
15. DD statement for each of the new system data sets to be reallocated. This DD statement must be the same used in the input deck for the preparation for system generation.
16. DD \* statement (SYSIN).

17. CATLG statement for each new system data set to be reallocated. The CATLG statement must be the same used in the input deck for the preparation for system generation.

18. /\* statement.

For example, SYS1.SVCLIB and SYS1.LINKLIB must be reallocated. Figure 39 shows the DD statements that define these two libraries during the preparation for system generation:

---

```
//SVCLIB DD DSN=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=AAA111), X
// UNIT=2301,DISP=(,KEEP),SPACE=(TRK,(40,10,75)), X
// LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//LINKLIB DD DSN=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=AAA112), X
// UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(125,10,100)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
```

---

Figure 39. Example of Reallocation for SYS1.SVCLIB and SYS1.LINKLIB

The generating system-residence volume is a 2311 volume whose serial number is 111111. Figure 40 shows the input deck for this reallocation.

---

```
//SAME JOB -REALLOCATE ON SAME SPACE-
//STEP1 EXEC PGM=IEHPROGM -SCRATCH-
//GENRES DD UNIT=2311,VOLUME=SER=111111,DISP=OLD
//NEWRES DD UNIT=2301,VOLUME=SER=AAA111,DISP=OLD
//LINVOL DD UNIT=(2311,,DEFER),VOLUME=PRIVATE,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
SCRATCH DSN=SYS1.SVCLIB,VOL=2301=AAA111
UNCATLG DSN=SYS1.SVCLIB,CVOL=2301=AAA111
SCRATCH DSN=SYS1.LINKLIB,VOL=2311=AAA112
UNCATLG DSN=SYS1.LINKLIB,CVOL=2301=AAA111

/*
//STEP2 EXEC PGM=IEHPROGM -ALLOCATE-
//GENRES DD UNIT=2311,VOLUME=SER=111111,DISP=OLD
//NEWRES DD UNIT=2301,VOLUME=SER=AAA111,DISP=OLD
//LINVOL DD UNIT=(2311,,DEFER),VOLUME=PRIVATE,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SVCLIB DD DSN=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=AAA111), X
// UNIT=2301,DISP=(,KEEP),SPACE=(TRK,(40,10,75)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024,DSORG=POU)
//LINKLIB DD DSN=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=AAA112), X
// UNIT=2311,DISP=(,KEEP), X
// SPACE=(CYL,(125,10,100)),LABEL=EXPDT=99350, X
// DCB=(RECFM=U,BLKSIZE=3625)
//SYSIN DD *
CATLG DSN=SYS1.SVCLIB,CVOL=2301=AAA111,VOL=2301=AAA111
CATLG DSN=SYS1.LINKLIB,CVOL=2301=AAA111,VOL=2311=AAA112

/*
//
```

---

Figure 40. Reallocating on Same Space

Reallocating With More Space

The method for reallocating with more space depends on whether the space for the data set must be continuous. If the space need not be continuous, follow the

procedure described in "Reallocating on the Same Space," changing the SPACE parameter of the DD statement (statement 15) for the new system data set in the second step. This same procedure can be followed for a continuous data set if there is enough continuous space for the new allocation elsewhere on the volume.

The second case can be illustrated with the following example. One of the new system volumes is organized as shown in Figure 41.

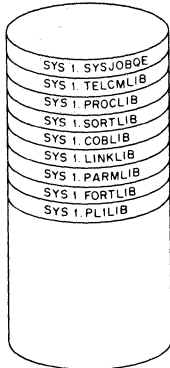


Figure 41. Reallocation on Same Volume

Not enough space was allocated for SYS1.PARMLIB, and there is not enough continuous space after SYS1.PL1LIB for a new allocation of SYS1.PARMLIB. One of two procedures can be followed to free space for the new allocation for SYS1.PARMLIB.

- If there is no system data in SYS1.PARMLIB, SYS1.FORTLIB, and SYS1.PL1LIB, follow the procedures described in "Reallocating in the Same Space" to reallocate SYS1.PARMLIB, SYS1.FORTLIB, and SYS1.PL1LIB. Correct the SPACE parameter in the DD statement for SYS1.PARMLIB.
- If there is system data in SYS1.PARMLIB, SYS1.FORTLIB, or SYS1.PL1LIB follow these procedures:

Copy the library that contains system data onto a scratch volume by using the IEBCOPY utility program.

Follow the procedures described in "Reallocating in the Same Space" to scratch SYS1.PARMLIB, SYS1.FORTLIB, and SYS1.PL1LIB, correcting the SPACE parameter in the DD statement for SYS1.PARMLIB.

Copy the library that contains system data from the scratch volume to its original volume.

If a system data set (continuous or not) contains system data, and if there is enough space elsewhere in the volume, the following procedure can be used:

1. Rename the system data set.
2. Allocate and catalog space for the system data set (with its correct name) on the same volume.
3. Copy the data in the renamed data set onto the newly allocated data set.
4. Scratch and uncatalog the renamed data set.

Figure 42 is an example of reallocation on the same volume of a system data set that contains system data. The system data set to be reallocated is SYS1.PARMLIB. It was allocated space during the preparation for system generation with the following DD statement:

```
//PARMLIB DD DSN=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=2301,DISP=(,KEEP),SPACE=(TRK,(1,,1),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
```

The new system-residence volume is a 2301 volume whose serial number is SYSTEM. The renamed SYS1.PARMLIB will be called SYS1.HARRY. The generating system-residence volume is a 2311 volume whose serial number is 111111.

---

```
//MOVE JOB -REALLOCATE WITH DATA-
//STEP1 EXEC PGM=IEHPROGM -RENAME-
//SYSPRINT DD SYSOUT=A
//GENRES DD UNIT=2311,VOLUME=SER=111111,DISP=OLD
//NEWRES DD UNIT=2301,VOLUME=SER=SYSTEM,DISP=OLD
//SYSIN DD *
RENAME DSN=SYS1.PARMLIB,VOL=2301=SYSTEM, X
NEWNAME=SYS1.HARRY
UNCATLG DSN=SYS1.PARMLIB,CVOL=2301=SYSTEM
CATLG DSN=SYS1.HARRY,CVOL=2301=SYSTEM,VOL=2301=SYSTEM
/*
//STEP2 EXEC PGM=IEHPROGM -REALLOCATE-
//SYSPRINT DD SYSOUT=A
//GENRES DD UNIT=2311,VOLUME=SER=111111,DISP=OLD
//NEWRES DD UNIT=2301,VOLUME=SER=SYSTEM,DISP=OLD
//PARMLIB DD DSN=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=2301,DISP=(,KEEP), X
// SPACE=(TRK,(5,,1),,CONTIG),LABEL=EXPDT=99350, X
// DCB=(RECFM=F,BLKSIZE=80)
//SYSIN DD *
CATLG DSN=SYS1.PARMLIB,CVOL=2301=SYSTEM,VOL=2301=SYSTEM
/*
//STEP3 EXEC PGM=IEBCOPY -COPY-
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.HARRY,VOLUME=SER=SYSTEM,UNIT=2301, X
// DCB=(RECFM=F,BLKSIZE=80)
//SYSUT2 DD DSN=SYS1.PARMLIB,VOLUME=SER=SYSTEM,UNIT=2301, X
// DCB=(RECFM=F,BLKSIZE=80)
//SYSIN DD DUMMY
//STEP4 EXEC PGM=IEHPROGM -SCRATCH-
//SYSPRINT DD SYSOUT=A
//GENRES DD UNIT=2311,VOLUME=SER=111111,DISP=OLD
//NEWRES DD UNIT=2301,VOLUME=SER=SYSTEM,DISP=OLD
//SYSIN DD *
SCRATCH DSN=SYS1.HARRY,VOL=2301=SYSTEM
UNCATLG DSN=SYS1.HARRY,VOL=2301=SYSTEM
/*
//
```

---

Figure 42. Reallocate Data Sets With System Data

## SYS1.SYSJOBQE Considerations

If the job stream is too large for the SYS1.SYSJOBQE of the generating system, one of the following procedures can be followed:

- Convert the job stream into two or more jobs
- Reformat the job queue
- Enlarge SYS1.SYSJOBQE

The job stream can be converted into two or more jobs by placing additional JOB cards throughout the job stream. For example, you might insert a JOB card just before the first IEWL step. This method requires that the job stream be in cards.

A description of the values to specify when reformatting can be found in the "MVT Job Queue Formatting" section of the OS System Programmer's Guide.

The size of SYS1.SYSJOBQE can be changed following this procedure:

1. Determine the appropriate size of SYS1.SYSJOBQE using the formula given in OS Storage Estimates.
2. Run the job shown in Figure 43.

---

```
//LASTJOB JOB
//STEP EXEC PGM=IEHPROGM
//DUMMY DD DSNAME=SYS1.DUMMY,VOLUME=(,RETAIN,SER=serial), X
// UNIT=unit,DISP=(,CATLG),SPACE=(TRK,(xxx),,CONTIG)
//GENRES DD UNIT=unit,VOLUME=SER=serial,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
SCRATCH DSNAME=SYS1.SYSJOBQE,VOL=unit=serial
UNCATLG DSNAME=SYS1.SYSJOBQE
RENAME DSNAME=SYS1.DUMMY,VOL=unit=serial, X
NEWNAME=SYS1.SYSJOBQE
CATLG DSNAME=SYS1.SYSJOBQE,VOL=unit=serial
/*
//
```

---

Figure 43. Control Statements to Change SYS1.SYSJOBQE

In the preceding statements,

- SYS1.DUMMY is allocated space on the same volume on which the generating SYS1.SYSJOBQE resides. The amount of space allocated for SYS1.DUMMY must be the appropriate size for SYS1.SYSJOBQE.
  - The GENRES DD statement defines the generating system-residence volume.
  - The UNCATLG and CATLG statements should be omitted if SYS1.SYSJOBQE is not cataloged in the generating system.
  - The VOL parameter of the utility statements defines the volume on which SYS1.SYSJOBQE resides.
3. Stop the system and restart (IPL again).



## Testing the New System

This section describes the sample programs provided by IBM to test the functioning of various components of the new system after system generation.

A description of the sample program, its operating instructions, and a list of the output results are provided.



The sample programs are contained in SYS1.SAMPLIB of the starter operating system package. The "System Generation Using the Starter Operating System" section describes the procedure used to punch sample program card decks from SYS1.SAMPLIB. These decks can then be used whenever an operating system component is to be tested.

The following list shows the names of the sample programs provided in SYS1.SAMPLIB and the components they test:

<u>Program</u>	<u>Component</u>
IEXSAMP	ALGOL
IKFSAMP	American National Standard COBOL
IEUESP	Assembler F
IEPSAMP	COBOL E
IEBDATGN	Data Generator Sample Program
IKDSAMPL	Data transmission and conversion subroutines for FORTRAN IV
IEYSP	FORTRAN G and H
SAMP2250	Graphics - 2250 Display Units
SAMP2260	Graphics - 2260 Display Stations
GSPSAMP	Graphic Subroutine Package for FORTRAN IV
COBSAMP	Graphic Subroutine Package for COBOL
PL1SAMP	Graphic Subroutine Package for PL/1
DRISAMP	IEHDASDR system utility program Loader Program Test Program
IEMSP2	PL/I F
RPGSMPL	RPG
SAMACTRT	Sample Account Routine
IERSP	Sort/Merge System Management Facilities Sample Programs (see below)
IHGSAMP	Update Analysis Program
USERLABL	User Label Sample Program
IBCDASDI	Direct Access Initialization
IBCRCVRP	Direct Access Recover/Replace
IBCDMPRS	Direct Access Dump Restore

DASDI, DUMPREST, and RECOVER (alias for IBCDASDI, IBCDMPRS, IBCRCVRP respectively) are included in SYS1.SAMPLIB. These programs are identical to IBCDASDI, IBCDMPRS, and IBCRCVRP. Either DASDI or IBCDASDI, either DUMPREST or IBCDMPRS, or either RECOVER or IBCRCVRP can be specified in a PUNCH statement.

The following pages contain descriptions of the sample programs, operating instructions, and a description of the program execution results, except for the system management facilities (SMF) sample programs. The documentation for the SMF sample programs is in OS SMF. More detailed operating instructions for all the sample programs except the SMF sample programs are in OS Operator's Reference.

**Note:** The DD statements of the sample programs specify UNIT=2311. If the system data sets affected by the sample program do not reside on a 2311, the UNIT parameter should be changed accordingly. If the system data set is cataloged, the UNIT parameter of the corresponding DD statement can be deleted rather than changed.

## ALGOL Sample Program (IEXSAMP)

The IEXSAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements for an ALGOL compilation, a link-edit, and execution.
2. ALGOL sample program source statements.

The sample program generates the first twenty lines of Pascal's Triangle. Comments included in the program may be used for checking the results. (The sample program is included as an example in OS ALGOL Language.)

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the IEXSAMP card deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

The normal output from the compiler is:

1. A list of all job control language statements that apply to the compiler.
2. A list of all source statements supplemented by a count of the semicolons occurring in the program.
3. The identifier table in symbolic form, giving details of all identifiers used in the program.
4. Information on main-storage requirements at execution time.

The normal output from the linkage editor will be:

1. A list of all job control language statements that apply to the linkage editor.
2. A cross-reference table of the load module.

The normal output at execution time will be:

1. A list of all job control language statements that apply to the execution.
2. The first twenty lines of Pascal's Triangle, corresponding to the comments included in the source statements.

A more detailed description of the system output is contained in OS ALGOL Programmer's Guide.

## American National Standard Cobol, X3.23-1968 Sample Program (IKFSAMP)

The IKFSAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements to call a cataloged procedure to compile, link-edit, and execute the sample program.
2. COBOL sample program source statements.

The sample program tests the operation of the IBM American National Standard COBOL compiler; it generates an output data set on tape and then reads, processes, and exhibits this data set. The program uses the COBOL verbs IF, OPEN, READ, WRITE, CLOSE, DISPLAY, MOVE, ADD, STOP, GO TO, PERFORM, NOTE, TRACE, and EXHIBIT. A complete output listing of the program is in the publication, OS American National Standard COBOL Programmer's Guide.

### OPERATING INSTRUCTIONS

1. Mount the operating system.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the IKFSAMP deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

1. Listing of all job control language statements
2. Listing of the source program
3. Listing of:
  - a. Heading including date and level
  - b. Source program
  - c. Data map of the Data Division
4. Module map and cross-reference list

## Assembler F Sample Program (IEUESP)

The IEUESP card deck (punched from SYS1.SAMPLIB) is the sample program source deck.

The sample problem demonstrates the use of the assembler and serves as a minimal test of the functioning of the assembler. In addition, it provides sample coding that demonstrates the definition and use of user-written macro instructions, the calling of system macro instructions, and the proper method of saving and restoring registers upon entry to and exit from a problem program.

The input is assembled into the program in the form of a table (TABLE) and a list (LIST) of entries that are to be compared with the table entries. Each item in the table contains an argument name such as ALPHA and space in which information concerning that name is to be placed. Each entry in the LIST contains an argument name and function values. The formats of the TABLE entries and the LIST entries are different, and both formats are described by means of DSECTs. The program searches the TABLE for an argument name in the list. If a match is found, the function values are reformatted and moved to the appropriate TABLE entry. If an argument name in the LIST cannot be found in the TABLE, a switch is set in the LIST entry. After all LIST entries have been processed, both the LIST and TABLE areas are compared with TESTTABL, which contains the predefined results. If the two are equal, the routine is executed properly and a message is written to indicate this.

The program for the assembler may be called by the IBM-supplied cataloged procedure ASMFCLG in the following manner:

```
//jobname JOB
//stepname EXEC PROC=ASMFCLG
//ASM.SYSIN DD *
    Sample program source deck
/*
```

This procedure calls for an assembly, link-edit, and execution of the sample program. A more detailed explanation of the contents of the procedure and use of the assembly program is given in OS Assembler (F) Programmer's Guide.

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the sample program source deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

The check for the successful execution of the sample program is a message on the operator's console. The message says either "ASSEMBLER SAMPLE PROGRAM SUCCESSFUL" or "ASSEMBLER SAMPLE PROGRAM UNSUCCESSFUL." The program gives a normal return to the operating system with a return code of 0.

## COBOL E Sample Program (IEPSAMP)

The IEPSAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements to execute the COBOL E cataloged procedure COBECLG to compile, link-edit, and execute.
2. COBOL sample program source statements.
3. DD statements for tape data sets required at execution time.

The sample program tests the COBOL compiler's ability to write to and read from tape, checking proper data alignment where the data is a mixture of DISPLAY COMPUTATIONAL and COMPUTATIONAL-3.

The verbs used in this program are OPEN, CLOSE, READ, WRITE, PERFORM, IF, MOVE, GO TO, DISPLAY, and ADD. There are 575 source statements.

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Mount unlabeled tapes on 182, 183, and 282.
3. Set the load address switches and press the Initial Program Load key to load the operating system.
4. Place the IEPSAMP card deck in the card reader, ready the reader, and press the End-of-File key.
5. Ready the printer.
6. Execute the job.

### OUTPUT

1. The scheduler reads, processes, and prints (on the device specified as SYSOUT by the operator) all job control language statements.
2. The compiler reads and lists the source program on the device specified as SYSOUT by the operator.
3. The compiler prepares and lists the following (on SYSOUT):
  - a. Heading including date and level
  - b. The source program
  - c. A data map of the Data Division
  - d. An object code listing (PMAP) of the Procedure Division
4. The linkage editor prepares and lists the following (on SYSOUT):
  - a. A module map
  - b. A cross-reference list
5. The COBOL load module lists the following (on SYSOUT):

```
GROUP B LEVEL P TEST CASE 1
END OF PROGRAM
```

## Data Generator Sample Program (IEBDATGN)

The IEBDATGN card deck, punched from SYS1.SAMPLIB, consists of:

1. Job control language statements for the execution of the data generator system utility program
2. Comment statements describing the control cards for the system utility program
3. The sample program utility statements

The sample program generates patterns of data on the printer illustrating the generating capabilities of the data generator system utility program and the various ways of altering this data.

### OPERATING PROCEDURES

1. Mount the operating system.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the IEBDATGN card deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

1. A list of all job control language statements that apply to the execution.
2. The comment statements explaining the function performed by the system utility control cards.
3. The data generator sample program utility statements.
4. The generated output showing the results of the action parameter.
5. A second group of utility statements.
6. The generated output showing the results of the picture and index parameter of the data generator system utility program.



## Data Transmission and Conversion Subroutines for FORTRAN IV Sample Program (IKDSAMPL)

The System/360 portion of the IKDSAMPL sample program for data transmission and conversion subroutines can be punched from SYS1.SAMPLIB. The IKDSAMPL card deck consists of:

1. The following job control language statements, which call the FORTGCLG cataloged procedure to compile, link-edit, and execute the program when the FORTRAN G compiler is used in the operating system.

```
//PTOPSAMP JOB 360T1130,SAMPLE,MSGLEVEL=1
//PTOPSTEP EXEC FORTGCLG,PARM.FORT=(NODECK,MAP,LOAD)
//FORT.SYSIN DD *
```

If the FORTRAN H compiler is to be used, the FORTGCLG parameter of the EXEC statement must be changed to FORTHCLG.

2. A sample program input symbolic deck. (The contents of this deck are shown in IBM System/360 Operating System and 1130 Disk Monitor System: System/360-1130 Data Transmission for FORTRAN.)
3. The following cards after the FORTRAN language END card:

```
/*
//LKED.SYSLMOD DD SPACE=(1024,(50,20,1),RLSE)
/*
//GO.SYSABEND DD SYSOUT=A
//GO.FT09F001 DD UNIT=XXX
//GO.SYSIN DD *
AXLEP1 (This is the data card that identifies the password)
/*
```

The UNIT parameter of the GO.FT09F001 DD statement must be changed to specify the appropriate address or symbolic name previously assigned to the 1130 system with which this sample program is to communicate. You must then substitute this newly punched card for the GO.FT09F001 DD card in the existing card deck.

### OPERATING INSTRUCTIONS

To process the PTOP sample program card deck, the following steps are necessary:

1. Ensure that the 1130 portion of the sample program for the data transmission and conversion subroutines (program number 1130-LM-011) exists in an associated 1130 system, as described in IBM System/360 Operating System and 1130 Disk Monitor System: System/360-1130 Data Transmission for FORTRAN.
2. Place the PTOPSAMP card deck in the card reader.
3. Mount the operating system.
4. Set the load address switches and press the Initial Program Load key to load the system.

### OUTPUT

The 1130 program prints out the results of the computations (the weight distributions on the front and rear axles of a truck). From these results, the optimum dimensions for the front and rear axles of a truck can be determined.

## **FORTRAN G and H Sample Program (IEYSP)**

The IEYSP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements for a FORTRAN G or H compilation.
2. FORTRAN sample program source statements.
3. Job control language statements for link-edit and execution of the sample program.
4. A data card for FORTRAN sample program execution.

The sample program consists of one main program and one function subprogram, which together compute and print out binomial coefficients. The deck is composed of the following source statements:

	<u>Main Program</u>	<u>Subprogram</u>
Comments	42	19
Specification	1	1
Format	5	
Read/Write	5	
Control	8	13
Arithmetic	2	13
Total	63	46

The program processes one data card as its input. The comment cards show a complete list of the expected output from the execution of the program and may be used for checking the output.

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the IEYSP card deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

1. The scheduler will read, process, and print (on the device specified as SYSOUT by the operator) all job control statements.
2. The compiler will read and list the source program on the device specified as SYSOUT by the operator.
3. The compiler will prepare and list the following (on SYSOUT):
  - a. Heading, including date and level
  - b. The source program
  - c. Storage map including size of COMMON and size of program
  - d. The generated object code
4. The linkage editor will prepare and list (on SYSOUT) a storage map including the relative address of each external reference.
5. The FORTRAN load module will list (on SYSOUT) the results of execution. The results should correspond to the comments in the source program.

## Graphics Sample Programs (SAMP2250 and SAMP2260)

The SAMP2250 card deck (sample program to exercise the 2250 display unit) and the SAMP2260 card deck (sample program to exercise the 2260 display station, local) can be punched from SYS1.SAMPLIB.

Each card deck consists of:

1. Appropriate job control language statements. Included is a cataloged procedure (ASMFCLG) to assemble, link-edit, and execute the compiled program. Before compiling either deck, the four override cards below must be added to the end of the deck after the assembler language END card.

```
/*  
//LKED.SYSLIB DD DSNAME=SYS0.LINKLIB,DISP=OLD  
//GO.GRAPHIC DD UNIT=2250-1  
OR  
//GO.GRAPHIC DD UNIT=(2260-1,2)  
/*
```

In the preceding statements, the underlined values must be replaced with the device type of the 2250 or 2260 being tested. (See Appendix A.) Note that the GO.GRAPHIC DD statement for the 2260 specifies two devices. (For further information about specifying more than one device, see OS Job Control Language Reference.)

2. Sample program input symbolic deck.

### OPERATING INSTRUCTIONS

1. Ensure that the graphic device is "on."
2. Place the sample program deck in the card reader.
3. Mount the operating system.
4. Set the load address switches and press the Initial Program Load key to load the system.
5. Perform the assemble, link-edit, and go procedure.
6. Follow the instructions that appear on the display screen.

### OUTPUT

The displays that appear on the 2250 display unit are shown in Figure 44. The displays that appear on the 2260 display station are shown in Figure 45.

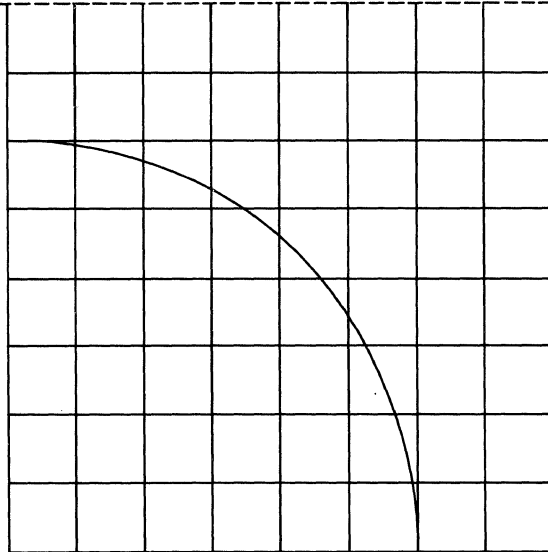
2250 SAMPLE PROGRAM - END ATTENTION TO BEGIN

a. First Display

THIS PROGRAM WILL PLOT A GRID WITH CORNERS AT 2400,2400-2400,4000-4000,4000-4000,2400. IT WILL PLOT 90 DEGREES OF ARC WHOSE RADIUS IS 1200 RASTER UNITS. IT ILLUSTRATES THE USE OF ORDER MACROS TO CREATE THIS DISPLAY AND PORS TO CREATE THE NEXT DISPLAY.

DEPRESS KEY D AND THEN THE END KEY ON THE ALPHAMERIC KEYBOARD TO INITIATE THE NEXT DISPLAY.

b. Second Display



THIS DISPLAY ILLUSTRATES THE OUTPUT OF THE GCGRID, GARC, AND GCPRNT PROBLEM ORIENTED ROUTINES.

DEPRESS KEY E AND THEN THE END KEY ON THE ALPHAMERIC KEYBOARD TO TERMINATE THIS SAMPLE PROGRAM.

c. Third Display

Figure 44. 2250 Displays

---

DISPLAY STATION OPEN - INSERT DS MI REQUEST

a. First Display

PROCESSING COMPLETE - INSERT NEXT MI REQUEST

b. Second Display

END OF JOB

c. Third Display

---

Figure 45. 2260 Displays

## Graphic Subroutine Package for FORTRAN, COBOL, and PL/I Sample Program (GSPSAMP, COBSAMP, and PLISAMP)

There are two sample programs for the Graphic Subroutine Package: GSPSAMP and PL1SAMP written in FORTRAN IV and PL/I, respectively. Both programs produce the same displays and perform the same functions.

The GSPSAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. The following job control language statements that call the FORTGCLG cataloged procedure to compile, link-edit, and execute the program when the FORTRAN G compiler is used in the operating system:

```
//GSPSAMY      JOB      1234567,GSPSAMP,MSGLEVEL=1
//SAMPL       EXEC     FORTGCLG,PARM.FORT=(NODECK,MAP,LOAD)
//PORT.SYSIN  DD       *
```

If the FORTRAN H compiler is to be used, the FORTGCLG parameter of the EXEC statement must be changed to FORTHCLG.

2. A sample program input symbolic deck.
3. The following cards after the FORTRAN-language END card:

```
/*
//LKED.SYSIN  DD       *
  INCLUDE     SYSLIB(IHCGSP03)
/*
//GO.SYSABEND DD       SYSOUT=A
//GO.FT10F001 DD      UNIT=(2250-1)
//GO.SYSIN    DD       *
X
NUM1NUM2NUM3NUM4NUM5NUM6NUM7NUM8
/*
```

The PL1SAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Appropriate job control language statements that call a cataloged procedure (PL1LFCLG) to compile, link-edit, and execute the program. The following cards are required after the PL/I language END card:

```
/*
//LKED.SYSIN  DD       *
  INCLUDE     SYSLIB(IHEGSP01)
  INCLUDE     SYSLIB(IHEGSP02)
  INCLUDE     SYSLIB(IHEGSP03)
/*
//GO.PL1DUMP  DD       SYSOUT=A
//GO.SYSUDUMP DD       SYSOUT=A
//GO.FT10F001 DD      UNIT=xxx (where xxx is the 2250 unit
//                                     address)
/*
```

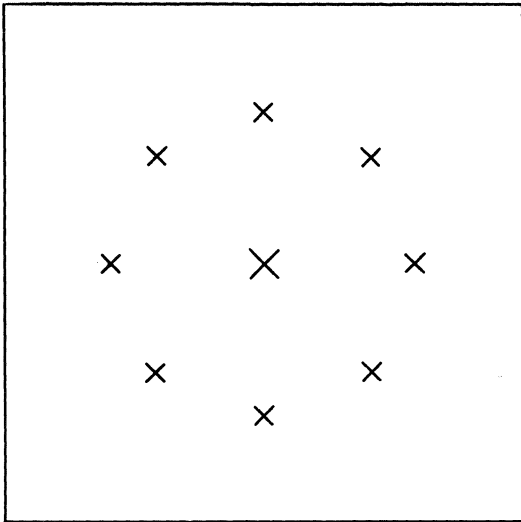
2. A sample program input symbolic deck.

## OPERATING INSTRUCTIONS

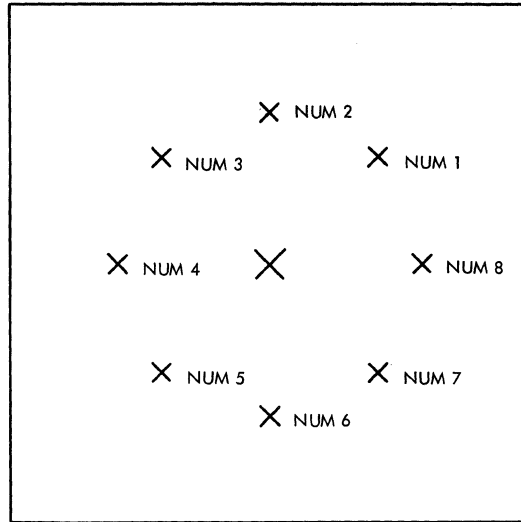
1. Ensure that the graphic device is "on."
2. Place the sample program deck in the card reader.
3. Mount the operating system.
4. Set the load address switches and press the Initial Program Load key to load the system.
5. Perform the compile, link-edit, and go procedure.
6. Follow instructions that appear with the sample program in OS GSP for FORTRAN and PL/I.

## OUTPUT

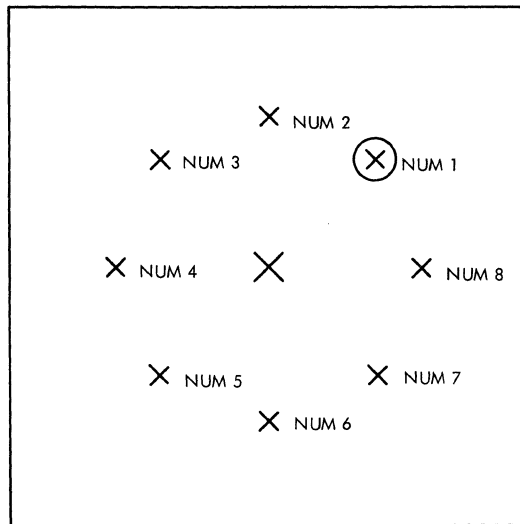
The displays that appear on the 2250 display unit are shown in Figure 46.



A. Display Produces After Call to EXEC for IGDSI.



B. Display Produced After Light Pen Attention on Center X.



C. Display Produced After Light Pen Attention on X Labeled NUM1.

Figure 46. Display for GSP Sample Program



## IEHDASDR Sample Program (DRISAMP)

The DRISAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements for execution of the IEHDASDR system utility program
2. IEHDASDR sample program utility statements

The sample program initializes a direct-access volume chosen by the installation, dumps this volume to a magnetic tape volume also chosen by the installation, and restores from that magnetic tape volume to the dumped direct-access volume.

Two of the DRISAMP DD statements must be changed to indicate the direct-access volume and tape volume chosen by the installation. The statements are:

```
//DADEV DD UNIT=XXXX,VOLUME=SER=YYYYYY,DISP=OLD
//TAPDEV DD UNIT=MMMM,VOLUME=SER=ABCXYZ,DISP=OLD,LABEL=(,NL)
```

where:

XXXX

is the unit name of the direct-access volume (See Appendix B).

YYYYYY

is the serial number of the direct-access volume. The system-residence volume must not be specified.

MMMM

is the unit name of the magnetic tape volume (see Appendix B).

The TAPDEV DD statement defines an unlabeled tape with an external serial number. If desired, this external serial number (ABCXYZ) may be changed.

### OPERATING INSTRUCTIONS

1. Mount the operating system and the desired direct-access volume.
2. Mount a magnetic tape volume (unlabeled).
3. Set the load address switches and press the Initial Program Load key to load the operating system.
4. Place the DRISAMP card deck in the card reader, ready the reader, and press the End-of-File key.
5. Ready the printer.
6. Execute the job.

### OUTPUT

1. A list of all job control language statements that apply to execution.
2. The following appears on SYSOUT:

```
ANALYZE TODD=DADEV,VTOC=00050,EXTENT=00010,PURGE=YES
DUMP FROMDD=DADEV,TODD=TAPDEV
IEH8061 ANALYZE OF DDNAME=DADEV IS COMPLETE. VOLUME SERIAL NO=YYYYYY

RESTORE FROMDD=TAPDEV,TODD=DADEV

IEH8061 DUMP TO DDNAME=TAPDEV IS COMPLETE
IEH8061 RESTORE TO DDNAME=DADEV IS COMPLETE. VOLUME SERIAL NO=YYYYYY

IEH8391 HIGHEST RETURN CODE ENCOUNTERED WAS 00
```

Note: The serial number you indicate in the DADEV DD statement will appear instead of YYYYYY.

## Loader Program Test Program

In order to test the loader program, punch the Assembler sample program, IEUESP, from SYS1.SAMPLIB following the procedure described in the section "System Generation Using the Starter Operating System." For detailed information about the sample program, see the writeup for "Assembler F Sample Program (IEUESP)."

To test the sample program and the loader program, call the IBM-supplied cataloged procedure, ASMFPG, using the following JCL statements.

```
//jobname   JOB
//stepname  EXEC   ASMFPG
//ASM.SYSIN DD   *
            (sample program source deck)
/*
```

A more detailed explanation of the contents of the procedure is given in OS Assembler (F) Programmer's Guide.

All IBM-supplied cataloged procedures use the ddnames SYSLIN, SYSLIB, and SYSLOUT for the loader program. If other ddnames are assigned by the installation at system generation time, then these alternate ddnames must be substituted in the ASMFPG cataloged procedure.

### OPERATING INSTRUCTIONS

The operating instructions to be followed are those provided with the IEUESP sample program.

### OUTPUT

The output will be the same as the output for the IEUESP sample program. The loader output, however, will be different from the linkage-editor output:

1. No cross reference list is produced.
2. The storage map contains the absolute addresses of the external names rather than the relative addresses.

## PL/I F Sample Program (IEMSP2)

The IEMSP2 card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements for a PL/I compilation, link-edit, and execution of the compiled program. The cataloged procedure PL1LFCLG is used.
2. PL/I sample program source statements.
3. Input data.

The purpose of the sample program is to illustrate the use of record-oriented input/output, initialization of STATIC arrays, and tabulation of data-directed output. Statements used include PROCEDURE, DECLARE, BEGIN, END, ON, GOTO, READ, WRITE, PUT EDIT, and PUT DATA.

### OPERATING PROCEDURES

1. Mount the operating system.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the IEMSP2 card deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

1. Job control language statements in the cataloged procedure.
2. Normal compilation output:
  - a. All source statements.
  - b. An attribute and cross-reference listing.
  - c. A list of errors found during the compilation.
3. Normal link-edit step output.
4. The correct generated output is shown in Figure 47.

---

#### OUTPUT OF PL/I SAMPLE PROGRAM

THIS CARD IGNORED BECAUSE TYPE IS NOT NUMERIC:=8 DELIBERATE DUD CARD

TYPE(0) = 1    TYPE(1) = 0    TYPE(2) = 4    TYPE(3) = 0    TYPE(4) = 1  
TYPE(5) = 0    TYPE(6) = 10    TYPE(7) = 1    TYPE(8) = 1    TYPE(9) = 0;

END OF SAMPLE OUTPUT

---

Figure 47. PL/I Generated Output

## RPG Sample Program (RPGSMPL)

The RPGSMPL card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements to call a cataloged procedure for a compilation, link-edit, and execution of the sample program.
2. Forty-five report program generator source statements.
3. Thirteen data cards for the program to process; no other input is necessary.

The program processes the 13 data cards, which contain details of customer transactions, and prepares the report shown in Figure 48. A detailed description of the source program appears in OS RPG Language.

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the RPGSMPL card deck in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

Program output consists of a printed listing of the source program and the printed report shown in Figure 48.

A C C O U N T S R E C E I V A B L E R E G I S T E R

CUSTOMER NUMBER	CUSTOMER NAME	LOCATION STATE	CITY	INVOICE NUMBER	INVOICE NO.	DATE DAY	INVOICE AMOUNT
10712	AMALGAMATED CORP	33	61	11603	11	10	\$ 389.25 \$ 389.25*
11315	BROWN WHOLESALE	30	231	12324	12	28	\$ 802.08
11315	BROWN WHOLESALE	30	231	99588	12	14	\$ 261.17 \$ 1,063.25*
11897	FARM IMPLEMENTS	47	77	10901	10	18	\$ 27.63 \$ 27.63*
18530	BLACK OIL	16	67	11509	11	8	\$ 592.95
18530	BLACK OIL	16	67	12292	12	23	\$ 950.97 \$ 1,543.92*
20716	LEATHER BELT CO	36	471	11511	11	8	\$ 335.63
20716	LEATHER BELT CO	36	471	12263	12	17	\$ 121.75 \$ 457.38*
29017	GENERAL MFG CO	6	63	11615	11	14	\$ 440.12
29017	GENERAL MFG CO	6	63	11676	11	23	\$ 722.22 \$ 1,162.34*
29054	A-B-C DIST CO	25	39	9689	9	11	\$ 645.40
29054	A-B-C DIST CO	25	39	11605	11	11	\$ 271.69
29054	A-B-C DIST CO	25	39	12234	12	14	\$ 559.33 \$ 1,476.42* \$ 6,120.19**

Figure 48. RPG Sample Program Printed Report

## Sample Account Routine (SAMACTRT)

This sample account routine is intended for use as a model of an account routine. Listing the cards that have been punched out from SYS1.SAMPLIB provides sample coding that can be used for reference while the user is writing his own account routine.

## Sort/Merge Sample Program (IERSP)

The IERSP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements for TAPESORT
2. Job control language statements for DISKSORT
3. Data to be sorted

The data to be sorted consists of 500 80-character records, each containing a six-digit sequence number and a ten-character control field. As provided, the data deck is in the six-digit sequence number order. The SORT control card specifies that the data is to be sorted on the ten-character control field. The output of either a TAPESORT or a DISKSORT will be in ten-character control field sequence.

The TAPESORT job control language statements provided in IERSP call for five tape units (three for scratch, one for SORTIN, and one for SORTOUT). To use more than three work tapes, add additional cards behind the SORTWK03 DD card. They should be identical to other SORTWK DD cards except for DD names, which must be consecutive, that is, SORTWK04, SORTWK05, ..., SORTWK32. SORTOUT is specified as a 9-track 800-BPI tape.

The DISKSORT job control language statements provided in IERSP call for six work areas of 30 consecutive tracks each on a 2311. If the output from the DISKSORT is to be printed in a print step, the DISP parameters in the SORTOUT DD card must be changed to (,PASS) before the sort run.

### OPERATING INSTRUCTIONS

1. Mount the operating system and an initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. For a TAPESORT, the data deck should be placed on tape as one unblocked sequential data set. An operator message (during step #7) will give mounting instructions.

For a DISKSORT, the data deck should be placed on the scratch pack (mounted in step #1) as one unblocked sequential data set and cataloged under the name TEST.

4. For a TAPESORT, place the TAPESORT job control language statements in the card reader followed by a /\* card.

For a DISKSORT, place the DISKSORT job control language statements in the card reader followed by a /\* card.

5. Ready the reader and press the End-of-File key.
6. Ready the printer.
7. Execute the job.

### OUTPUT

A message to the operator will indicate the unit assigned to SORTOUT (unit containing output). This output can be printed and reviewed. The records will be in 10-character control field sequence.

Output of TAPESORT will be on tape. Output of DISKSORT will be on disk.

## Update Analysis Program (IHGSAMP)

The IHGSAMP card deck (punched from SYS1.SAMPLIB) consists of:

1. Job control language statements and data to accomplish Stage I, steps 1, 1A, 2, 3, 4, 5, 6, and 7 described in the "Output" section.
2. Job control language statements to accomplish Stage II, steps 3, 4, 5, and 6.

This sample problem is a two-stage operation requiring two input card decks. The Stage I input deck is complete. However, the output created by Stage I must be placed in front of the Stage II input deck before Stage II can be executed. (The output of Stage I is the job control language statements to accomplish Stage II, steps 1 and 2.)

To demonstrate the functions of Update Analysis, it is necessary to create some data sets on a disk pack of the operating system. However, the sample problem has been constructed so that at its completion all data sets created by it are deleted. Therefore, the program can be run without leaving residue from its operations on the system disk packs. The printer output acts as a graphic demonstration of the changing programs.

Stage I builds two sample symbolic libraries and one sample change library. It then executes the Update Analysis Program, which creates the job stream required to update the sample libraries. Stage II utilizes the output of the Update Analysis to update the two sample libraries, then proceeds to eliminate the data sets from the system. This is described in "Output."

### OPERATING INSTRUCTIONS

1. Mount the operating system and initialized scratch pack.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the Stage I card deck in the reader and ready the reader, printer, and punch.
4. Execute the Stage I job stream.
5. At the end of the job, take the output of Stage I from the punch stacker and place in front of Stage II deck.
6. Place the Stage II deck in the reader and ready the reader.
7. Issue commands:  
START RDR, 00C  
START  
to execute the Stage II job stream.

### OUTPUT

#### Stage I

- Step 1 EXEC IEBUPDTE  
This step creates a temporary data set that contains the SYSIN control cards for step 1A.



- Step 1A EXEC IEBGENER  
This step creates the PDS SAMPLCHG, which is the change PDS containing three change members to be applied to the two sample libraries.
- Step 2 EXEC IEBUPDTE  
This step creates the PDS SAMPLIB1, which is a sample system library containing two members: MEMBER1 and MEMBER2.
- Step 3 EXEC IEBUPDTE  
This step creates the PDS SAMPLIB2, which is a sample system library containing one member: MEMBER3.
- Step 4 EXEC IEHLIST  
This step lists the directories of the two sample libraries. The list shows the SSI bytes of the members prior to updating.
- Step 5 EXEC IHGUAP  
This step causes the SAMPLCHG PDS to be analyzed and creates the job stream that updates the sample system libraries. For the purpose of the sample problem, this output is temporarily stored on disk to be available for steps 6 and 7.
- Step 6 EXEC IEBPTPCH  
This prints the output of step 5.
- Step 7 EXEC IEBPTPCH  
This punches the output of step 5.

#### Stage II

- Steps 1 (Using punched output from step 7  
and 2 of Stage I)
- EXEC IEBUPDTE            Update SAMPLIB1
- EXEC IEBUPDTE            Update SAMPLIB2
- Step 3 EXEC IEHLIST  
This step lists the directories of the two sample libraries. The list shows the SSI bytes of the members after they have been updated.
- Step 4 EXEC IEBUPDTE  
This reproduces SAMPLIB1 in its updated form and creates a printed list that shows the effect of the change.
- Step 5 EXEC IEBUPDTE  
This reproduces SAMPLIB2 in its updated form and creates a printed list that shows the effect of the change.
- Step 6 EXEC IEHPRGM  
SAMPLCHG, SAMPLIB1, and SAMPLIB2 data sets are deleted by means of job control language statements in step 5 of Stage I and step 5 of Stage II. As a check, however, step 6 attempts to delete the same data sets and generates the diagnostic message, "DATA SET OR MEMBER NOT FOUND."

## User Label Sample Program (USERLABL)

The USERLABL card deck (punched from SYS1.SAMPLIB) consists of sample program source statements.

The sample program demonstrates the user label and user totaling functions and serves as a minimal test of these functions. In addition, it provides sample coding that demonstrates how to code a routine to process user labels.

The sample program creates a data set that contains eight user header labels, 500 data records, and eight user trailer labels on a direct access or magnetic tape volume. It also maintains a record count as an example of the use of the totaling facility. A listing of the source program card deck provides you with a sample of the basic coding required as an interface between the control program and a user label routine.

You must provide the job control language required to assemble, link-edit, and run the sample program. A suggested sequence of JCL to assemble using Assembler F is:

```
//jobname      JOB
//stepname     EXEC  PROC=ASMFCLG
//ASM.SYSIN    DD   *
.
. Sample Program Source Deck
.
/*
//GO.OUTPUT    DD   (parameters necessary for a volume output device)
//GO.SYSPRINT  DD   (parameters necessary for your printer)
```

You must prepare two DD statements for the GO step: one for the data set to be created and one for the messages written by the sample program. The first DD statement must have a ddname of OUTPUT and contain LABEL=(,SUL) plus the additional parameters required for the output device desired. The second DD statement must have a ddname of SYSPRINT and contain the additional parameters required for the system printer.

### OPERATING INSTRUCTIONS

1. Mount the operating system and the desired direct-access device or magnetic tape volume.
2. Set the load address switches and press the Initial Program Load key to load the operating system.
3. Place the USERLABL card deck with the JCL in the card reader, ready the reader, and press the End-of-File key.
4. Ready the printer.
5. Execute the job.

### OUTPUT

The following is written on the output device specified in the DD statement for the output device:

1. Eight user header labels.
2. A one-volume data set consisting of eight records.
3. Eight user trailer labels.

In order to verify the results of this sample program it is suggested that you print the contents of the data set and its tables.

This section contains three examples of system generation.

- Example 1 shows the generation of the 3330 starter operating system. It illustrates a complete system generation and a nucleus generation.
- Example 2 shows a complete operating system generation and an I/O device generation using the starter operating system.
- Example 3 shows a TSO generation.

The machine configurations and operating systems shown in these examples are not meant to represent the needs of an average installation, but were chosen because of their value as examples. The numbers chosen for space allocation in these examples are for illustrative purposes only. Exact space requirements for the system data sets can be found in OS Storage Estimates.



## Example 1

This example shows the contents of the 3330 starter operating system. The example consists of:

- A diagram of a machine configuration
- A diagram of the volumes that contain generating and new system data sets
- An input deck listing used for initializing new system data sets
- A deck used for system generation

In addition, sections describe the following:

- Creating backup of the new system
- Scratching utility data sets
- Cataloging the generating SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and component libraries in the new system
- Using the new system to add a second nucleus to itself

### Volumes Used for System Generation

Figure 49 shows the volumes that contain the generating and new system data sets. The serial number of the generating system-residence volume is 111111. Volume DLIB01 contains the remaining generating system data sets. The serial number of the new system-residence volume is SYSTEM. It is assumed that SYSTEM has been previously initialized. The five utility data sets used during system generation are named SYS1.OBJMOD, SYS1.UT10, SYS1.UT20, SYS1.UT30, and SYS1.UT40.

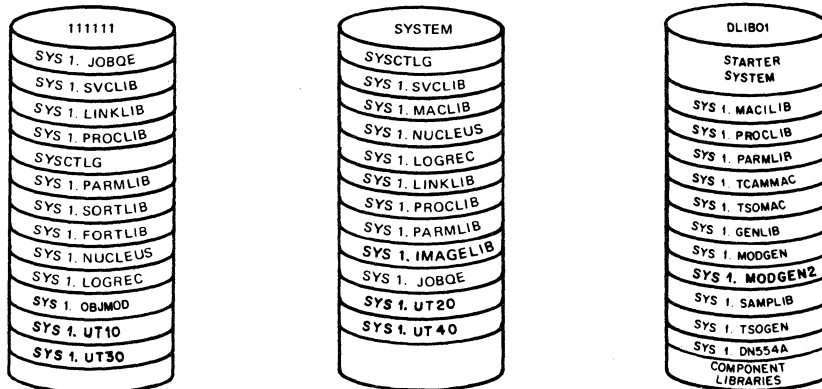


Figure 49. Example 1: Generating and New System Volumes

### Machine Configuration

Figure 50 shows the machine configuration used in this example.

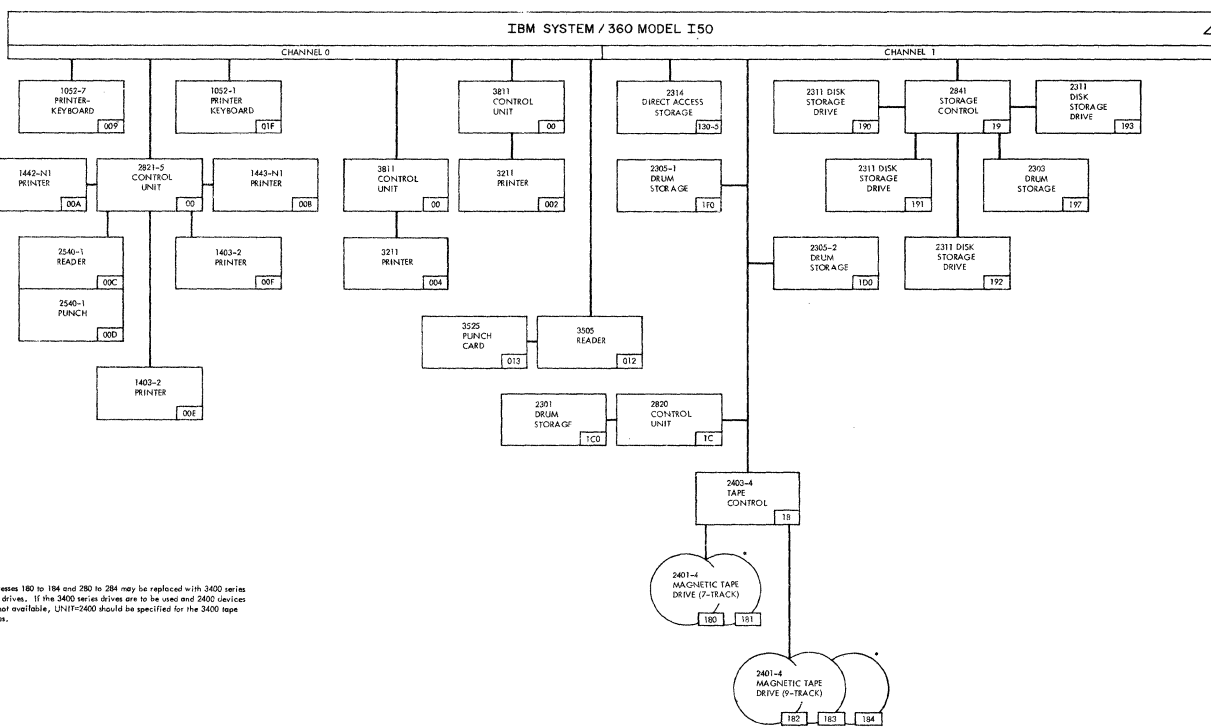


Figure 50 (Part 1 of 2). Example 1: 3330 Starter Machine Configuration

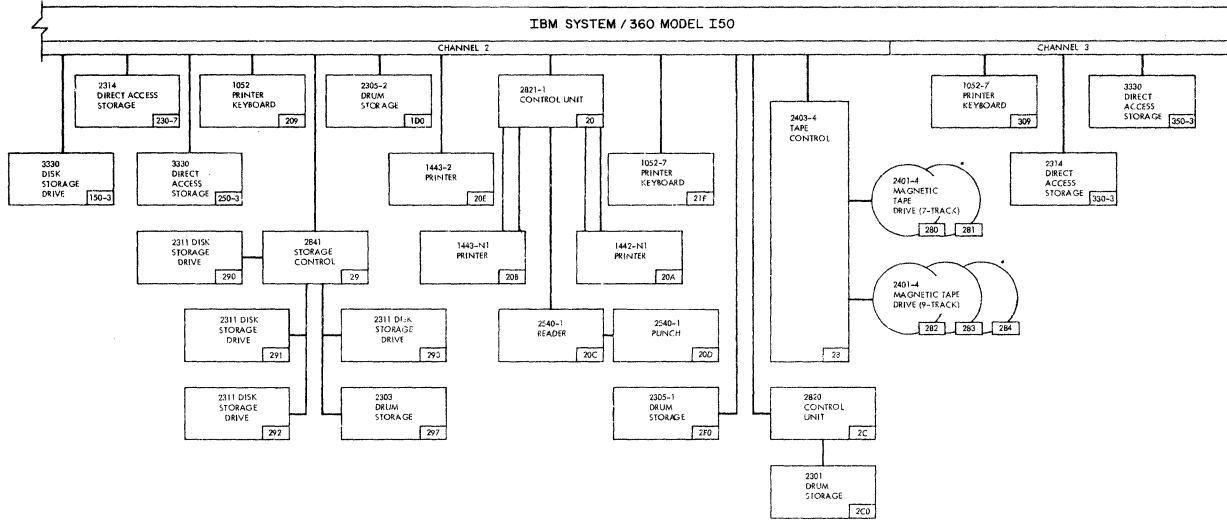


Figure 50 (Part 2 of 2). Example 1: 3330 Starter Machine Configuration

Deck for Initializing New System Data Sets

Figure 51 shows a listing of the input deck used for initializing the new system data sets. During this initialization, the SYSTEM volume must be mounted. It is assumed that there is no dependency on device addresses and that the volume can be mounted on any 3330 drive. For illustrative purposes, assume that volume 111111, DLIB01, DLIB04, and SYSTEM are mounted on drives 131, 350, and 351, respectively.

```

//SYSGEN   JOB MSGLEVEL=1           -EXAMPLE 1-
//STEP0    EXEC PGM=IEHPROGM        -ALLOCATE ON 3330-
//SYSPRINT DD SYSOUT=A
//CATALOG  DD DSNAME=SYSCTLG,VOLUME=(RETAIN,SER=SYSTEM),      X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(10,1)),        X
//          LABEL=EXPDT=99350
//SVCLIB   DD DSNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=SYSTEM),  X
//          UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(4,1,90)),      X
//          LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//NUCLEUS  DD DSNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=SYSTEM), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(10,,2),,CONTIG), X
//          LABEL=EXPDT=99350
//JOBQE    DD DSNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=LINVOL), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(300),,CONTIG)
//LINKLIB  DD DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=LINVOL), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(CYL,(8,5,50)),      X
//          LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//PROCLIB  DD DSNAME=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=LINVOL), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(15,6,9),      X
//          LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//IMAGELIB DD DSNAME=SYS1.IMAGELIB,VOLUME=(,RETAIN,SER=LINVOL), X
//          UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(5,,3),,CONTIG), X
//          LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024)
//PARMLIB  DD DSNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=LINVOL), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(5,,1),,CONTIG), X
//          LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//IMAGELIB DD DSNAME=SYS1.IMAGELIB,VOLUME=(,RETAIN,SER=SYSTEM), X
//          UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(2,,3),,CONTIG), X
//          LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024)
//SYSIN    DD * -INPUT FOR CATALOGING SYSTEM DATA SETS-
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.LINKLIB
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.PROCLIB
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.PARMLIB
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.NUCLEUS
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.JOBQE
           CATLG   CVOL=3330,SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.MACLIB
           CATLG   CVOL=3330,SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.IMAGELIB
           CATLG   CVOL=3330=SYSTEM,VOL=3330=SYSTEM,DSNAME=SYS1.SVCLIB
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.AS037           ASSEMBLER F
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.DM509           BDAM
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.CI505           CONTROL PROGRAM
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.DM508           DATA MANAGEMENT
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.UT506           UTILITIES
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.FO520           FORTRAN G
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.LM501           FORTRAN LIBRARY
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.IO526           ISAM
           CATLG   VOL=3330=DLIB01,DSNAME=SYS1.ED521           LINKAGE EDITOR F

```

Figure 51 (Part 1 of 2). Example 1: Initializing New System Data Sets



```

CATLG VOL=3330=DLIB01,DSNAME=SYS1.LD547      LOADER
CATLG VOL=3330=DLIB01,DSNAME=SYS1.RG038      RPG
CATLG VOL=3330=DLIB01,DSNAME=SYS1.SM023      SORT/MERGE
CATLG VOL=3330=DLIB01,DSNAME=SYS1.PROCLIB
CATLG VOL=3330=DLIB01,DSNAME=SYS1.PARMLIB
CATLG VOL=3330=DLIB01,DSNAME=SYS1.MODGEN
CATLG VOL=3330=DLIB01,DSNAME=SYS1.GENLIB

```

/\*

Figure 51 (Part 2 of 2). Example 1: Initializing New System Data Sets

If the IBM-supplied copies of SYS1.PROCLIB and SYS1.PARMLIB from DLIB01 are to be in the new system, they must be recataloged on the existing system. They are recataloged on the existing system to point to DLIB01. During system generation, SYS1.PROCLIB and SYS1.PARMLIB are copied from the data sets cataloged on the generating system.

#### Input Deck for Stage I

Figure 52 shows a listing of the input deck used for system generation. The utility data sets are allocated space on the volumes indicated in Figure 49. The job stream will be written on an unlabeled magnetic tape that resides on drive 183 (see Figure 50). Unit names are used. This input deck is the second step of the job illustrated in Figure 51.

For Stage I of the system generation process, the volumes that are mounted are 111111, DLIB01, and SYSTEM. Stage II will be started automatically after Stage I is completed.

The new system supports all the devices shown in Figure 50. The MFT control program is used. The compiler has all standard default options for compilation time. The procedure library and the unit names it requires are also included.

```

//STEP1 EXEC PGM=ASMBLR -STAGE I INPUT-
//SYSLIB DD DSN=SYS1.GENLIB,DISP=OLD
//OBJPDS DD DSN=SYS1.OBJMOD,VOLUME=(,RETAIN,SER=111111), X
// DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(30,20,8))
//SYSUT10 DD DSN=SYS1.UT10,VOLUME=(,RETAIN,SER=111111), X
// DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(50,10))
//SYSUT20 DD DSN=SYS1.UT20,VOLUME=(,RETAIN,SER=SYSTEM), X
// DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(150,10))
//SYSUT30 DD DSN=SYS1.UT30,VOLUME=(,RETAIN,SER=111111), X
// DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(150,10))
//SYSUT40 DD DSN=SYS1.UT40,VOLUME=(,RETAIN,SER=SYSTEM), X
// DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(5,10))
//SYSPUNCH DD UNIT=183,LABEL=(,NL)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
CENPROC CENPROCS INSTSET=UNIV,STORAGE=G,MODEL=50
CONTRL CTRLPROG MAXIO=20,OVERLAY=BASIC,TYPE=MFT,SYSQUE=5000
SUP SUPRVSR OPTIONS=(TRSVCTBL,COMM),SER=SERO, X
RESIDNT=ACSMETH
SCHED SCHEDULR DESIGN=44K,CONSOLE=C09,ALTCONS=01F,TYPE=MFT, X
WTOBFRS=2,REPLY=2,JOBQLMT=960,OPTIONS=NOLOG
PARTS PARTITNS P0(C-D,S-8K),P1(C-ABC,S-65K)
CHAN0 CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
IOC0811 IOCONTRL UNIT=3811,ADDRESS=00
IOD3211 IODEVICE UNIT=3211,ADDRESS=002
IOC3811 IOCONTRL UNIT=3811,ADDRESS=00
IOD3211 IODEVICE UNIT=3211,ADDRESS=004
IOD10527 IODEVICE UNIT=1052,ADDRESS=009,MODEL=7

```

Figure 52 (Part 1 of 3). Example 1: Stage I Input Deck

IOC28215	IOCONTRL	UNIT=2821, ADDRESS=00, MODEL=5	
IOD1442N	IODEVICE	UNIT=1442, ADDRESS=00A, MODEL=N1	
IOD1443N	IODEVICE	UNIT=1443, ADDRESS=00B, MODEL=N1	
IOD2540R	IODEVICE	UNIT=2540R, ADDRESS=00C, MODEL=1	
IOD2540P	IODEVICE	UNIT=2540P, ADDRESS=00D, MODEL=1	
IOD14032	IODEVICE	UNIT=1403, ADDRESS=00E, MODEL=2	
IOD14032	IODEVICE	UNIT=1403, ADDRESS=00F, MODEL=2	
IOD3525P	IODEVICE	UNIT=3525, ADDRESS=013	
IOD3505R	IODEVICE	UNIT=3505, ADDRESS=012	
IOD10527	IODEVICE	UNIT=1052, ADDRESS=01F, MODEL=7	
CHAN1	CHANNEL	ADDRESS=1, TYPE=SELECTOR	
IOD2314	IODEVICE	UNIT=2314, ADDRESS=(130,6)	
IOD3330	IODEVICE	UNIT=3330, ADDRESS=(150,4)	
IOC24034	IOCONTRL	UNIT=2403, ADDRESS=18, MODEL=4	X
		FEATURE=(DATA CONV, 7-TRACK, 800BPI)	
IOD24014	IODEVICE	UNIT=2401, ADDRESS=(180,2), MODEL=4,	X
		FEATURE=7-TRACK	
IOD24014	IODEVICE	UNIT=2401, ADDRESS=(182,3), MODEL=4,	X
		FEATURE=(9-TRACK, DUALDENS)	
IOC2841	IOCONTRL	UNIT=2841, ADDRESS=19	
IOD2311	IODEVICE	UNIT=2311, ADDRESS=(190,4)	
IOD2303	IODEVICE	UNIT=2303, ADDRESS=197	
IOC2820	IOCONTRL	UNIT=2820, ADDRESS=1C	
IOD2301	IODEVICE	UNIT=2301, ADDRESS=1C0	
IOD23052	IODEVICE	UNIT=2305, ADDRESS=1D0, MODEL=2	
IOD23051	IODEVICE	UNIT=2305, ADDRESS=1F0, MODEL=1	
CHAN2	CHANNEL	ADDRESS=2, TYPE=SELECTOR	
IOD90527	IODEVICE	UNIT=1052, ADDRESS=209, MODEL=7	
IOC28211	IOCONTRL	UNIT=2821, ADDRESS=20, MODEL=1	
IOD1442	IODEVICE	UNIT=1442, ADDRESS=20A, MODEL=N1	
IOD1443	IODEVICE	UNIT=1443, ADDRESS=20B, MODEL=N1	
IOD2540R	IODEVICE	UNIT=2540R, ADDRESS=20C, MODEL=1	
IOD2540P	IODEVICE	UNIT=2540P, ADDRESS=20D, MODEL=1	
IOD1443	IODEVICE	UNIT=1443, ADDRESS=20E, MODEL=2	
IOD10527	IODEVICE	UNIT=1052, ADDRESS=21F, MODEL=7	
IOD2314	IODEVICE	UNIT=2314, ADDRESS=(230,6)	
IOD3330	IODEVICE	UNIT=3330, ADDRESS=(250,4)	
IOC24034	IOCONTRL	UNIT=2403, ADDRESS=28, MODEL=4,	X
		FEATURE=(DATA CONV, 7-TRACK, 800BPI)	
IOD24014	IODEVICE	UNIT=2401, ADDRESS=(280,2), MODEL=4,	X
		FEATURE=7-TRACK	
IOD24014	IODEVICE	UNIT=2401, ADDRESS=(282,3), MODEL=4,	X
		FEATURE=(9-TRACK, DUALDENS)	
IOC2841	IOCONTRL	UNIT=2841, ADDRESS=29	
IOD2311	IODEVICE	UNIT=2311, ADDRESS=(290,4)	
IOD2303	IODEVICE	UNIT=2303, ADDRESS=297	
IOC2820	IOCONTRL	UNIT=2820, ADDRESS=2C	
IOD2301	IODEVICE	UNIT=2301, ADDRESS=2C0	
IOD23052	IODEVICE	UNIT=2305, ADDRESS=2D0, MODEL=2	
IOD23051	IODEVICE	UNIT=2305, ADDRESS=2F0, MODEL=1	
CHAN3	CHANNEL	ADDRESS=3, TYPE=SELECTOR	
IOD10527	IODEVICE	UNIT=1052, ADDRESS=309, MODEL=7	
IOD2314	IODEVICE	UNIT=2314, ADDRESS=(330,4)	
IOD3330	IODEVICE	UNIT=3330, ADDRESS=(350,4)	
SYSSQ	UNITNAME	NAME=SYSSQ, UNIT=((182,4), (282,3), (190,4),	X
		(290,4))	

Figure 52 (Part 2 of 3). Example 1: Stage I Input Deck

```

SYSDA      UNITNAME  NAME=SYSDA,UNIT=((190,4),(290,4),(130,6),
                (230,6),(330,4),(150,4),(250,4),197,1C0,
                1D0,1F0,297,2C0,2D0,2F0,(350,4))
                                                    X
                                                    X
SYSCP      UNITNAME  NAME=SYSCP,UNIT=(00D,00A,20A,20D)
ASM        ASSEMBLR  DESIGN=F
EDIT       EDITOR    DESIGN=F44,SIZE=(100K,6K)
UTILS     SYSUTILS  SIZE=44K
UCS3211   UCS        UNIT=2311,IMAGE=(A11,H11,G11,P11,T11),
                DEFAULT=(A11,P11)
                                                    X

IMGLIB     IMAGELIB
PARM       PARMLIB
PROC       PROCLIB
GEN        GENERATE  UT1SDS=SYS1.UT10,UT2SDS=SYS1,UT20,
                UT3SDS=SYS1.UT30,UT4SDS=SYS1.UT40,
                OBJPDS=SYS1.OBJMOD,RESTYPE=3330,
                RESNAME=3330,RESVOL=111111,
                LEPRT=(LIST,KREF),ASMPRT=ON,LBMAINT=E,
                DIRDATA=PDS
                                                    X
                                                    X
                                                    X
                                                    X
                                                    X
/*         END
//         START   RDR,183

```

Figure 52 (Part 3 of 3). Example 1: Stage I Input Deck

### Backup of New System

Figure 53 shows a listing of the IEHDASDR input deck used for creating the backup of the new system. IEHDASDR operates under control of the generating system on 111111. The backup copy of SYSTEM is made on the SYS001 magnetic tape volume. After the backup copy has been made, SYS001 should be removed and stored in the tape library.

```

//BACKUP   JOB      MSGLEVEL=1          -BACKUP-
//         EXEC     PGM=IEHDASDR
//SYSPRINT DD      SYSOUT=A
//SYSPACK  DD      UNIT=3330,DISP=OLD,VOLUME=SER=SYSTEM
//SYSTAPE  DD      UNIT=2400,DISP=OLD,VOLUME=SER=SYS001,LABEL=(,NL)
//SYSIN    DD      *
          DUMP     FROMDD=SYSPACK,TODD=SYSTAPE
/*

```

Figure 53. Example 1: Creating Backup of SYSTEM

## Scratching Utility Data Sets

After the new operating system is generated, SYS1.UT10, SYS1.UT20, SYS1.UT30, and SYS1.UT40 should be scratched and uncataloged. The job stream and SYS1.OBJMOD should be saved. Figure 54 shows a listing of the input deck listing used for scratching and uncataloging the three sequential data sets. The magnetic tape volume that contains the job stream can be removed from unit 183 and stored in the tape library. The IEHPROGM utility program is executed under control of the generating system.

```
//SCRATCH JOB MSGLEVEL=1 -SCRATCH DECK-
//STEP0 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//SYSUT10 DD DSNAME=SYS1.UT10,DISP=(OLD,DELETE)
//SYSUT20 DD DSNAME=SYS1.UT20,DISP=(OLD,DELETE)
//SYSUT30 DD DSNAME=SYS1.UT30,DISP=(OLD,DELETE)
//SYSUT40 DD DSNAME=SYS1.UT40,DISP=(OLD,DELETE)
//SYSIN DD DUMMY
```

Figure 54. Example 1: Scratching Utility Data Sets

## Obtaining SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and Component Libraries

The new system (volume SYSTEM) meets all the requirements for a generating system except that it does not have SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the component libraries. These libraries can be copied from the generating system to the new system and then cataloged. Figure 55 shows a listing of the input deck used for cataloging SYS1.GENLIB and the component libraries in the new system.

```
//SYSGEN JOB MSGLEVEL=1 -PROCESSOR/LIBRARY GENERATION-
//STEP1 EXEC PGM=IEHPROGM -ALLOCATE SYS1.PL1LIB-
//SYSPRINT DD SYSOUT=A
//PL1LIB DD DSNAME=SYS1.PL1LIB,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=3330,DISP=(,KEEP),SPACE=(TRK,(80,10,65)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//SYSIN DD *
CATLG VOL=3330=DLIB01,DSNAME=SYS1.PL1LIB
CATLG DSNAME=SYS1.GENLIB,VOL=3330=DLIB01
CATLG DSNAME=SYS1.MODGEN,VOL=3330=DLIB01
CATLG DSNAME=SYS1.MODGEN2,VOL=3330=DLIB01
CATLG DSNAME=SYS1.NL511,VOL=3330=DLIB01
CATLG DSNAME=SYS1.LM512,VOL=3330=DLIB01
/*
```

Figure 55. Example 1: Cataloging SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and Component Libraries

## Nucleus Generation

A second nucleus (02) is added to the new system during this generation. The same CENPROCS, CHANNEL, IOCTRL, and IODEVICE macro instructions included in the generation of the new system (see Figure 52) must be included in this generation. The new nucleus will then support the same access methods as the first nucleus. Each nucleus must support the same control program (MFT), because the resident portion of the control program (nucleus) must be compatible with the nonresident portion.

Figure 56 shows the volumes involved in this generation. The five system generation utility data sets are named SYS1.OBJECT, SYS1.ONE, SYS1.TWO, SYS1.THREE, and SYS1.FOUR.

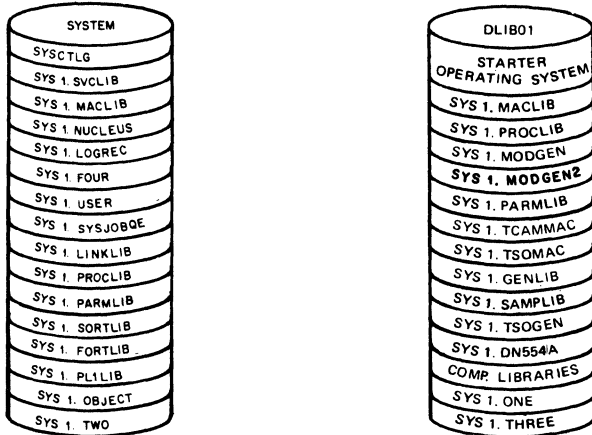


Figure 56. Example 1: Volumes for Nucleus Generation

Figure 57 shows a listing of the input deck used for this generation. STEP1 catalogs the component libraries required. SYS1.GENLIB and SYS1.MODGEN are already cataloged in the system so only the component libraries required for this system generation need be added. STEP2 is the input deck for Stage 1. Five utility data sets are defined. Their names are SYS1.ONE, SYS1.TWO, SYS1.THREE, SYS1.FOUR, and SYS1.OBJECT. The job stream will be written on an unlabeled magnetic tape that is located on drive 183. A user-written routine named NUCID2 is included in the nucleus. NUCID2 is a member of the SYS1.USER data set.

```

//SYSGEN   JOB   MCR,67,MSGLEVEL=1           -NUCLEUS GENERATION-
//STEP1    EXEC  PGM=IEHPRGM                 -COMPONENT LIBRARIES-
//NEWRES   DD    UNIT=3330,VOLUME=SER=SYSTEM,DISP=OLD
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
      CATLG   DSNAME=SYS1.USER,VOL=3330=SYSTEM
      CATLG   DSNAME=SYS1.CI505,VOL=3330=DLIB01
      CATLG   DSNAME=SYS1.DM509,VOL=3330=DLIB01
      CATLG   DSNAME=SYS1.IO526,VOL=3330=DLIB01
      CATLG   DSNAME=SYS1.PT516,VOL=3330=DLIB01
//STEP2    EXEC  PGM=ASMBLR                 -STAGE 1 INPUT-
//SYSLIB   DD    DSNAME=SYS1.GENLIB,DISP=OLD
//OBJPDS   DD    DSNAME=SYS1.OBJECT,VOLUME=(,RETAIN,SER=SYSTEM),      X
//          DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(40,20,8))
//SYSUT1    DD    DSNAME=SYS1.ONE,VOLUME=(,RETAIN,SER=DLIB01),      X
//          DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(30,20,8))
//SYSUT2    DD    DSNAME=SYS1.TWO,VOLUME=(,RETAIN,SER=SYSTEM),      X
//          DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(150,10))
//SYSUT3    DD    DSNAME=SYS1.THREE,VOLUME=(,RETAIN,SER=DLIB01),    X
//          DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(150,10))
//SYSUT4    DD    DSNAME=SYS1.FOUR,VOLUME=(,RETAIN,SER=SYSTEM),     X
//          DISP=(,CATLG),UNIT=3330,SPACE=(TRK,(5,10))
//SYSPUNCH DD    UNIT=183,LABEL=(,NL)
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
      CENPROCS MODEL=50,STORAGE=G,INSTSET=UNIV
      CTRLPROG TYPE=MFT,MAXIO=20,OVERLAY=BASIC,SYSQUE=5000
      SCHEDULR CONSOLE=009,DESIGN=44K,ALTCONS=01F,                  X
              TYPE=MFT,JOBQLMT=960,OPTIONS=NOLOG
      SUPRVSOR RESIDNT=ACSMETH,                                       X
              OPTIONS=(TRSVCTBL,COMM),                               X
              TRACE=100,SER=SER0                                     X
      RESMODS   PDS=SYS1.USER,MEMBERS=NUCID2
      GENERATE  GENTYPE=(NUCLEUS,2),UT1SDS=SYS1.ONE,                 X
              UT2SDS=SYS1.TWO,UT3SDS=SYS1.THREE,UT4SDS=SYS1.FOUR,  X
              OBJPDS=SYS1.OBJECT,RESNAME=3330,RESVOL=SYSTEM,       X
              RESTYPE=3330,LBMAINT=E,ASMPTR=ON,LEPRT=(LIST,XREF)
      END
/*
//          START      RDR,183

```

Figure 57. Example 1: Input Deck for Nucleus Generation

## Example 2

This example illustrates and describes:

- A machine configuration
- Deck listings used for initializing the starter operating system (2314 distribution)
- Deck listings used for initializing direct-access volumes for the new system
- Volumes that contain generating and new system data sets
- A deck listing used for initializing new system data sets
- A deck listing used for system generation
- Deck listings used for system residence on 2303
- A deck listing used for I/O Device generation

### Machine Configuration

Figure 58 shows the machine configuration used in this example. Only those devices in the shaded areas are supported by the starter operating system.

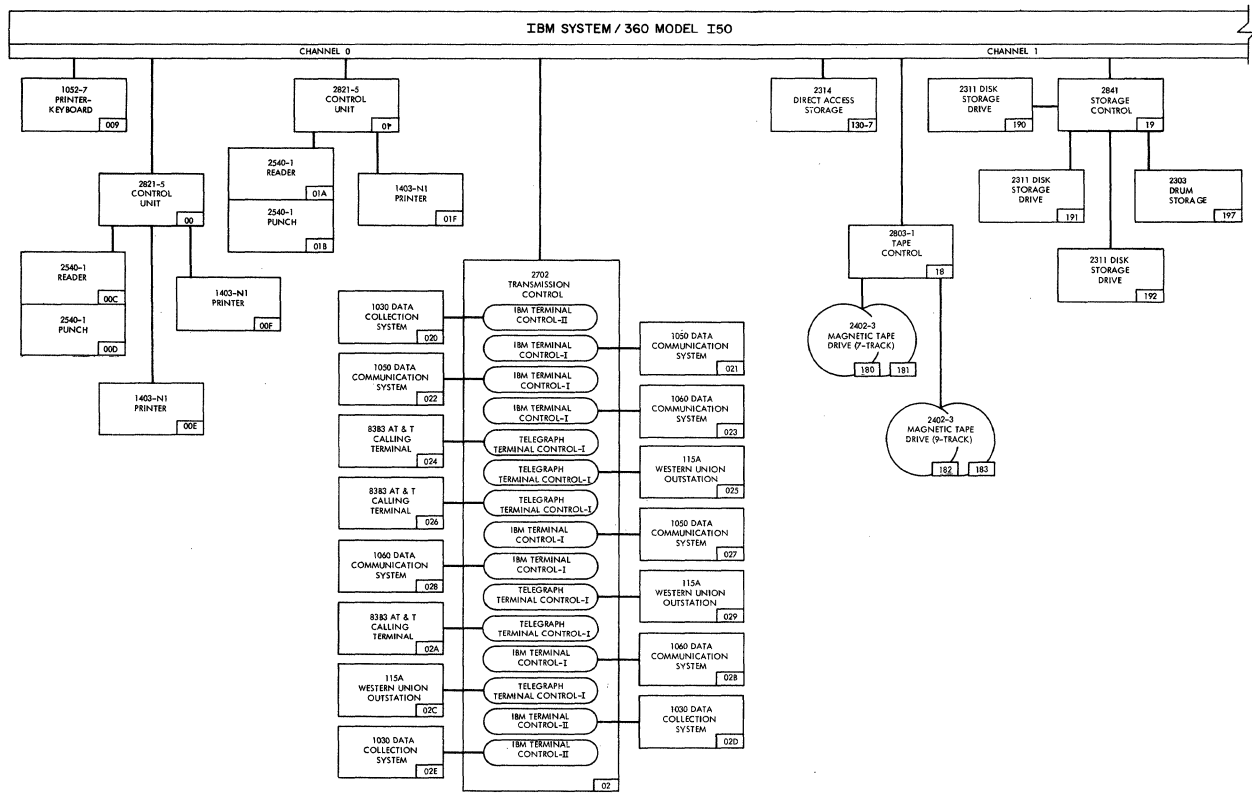


Figure 58 (Part 1 of 2). Example 2: Machine Configuration



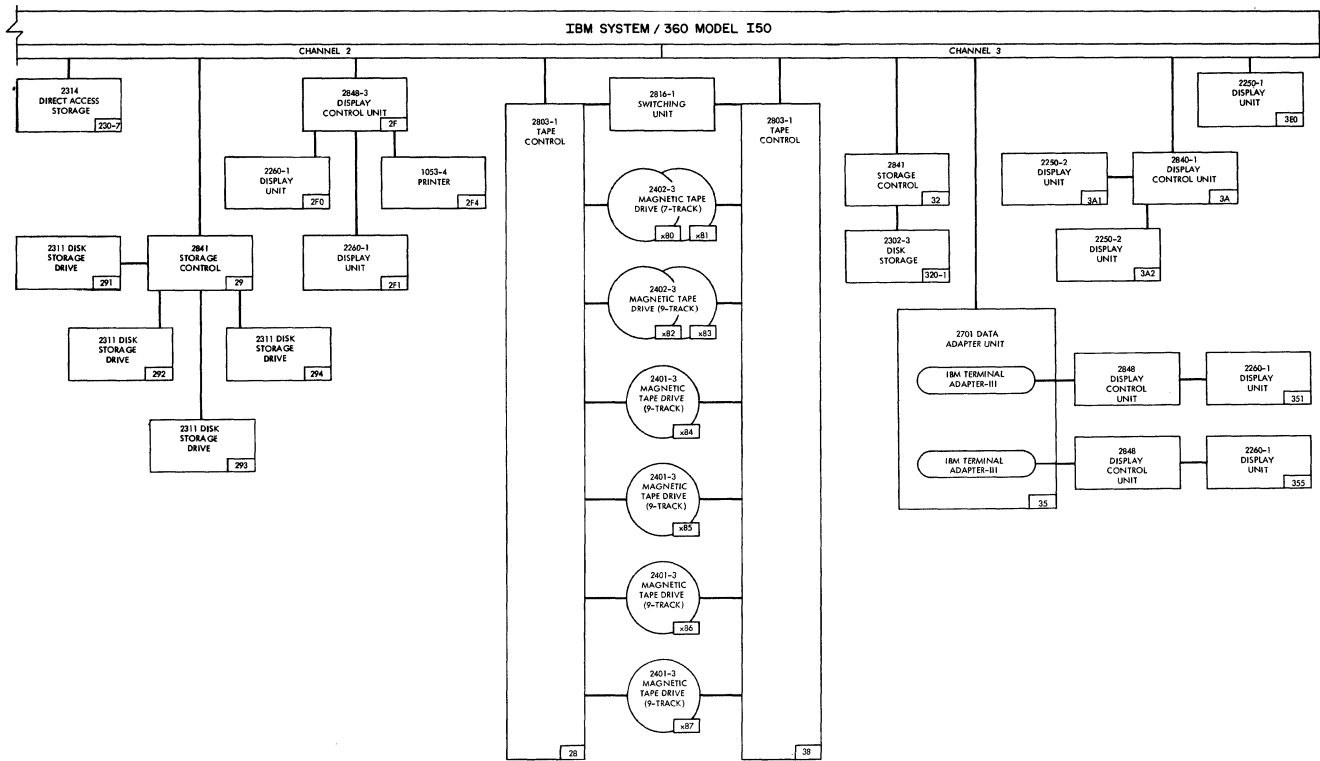


Figure 58 (Part 2 of 2). Example 2: Machine Configuration

## Decks Required to Use the Starter Operating System

Four steps are required for the starter operating system to be used as the generating system. The first step is to initialize the volumes that are to contain DLIB01 and DLIB02. The second step is to transfer the data from the distribution tapes to the volumes. The third step is to punch the programs desired from SYS1.SAMPLIB. The fourth step is to list the contents of SYS1.SAMPLIB, SYS1.NUCLEUS, and SYS1.PROCLIB.

Figure 59 shows a listing of the input deck used for the first step. The IBCDASDI program is used to initialize the volumes that are to contain DLIB01 and DLIB02. The serial numbers of the volumes are NNNNNN and XXXXXX. Both volumes are being initialized for the first time. The NNNNNN volume is located on unit 130 and the XXXXXX volume is located on unit 131. The DASDI deck is placed in the card reader located at address 00C. Messages will be printed on the printer located at address 00E.

---

```
DLIB01  JOB
        MSG      TODEV=1403,TOADDR=00E
        DADEF    TODEV=2314,TOADDR=130,VOLID=SCRATCH,      X
                FLAGTEST=NO
        VLD      NEWVOLID=NNNNNN,OWNERID=DEPTD58
        VTOCD    STARTADR=50,EXTENT=10
        END
DLIB02  JOB
        MSG      TODEV=1403,TOADDR=00E
        DADEF    TODEV=2314,TOADDR=131,VOLID=SCRATCH,      X
                FLAGTEST=NO
        VLD      NEWVOLID=XXXXXX,OWNERID=DEPTD58
        VTOCD    STARTADR=50,EXTENT=10
        END
```

---

Figure 59. Example 2: Initializing Volumes for DLIB01 and DLIB02

Figures 60 and 61 show the listings of the decks used for the second step. The IBCDMPRS program is used to transfer the data from the 2314 distribution tapes to the NNNNNN and XXXXXX volumes. The distribution tapes are located on units 282 and 283. The deck is placed in the card reader located at address 00C, and messages are printed on the printer-located address 00E.

Figure 60 shows the listings of the IBCDMPRS input deck used to transfer the first two files from the distribution tapes to volume NNNNNN. These two files contain the data for DLIB01.

---

```
TAPE    JOB
        MSG      TODEV=1403,TOADDR=00E
        RESTORE  FROMDEV=2400,FROMADDR=282,TODEV=2314,      X
                TOADDR=130,VOLID=NNNNNN
        END
TAPE2   JOB
        MSG      TODEV=1403,TOADDR=00E
        RESTORE  FROMDEV=2400,FROMADDR=283,TODEV=2314,      X
                TOADDR=130,VOLID=DLIB01
        END
```

---

Figure 60. Example 2: Restoring DLIB01

After executing the IBCDMPRS program, the volume on unit 130 is now DLIB01. Remove the tape on unit 282 and store it in the tape library as a backup for the starter system. Carry out the IPL procedure for the DLIB01 volume. The starter operating system on DLIB01 is used for the third and fourth steps.

Figure 61 shows a listing of the input deck used for the IEHDASDR program that transfers the data in the fourth file on the second tape of the distribution package to the XXXXXX volume to create DLIB02.

---

```
//DLIB02 JOB MSGLEVEL=1
//REST EXEC PGM=IEHDASDR,PARM='N=1'
//SYSPRINT DD SYSOUT=A
//FROM DD UNIT=283,DISP=OLD,LABEL=(4,NL), X
// VOLUME=SER=REST02
//DLIB02 DD UNIT=131,DISP=OLD,VOLUME=SER=DLIB02
//SYSIN DD *
RESTORE FROMDD=FROM,TODD=DLIB02,CPYVOLID=YES, X
PURGE=YES
*
```

---

Figure 61. Example 2: Restoring DLIB02

Figure 62 shows a listing of the input deck used for the third step. The IEBTPCH program is used to punch the desired contents of SYS1.SAMPLIB. The independent utility programs, the sample programs for FORTRAN, PL/I, and Sort/Merge, and the IPL program are punched.

---

```
//PUNCH JOB MSGLEVEL=1 -PUNCH SYS1.SAMPLIB-
// EXEC PGM=IEBTPCH
//SYSUT1 DD DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP), X
// UNIT=2314,VOLUME=SER=DLIB02
//SYSUT2 DD UNIT=2540-2
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
PUNCH TYPORG=PO,MAXNAME=7
MEMBER NAME=IBCDMPRS
MEMBER NAME=IBCDASDI
MEMBER NAME=IBRCVPR
MEMBER NAME=IEAIPLOO
MEMBER NAME=IEYSP
MEMBER NAME=IEMSP2
MEMBER NAME=IERSP
/*
```

---

Figure 62. Example 2: Punching Members of SYS1.SAMPLIB

After these seven decks are punched, the first card (member name card) of each deck must be removed.

Figure 63 shows a listing of the input deck used for the system data describing the starter system's SYS1.NUCLEUS, SYS1.SAMPLIB, and SYS1.PROCLIB.

---

```
//LIST JOB MSGLEVEL=1
// EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1 DD DISP=OLD,UNIT=2314,VOLUME=SER=DLIB01
//DD2 DD DISP=OLD,UNIT=2314,VOLUME=SER=DLIB02
//SYSIN DD *
LISTCTLG
LISTVTOC VOL=2314=DLIB01,DUMP
LISTVTOC VOL=2314=DLIB02,DUMP
LISTPDS DSNAME=SYS1.NUCLEUS,VOL=2314=DLIB01
LISTPDS DSNAME=SYS1.PROCLIB,VOL=2314=DLIB01
LISTPDS DSNAME=SYS1.SAMPLIB,VOL=2314=DLIB02
/*
```

---

Figure 63. Example 2: Listing Data on DLIB01 and DLIB02

## Initializing Volumes for New System

The volumes for the new system must be initialized before system generation. Five volumes (besides DLIB01 and DLIB02) are used whose serial numbers are MVT111, MVT222, MVT333, MVT444, and MVT555. These volumes are mounted on units 131, 230, 231, 232, and 132, respectively. MVT111 is to be the new system-residence volume. It is assumed that all these volumes have been previously initialized.

Figure 64 shows five listings of the IBCDASDI input decks used for initializing the new volumes. The deck for MVT111 contains the IPL program (IEAIPL00). The IPL cards were obtained from SYS1.SAMPLIB (See Figure 62).

The IBCDASDI object program cards must be placed in the card reader on unit 00C and loaded into main storage. The input decks shown in Figure 61 are placed in the card reader immediately following the IBCDASDI object program deck. (For further details on the operating procedure for IBCDASDI, refer to OS Utilities.)

---

```
MVT111 JOB
  MSG      TODDEV=1403,TOADDR=00E
  DADEF    TODDEV=2314,TOADDR=131,IPL=YES,VOLID=SCRATCH
  VLD      NEWVOLID=MVT111,OWNERID=DEPTD58
  VTOCD    STRTADR=2,EXTENT=8
  IPLTXT
  .
  .      IEAIPL00 cards
  .
END
MVT222 JOB
  MSG      TODDEV=1403,TOADDR=00E
  DADEF    TODDEV=2314,TOADDR=230,VOLID=SCRATCH
  VLD      NEWVOLID=MVT222,OWNERID=DEPTD58
  VTOCD    STRTADR=2,EXTENT=8
  END
MVT333 JOB
  MSG      TODDEV=1403,TOADDR=00E
  DADEF    TODDEV=2314,TOADDR=231,VOLID=SCRATCH
  VLD      NEWVOLID=MVT333,OWNERID=DEPTD58
  VTOCD    STRTADR=2,EXTENT=8
  END
MVT444 JOB
  MSG      TODDEV=1403,TOADDR=00E
  DADEF    TODDEV=2314,TOADDR=232,VOLID=SCRATCH
  VLD      NEWVOLID=MVT444,OWNERID=DEPTD58
  VTOCD    STRTADR=2,EXTENT=8
  END
MVT555 JOB
  MSG      TODDEV=1403,TOADDR=00E
  DADEF    TODDEV=2314,TOADDR=132,VOLID=SCRATCH
  VLD      NEWVOLID=MVT555,OWNERID=DEPTD58
  VTOCD    STRTADR=2,EXTENT=8
  END
```

---

Figure 64. Example 2: Initializing New System Volumes

## Volumes Used for System Generation

Figure 65 shows the volumes that contain the generating and new system data sets. The serial number of the generating system-residence volume is DLIB01. The serial number of the new system-residence volume is MVT111. Volumes MVT222, MVT333, and MVT444 contain the remaining new system data sets. The five utility data sets used during system generation are named SYS1.SGOBJ, SYS1.SG1, SYS1.SG2, SYS1.SG3, and SYS1.SG4.

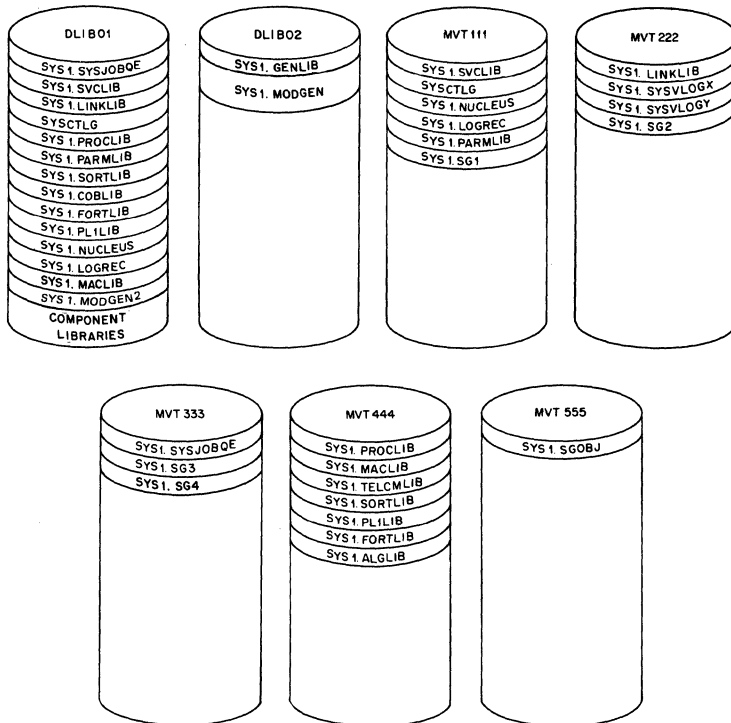


Figure 65. Example 2: Generating and New System Volumes (2314)

## Initializing New System Data Sets

Figure 66 shows a listing of the input deck used for initializing the new system data sets. During this initialization, volumes DLIB01, MVT111, MVT222, MVT333, and MVT444 must be mounted. The configuration used in this example (Figure 58) allows the simultaneous mounting of all seven volumes (Figure 65) involved in this generation. Therefore, to simplify operating procedures, all seven volumes should be mounted at this point. It is assumed that there is no dependency on device addresses, and that the volumes can be mounted on any 2314 drive. For illustrative purposes, assume that volumes DLIB01, MVT111, and MVT555 are mounted on drives 130, 131, and 132, and that volumes MVT222, MVT333, MVT444, and DLIB02 are mounted on drives 230, 231, 232, and 233, respectively.

```

//SYSGEN JOB MSGLEVEL=1 -EXAMPLE 2-
//STEP EXEC PGM=IEHPROGM -ALLOCATE ON 2314-
//SYSPRINT DD SYSOUT=A
//SVCLIB DD DSN=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=MVT111), X
// UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(7,1,75)), X
// LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//CATALOG DD DSN=SYS1.SYCTLG,VOLUME=(,RETAIN,SER=MVT111), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(6,1)), X
// LABEL=EXPDT=99350
//NUCLEUS DD DSN=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=MVT111), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(20,,1),,CONTIG), X
// LABEL=EXPDT=99350
//LINKLIB DD DSN=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=MVT222), X
// UNIT=2314,DISP=(,KEEP),SPACE=(CYL,(43,5,100)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//LOGX DD DSN=SYS1.SYSVLOGX,VOLUME=(,RETAIN,SER=MVT222), X
// UNIT=2314,DISP=(,KEEP),SPACE=(120,(100),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=VB,BLKSIZE=3625)
//LOGY DD DSN=SYS1.SYSVLOGY,VOLUME=(,RETAIN,SER=MVT222), X
// UNIT=2314,DISP=(,KEEP),SPACE=(120,(100),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=VB,BLKSIZE=3625)
//JOBQE DD DSN=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=MVT333), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(200),,CONTIG)
//PROCLIB DD DSN=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(20,6,9)), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//PARMLIB DD DSN=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=MVT111), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(5,,1),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//MACLIB DD DSN=SYS1.MACLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(300,30,25)), X
// LABEL=EXPDT=99350,DCB=(RECFM=FB,BLKSIZE=7280,LRECL=80)
//TELCLIB DD DSN=SYS1.TELCLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(30,2,10)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//SORTLIB DD DSN=SYS1.SORTLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(40,1,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//PL1LIB DD DSN=SYS1.PL1LIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(55,2,65)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//FORTLIB DD DSN=SYS1.FORTLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(25,1,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//ALGLIB DD DSN=SYS1.ALGLIB,VOLUME=(,RETAIN,SER=MVT444), X
// UNIT=2314,DISP=(,KEEP),SPACE=(TRK,(26,2,15)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=7294)
//SYSIN DD * -INPUT FOR CATALOGING SYSTEM DATA SETS-
CATLG CVOL=2314=MVT111,VOL=2314=MVT222,DSN=SYS1.LINKLIB
CATLG CVOL=2314=MVT111,VOL=2314=MVT111,DSN=SYS1.SVCLIB
CATLG CVOL=2314=MVT111,VOL=2314=MVT333,DSN=SYS1.SYSJOBQE
CATLG CVOL=2311=MVT111,VOL=2314=MVT444,DSN=SYS1.PROCLIB
CATLG CVOL=2314=MVT111,VOL=2314=MVT111,DSN=SYS1.PARMLIB
CATLG CVOL=2314=MVT111,VOL=2314=MVT444,DSN=SYS1.MACLIB
CATLG CVOL=2314=MVT111,VOL=2314=MVT444,DSN=SYS1.TELCLIB

```

Figure 66 (Part 1 of 2). Example 2: Initializing New System Data Sets

---

```

CATLG   CVOL=2314=MVT111,VOL=2314=MVT444,DSNAME=SYS1.SORTLIB
CATLG   CVOL=2314=MVT111,VOL=2314=MVT444,DSNAME=SYS1.PL1LIB
CATLG   CVOL=2314=MVT111,VOL=2314=MVT444,DSNAME=SYS1.FORTLIB
CATLG   CVOL=2314=MVT111,VOL=2314=MVT444,DSNAME=SYS1.ALGLIB
CATLG   CVOL=2314=MVT111,VOL=2314=MVT222,DSNAME=SYS1.SYSVLOGX
CATLG   CVOL=2314=MVT111,VOL=2314=MVT222,DSNAME=SYS1.SYSVLOGY

```

---

/\*

Figure 66 (Part 2 of 2). Example 2: Initializing New System Data Sets

Input Deck for Stage I

Figure 67 shows a listing of the input deck used for system generation. The utility data sets are allocated space on the volumes as indicated in Figure 65. The job stream is written on an unlabeled magnetic tape that resides on drive 182 (See Figure 58). Stage II will be started automatically after Stage I is completed. Unit names are used.

The new system supports all devices shown in Figure 58. Multiprogramming with a variable number of tasks (MVT) is specified. All access methods and optional system data sets are generated. Assembler F, linkage-editor F44, Sort/Merge, ALGOL, FORTRAN G, PL/I F, and RPG are included in the new system. All standard default options are selected for the processors.

After Stage II has terminated, SYS1.SG1, SYS1.SG2, SYS1.SG3, and SYS1.SG4 can be scratched and uncataloged. The job stream and SYS1.SGOBJ should be saved. DLIB01 and DLIB02 can be removed. A backup copy of the new system should be made. The new system is ready to operate after initial program load (IPL).

---

```

//STEP1 EXEC PGM=ASMBLR -STAGE I INPUT-
//SYSLIB DD DSNAME=SYS1.GENLIB,DISP=OLD
//OBJPDS DD DSNAME=SYS1.SGOBJ,VOLUME=(,RETAIN,SER=MVT555), X
// DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(45,5,12))
//SYSUT1 DD DSNAME=SYS1.SG1,VOLUME=(,RETAIN,SER=MVT111), X
// DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(240,5))
//SYSUT2 DD DSNAME=SYS1.SG2,VOLUME=(,RETAIN,SER=MVT222), X
// DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(160,10))
//SYSUT3 DD DSNAME=SYS1.SG3,VOLUME=(,RETAIN,SER=MVT333), X
// DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(230,5))
//SYSUT4 DD DSNAME=SYS1.SG4,VOLUME=(,RETAIN,SER=MVT333), X
// DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(3,3))
//SYSPUNCH DD UNIT=182,LABEL=(,NL)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
CENPROCS MODEL=50,STORAGE=1,FEATURE=PROTECT
MPX CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CONSOLE IODEVICE UNIT=1052,MODEL=7,ADDRESS=009
CNT00 IOCONTRL UNIT=2821,MODEL=5,ADDRESS=00,FEATURE=COLBNRY, X
 TRNMODE=BYTE
DEV00C IODEVICE UNIT=2540R,MODEL=1,ADDRESS=00C
DEV00D IODEVICE UNIT=2540P,MODEL=1,ADDRESS=00D
DEV00E IODEVICE UNIT=1403,MODEL=N1,ADDRESS=00E

```

---

Figure 67 (Part 1 of 4). Example 2: Input Deck for Stage I

DEV00F	IODEVICE	UNIT=1403,MODEL=N1,ADDRESS=00F	
CNT01	IOCONTRL	UNIT=2821,MODEL=5,ADDRESS=01,FEATURE=COLBNRY, TRNMODE=BYTE	X
DEV01A	IODEVICE	UNIT=2540R,MODEL=1,ADDRESS=01A	
DEV01B	IODEVICE	UNIT=2540P,MODEL=1,ADDRESS=01B	
DEV01F	IODEVICE	UNIT=1403,MODEL=N1,ADDRESS=01F	
CNT02	IOCONTRL	UNIT=2702,ADDRESS=02	
LINE020	IODEVICE	UNIT=1030,ADDRESS=020,ADAPTER=IBM2,SETADDR=3, FEATURE=AUTOPOLL	X
LINE021	IODEVICE	UNIT=1050,ADDRESS=021,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE022	IODEVICE	UNIT=1050,ADDRESS=022,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE023	IODEVICE	UNIT=1060,ADDRESS=023,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE024	IODEVICE	UNIT=83B3,ADDRESS=024,ADAPTER=TELE1,SETADDR=3	
LINE025	IODEVICE	UNIT=115A,ADDRESS=025,ADAPTER=TELE1,SETADDR=3	
LINE026	IODEVICE	UNIT=83B3,ADDRESS=026,ADAPTER=TELE1,SETADDR=3	
LINE027	IODEVICE	UNIT=1050,ADDRESS=027,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE028	IODEVICE	UNIT=1060,ADDRESS=028,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE029	IODEVICE	UNIT=115A,ADDRESS=029,ADAPTER=TELE1,SETADDR=3	
LINE02A	IODEVICE	UNIT=83B3,ADDRESS=02A,ADAPTER=TELE1,SETADDR=3	
LINE02B	IODEVICE	UNIT=1060,ADDRESS=028,ADAPTER=IBM1,SETADDR=1, FEATURE=AUTOPOLL	X
LINE02C	IODEVICE	UNIT=115A,ADDRESS=02C,ADAPTER=TELE1,SETADDR=3	
LINE02D	IODEVICE	UNIT=1030,ADDRESS=02D,ADAPTER=IBM2,SETADDR=2, FEATURE=AUTOPOLL	X
LINE02E	IODEVICE	UNIT=1030,ADDRESS=02E,ADAPTER=IBM2,SETADDR=2, FEATURE=AUTOPOLL	X
CH1	CHANNEL	ADDRESS=1,TYPE=SELECTOR	
DEV130	IODEVICE	UNIT=2314,ADDRESS=130	
CNT18	IOCONTRL	UNIT=2803,MODEL=1,ADDRESS=18, FEATURE=(DATA CONV,7-TRACK)	X
DEV180	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=(180,2), FEATURE=7-TRACK	X
DEV182	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=(182,2), FEATURE=9-TRACK	X
CNT19	IOCONTRL	UNIT=2841,ADDRESS=19	
DEV190	IODEVICE	UNIT=2311,ADDRESS=(190,3)	
DEV193	IODEVICE	UNIT=2303,ADDRESS=193	
CH2	CHANNEL	ADDRESS=2,TYPE=SELECTOR	
DEV230	IODEVICE	UNIT=2314,ADDRESS=230	
CNT28	IOCONTRL	UNIT=2803,MODEL=1,ADDRESS=28, FEATURE=(DATA CONV,7-TRACK)	X
DEVX80	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=280,FEATURE=7-TRACK, OPTCHAN=3	X
DEVX81	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=281,FEATURE=7-TRACK OPTCHAN=3	X
DEVX82	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=282,FEATURE=9-TRACK, OPTCHAN=3	X
DEVX83	IODEVICE	UNIT=2402,MODEL=3,ADDRESS=283,FEATURE=9-TRACK, OPTCHAN=3	X
DEVX84	IODEVICE	UNIT=2401,MODEL=3,ADDRESS=284,FEATURE=9-TRACK, OPTCHAN=3	X
DEVX85	IODEVICE	UNIT=2401,MODEL=3,ADDRESS=285,FEATURE=9-TRACK, OPTCHAN=3	X

Figure 67 (Part 2 of 4). Example 2: Input Deck for Stage I



---

DEVX86	IODEVICE UNIT=2401,MODEL=3,ADDRESS=286,FEATURE=9-TRACK, OPTCHAN=3	X
DEVX87	IODEVICE UNIT=2401,MODEL=3,ADDRESS=287,FEATURE=9-TRACK, OPTCHAN=3	X
CNT29	IOCONTRL UNIT=2841,ADDRESS=29	
CNT291	IODEVICE UNIT=2311,ADDRESS=(291,4)	
CNT2F	IOCONTRL UNIT=2848,MODEL=3,ADDRESS=2F,FEATURE=NODESCUR	
DEV2F0	IODEVICE UNIT=2260,MODEL=1,ADDRESS=2F0,FEATURE=ALKYB2260	
DEV2F1	IODEVICE UNIT=2260,MODEL=1,ADDRESS=2F1,FEATURE=NMKYB2260	
DEV2F4	IODEVICE UNIT=1053,MODEL=4,ADDRESS=2F4	
CH3	CHANNEL ADDRESS=3,TYPE=SELECTOR	
CNT38	IOCONTRL UNIT=2803,MODEL=1,ADDRESS=38, FEATURE=(DATA CONV,7-TRACK)	X
CNT32	IOCONTRL UNIT=2841,ADDRESS=32	
DEV320	IODEVICE UNIT=2302,MODEL=3,ADDRESS=320	
DEV321	IODEVICE UNIT=2302,MODEL=3,ADDRESS=321	
CNT35	IOCONTRL UNIT=2701,ADDRESS=35	
DEV351	IODEVICE UNIT=2260,MODEL=1,FEATURE=ALKYB2260,ADAPTER=IBM3, ADDRESS=351	X
DEV355	IODEVICE UNIT=2260,MODEL=1,FEATURE=ALKYB2260,ADAPTER=IBM3, ADDRESS=355	X
CNT3A	IOCONTRL UNIT=2840,MODEL=1,ADDRESS=3A	
DEV3A1	IODEVICE UNIT=2250,MODEL=2,ADDRESS=3A1,NUMSECT=5, FEATURE=(ALKYB2250,LIGHTPEN,PRGMKYBD)	X
DEV3A2	IODEVICE UNIT=2250,MODEL=2,ADDRESS=3A2,NUMSECT=5, FEATURE=(ALKYB2250,LIGHTPEN,PRGMKYBD)	X
DEV3E0	IODEVICE UNIT=2250,MODEL=1,ADDRESS=2E0, FEATURE=(ALKYB2250,BUFFER8K,LIGHTPEN,DESIGNFEAT)	X
	UNITNAME NAME=SYSSQ,UNIT=((180,4),(280,4),(291,3))	
	UNITNAME NAME=SYSDA,UNIT=((190,4),(291,4),(235,3))	
	UNITNAME NAME=SYSCP,UNIT=00D	
	UNITNAME NAME=TAPE,UNIT=((282,6))	
	UNITNAME NAME=RECORDS,UNIT=((320,2))	
	UNITNAME NAME=FILE,UNIT=((134,4),(234,4))	
	CTRLPROG TYPE=MVT,MAXIO=61,QSPACE=20,ADDTRAN=4	
	SCHEDULR TYPE=MVT,CONSOLE=009,ALTCONS=(I-00D,O=00F),	X
	STARTR=A-00D,STARTW=A-00E,	X
	ACCTRTN=BASIC,STARTI=AUTO,WTLBFRS=10,	X
	PROCRES=232,JOBQRES=231,INITQBF=10,MINPART=54	
	SUPRVSR RESIDENT=(BLDLTAB,RENTCODE,TR SVC),TRACE=100,	X
	OPTIONS=(PROTECT,COMM),TIMER=JOBSTEP,SER=SER1	
	PROCLIB UNIT=2314,VOLNO=MVT444	
	PARMLIB UNIT=2314,VOLNO=MVT111	
	DATAMGT ACSMETH=(BDAM,ISAM,BTAM,QTAM)	
	TELCMLIB UNIT=2314,VOLNO=MVT444	
	GRAPHICS	
	EDITOR DESIGN=F44	
	ASSEMBLR DESIGN=F	
	MACLIB UNIT=2314,VOLNO=MVT444,EXCLUDE=(OCR,TSO)	
	CKPTREST ELIGBLE=(5,6,7,8)	
	SORTMERG SIZE=24000,SORTOPT=FULLIB	
	SORTLIB UNIT=2314,VOLNO=MVT444	

---

Figure 67 (Part 3 of 4). Example 2: Input Deck for Stage I

```

ALGOL
ALGLIB      UNIT=2314,VOLNO=MVT444
FORTRAN     DESIGN=G
FORTLIB     DESIGN=G,UNIT=2314,VOLNO=MVT444
PL1         DESIGN=F
PL1LIB      UNIT=2314,VOLNO=MVT444,LIBFCNS=COMPLEX
GENERATE    UT1SDS=SYS1.SG1,UT2SDS=SYS1.SG2,UT3SDS=SYS1.SG3, X
           UT4SDS=SYS1.SG4,OBJPDS=SYS1.SGOBJ,RESNAME=2314, X
           RESVOL=MVT111,RESTYPE=2314,LNKNAME=2314,222, X
           LBMAINT=E,LNKVOL=MVT222,LBMAINT=E,ASMPRT=ON, X
           DIRDATA=PDS,LEPRT=(LIST,XREF)
END

/*
// START   RDR,182

```

Figure 67 (Part 4 of 4). Example 2: Input Deck for Stage I

Decks for System Residence on 2303

The 2303 unit at address 197 (See Figure 58) can be used for system residence instead of volume MVT111. The IBCDASDI deck listing shown in Figure 68 replaces the deck listing for MVT111 shown in Figure 64. The serial number of the 2303 volume is SYSRES.

```

SYSRES JOB
MSG      TODEV=1403,TOADDR=00E
DADEF    TODEV=2303,TOADDR=197,IPL=YES,VOLID=SCRATCH
VLD      NEWVOLID=SYSRES,OWNERID=DEPTD58
VTOCD    STRTADR=2,EXTENT=8
IPLTXT

.
. IEAIPL00 cards
.

END

```

Figure 68. Example 2: IBCDASDI Deck for 2303

The DD statements in Figure 69 must replace the DD statements for SYS1.SVCLIB, SYSCTLG, and SYS1.NUCLEUS in Figure 66. The CATLG statements replace all the CATLG statements in Figure 66.

Only two changes must be made to the Stage I input deck in Figure 67:

- The SYS1.SG1 utility data set must be allocated space on SYSRES rather than on MVT111 as follows:

```

//SYSUT1 DD DSNAME=SYS1.SG1,VOLUME=(,RETAIN,SER=SYSRES), X
//          DISP=(,CATLG),UNIT=2303,SPACE=(TRK,(180,16))

```

- The RESNAME, RESVOL, and RESTYPE parameters of the GENERATE macro instruction must be coded as follows:

```
RESNAME=2303,RESVOL=SYSRES,RESTYPE=2303
```

---

```

.
.
//SVCLIB DD DSNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=SYSRES), X
// UNIT=2303,DISP=(,KEEP),SPACE=(TRK,(120,10,75)), X
// LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//CATALOG DD DSNAME=SYSCTLG,VOLUME=(,RETAIN,SER=SYSRES), X
// UNIT=2303,DISP=(,KEEP),SPACE=(TRK,(8,1)), X
// LABEL=EXPDT=99350
//NUCLEUS DD DSNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=SYSRES), X
// UNIT=2303,DISP=(,KEEP),SPACE=(TRK,(25,,1),,CONTIG), X
// LABEL=EXPDT=99350
.
.
//SYSIN DD * -INPUT FOR CATALOGING SYSTEM DATA SETS-
CATIG CVOL=2303=SYSRES,VOL=2314=MVT333,DSNAME=SYS1.SYSJOBQE
CATLG CVOL=2303=SYSRES,VOL=2314=MVT222,DSNAME=SYS1.LINKLIB
CATLG CVOL=2303=SYSRES,VOL=2303=SYSRES,DSNAME=SYS1.SVCLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.PROCLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.PARMLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.MACLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.TELCMLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.SORTLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.PL1LIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.FORTLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT444,DSNAME=SYS1.ALGLIB
CATLG CVOL=2303=SYSRES,VOL=2314=MVT222,DSNAME=SYS1.SYSVLOGX
CATLG CVOL=2303=SYSRES,VOL=2314=MVT222,DSNAME=SYS1.SYSVLOGY
/*

```

---

Figure 69. Example 2: Allocation on 2303

#### Input Deck for I/O Device Generation

During I/O device generation, the arrangement of the communications terminals is changed, a 2803 and 2302 are deleted, and a 1052, 1403, 2314, 2841, and 2321 are added to the system. The device, channel, and addresses are changed and new groups of devices are named.

The CENPROCS, CTRLPROG, SUPRVSOR, DATAMGT, and GRAPHICS, macro instructions included in the complete operating system generation must be included in this generation. The CONSOLE and ALTCONS addresses are changed in the SCHEDULR macro instruction, but the STARTR, STARTW, PROCRES, and JOBQRES addresses cannot be changed until IPL time. The UNITNAME macro instruction is used to rename and give new names to groups of devices.

Figure 70 shows a listing of the input deck used for the I/O device generation. There is no allocation step before Stage I because there should be enough space available on the affected libraries, SYS1.NUCLEUS, SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.LOGREC. Because in an I/O device generation the system being modified cannot be the generating system, the starter operating system is used. DLIB01 and DLIB02 must, therefore, be mounted before the generation starts.

The job stream will be written on an unlabeled tape on drive 183. Stage II will be started automatically after Stage I is completed. SYS1.SG1 and SYS1.SG2 are

defined on volumes MVT111 and MVT222, respectively, whereas SYS1.SG3 and SYS1.SG4 are defined on volume MVT333. A partitioned data set named SYS1.SGOBJ is allocated on volume MVT555. SYS1.SG1, SYS1.SG2, SYS1.SG3, and SYS1.SG4 will be deleted after they are no longer needed in Stage II.

```

//SYSGEN      JOB MSGLEVEL=1                      -I/O DEVICE GENERATION-
//STEP1       EXEC PGM=ASMBLR                      -STAGE 1 INPUT-
//SYSLIB      DD DSN=SYS1.GENLIB,DISP=OLD
//OBJPDS      DD DSN=SYS1.SGOBJ,DISP=OLD
//SYSUT1      DD DSN=SYS1.SG1,VOLUME=(,RETAIN,SER=MVT111),      X
//              DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(240,5))
//SYSUT2      DD DSN=SYS1.SG2,VOLUME=(,RETAIN,SER=MVT222),      X
//              DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(160,10))
//SYSUT3      DD DSN=SYS1.SG3,VOLUME=(,RETAIN,SER=MVT333),      X
//              DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(280,5))
//SYSUT4      DD DSN=SYS1.SG4,VOLUME=(,RETAIN,SER=MVT333),      X
//              DISP=(,CATLG),UNIT=2314,SPACE=(TRK,(3,5))
//SYSPUNCH    DD UNIT=183,LABEL=(,NL)
//SYSPRINT    DD SYSOUT=A
//SYSIN       DD *
              CENPROCS MODEL=50,STORAGE=I,FEATURE=PROTECT
MPX           CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CON00A        IODEVICE UNIT=1052,MODEL=7,ADDRESS=00A
CON01A        IODEVICE UNIT=1052,MODEL=7,ADDRESS=01A
CNT00         IOCONTRL UNIT=2821,MODEL=5,ADDRESS=00,FEATURE=COLBNRY
DEV00C        IODEVICE UNIT=2540R,MODEL=1,ADDRESS=00C
DEV00D        IODEVICE UNIT=2540P,MODEL=1,ADDRESS=00D
DEV00E        IODEVICE UNIT=1403,MODEL=N1,ADDRESS=00E
DEV00F        IODEVICE UNIT=1403,MODEL=N1,ADDRESS=00F
CNT01         IOCONTRL UNIT=2821,MODEL=5,ADDRESS=01,FEATURE=COLBNRY
DEV01C        IODEVICE UNIT=2540R,MODEL=1,ADDRESS=01C
DEV01D        IODEVICE UNIT=2540P,MODEL=1,ADDRESS=01D
DEV01E        IODEVICE UNIT=1403,MODEL=N1,ADDRESS=01E
DEV01F        IODEVICE UNIT=1403,MODEL=N1,ADDRESS=01F
CNT02         IOCONTRL UNIT=2702,ADDRESS=02
LINE020       IODEVICE UNIT=1030,ADDRESS=020,ADAPTER=IBM2,SETADDR=3,      X
              FEATURE=AUTOPOLL
LINE021       IODEVICE UNIT=1050,ADDRESS=021,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE022       IODEVICE UNIT=1060,ADDRESS=022,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE023       IODEVICE UNIT=1050,ADDRESS=023,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE024       IODEVICE UNIT=1060,ADDRESS=024,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE025       IODEVICE UNIT=1030,ADDRESS=025,ADAPTER=IBM2,SETADDR=3,      X
              FEATURE=AUTOPOLL
LINE026       IODEVICE UNIT=1050,ADDRESS=026,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE027       IODEVICE UNIT=1060,ADDRESS=027,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE028       IODEVICE UNIT=1050,ADDRESS=028,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
LINE029       IODEVICE UNIT=1050,ADDRESS=029,ADAPTER=IBM1,SETADDR=1,      X
              FEATURE=AUTOPOLL
CH1           CHANNEL ADDRESS=1,TYPE=SELECTOR
DEV130        IODEVICE UNIT=2314,ADDRESS=130
DEV140        IODEVICE UNIT=2314,ADDRESS=140
CNT18         IOCONTRL UNIT=2803,MODEL=1,ADDRESS=18,              X
              FEATURE=(DATACONV,7-TRACK)

```

Figure 70 (Part 1 of 3). Example 2: Input for I/O Device Generation

```

DEV180  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=180,FEATURE=7-TRACK
DEV181  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=181,FEATURE=7-TRACK
DEV182  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=182,FEATURE=9-TRACK
DEV183  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=183,FEATURE=9-TRACK
CNT19   IOCONTRL  UNIT=2841,ADDRESS=19
DEV190  IODEVICE  UNIT=2311,ADDRESS=(190,3)
DEV193  IODEVICE  UNIT=2303,ADDRESS=193
CNT1A   IOCONTRL  UNIT=2841,ADDRESS=1A
DEV1A0  IODEVICE  UNIT=2321,MODEL=1,ADDRESS=1A0
CH2     CHANNEL  ADDRESS=2,TYPE=SELECTOR
DEV230  IODEVICE  UNIT=2314,ADDRESS=230
CNT28   IOCONTRL  UNIT=2803,MODEL=1,ADDRESS=28,          X
          FEATURE=(DATA CONV,7-TRACK)
DEVX80  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=(280,2),        X
          FEATURE=7-TRACK,OPTCHAN=3
DEVX82  IODEVICE  UNIT=2402,MODEL=3,ADDRESS=(282,2),        X
          FEATURE=9-TRACK,OPTCHAN=3
DEVX84  IODEVICE  UNIT=2401,MODEL=3,ADDRESS=(284,4),        X
          FEATURE=9-TRACK,OPTCHAN=3
CNT29   IOCONTRL  UNIT=2841,ADDRESS=29
DEV291  IODEVICE  UNIT=2311,ADDRESS=(291,4)
CNT2F   IOCONTRL  UNIT=2848,MODEL=3,ADDRESS=2F,FEATURE=NODESCUR
DEV2F0  IODEVICE  UNIT=2260,MODEL=1,ADDRESS=2F0,FEATURE=ALKYB2260
DEV2F1  IODEVICE  UNIT=2260,MODEL=1,ADDRESS=2F1,FEATURE=NMKYB2260
DEV2F4  IODEVICE  UNIT=1053,MODEL=4,ADDRESS=2F4
CH3     CHANNEL  ADDRESS=3,TYPE=SELECTOR
CNT32   IOCONTRL  UNIT=2841,ADDRESS=32
DEV320  IODEVICE  UNIT=2311,ADDRESS=(320,4)
DEV324  IODEVICE  UNIT=2302,MODEL=3,ADDRESS=324
CNT35   IOCONTRL  UNIT=2701,ADDRESS=35
DEV351  IODEVICE  UNIT=2260,MODEL=1,FEATURE=ALKYB2260,      X
          ADAPTER=IBM3,ADDRESS=351
DEV355  IODEVICE  UNIT=2260,MODEL=1,FEATURE=ALKYB2260,      X
          ADAPTER=IBM3,ADDRESS=355
CNT3A   IOCONTRL  UNIT=2840,MODEL=1,ADDRESS=3A
DEV3A1  IODEVICE  UNIT=2250,MODEL=2,ADDRESS=(3A1,2) NUMSECT=5,  X
          FEATURE=(ALKYB2250,LIGHTPEN,PRGMKYBD)
DEV3A3  IODEVICE  UNIT=2250,MODEL=1,ADDRESS=(3A3,2),        X
          FEATURE=(ALKYB2250,BUFFER8K,LIGHTPEN,DESIGNFEAT)
UNITNAME NAME=SYSSQ,UNIT=((180,4),(280,4))
UNITNAME NAME=SYSDA,UNIT=((190,4),(291,4))
UNITNAME NAME=SYSCP,UNIT=00D
UNITNAME NAME=TAPE,UNIT=(284,4)
UNITNAME NAME=DISK,UNIT=(320,4)
UNITNAME NAME=RECORDS,UNIT=324
UNITNAME NAME=FILE,UNIT=1A0
UNITNAME NAME=SYSCR,UNIT=00C
UNITNAME NAME=PRINT,UNIT=(00E,1)
UNITNAME NAME=DATA,UNIT=(130,140,230)
CTRLPROG TYPE=MVT,MAXIO=61,QSAPCE=20,ADDTRAN=4
SCHEDULR TYPE=MVT,CONSOLE=(I=00A,O=01A),          X
          ALTCONS=(I-00D,O-00F,I-01D,O-01F),          X
          STARTR=A-00D,STARTW=A-00E,                  X
          ACCTRTN=BASIC,STARTI=AUTO,MINPART=54,        X
          WTLBFRS=10,PROGRES=232,JOBQRES=231,INITQBF=10
SUPRVSR  RESIDNT=(BLDLTAB,RENTCODE,TR SVC),TRACE=100,      X
          OPTIONS=(PROTECT,COMM),TIMER=JOBSTEP,SER=SER1
DATAMGT  ACSMETH=(BDAM,ISAM,BTAM,QTAM)
GRAPHICS

```

Figure 70 (Part 2 of 3). Example 2: Input for I/O Device Generation

---

```
GENERATE UT1SDS=SYS1.SG1,UT2SDS=SYS1.SG2,UT3SDS=SYS1.SG3, X
         UT4SDS=SYS1.SG4,OBJPDS=SYS1.SGOBJ,GENTYPE=(IO,1), X
         RESNAME=2314,RESTYPE=2314,LNKNAME=2314, X
         LNKVOL=MVT222,LBMAINT=E,ASMPRT=ON,DIRDATA=PDS, X
         LEPRT=(LIST,XREF),RESVOL=MVT111
END
```

```
/*
// START RDR,183
```

---

Figure 70 (Part 3 of 3). Example 2: Input for I/O Device Generation

### Example 3

This example illustrates and describes:

- Volumes that contain generating and new system data sets
- A deck listing used for initializing new system data sets
- A deck listing used for system generation
- A method for creating backup of a new system
- A method for scratching utility data sets

#### Volumes Used for System Generation

Figure 71 shows the volumes that contain the generating and new system data sets. The serial number of the generating system-residence volume is 111111. Volumes DLIB01, DLIB02, DLIB03, DLIB04, DLIB05, and DLIB06 contain the remaining generating system data sets. The serial number of the new system-residence volume is SYSTEM. It is assumed that SYSTEM has been previously initialized. The five utility data sets used during system generation are named SYS1.OBJECT, SYS1.ONE, SYS1.TWO, SYS1.THREE, and SYS1.FOUR. The SYS1.USER data sets contain user-written routines for the new system.

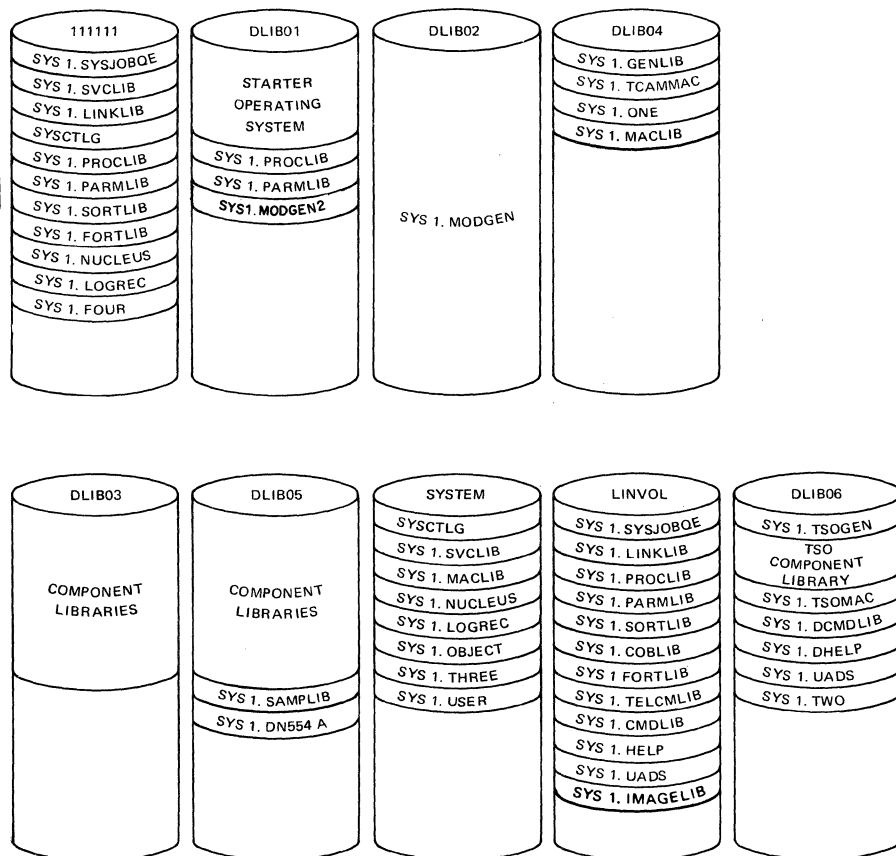


Figure 71. Example 3: Generating and New System Volumes

Deck for Initializing New System Data Sets

Figure 72 shows a listing of the input deck used for initializing the new system data sets. During this initialization, volumes 111111, SYSTEM, and LINVOL must be mounted. It is assumed that there is no dependency on device addresses, and that the volumes can be mounted on any 2311 drive. For illustrative purposes, assume that volumes 111111, DLIB01, DLIB03, DLIB04, SYSTEM, and LINVOL are mounted on drives 161, 162, 163, 251,252, and 253, respectively.

```
//SYSGEN JOB MSGLEVEL=1 -EXAMPLE 3-
//STEP0 EXEC PGM=IEHPROGM -ALLOCATE ON 2311-
//SYSPRINT DD SYSOUT=A
//CATALOG DD DSNAME=SYSCTLG,VOLUME=(RETAIN,SER=SYSTEM), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,=10,1), X
// LABEL=EXPDT=99350
//SVCLIB DD DSNAME=SYS1.SVCLIB,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(15,1,75)), X
// LABEL=EXPDT=99350,DCB=(DSORG=POU,RECFM=U,BLKSIZE=1024)
//MACLIB DD DSNAME=SYS1.MACLIB,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(440,50,25)), X
// LABEL=EXPDT=99350,DCB=(RECFM=FB,BLKSIZE=3360,LRECL=80)
//NUCLEUS DD DSNAME=SYS1.NUCLEUS,VOLUME=(,RETAIN,SER=SYSTEM), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(60,,2),,CONTIG), X
// LABEL=EXPDT=99350
//JOBQE DD DSNAME=SYS1.SYSJOBQE,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(120),,CONTIG)
//LINKLIB DD DSNAME=SYS1.LINKLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(CYL,(10,10,100)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//PROCLIB DD DSNAME=SYS1.PROCLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(30,10,9)), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//PARMLIB DD DSNAME=SYS1.PARMLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(5,,1),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=F,BLKSIZE=80)
//SORTLIB DD DSNAME=SYS1.SORTLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,=60,2,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//FORTLIB DD DSNAME=SYS1.FORTLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(50,2,40)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//IMAGELIB DD DSNAME=SYS1.IMAGELIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(5,,3),,CONTIG), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=1024)
//HELP DD DSNAME=SYS1.HELP,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(35,2,30)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=800)
//UADS DD DSNAME=SYS1.UADS,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(35,2,30)), X
// LABEL=EXPDT=99350,DCB=(DSORG=PD,LRECL=80,RECFM=FB)
//CMDLIB DD DSNAME=SYS1.CMDLIB,VOLUME=(,RETAIN,SER=LINVOL), X
// UNIT=2311,DISP=(,KEEP),SPACE=(TRK,(35,2,30)), X
// LABEL=EXPDT=99350,DCB=(RECFM=U,BLKSIZE=3625)
//SYSIN DD * -INPUT FOR CATALOGING SYSTEM DATA SETS
UNCATLG DSNAME=SYS1.PROCLIB
UNCATLG DSNAME=SYS1.PARMLIB
UNCATLG DSNAME=SYS1.SORTLIB
UNCATLG DSNAME=SYS1.FORTLIB
```

Figure 72 (Part 1 of 2). Example 3: Initializing New System Data Sets



---

```

CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.LINKLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.PROCLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.PARMLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=SYSTEM,DSNAME=SYS1.MACLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.SORTLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.FORTLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=SYSTEM,DSNAME=SYS1.SVCLIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.HELP
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.IMAGELIB
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.UADS
CATLG CVOL=2311=SYSTEM,VOL=2311=LINVOL,DSNAME=SYS1.CMDLIB
CATLG VOL=2311=DLIB05,DSNAME=SYS1.AS037 ASSEMBLER F
CATLG VOL=2311=DLIB03,DSNAME=SYS1.DM509 BDAM
CATLG VOL=2311=DLIB03,DSNAME=SYS1.CI505 CONTROL PROGRAM
CATLG VOL=2311=DLIB03,DSNAME=SYS1.DM508 DATA MANAGEMENT
CATLG VOL=2311=DLIB03,DSNAME=SYS1.UT506 UTILITIES
CATLG VOL=2311=DLIB05,DSNAME=SYS1.FO520 FORTRAN G
CATLG VOL=2311=DLIB05,DSNAME=SYS1.LM501 FORTRAN LIBRARY
CATLG VOL=2311=DLIB03,DSNAME=SYS1.IO526 ISAM
CATLG VOL=2311=DLIB05,DSNAME=SYS1.ED521 LINKAGE EDITOR F
CATLG VOL=2311=DLIB05,DSNAME=SYS1.LD547 LOADER
CATLG VOL=2311=DLIB05,DSNAME=SYS1.RG038 RPG
CATLG VOL=2311=DLIB05,DSNAME=SYS1.SM023 SORT/MERGE
CATLG VOL=2311=DLIB03,DSNAME=SYS1.CQ548 TCAM
CATLG VOL=2311=DLIB06,DSNAME=SYS1.CI555 TSO
CATLG VOL=2311=DLIB06,DSNAME=SYS1.DHELP TSO
CATLG VOL=2311=DLIB06,DSNAME=SYS1.DCMDLIB TSO
CATLG VOL=2311=DLIB06,DSNAME=SYS1.DUADS TSO
CATLG VOL=2311=DLIB01,DSNAME=SYS1.PROCLIB
CATLG VOL=2311=DLIB01,DSNAME=SYS1.PARMLIB
CATLG VOL=2311=DLIB05,DSNAME=SYS1.SORTLIB
CATLG VOL=2311=DLIB05,DSNAME=SYS1.FORTLIB

```

---

/\*

Figure 72 (Part 2 of 2). Example 3: Initializing New System Data Sets

### Input Deck for Stage I

Figure 73 shows a listing of the input deck used for system generation. The utility data sets are allocated space on the volumes indicated in Figure 71. The job stream will be written on an unlabeled magnetic tape that resides on drive 190. Unit names are used. This input deck is the second step of the job defined in Figure 72.

For Stage I of the system generation process the volumes that are mounted are: 111111, DLIB01, DLIB02, DLIB04, SYSTEM, and LINVOL. Stage II will be started automatically after Stage I is completed. When it is required, the scheduler will request the demounting of DLIB02 and the mounting of DLIB06, the demounting of DLIB06 and the mounting of DLIB03, and the demounting of DLIB03 and the mounting of DLIB05.

BDAM, BISAM, QISAM, TCAM, Assembler F, FORTRAN G, HELP, UADS, Sort/Merge, and RPG are included in the new system. The compilers have all standard default options for compilation time. The procedure library and the unit names it requires are also included. A user-written routine named NUCID is included in the nucleus. NUCID is a member of the SYS1.USER data set.

```

//STEP1 EXEC PGM=ASMBLR -STAGE I INPUT-
//SYSLIB DD DSNAME=SYS1.GENLIB,DISP=OLD
//OBJPDS DD DSNAME=SYS1.MOD,VOLUME=(,RETAIN,SER=SYSTEM), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(60,10,12))
//SYSUT1 DD DSNAME=SYS1.ONE,VOLUME=(,RETAIN,SER=DLIB04), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(350,10))
//SYSUT2 DD DSNAME=SYS1.TWO,VOLUME=(,RETAIN,SER=DLIB02), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(240,20))
//SYSUT3 DD DSNAME=SYS1.THREE,VOLUME=(,RETAIN,SER=SYSTEM), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(400,10))
//SYSUT4 DD DSNAME=SYS1.FOUR,VOLUME=(,RETAIN,SER=111111), X
// DISP=(,CATLG),UNIT=2311,SPACE=(TRK,(5,10))
//SYSPUNCH DD UNIT=190,LABEL=(,NL)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
CENPROCS MODEL=65,STORAGE=IH
CHAN0 CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR
CONSOLE IODEVICE UNIT=1052,MODEL=7,ADDRESS=010
CONT00 IOCONTRL UNIT=2821,MODEL=5,ADDRESS=00
READ00A IODEVICE UNIT=2540R,MODEL=1,ADDRESS=00A
PUNCH00B IODEVICE UNIT=2540P,MODEL=1,ADDRESS=00B
PRINT00C IODEVICE UNIT=1403,MODEL=3,ADDRESS=00C,FEATURE=UNVCHSET
PRINT00D IODEVICE UNIT=1403,MODEL=3,ADDRESS=00D,FEATURE=UNVCHSET
CONT01 IOCONTRL UNIT=3811,ADDRESS=00
PRINT00F IODEVICE UNIT=3211,ADDRESS=00F
CHAN1 CHANNEL ADDRESS=1,TYPE=SELECTOR
CONT16 IOCONTRL UNIT=2841,ADDRESS=16
DISK161 IODEVICE UNIT=2311,ADDRESS=(161,3)
CONT19 IOCONTRL UNIT=2403,MODEL=3,ADDRESS=19
TAPE190 IODEVICE UNIT=2403,MODEL=3,ADDRESS=190,FEATURE=9-TRACK
TAPE191 IODEVICE UNIT=2401,MODEL=3,ADDRESS=(191,3), X
          FEATURE=9-TRACK
CHAN2 CHANNEL ADDRESS=2,TYPE=SELECTOR
CONT25 IOCONTRL UNIT=2841,ADDRESS=25
DISK251 IODEVICE UNIT=2311,ADDRESS=(251,3)
          UNITNAME NAME=SYSSQ,UNIT=((161,3),(190,4),253)
          UNITNAME NAME=SYSDA,UNIT=((161,3),253)
          UNITNAME NAME=SYSCP,UNIT=00B
          UNITNAME NAME=TAPE,UNIT=((190,4))
          CTRLPROG TYPE=MVT,MAXIO=30
          SCHEDULR CONSOLE=010,STARTR=A-00C,STARTW=A-00C, X
          TYPE=MVT,OPTIONS=(TSO),JOBQRES=253,SUBMITQ=500
          SUPRVSOR RESIDNT=(TR SVC,ACSMETH), X
          OPTIONS=(PROTECT,TRSVCTBL), X
          WAIT=MULTIPLE,TIMER=JOBSTEP,SER=MCH
          IMAGELIB UNIT=2311,VOLNO=LINVOL
          UCS UNIT=3211,IMAGE=(A11,H11,P11),DEFAULT=A11,P11
          UCS UNIT=1403,IMAGE=(AN,PN,TN),DEFAULT=(AN,PN)
          PARMLIB UNIT=2311,VOLNO=LINVOL
          PROCLIB UNIT=2311,VOLNO=LINVOL
          RESMODS PDS=SYS1.USER,MEMBERS=NUCID
          DATAMGT ACSMETH=(BDAM,ISAM,TCAM)
          EDITOR DESIGN=F44
          LOADER SIZE=50K,PARM=MAP
          ASSEMBLR DESIGN=F
          MACLIB EXCLUDE=(TRAM,QTAM,GPS,OCR)

```

Figure 73 (Part 1 of 2). Example 3: Stage I Input Deck

---

```

CKPTREST  ELIGBLE=(1,2,3,4,5)
SORTMERG  SORTOPT=FULLIB,SIZE=51200
SORTLIB   UNIT=2311,VOLNO=LINVOL
FORTRAN   DESIGN=G
FORTLIB   DESIGN=G,UNIT=2311,VOLNO=LINVOL,OBJERR=03
RPG
TELCMLIB
TSOPTION  LOGLINE=100
UADS      UNIT=2311,VOLNO=LINVOL
CMDLIB
EDIT      BLOCK=(FORT-450,TEXT-100)
OUTPUT    CLASS=(Z,Y)
HELP      UNIT=2311,VOLNO=LINVOL
GENTSO    UT1SDS=SYS1.ONE,UT2SDS=SYS1.TWO,UT3SDS=SYS1.THREE,X
          UT4SDS=SYS1.FOUR,OBJPDS=SYS1.OBJECT,RESNAME=2311,X
          RESVOL=SYSTEM,RESTYPE=2311,LINKNAME=2311,X
          LINKVOL=LINVOL,LBMAINT=E,ASMPRT=ON,X
          DIRDATA=PDS,LEPRT=(LIST,XREF),CMDNAME=2311,X
          CMDVOL=LINVOL
END
/*
//        START      RDR,190

```

---

Figure 73 (Part 2 of 2). Example 3: Stage I Input Deck

### Backup of New System

Figure 74 shows a listing of the IEHDASDR input deck used for creating the backup of the new system. IEHDASDR operates under control of the generating system on 111111. The backup copy of SYSTEM is made on the SYS001 magnetic tape volume. The backup copy of LINVOL is made on the SYS002 magnetic tape volume. After the two backup copies have been made, SYS001 and SYS002 should be removed and stored in the tape library.

---

```

//BACKUP  JOB MSGLEVEL=1          -BACKUP-
//        EXEC PGM=IEHDASDR
//SYSPRINT DD  SYSOUT=A
//SYSPACK DD  UNIT=2311,DISP=OLD,VOLUME=SER=SYSTEM
//LINPACK  DD  UNIT=2311,DISP=OLD,VOLUME=SER=LINVOL
//SYSTAPE  DD  UNIT=2400,DISP=OLD,VOLUME=SER=SYS001,LABEL=(,NL)
//LINTAPE  DD  UNIT=2400,DISP=OLD,VOLUME=SER=SYS002,LABEL=(,NL)
//SYSIN    DD  *
          DUMP FROMDD=SYSPACK,TODD=SYSTAPE
          DUMP FROMDD=LINPACK,TODD=LINTAPE
/*

```

---

Figure 74. Example 3: Creating Backup of SYSTEM and LINVOL

### Scratching Utility Data Sets

After the new operating system is generated, SYS1.ONE, SYS1.TWO, SYS1.THREE, and SYS1.FOUR should be scratched and uncataloged. The job stream and SYS1.OBJECT should be saved. Figure 75 shows a listing of the input deck used for scratching and uncataloging the three sequential data sets. The magnetic tape that contains the job stream can be removed from unit 190 and stored in

the tape library. SYS1.MOD remains in the new system-residence volume (SYSTEM) to be used later in this example. The IEHPROGM utility program is executed under control of the generating system.

---

```
//SCRATCH JOB MSGLEVEL=1 -SCRATCH DECK-  
//STEP0 EXEC PGM=IEHPROGM  
//SYSPRINT DD SYSOUT=A  
//SYSUT1 DD DSNAME=SYS1.ONE,DISP=(OLD,DELETE)  
//SYSUT2 DD DSNAME=SYS1.TWO,DISP=(OLD,DELETE)  
//SYSUT3 DD DSNAME=SYS1.THREE,DISP=(OLD,DELETE)  
//SYSUT4 DD DSNAME=SYS1.FOUR,DISP=(OLD,DELETE)  
//SYSIN DD DUMMY
```

---

Figure 75. Example 3: Scratching Utility Data Sets

## Program Options Descriptions

The descriptions of the control program options are provided in this chapter and arranged alphabetically. Each description contains a brief summary of the option's function, shows how to specify the particular option at system generation time, and provides a list of publications where further information can be found.



You can select various control program options to be included in your new system during system generation. What you select depends on the type of control program you plan to use, the amount of main storage available, and the types of job processing that will be done at your installation.

## **Additional Pairs of Transient Areas**

Optional For: MVT (including M65MP) systems.

One pair of supervisor (SVC) transient areas is always provided in an MVT system; additional pairs may be added. When a nonresident SVC routine is required during job execution, it is loaded into an available transient area. If no transient areas are available, then the task requiring the routine is placed in a wait state until one becomes available.

### Where to Find Information

The information necessary to use the facility discussed above can be found in:

- Storage requirements -- OS Storage Estimates

### What to Code When Generating a System

The following code is required to have additional pairs of transient areas included:

```
CTRLPROG      ...,ADDTRAN=number of pairs
```

## **Alternate Path Retry (APR)**

Standard For: M65MP systems

Optional For: MFT and MVT systems

The alternate path retry (APR) option allows an I/O operation that has developed an error on one channel path to a device to be retried on another channel path to the same device. This can be done only if another channel path has been assigned to the device performing the I/O operation. APR also provides the capability to vary a path to a device online or offline by use of the VARY command. The VARY PATH is a standard function of APR.

APR can handle:

- Up to four paths to one device.
- Two paths to a CPU for a multiprocessing system.
- The ninth drive on a 2314.

APR performs its function usefully only in a system that has the channel-check handler (CCH) and alternate paths to at least some of the I/O devices.

### How to Use

The operation of the selective retry function of APR, in conjunction with the I/O supervisor, does not depend on anything you do. The operator can initiate the VARY path function by entering the VARY PATH command in the input stream or at the console.

### Where to Find Information

Further information about APR can be found in:

- Storage requirements -- OS Storage Estimates
- How to enter commands -- OS Operator's Procedures, OS Operator's Reference

### What to Code When Generating a System

The following code is required to include alternate path retry (APR) in MFT and MVT systems:

```
SUPRVSOR ...,OPTIONS=(APR)
```

For APR to be able to determine whether a device has an alternate path in an M65MP system, the following code should be specified for magnetic tape drives and 2314 drives. No other devices need this additional code.

For magnetic tape drives:

```
IOCONTRL ...,FEATURE=(ALTCTRL)
```

For 2314 drives:

```
IODEVICE ...,FEATURE=(ALTCTRL)
```

Note: If none of the I/O RMS options (APR, DDR, CCH) are chosen, IGF201 (the I/O RMS communications area) appears as an unresolved symbol.

## **Attach Function**

Without Subtasking Standard For: MFT systems

With Subtasking Standard For: MVT (including M65MP) systems

With Subtasking Optional For: MFT systems

There are two versions of the ATTACH function -- with and without the subtasking capability. One of these versions is always part of every control program.



The ATTACH function without the subtasking capability passes control to another load module; when the requested load module terminates, control is returned to the program that issued the ATTACH macro instruction.

The ATTACH function with the subtasking capability creates subtasks, so that the issuing program and the program requested in the ATTACH macro instruction compete for system resources.

#### How to Use

MFT systems can have either form of the ATTACH function. The version without subtasking is standard, and the version with subtasking can be specified. This ATTACH function allows more than one task to be executed within one partition.

MVT systems have the ATTACH function with subtasking as a standard part of the control program.

#### Where to Find Information

Further information about the ATTACH function can be found in:

- Storage requirements -- OS Storage Estimates

#### What to Code When Generating a System

The following code is required to specify the ATTACH function with the subtasking capability for MFT systems:

```
SUPRVSOR ...,OPTIONS=(ATTACH)
```

MVT systems always have the ATTACH function with the subtasking capability; this is standard and should not be specified.

### **Attach Function Made Resident**

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The routines that make up the ATTACH function can be made resident in main storage as part of the nucleus. If this function is not resident, every time an ATTACH macro instruction is issued the ATTACH routines are brought into the supervisor transient area. Making the ATTACH routines resident can be specified for MFT systems. It is always included as a standard feature in MVT and M65MP systems, and the storage required is included in the basic fixed-main-storage requirements for those systems.

#### Where to Find Information

Further information about how the ATTACH function routines is made resident can be found in:

- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required to make the ATTACH routines resident for MFT systems:

```
SUPRVSOR      ...,RESIDNT=(ATTACH)
```

## **Basic Direct Access Method (BDAM)**

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The basic direct access method (BDAM) can be included in the new operating system. If you plan to use the SVC 2B (CIRB) in your programs, then BDAM, ISAM, or BTAM must be specified. (The MACLIB macro instruction must also be specified if CIRB is included.)

## Where to Find Information

Further information about BDAM can be found in:

- Storage requirements -- OS Storage Estimates
- How to use the CIRB macro instruction -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required to have BDAM included in new MFT systems:

```
DATANGT      ACSMETH=(BDAM)
```

If the CIRB macro instruction is going to be used, then code:

```
MACLIB
```

## **BLDL Table Made Resident**

Optional For: All systems

Any or all of the SYS1.LINKLIB or SYS1.SVCLIB directory entries can be made resident in fixed main storage. No List is supplied for SYS1.SVCLIB. If you want a list for SYS1.SVCLIB, you must specify the operator communication option in the SUPRVSOR macro. You can modify this list to fit your requirements. If you create your own list, the operator communication option in the SUPRVSOR macro instruction must be specified so that your list can be brought in during system initialization.

## How to Use

The standard list of SYS1.LINKLIB directory entries, IEABLD00, can be made resident. This BLDL list has nine entries. If you plan to use your own BLDL list or a list for SYS1.SVCLIB, the operator communication option must be specified at system generation time in the OPTIONS parameter of the SUPRVSOR macro instruction. This causes the 'SPECIFY SYSTEM PARAMETERS' message (IEA101A) to be printed during execution of NIP and provides the operator with the opportunity to specify a BLDL list other than the standard one to be used during the loading of the nucleus.

## Where to Find Information

The information necessary to use the BLDL table can be found in:

- Storage requirements -- OS Storage Estimates
- SYS1.LINKLIB -- OS System Generation
- Standard BLDL table -- OS Storage Estimates
- Modifying the BLDL table -- OS MFT Guide, OS MVT Guide
- Operator communication option -- OS System Generation, OS Messages and Codes, OS Operator's Procedures, OS Operator's Reference

## What to Code When Generating a System

The following code is required to make the BLDL table resident:

```
SUPRVSOR      ...,RESIDNT=(BLDLTAB)
```

When you create your own list or a list for SYS1.SVCLIB, this code must be used:

```
SUPRVSOR      ...,RESIDNT=(BLDLTAB),OPTIONS=(COMM)
```

## **Channel-Check Handler (CCH)**

Optional For: MFT and MVT Systems

CCH supports the 2860, 2870, 2880, 135, 145, and 155 channels. CCH intercepts channel-check conditions, performs an analysis of the environment, and facilitates recovery from channel-check conditions by allowing for the scheduling of device-dependent error recovery procedures by the input/output supervisor, which will determine whether the failing channel operation can be retried. If CCH is not present in the system, one of the other recovery management facilities receives control and writes an error record for the channel failure. In this case, the error causes system termination.

## How to Use

This feature is optional in the S/360 Models 65, 75, and 91 if the models are specified in the CENPROCS macro instruction.

It is automatically included in the S/360 Models 85 and 195 and S/370 Models 135, 145, 155, 165, and 195 if the models are specified in the CENPROCS macro instruction.

CCH must be specified for a S/360 Model M65MP if the model is specified in the CENPROCS macro instruction.

### Where to Find Information

The information necessary to use the channel-check handler can be found in:

- Storage requirements -- OS Storage Estimates
- Operator action -- OS Operator's Reference
- Channel-check record processing -- OS Utilities

### What to Code When Generating a System

The following code is required to specify this option for the Models 65, M65MP, 75, and 91:

```
SUPRVSOR ...,OPTIONS=CCH
```

## Checkpoint/Restart

Optional For: All systems

Checkpoint/restart expands the use of the restart capabilities that are provided by the RD parameter of the JOB and EXEC statements. The RD parameter permits execution of jobs to be restarted automatically at a job step after abnormal termination occurs.

Checkpoint/restart enables you to write checkpoint macro instructions (CHKPT) at various points in your program to record job status information. Then when an ABEND occurs, your program can be restarted automatically at the last of these points, or restart can be deferred until a later time, when the job can be resubmitted and the RESTART parameter in the JOB statement used. The RD parameter can also be used to partially or totally suppress checkpoint/restart.

The following restrictions apply to the establishment of a checkpoint by the CHKPT macro instruction.

- When the checkpoint is established, the job step must comprise a single task. The job step task must be the only task when the job step is restarted.
- A checkpoint cannot be established by an exit routine that returns control to the control program.
- If a STIMER or WTOR macro instruction has been issued, a checkpoint cannot be established before the time interval is completed or the operator's reply is received.
- In an MVT system with rollout/rollin, a checkpoint cannot be established when the job step has been allocated storage from outside its region.

### How to Use

To use checkpoint/restart, you must indicate that you plan to use it at system generation time in the RESIDNT parameter of the SUPRVSOR macro instruction.

The basic modules required from the SVC library (SYS1.SVCLIB) for checkpoint/restart will then be loaded automatically at NIP time. In the program that contains CHKPT macro instructions, a checkpoint data set and work area must be defined. The checkpoint/restart cataloged procedure (IEFREINT) must be included in SYS1.PROCLIB either before or after system generation.

Additional modules from the SVC library will be required if chained scheduling or track overflow are going to be used. You can obtain the additional modules by constructing your own access method option list (IEAIGGxx) and include it in the parameter library (SYS1.PARMLIB). To use your own access method list, you must specify the operator communication option at system generation time in the OPTIONS parameter of the SUPRVSOR macro instruction. This causes the 'SPECIFY SYSTEM PARAMETERS' message (IEA101A) to be printed during NIP and provides the operator with the opportunity to specify an access method option list other than the standard one to be used during the loading of the nucleus.

### Where to Find Information

Further information about checkpoint/restart can be found in:

- Detailed information -- OS Advanced Checkpoint/Restart
- RD parameter -- OS Job Control Language Reference
- Checkpoint macro instruction -- OS Supervisor Services and Macro Instructions
- RESTART parameter -- OS Job Control Language Reference
- SUPRVSOR macro instruction -- OS System Generation
- Access method option list -- OS MFT Guide, OS MVT Guide
- Storage requirements -- OS Storage Estimates
- Messages -- OS Messages and Codes

### What to Code When Generating a System

The following code is required to make it possible for you to use checkpoint/restart in your system:

For MFT systems:

```
SUPRVSOR    ...,RESIDNT=(ACSMETH)
```

For MVT or M65MP systems:

```
SUPRVSOR    ...,RESIDNT=(RENTCODE)
```

If your list has been built and added to SYS1.PARMLIB:

```
SUPRVSOR    ...,RESIDNT= ( ACSMETH  ),OPTIONS=(COMM)
              RENTCODE
```

To modify the standard list of ABEND codes that are eligible for automatic restart, or to add ABEND codes and make them eligible, use:

```
CKPTREST    ...
```

## Consoles — Alternate and Composite Console Options

Optional For: All systems

One primary console must always be specified for any operating system except the M65MP system. M65MP must have two primary consoles specified except when the multiple console support (MCS) option is specified. (See the description in "Consoles - Multiple Console Support (MCS).") One alternate console can be specified, or two for the M65MP system when MCS is not selected. A composite console (such as a card reader and a printer) can be specified as a primary or an alternate console. The composite console is considered one console even though it may consist of two different physical devices.

The following guidelines must be used when MCS is not selected:

- A primary console must be specified in the SCHEDULR macro instruction.  
For M65MP: Two primary consoles must be specified in the SCHEDULR macro instruction.
- A composite console can be used as a primary or an alternate console.  
For M65MP: No more than two of the total number of consoles specified can be composite consoles.
- When a graphic device is going to be active as a console, a device that produces printed output must be specified.

### Where to Find Information

Further information regarding alternate and composite console options can be found in:

- Storage requirements -- OS Storage Estimates

### What to Code When Generating a System

The following code is required to specify consoles and alternate consoles when MCS is not specified:

For MFT or MVT:

```
SCHEDULR    ...,CONSOLE={address
                {(I-address,0-address)}
            , ALTCONS={address
                {(I-address,0-address)}
            }
```

For M65MP:

```
SCHEDULR    ...,CONSOLE=( {address
                            {(I-address,0-address)} {, address
                            {(I-address,0-address)}} )
            [ ,ALTCONS=( {address
                            {(I-address,0-address)} {, address
                            {(I-address,0-address)}} ) ]
```

## Consoles — Multiple Consoles Support (MCS)

Optional For: MFT and MVT (including M65MP) systems

If you want two or more consoles active during execution of an MFT or MVT system, you must specify the multiple console support (MCS) option in the CONOPTS parameter of the SCHEDULR macro instruction.

### How to Use

One console must be specified in the SCHEDULR macro instruction; it is called the "master" console. An alternate console for the master console must be specified in the ALTCONS parameter of the SCHEDULR macro instruction. A SECONSLE macro instruction must be coded defining the alternate as a secondary console. Additional secondary consoles can be defined with SECONSLE macro instructions -- up to a maximum of 31 secondary consoles. For all consoles for which no alternate console is specified, the master console is automatically assigned as the alternate.

When MCS is specified, the M65MP system requirements are exactly the same as for the other MCS systems. However, it is recommended that the console for the other CPU be assigned as the alternate for the master console.

A hard-copy log can be specified either at system generation or by the operator during system initialization or execution. A hard-copy log is required when there is more than one active console during initialization or execution, or when there is an active display console. The hard-copy log can be the system log that is contained on SYS1.SYSVLOGX and SYS1.SYSVLOGY, or it can be a console with output capability. If the log is required, the system records the operator commands, the system commands and responses, and the messages with routing codes of 1, 2, 3, 4, 7, 8, and 10 on the hard-copy log. Additional messages can be recorded if desired.

Routing codes and descriptor codes are required for all messages handled by a system using MCS. Messages that already exist can be assigned routing codes at system generation time or, by default, they will be sent to the master console.

Routing codes are assigned to all new operator messages (WTO and WTOR). They designate what function the message is connected with and determine where a message is to be sent. A system generation parameter provides the ability to supply routing codes to all operator messages that already exist and do not have routing codes.

Each console is assigned one or more routing codes. The routing codes assigned to a console are matched to the routing codes assigned to WTO and WTOR messages. If there is a match, the message is sent to the console. Some messages, such as a message that is broadcast to all active consoles, are not routed by the routing code.

Display areas may be defined for display consoles. Display areas are contiguous groups of four or more screen lines which are designated for use by status displays.

Descriptor codes must be specified for all new operator messages. They are specified in the WTO or WTOR macro instructions. They designate how a message is to be printed or displayed.

All commands have been arranged by function into four command code groups: informational, system control, I/O control, and console control.

An exit routine is provided just before the routing codes of a message are checked to enable you to supply your own routine to add, delete, or change routing and descriptor codes. (See OS MPT Guide or OS MVT Guide for a description of the exit routine.)

The following guidelines must be used:

- If HARDCPY=SYSLOG is specified in the SCHEDULR macro instruction during system generation, then at IPL time the operator must change the HARDCPY parameter to refer to the address of an operator console that has output capability. The device should not be the master console. The HARDCPY specification can be changed back after the message IEE141I has been received. (For detailed operating instructions, see OS Operator's Reference.)
- A master console must be specified in the CONSOLE keyword parameter of the SCHEDULR macro instruction.
- An alternate console to the master console must be specified in the ALTCONS keyword parameter of the SCHEDULR macro instruction.
- The alternate for the master console must be defined in the CONSOLE parameter of a SECONSLE macro instruction to make it a secondary console.
- A console with at least printing output capability must be specified as the hard-copy log. Although the system log is not a console, and does not directly produce printed output, it can be used.
- A record of the operator commands, system commands and responses, and routing codes 1, 2, 3, 4, 7, 8, and 10 should be maintained.
- Up to 31 secondary consoles can be specified with SECONSLE macro instructions. They can all have alternate consoles specified. If no alternate is defined, then the master console automatically becomes the alternate.
- A 2250 display unit can be specified as a master, secondary, or alternate console. A display console with input and output capability (a full capability console) can be specified as a master, secondary, or alternate console. A display console with no input capability (an output only console) or limited light pen input capability (a limited capability console) should not be specified as a master or alternate console.
- A 2260 display station can be specified as a master, secondary, or alternate console. It may also be designated as an output-only console for display of either general messages or status displays.
- One or more display areas may be defined for each display console in the system.
- Any number of the consoles can be composite consoles.
- Routing and descriptor codes are assigned to all new operator messages that are written.



## Where to Find Information

Further information about MCS can be found in:

- SCHEDULR and SECONSLE macro instructions -- OS System Generation
- Hard-copy log -- OS Operator's Reference, OS System Generation
- System log -- OS Operator's Reference, OS System Generation
- Operator commands -- OS Operator's Reference
- Display (CRT) consoles -- OS Operator's Guide for Display Consoles
- Messages -- OS Messages and Codes
- Routing codes -- OS Messages and Codes, OS Supervisor Services and Macro Instructions, OS Operator's Reference
- Descriptor codes -- OS Messages and Codes, OS Supervisor Services and Macro Instructions, OS Operator's Reference
- Master, secondary, and alternate consoles -- OS Operator's Reference, OS System Generation
- Storage requirements -- OS Storage Estimates
- WTO and WTOR macro instructions -- OS Supervisor Services and Macro Instructions
- Command code group -- OS Operator's Reference
- SYS1.SYSVLOGX and SYS1.SYSVLOGY -- OS System Generation
- User exit routine -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required to specify the master, the secondary, and the alternate consoles:

```
SCHEDULR  ....,CONSOLE={address
                      {(I-address,0-address)}
,ALTCONS={address
          {(I-address,0-address)}

CONOPTS=  (MCS{ ,NOEXIT}
           {EXIT

           [ROUTCDE={1
                   {(routing code[,routing code]...)}]}

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

           [ HARDCPY=({SYSLOG }
                     {address}
                     { ,ALL
                       { , (routing code[, routing code]...)} }
                     { ,NOCMDS } { , INCMDS }
                     { ,CMDS }  { , STCMDS }
           ]
           [ AREA=(nn[ ,nn... ] ) ]
           [ OPSTRAN=nn ]
           [ PFK=nn ]
```

```

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

```

## Conversational Remote Job Entry (CRJE)

Optional For: MFT and MVT (including M65MP) systems.

Conversational remote job entry (CRJE) provides remote access to the operating system from printer-keyboard terminals. Authorized terminal users can conversationally prepare and update programs and data, submit them for OS background processing, and receive the output either at the central installation or at the remote terminal.

### How to Use

Conversational remote job entry (CRJE) requires the basic telecommunication access method (BTAM) routines. Background execution of CRJE-submitted jobs is accomplished concurrently with normal batch processing under the supervision of the OS job management routines. The valid CRJE terminal user is one that has been defined in the system at CRJE assembly time in the CRJEUSER macro instruction or has been added to the system by the central operator using the USERID central command.

The terminal user can insert, replace, delete, or change information to be submitted in jobs by using the CRJE data set updating facilities. He can have PL/I or FORTRAN source statements checked for syntax errors before submitting the job. The syntax checking program(s) are included at system generation time by the CHECKER macro instruction.

The terminal user can inquire about the status of the system or remotely submitted jobs. There is also a message facility for two-way communication between terminal users, and between terminal users and the central operator.

CRJE is specified at system generation time in order to have the necessary modules included in the system. After generation, you must create the specific CRJE system required for your installation. There are three macro instructions available for this job -- CRJELINE, CRJETABL, and CRJEUSER. You set up a job that includes the CRJE macro instructions necessary to specify your system; you may include your own routines. The assembler translates these macro instructions and creates the required modules. The linkage editor incorporates the modules into the operating system.

SYS1.MACLIB must be in the operating system so that the assembler can expand the macro instructions. SYS1.TELCMLIB must be in the system to hold some of the CRJE load modules as well as the telecommunication subroutines. Enough

system queue space must be specified in the CTRLPROG macro instruction during system generation to handle the necessary CRJE space requirements.

### Where to Find Information

Further information about CRJE can be found in:

- Machine requirements -- OS CRJE System Programmer's Guide
- Generating the CRJE system -- OS CRJE System Programmer's Guide
- CRJE facilities -- OS CRJE System Programmer's Guide
- BTAM information -- OS CRJE System Programmer's Guide
- System queue space -- OS CRJE System Programmer's Guide, OS Storage Estimates
- CRJE macro instructions -- OS CRJE System Programmer's Guide
- System generation macro instructions -- OS System Generation
- Specifying system data sets -- OS System Generation
- Storage requirements -- OS CRJE System Programmer's Guide, OS Storage Estimates

### What to Code When Generating a System

The following system data sets and options must be specified:

SYS1.MACLIB  
SYS1.PROCLIB  
SYS1.TELCMLIB  
An Assembler  
A Linkage Editor

The following code is required to support CRJE:

```
DATAMGT          ACSMETH=BTAM

IOCONTRL -- One for each telecommunications control unit.

IODEVICE -- One for each telecommunications line.

SCHEDULR        ...,OPTIONS=CRJE,...

PROCLIB
```

For MFT:

```
SUPRVSOR        ...,OPTIONS=ATTACH
```

To include syntax checking:

```
CHECKER        ...
```

## Decimal Simulation Option for Model 91

Optional For: MVT systems

The decimal simulation option provides the Model 91 with the ability to handle decimal arithmetic instructions; the Model 91 is not equipped with decimal arithmetic instruction circuitry. This option requires both a long execution time and that the CPU not be operational during the simulation. The universal instruction set, which is standard for the Model 91, includes only the EDIT and EDMK decimal instructions; any other decimal instruction is simulated. This option should be specified if COBOL, PL/I, or RPG is to be included in the system, or if decimal arithmetic instructions are to be used in assembler language.

### Where To Find Information

Further information about the decimal simulation option can be found in:

- Storage requirements -- OS Storage Estimates

### What to Code When Generating a System

The following code is required to specify this option:

```
SUPRVSOR      ...,OPTIONS=(DECSIM)
```

Although the universal instruction set is assumed, it can be specified by:

```
CENPROCS      ...,INSTSET=UNIV
```

## Direct-Access Volume Serial Number Verification

Optional For: All systems

You can add direct-access volume serial number verification to your new system. If you do, the volume serial number of a direct-access device is checked after an unsolicited device-end interrupt condition has been corrected and the volume has been put back online again.

### How to Use

When an unsolicited device-end interrupt is received from a direct-access device, the I/O supervisor (IOS) ensures that the volume serial number of the mounted volume agrees with the volume serial in the unit control block (UCB).

The code to do the checking is included at system generation time unless NODAV is specified in the OPTIONS keyword parameter of the SUPRVSOR macro instruction.

## Where to Find Information

Further information about direct-access volume serial number verification can be found in:

- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required to delete this option:

```
SUPRVSOR      ...,OPTIONS=(NODAV)
```

Note: If OPTIONS=DDR is specified, this option is automatically included, so that if OPTIONS=NODAV is coded, it is ignored.

## **Dynamic Device Reconfiguration (DDR)**

Standard For: M65MP systems

Optional For: MFT and MVT systems

The dynamic device reconfiguration option allows a demountable volume to be moved from one device to another and repositioned if necessary without abnormally terminating the job or redoing IPL. A request to move a volume may be initiated by either the system or the operator.

### How to Use

The outboard recorder routine of the I/O Supervisor transfers control to the DDR routines when a permanent I/O error occurs. These routines then determine if it is possible to correct the error by moving a volume to another device. If it is, the system requests a volume swap by issuing a message to the operator. The operator must answer this message.

Sometimes the operator will determine that volumes need to be swapped. He can initiate this action by entering a SWAP command.

The DDR routines will be used if:

- DDR, DDRSYS, or DDRNSL has been specified in the OPTIONS keyword parameter of the SUPRVSOR macro instruction during system generation.
- The device that has a permanent I/O error is a 2311, 2314, 2321, 3330, any 2400 or 3400 series magnetic tape drive, a card reader, a printer, or a card punch. No teleprocessing devices are supported. Any volume on a device for which shared DASD has been specified can only be demounted and remounted on the same device. The DDR routines can be used for the unit record devices only if the operator issues the request by means of the SWAP command when intervention is required.
- The type of permanent I/O error is supported. The ones that are not supported are: wrong length record, no record found, unit exception, program check, protection check, IOB intercept condition, backing to load point, or a permanent I/O error caused by the channel program.

Notes:

- You should not code specific unit addresses in programs that will be processed on a system that has DDR.
- The direct-access serial number verification routines must be in the system that has the DDR routines.

For FETCH: When I/O errors occur while the FETCH routines are addressing the SVCLIB, the DDR system-residence routines receive control, and, if possible, request a swap. For this to occur, OPTIONS=DDRSYS must be specified in the SUPRVSOR macro instruction and the conditions listed above must exist.

For DDR System-Residence Routines: When these routines are specified in the OPTIONS keyword parameter of the SUPRVSOR macro instruction, another keyword parameter, ALTSYS, must also be specified.

If high availability is important to the installation, a duplicate system-residence volume is advisable. However, to use such a volume, writing on any part of the system-residence volume other than SYS1.LOGREC would have to be prohibited.

The system-residence device specified during system generation can be changed at IPL time by the operator. OPTIONS=COMM must be specified in the SUPRVSOR macro instruction during system generation to make this change possible.

For Nonstandard Labels: If you want DDR and have nonstandard magnetic tape labels, OPTIONS=DDRNSL must be specified. A nonstandard label routine with the name NSLREPOS must be supplied. This routine can either be added during system generation using the SVCLIB macro instruction, or be link-edited into SVCLIB after the system generation process is completed.

For DDR When EXCP is Used: When the EXCP macro instruction is used to address magnetic tape drives in a program that runs under a system with DDR, REPOS=Y or N must be coded in the DCB macro instruction to indicate whether an accurate block count is being maintained.

Where to Find Information

Further information about DDR can be found in:

- SWAP command -- OS Operator's Reference
- System generation macro instruction -- OS System Generation
- Nonstandard label routine -- OS Tape Labels
- Shared DASD devices -- OS System generation
- EXCP macro instruction -- OS Data Management for System Programmers
- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required to specify that the dynamic device reconfiguration (DDR) routines are to be included in MFT and MVT systems:

```
SUPRVSOR      ...,OPTIONS=(DDR)
```

When this option is specified, the direct-access volume serial number verification (DAV) is automatically included. If OPTIONS=NODAV is coded, it is ignored.

To include the DDR routines to handle system-residence volumes:

```
SUPRVSOR      ...,OPTIONS=(DDRSYS),  
              ALTSYS=address
```

To include the DDR routines to handle magnetic tape volumes with nonstandard labels:

```
SUPRVSOR      ...,OPTIONS=(DDRNSL)
```

To change the system-residence device address at IPL:

```
SUPRVSOR      ...,OPTIONS=(COMM)
```

Note: If none of the I/O RMS options (APAR, DDR, CCH) are chosen, IGF201 (the I/O RMS communication area) appears as an unresolved symbol.

## **Extract Function Made Resident**

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The routines that make up the EXTRACT function can be made resident in main storage as a part of the nucleus. If this function is not resident, every time an EXTRACT macro instruction is issued, the routines are brought into the supervisor transient area.

The EXTRACT function that is included in an MFT system with subtasking is the same as the EXTRACT function in an MVT system.

The EXTRACT macro instruction provides your program with information contained in specified fields of the task control block (TCB) of either the task that issued the macro instruction or, in a multiprogramming environment, one of its subtasks.

### Where to Find Information

Further information about how the extract function is made resident can be found in:

- Use of EXTRACT -- OS MFT Guide, OS MVT Guide, OS Data Management Services Guide
- EXTRACT macro instruction -- OS Supervisor Services and Macro Instructions
- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required to make the EXTRACT routines resident for MFT systems:

```
SUPRVSOR      ...,RESIDNT=(EXTRACT)
```

## Graphic Programming Services

Optional For: All systems

The graphic programming services control graphic input and output and a set of problem-oriented routines that are used as building blocks in the construction of graphic processing programs. The graphic subroutine package (GSP) allows the FORTRAN IV or PL/I F programmer to use the graphic programming services.

### How to Use

The problem-oriented routines generate graphic instructions for displaying various images and alphameric information on the 2250 display unit. These routines function as part of the problem program and are reached by a CALL or LINK macro instruction.

### Where to Find Information

Further information about graphic programming services can be found in:

- Storage requirements -- OS Storage Estimates
- Detailed information -- OS GPS for IBM 2250 Display Unit, OS GPS for IBM 2260 Display Station (Local Attachment), OS GPS for IBM 2280 and 2282 Display Units

## What to Code When Generating a System

The following code is required for this option:

```
GRAPHICS      PORRTNS=INCLUDE,GSP=INCLUDE
```

## Identify Function Made Resident

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The routines that make up the IDENTIFY function can be made resident in main storage as a part of the nucleus. If these routines are not resident, every time an IDENTIFY macro instruction is issued, the routines are brought into the supervisor transient area. If the IDENTIFY module is resident, performance improves, but the amount of required fixed main storage increases.



## How to Use

The IDENTIFY macro instruction is used to inform the supervisor of an embedded entry point within a load module.

After the IDENTIFY macro instruction has been executed, the entry point can be referred to by an ATTACH, LINK, XCTL, or LOAD macro instruction.

## Where to Find Information

Further information on how the identity function is made resident can be found in:

- IDENTIFY macro instruction -- OS Supervisor Services and Macro Instructions
- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required to specify the option for MFT:

```
SUPRVSR      ...,RESIDNT=(IDENTIFY)
```

## **Indexed Sequential Access Method (ISAM)**

Optional For: All systems

The indexed sequential access method can be included in the new system so that tasks can use the basic indexed sequential access method (BISAM) or the queued indexed sequential access method (QISAM). If you plan to use the SVC 2B (CIRB) in your programs, then BDAM, ISAM, or BTAM must be specified. (The MACLIB macro instruction must also be specified if CIRB is included.)

## Where to Find Information

Further information about ISAM can be found in:

- Storage requirements -- OS Storage Estimates
- How to use CIRB macro instruction -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required to include ISAM in the new system:

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

If CIRB is desired:

```
MACLIB
```

## Job Step Timing

Optional For: All systems

Each job step can be timed and the time limits enforced. The amount of time used is recorded after a job step is finished. In addition, the following are included in this option: the ability to request the date plus the time of day, to change the time at midnight, and to request, check, and cancel intervals of time. (See the description of "Timing Options" later in this section.)

### Where to Find Information

Further information about job-step timing can be found in:

- Amount of storage required -- OS Storage Estimates

### What to Code When Generating a System

The following code is required to specify this option:

```
SUPRVSOR      ...,TIMER=JOBSTEP
```

## Main Storage Hierarchy Support

Optional For: All systems

Main storage hierarchy support provides selective access to either processor storage or the IBM 2361 Core Storage.

### How to Use

Main storage is divided into two blocks known as hierarchies; hierarchy 0 is assigned to processor storage and hierarchy 1 to the 2361. To use hierarchies, program controlled interrupt (PCI) must always be specified. (See the description of "Program Controlled Interrupt (PCI) FETCH" later in this section.)

For MFT systems, if there is not a 2361 unit on the system, any references to hierarchy 1 will have to be redefined at IPL time. Hierarchies for partitions in MFT are defined in the PARTITNS system generation macro instruction.

For MVT systems, the hierarchy structure is maintained even though there may not be a 2361 unit on the system.

### Where to Find Information

Further information about main storage hierarchy support can be found in:

- Storage requirements -- OS Storage Estimates
- Detailed information -- Introduction to Main Storage Hierarchy Support for IBM 2361, Mod 1 and 2 under OS

## What to Code When Generating a System

The following code is required to have this option included in the new system:

For MVT:

```
CTRLPROG      ...,HIARCHY=INCLUDE,FETCH=PCI
```

For MFT:

```
CTRLPROG      ...,HIARCHY=INCLUDE,FETCH=PCI
```

```
PARTITNS      (operand[,operand]...)
```

## **PL/I F Shared Library Feature**

Optional For: MFT or MVT (including M65MP) systems

This feature permits you to create a resident load module that contains common routines and makes this load module accessible from all partitions or regions. These often-used modules would otherwise be link-edited into the compiler output module in each program that uses them. With this feature, they are made into a separate resident module available to any program.

This feature can be used only with an operating system that is of Release 18 level or later.

### How to Use

The modules that are to be placed in the resident load module are specified in the parameters for the PL1LIB macro instruction at system generation time. The load module is created at system generation and is placed in SYS1.LINKLIB with the member name of IHELTVA. You must construct a reentrant list (IEAIGGnn) with the member IHELTVA in it. The list is placed in SYS1.PARMLIB before IPL time. At IPL time, the message (IEA101A) 'SPECIFY SYSTEM PARAMETERS' is typed out. The operator must reply RAM=nn, where nn represents the two digits used when the list name was created. This will cause the PL/I shared library module to be included in the resident portion of main storage.

COMM must be specified in the OPTIONS parameter and RENTCODE must be specified in the RESIDNT parameter of the SUPRVSOR macro instruction.

### Where to Find Information

Further information about the PL/I shared library feature can be found in:

- PL1LIB and SUPRVSOR macro instructions -- OS System Generation
- Shared library -- OS PL/I (F) Programmer's Guide
- List creation -- OS MFT Guide, OS MVT Guide
- Messages -- OS Messages and Codes

- Storage requirements -- OS Storage Estimates
- Operating procedures -- OS Operator's Procedures, OS Operator's Reference

### What to Code When Generating a System

The following code is required to specify this option:

```

SUPRVSOR      RESIDNT=RENTCODE,
               OPTIONS=COMM

PL1LIB        MODES=( NOTK [,REAL][,CMPX])
               TASK
               [ STORG=(operand[,operand]...) ]
               [ CONVS=(operand[,operand]...) ]
               [ STRGS=(operand[,operand]...) ]
               [ ARRAY=(operand[,operand]...) ]
               [ MATHS=(operand[,operand]...) ]
               [ STRIO=(operand[,operand]...) ]
               [ RECIO=(operand[,operand]...) ]

```

### **Program Controlled Interrupt (PCI)**

Optional For: All systems

Program controlled interrupt (PCI) permits the program to cause an I/O interruption during execution of an I/O operation. PCI provides a means of alerting the program of the progress of chaining during an I/O operation. It also permits programmed dynamic main-storage allocation.

PCI fetch is able to bring a program into main storage with only one seek of the disk if:

- A buffer is always available for relocation dictionaries.
- No errors occur during the I/O operation.
- No cylinders are crossed while bringing in the program.
- The speed of the central processing unit allows PCI to modify the channel command word before it reaches the channel.

An additional WAIT and seek are required each time a buffer is not available. A seek is required each time an error occurs or a cylinder is crossed. If the speed of the central processing unit does not allow PCI to perform its function in time, the number of seeks needed by the standard fetch are required.

### Where to Find Information

Further information about PCI can be found in:

- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required in order to specify this option:

```
CTRLPROG      ...,FETCH=PCI
```

## **Reenterable Load Modules Made Resident**

Optional For: All systems

Reenterable load modules from the SYS1.LINKLIB and SYS1.SVCLIB can be made resident. MFT systems can have only user-written load modules and the loader program modules from LINKLIB made resident in the reenterable load module area; MVT and M65MP systems can have modules from either or both libraries made resident in the link pack area.

### How to Use

There are standard lists that are used during IPL to place the load modules from the libraries into the fixed portion of main storage: IEAIGG00 for SYS1.LINKLIB and IEARSV00 for SYS1.SVCLIB. If the user wishes to create his own list, then the operator communication option (OPTIONS=COMM) must be specified in the SUPRVSOR macro instruction. This causes the message (IEA101A) to print out 'SPECIFY SYSTEM PARAMETERS'. Then the operator provides the unique identification for the list. The reenterable load modules pointed to by the list are loaded into main storage at IPL time.

### Where to Find Information

Further information about reenterable load modules being made resident can be found in:

- Storage requirements -- OS Storage Estimates
- Standard list -- OS Storage Estimates, OS MFT Guide, OS MVT Guide
- Operator communication option -- OS System Generation
- Messages -- OS Messages and Codes
- Operator action -- OS Messages and Codes, OS Operator's Procedures, OS Operator's Reference

## What to Code When Generating a System

The following coding is required to specify the option:

```
SUPRVSOR      ...,RESIDNT=(RENTCODE)
```

If the operator communication option is required, the following coding is required:

```
SUPRVSOR      ...,RESIDNT=(RENTCODE),OPTIONS=(COMM)
```

## Remote Job Entry (RJE)

Optional For: All systems

The remote job entry (RJE) facility provides a method of entering jobs from remote work stations into the job stream. Once the jobs have been entered, execution proceeds under the supervision of the operating system. Any output data sets created by a remotely submitted job that the terminal user wants returned are placed in a separate output class and then sent to him.

### How to Use

The RJE facility operates under the operating system on a computer-based telecommunications system; it requires the basic telecommunications access method (BTAM) routines. RJE is specified at system generation time in order to have the necessary modules included in the system.

After generation you must create the specific RJE system required for your installation. There are four macro instructions available for this job -- RJETERM, RJELINE, RJEUSER, and RJETABL. You must set up a job that includes the RJE macro instructions necessary to specify your system; you may include your own routines. The assembler translates these macro instructions and creates the required modules. The linkage editor incorporates the modules into the operating system. SYS1.MACLIB must be in the operating system so that the assembler can expand the macro instructions. SYS1.TELCMLIB must be present in the operating system also to hold some of the RJE load modules as well as the telecommunications subroutines.

Enough system write-to-operator (WTO) buffers must be specified in the WTOBFRS parameter of the SCHEDULR macro instruction during system generation so that an RJE task will not have to wait to display a message. If a wait occurs, a work station timeout could result. A recommended value for the number of buffers is twice the number of telecommunications lines in the system.

### Where to Find Information

Further information about RJE can be found in:

- Machine requirements -- OS RJE
- RJE macro instructions -- OS RJE
- System generation macro instructions -- OS System Generation
- Specifying system data sets -- OS System Generation
- Storage requirements -- OS RJE, OS Storage Estimates

### What to Code When Generating a System

The following system data sets and options must be specified:

SYS1.MACLIB  
SYS1.PROCLIB  
SYS1.TELCMLIB  
An assembler  
A linkage editor

The following code is required to support RJE:

```
DATAMGT      ACSMETH=(BTAM,BDAM)

IOCONTROL -- One for each telecommunications control unit.

IODEVICE -- One for each telecommunications line.

PROCLIB

SCHEDULR     ...,OPTIONS=(RJE),WTOBFRS=number
```

## Resident Access Method Routines

Optional For: MFT systems

Reenterable access method load modules can be made resident from the SYS1.SVCLIB. Modules IGG019FN, IGG019FP, and IGG019C4 should be made resident if you are using the search direct option. When these modules are resident, performance is improved and required region size is decreased.

### How to Use

The standard list, IEAIGG00, contains the names of the access-method routines that are to be loaded and made resident by IPL. If you want to create your own list to load certain modules, then the operator communication option (OPTIONS=COMM) must be specified in the SUPRVSOR macro instruction. Then at IPL time the message (IEA101A) will print out 'SPECIFY SYSTEM PARAMETERS'. The operator answers with the unique identification numbers for your list. The access-method routines that are pointed to by this list will then be loaded into main storage at IPL time.

### Where to Find Information

Further information about resident access method routines can be found in:

- Storage requirements -- OS Storage Estimates
- Standard list -- OS Storage Estimates, OS MFT Guide, OS MVT Guide
- Operator communication option -- OS System Generation
- Messages -- OS Messages and Codes
- Operator action -- OS Messages and Codes, OS Operator's Procedures, OS Operator's Reference

### What to Code When Generating a System

The following code is required to specify this option:

```
SUPRVSOR     ...,RESIDNT=(ACSMETH)
```

If the operator communication option is also specified, the coding required is:

```
SUPRVSOR     ...,RESIDNT=(ACSMETH),OPTIONS=(COMM)
```

## Rollout/Rollin Option

Optional For: MVT (including M65MP) systems

A job step can temporarily expand its specified region. A job step's region size can be based on a minimum actual requirement, rather than a maximum one.

### How to Use

When a job step needs more main storage, an attempt is made to obtain it from unassigned storage. If none is available, another job step is rolled out, that is, its entire region is transferred to secondary storage, and its storage is made available to the first job step. When released by the first job step, the additional storage is again available as unassigned storage, if that was its source. The additional storage may also be available to receive the rolled-out job step, which is transferred back into main storage (rolled in). Through job control you specify jobs eligible to be rolled out or to cause rollout. Exits are provided at key decision points where installation-written routines can be added to expand, redirect, or limit, the feature's operation.

The data set SYS1.ROLLOUT must be cataloged in the new system before IPL.

### Where to Find Information

Further information about the rollout/rollin option can be found in:

- Storage requirements -- OS Storage Estimates
- Exit routines -- OS MFT Guide, OS MVT Guide
- SYS1ROLLOUT -- OS System Generation.
- Job control statements -- OS Job Control Language Reference

### What to Code When Generating a System

The following code is required to specify this option:

```
CTRLPROG      ...,OPTIONS=(ROLLOUT)
```

## Shared DASD

Optional For: All systems except the M65MP

Two central processing units can access the same direct-access device concurrently.

### Where to Find Information

Further information about shared DASD can be found in:

- General information -- OS MVT Guide



- Amount of storage required -- OS Storage Estimates
- Operator information -- OS Operator's Reference
- Functions, operating environment, and volume acceptability -- OS Data Management for System Programmers

#### What to Code When Generating a System

The following code is required to specify this option:

```
Either -- IOCTRL      ...,FEATURE=SHARED
Or -- IODEVICE       ...,FEATURE=SHARED (For 2314 only)
```

### SPIE Routines Made Resident

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The set program interruption element (SPIE) function can be made resident. If this function is not resident, it is brought into the supervisor transient area whenever a SPIE macro instruction is executed.

The SPIE macro instruction specifies the address of a routine to be used when specified program interruptions occur in the task that issued the macro instruction.

#### Where to Find Information

Further information about SPIE routines being made resident can be found in:

- Amount of storage required -- OS Storage Estimates
- SPIE macro instruction -- OS Supervisor Services and Macro Instructions

#### What to Code At System Generation Time

The following code is required in order to specify this option:

```
SUPRVSOR      ...,RESIDNT=(SPIE)
```

## Storage Protection Option

Required For: MVT (including M65MP) systems

Optional For: MFT systems

When the storage protection option is included in the central processing unit, it may be specified. The validity check option is included also, but as a standard feature when this option is specified.

### How to Use

When the storage protection option is specified, the size of each protected area must be a multiple of 2,048 bytes.

The storage protection machine option is a standard feature for models 50 and larger, but the storage protection programming option must be specified during system generation. For all systems, if the storage protection programming option is specified, the validity checking function is included and cannot be specified in the system generation coding.

### Where to Find Information

Further information about the storage protection option can be found in:

- Amount of storage required -- OS Storage Estimates

### What to Code When Generating a System

The following code is required in order to specify this option:

```
CENPROCS      ...,FEATURE=PROTECT (For Model 40)
SUPRVSOR      ...,OPTIONS=(PROTECT)
```

## System Management Facilities (SMF)

Optional For: All systems

The system management facilities (SMF) are a group of routines that collect and record data about how the system and the I/O devices were used by the jobs and the job steps. For the M65MP systems these routines collect and record data about the use of central processing units, channels, and storage, as well as I/O devices. The data that is collected by the SMF routines is put on one or two data sets (SYS1.MANX and SYS1.MANY) -- one if magnetic tape is used, or two if direct-access devices are used. Six exits are provided, so that you can supply your own exit routines to supplement the SMF option. The data collected can be recorded on your own or the SMF data sets.

### How to Use

To use SMF, you must specify the ACCTRTN parameter in the SCHEDULR macro instruction and the TIMER parameter in the SUPRVSOR macro instruction at system generation time. A definition list (SMFDEFLT) should be placed in the parameter

library (SYS1.PARMLIB) before the first IPL. (This list can be put in either before or after system generation.) The definitions in the list provide the factors that determine which functions SMF will perform and whether any of the six exits (IEFUJV, IEFUJI, IEFUSI, IEFACRT, IEFUTL, IEFUSO) is to be used. If you have written one or more routines to supplement SMF, they may be placed in SYS1.CI505 before system generation is started.

The SMF macro instruction (SMFWTM) and the SMF dump routine (IFASMFDP) are included automatically at system generation time as part of the SMF routines. The macro instruction is used to write your data records onto the SMF data sets from the SMF I/O buffer. The dump routine should be used, if the data sets are on direct-access devices, to dump the contents to magnetic tape. A sample program (TESTEXIT) to test the SMF routines and any of your routines is provided in the sample library (SYS1.SAMPLIB) of the starter operating system (see the SMF publication for a discussion of the sample program).

### Where to Find Information

Further information about SMF can be found in:

- SMF data sets defined -- OS System Generation
- User exits -- OS SMF
- Definition list -- OS SMF
- SCHEDULR and SUPRVSOR macro instructions -- OS System Generation
- SMF I/O buffer -- OS Storage Estimates
- SMF macro instruction (SMFWTM) -- OS SMF
- SMF dump program (IFASMFDP) -- OS SMF
- Parameter library (SYS1.PARMLIB) -- OS System Generation
- Sample programs -- OS SMF
- Specifying SMF -- OS System Generation
- Storage requirements -- OS Storage Estimates

### What to Code When Generating a System

The following code is required to permit you to use the SMF routines in your system:

```
SCHEDULR      ...,ACCTRTN=SMF
SUPRVSOR      ...,TIMER=JOBSTEP
```

## Telecommunications Option

QTAM, TCAM, and BTAM Optional For: All systems

A telecommunications option can be included in the new system so that tasks can use the basic telecommunications access method (BTAM), the queued telecommunications access method (QTAM), or the telecommunications access method (TCAM). If CIRB (SVC 2B) is desired in the new system, BDAM, ISAM, or BTAM must be specified. (The MACLIB macro instruction must also be specified if CIRB is to be included.)

### Where to Find Information

Further information about the telecommunications option can be found in:

- Amount of storage required -- OS Storage Estimates
- How to use the CIRB macro instruction -- OS MFT Guide, OS MVT Guide

### What to Code When Generating a System

The following code is required to have this option in the new system:

```
DATAMGT   ACSMETH=(BTAM) or
DATAMGT   ACSMETH=(QTAM) or
DATAMGT   ACSMETH=(TCAM) or
DATAMGT   ACSMETH=(BTAM,QTAM) or
DATAMGT   ACSMETH=(BTAM,TCAM) or
DATAMGT   ACSMETH=(QTAM,TCAM) or
DATAMGT   ACSMETH=(BTAM,QTAM,TCAM)
TELCMLIB  UNIT=name,VOLNO=serial
```

If CIRB is desired:

```
MACLIB
```

## Timing Options

TIME Optional For: MFT systems

INTERVAL Optional For: All systems

INTERVAL or JOBSTEP Required For: MVT (including M65MP) systems

JOBSTEP Optional For: MFT and MVT (including M65MP) systems

These options may be selected only when an interval timer is included in the central processing unit. Three levels of interval timer support are specified:

- Time (TIME), which provides the complete facilities of the TIME macro instruction (the date and time of day).

- Internal Timing (INTERVAL), which provides the ability to request, check, and cancel time intervals with the STIMER and TTIMER macro instructions plus the ability to change the time at midnight. This level of support also includes the facilities provided by the TIME macro instruction.
- Job Step Timing (JOBSTEP), which provides the ability to time each job step and enforce the time limits. This level of support also includes the facilities provided by the TIME, STIMER, and TTIMER macro instructions. (See "Job Step Timing Option" in this section.)

#### How to Use

If no timing options are specified, then just the time of day is available.

If the EMULATOR macro instruction is specified, TIMER=INTERVAL or JOBSTEP must be specified.

If system management facilities (SMF) are to be included in an MFT or MVT system, TIMER=JOBSTEP must be specified in the SUPRVSOR.

For MVT and M65MP systems, INTERVAL or JOBSTEP must be specified. The storage required is included in the basic fixed requirement.

An MFT system must have one of the time options selected if the "Timestamp" feature of the Generalized Trace Facility and the START/STOP capability of the IMDPRDMP service aid EDIT statement are desired.

#### Where to Find Information

Further information about timing options can be found in:

- Storage requirements -- OS Storage Estimates
- TIMER, STIMER, and TTIMER macro instructions -- OS Supervisor Services and Macro Instructions

#### What to Code When Generating a System

The following code is required to specify any of these options.

For MFT:

```
SUPRVSOR    ...,TIMER=    { TIMER
                           { INTERVAL
                           { JOBSTEP }
```

For MVT, including M65MP:

```
SUPRVSOR    ...,TIMER=    { INTERVAL
                           { JOBSTEP }
```

## Time-Sharing Option (TSO)

Optional For: M65MP and MVT systems

The time sharing option (TSO) adds general-purpose time sharing to the facilities already available through MVT. The option provides a command language that can be entered at remote terminals -- printer-keyboard and keyboard-screen devices connected to telephone or other communications lines to a computer. The command language provides the following capabilities and facilities:

- Data management
- Program development
- Program execution
- System control

### How to Use

The generation of the TSO functions, with the exception of the TSO program products, takes place during a complete operating system generation. If you desire, the TSO generation process can be divided into two generations with certain macro instructions and functions specified in a complete operating system generation and the others, at some later date, in a processor generation.

The following is a list of TSO related generating system data sets and the DLIB on which they are located.

#### DLIB04

- SYS1.TCAMMAC -- The distribution library for the TCAM system macro instructions

#### DLIB06

- SYS1.TSOGEN -- The distribution library that contains the macro definitions of the TSO system generation macro instructions used during Stage II
- SYS1.TSOMAC -- The distribution library for TSO system macro instructions
- SYS1CI555 -- The component library for TSO modules
- SYS1.DCMDLIB -- The distribution library for the TSO command library data set
- SYS1.DUADS -- The distribution library with the initial TSO user attributes
- SYS1DHELP -- The distribution library for the TSO HELP data set

#### DLIB03

- SYS1.CQ548 -- The component library for TCAM modules

### Where to Find Information

Further information about TSO can be found in:

- System configuration -- OS TSO Guide

- Storage requirements -- OS Storage Estimates
- TCAM information -- OS TCAM
- TSO commands -- OS TSO Command Language Reference
- TSO data sets -- OS System Generation, OS TSO Command Language Reference
- TSO facilities -- OS TSO Guide

#### What to Code When Generating a System

The following is a list and description of the required and optional TSO data sets for a new TSO system:

- SYS1.CMDLIB -- The TSO command processing programs reside in this data set.
- SYS1.HELP -- The TSO HELP function with a member for each TSO command resides in this data set.
- SYS1.UADS -- This data set will contain attribute entries for each authorized TSO user.
- SYS1.BROADCAST -- This data set will store two types of TSO messages.

Notices - messages available for all users of the system

Mail - messages available for specific users of the system

The following code is required to include TSO in the new system:

GENTSO

This macro instruction is used in place of the GENERATE macro instruction. See "System Generation Macro Instructions" for a discussion of GENTSO:

```

CMDLIB
EDIT          BLOCK=(name-n[ ,name-n]...)
OUTPUT        CLASS=( [ classname1 ][ ,classname2 ])
SCHEDULR      ... ,OPTIONS=(TSO) , [ SUBMITQ=maxno ]
TELCMLIB      [ UNIT=name ]
               [ VOLNO=serial ]

TSOOPTION     [ LOGTIME=time ]
               [ LOGLINE=integer ]
               [ BCLMT=integer ]

UADS          [ UNIT=name ]
               [ VOLNO=serial ]

```

To include the HELP function:

```

HELP          [ UNIT=name ]
               [ VOLNO=serial ]

```

To include syntax checking:

```

CHECKER      ...

```

## Time-Slicing

Optional For: All systems

When time-slicing is included in the system, you can establish a group of partitions or tasks (called a time-slice group) that are to share the use of the CPU, each for the same fixed interval of time. For MFT systems, this is done for jobs scheduled into a group of consecutive partitions that have been defined as the partitions to be used for time-slicing.

The priority of a job can be changed by the CHAP macro instruction so that its priority will fall within the range of the priorities for the partitions defined for time-slicing. This job will then be handled in the same manner as the other jobs in the time-slice group. For MVT systems, this is done for all tasks that have a given priority.

When a member of the time-slice group has been active for the fixed interval of time, it is interrupted and control is given to another member of the group, which will, in turn, have control of the CPU for the same length of time. In this way, all member tasks are given an equal slice of CPU time and no task or partition within the group can monopolize the CPU. In MVT, only tasks in the group are time-sliced, and they are time-sliced only when the priority level of the group is the highest priority level that has a task ready. Dispatching of tasks continues within the group until:

- All tasks are in a waiting state, or
- A task of higher priority than the one assigned to the group becomes ready.

In MFT, only partitions that are assigned to the time-slice group will be time-sliced, and they are time-sliced only when the first partition in the group is the highest priority ready task. Dispatching of the partitions continues within the group until all the partitions are in a waiting state, or until a partition with a higher priority is in a ready state.

### How to Use

The group of tasks to be time-sliced (selected by priority or partition range) and the length of the time-slice are specified at system generation time in the CTRLPROG macro instruction. This can be modified in MVT at system initialization time and in MFT through the DEFINE command. Any task or partition in the system that is not defined within the time-slice group is dispatched under the current priority structure; that is, the task or partition is dispatched only when it is the highest priority ready task or partition on the TCB queue. The maximum number of milliseconds, a number specified from a range of 20 to 9999, is the amount of time that each ready task is to have control of the CPU during one pass through the group.

For MVT (including M65MP) systems, any of the priorities, 0 to 13, can be used.

### Where to Find Information

Further information on time-slicing can be found in:

- Storage requirements -- OS Storage Estimates
- Detailed description of use -- OS MFT Guide, OS MVT Guide
- System generation macro instructions -- OS System Generation



- Operator commands -- OS Operator's Reference
- CHAP macro instruction -- OS MFT Guide, OS MVT Guide, OS Supervisor Services and Macro Instructions

### What to Code When Generating a System

The following coding is required to specify this option:

```
For MFT --          CTRLPROG  ...,TMSLICE=(Pn-Pm,SLC-time)
For MVT or M65MP -- CTRLPROG  ...,TMSLICE=(priority,SLC-time
                        [ ,priority,SLC-time]...)
```

### Trace Option

Optional For: All systems

A tracing routine that aids in debugging and maintenance can be added to the new system.

The trace option and the generalized trace facility (GTF) component of the control program can coexist in the same system, however the trace option functions are inhibited while GTF is active.

### How to Use

The tracing routine stores information pertaining to start I/O (SIO) instruction execution, supervisor (SVC) interruptions, external interruptions, program check interruptions, and I/O interruptions in the trace table. When the table has been completely filled, the succeeding entries overlay the existing ones.

During system generation, only the size of the table is specified. However, when this system generation parameter is specified, the trace program routines are also included as part of the control program.

### Where to Find Information

Further information about the trace option can be found in:

- Storage requirements -- OS Storage Estimates
- Trace table size -- OS System Generation, OS Storage Estimates
- Description -- OS MFT Guide, OS MVT Guide
- Trace routine -- OS MFT Guide, OS MVT Guide
- SIO, I/O, and SVC interruptions -- OS Principles of Operation
- Entry format -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required to specify the number of entries in the table, which will also include the routines in the control program.

```
SUPRVSOR...,TRACE=number
```

## **Transient SVC Table Made Resident**

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

The relative track address (TTR) of all transient supervisor (SVC) routines are included as part of the resident table of control program SVC routines. (See the description in "Types 3 and 4 SVC Routines Made Resident" in a succeeding section.)

### How to Use

If types 3 and 4 SVC routines are being made resident, this option must be specified also.

During a nucleus generation, this option can be added to or deleted from the options specified during a complete operating system generation.

### Where to Find Information

Further information about the transient SVC table being made resident can be found in:

- Amount of storage required -- OS Storage Estimates
- How to use the tracing routine -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required in order to specify this option:

```
SUPRVSOR ...,OPTIONS=(TRSVCTBL)
```

If types 3 and 4 SVC routines are also to be made resident:

```
SUPRVSOR ...,RESIDNT=(TRSVC),OPTIONS=(TRSVCTBL)
```

## **Type 3 and 4 SVC Routines Made Resident**

Optional For: All systems

Modules of types 3 and 4 supervisor (SVC) routines can be made permanently resident in the fixed area of storage.

## How to Use

Types 3 and 4 SVC modules are loaded and made resident at IPL time. When this option is specified, the transient SVC table option must also be specified. The SVC table is a table containing the relative track addresses of all transient SVCs. This table is also stored in the resident portion of the control program.

The names and sizes of types 3 and 4 SVC routine modules are given in OS Storage Estimates. (See also the preceding description "Transient SVC Table Made Resident.")

During a nucleus generation, this option can be added or deleted from the options specified during a complete operating system generation, but the transient SVC table option will have to be specified the same way it was specified in the last complete operating system generation.

## Where to Find Information

Further information about Types 3 and 4 SVC routines being made resident can be found in:

- Storage requirements -- OS Storage Estimates

## What to Code When Generating a System

The following code is required in order to specify this option:

```
SUPRVSR ...,RESIDNT=(TR SVC),OPTIONS=(TR SVCTBL)
```

## **User-Added SVC Routines**

Optional For: All systems

User-written supervisor (SVC) routines can be added to the control program.

## How to Use

All of the SVC routines, whether they are to be transient or resident, must be listed in the operand of the SVCTABLE system generation macro instruction.

Any resident SVC routines that are to be added must be specified in the system generation RESMODS macro instruction. The fixed storage requirement is increased by the total of the sizes of the routines that are going to be added plus the size of the control information.

Any transient SVC routines that are to be added must be specified in the SVCLIB system generation macro instruction in the operand. In this case, only the size of the control information is added to the fixed storage requirements.

Nonstandard error routines can be one of the types of routines that are added. User-written routines must have a value from 220 to 229. This value is the suffix of the name IGEOC by which the error routine is named in SYS1.SVCLIB.

## Where to Find Information

Further information about user-added SVC routines can be found in:

- SVC routines, writing -- OS MFT Guide, OS MVT Guide

## What to Code When Generating a System

The following code is required in order to specify this option:

```
SVCTABLE      operand[,operand]...
RESMODS       PDS=SYS1.name,MEMBERS=(name[,name]...)
SVCLIB        PDS=SYS1.name,MEMBERS=(name[,name]...)
```

If nonstandard error routines are specified, the following must also be coded:

```
IODEVICE     ...,ERRTAB=nnn
```

## **Validity Check Option**

Standard For: MVT (including M65MP) systems

Optional For: MFT systems

Extra validity checking can be added to the new system to determine whether addresses are located within proper boundaries. The validity checking is provided for the WAIT, POST, and GETMAIN/FREEMAIN modules. The checking for WAIT also checks for the number of events.

## How to Use

For MFT systems, this option is specified in the SUPRVSOR system generation macro instruction. For MVT or M65MP systems, this option is a standard feature and the storage required is included in their basic fixed requirement.

During a nucleus generation, this option can be added to or deleted from the options specified during a complete operating system generation.

## Where to Find Information

Further information about the validity check option can be found in:

- Amount of storage required -- OS Storage Estimates

## What to Code When Generating a System

The following code is required in order to specify this option by itself:

```
SUPRVSOR     ...,OPTIONS=(VALIDCHK)
```

## Volume Statistics Facility

Optional For: All systems

Volume statistics may be kept only for magnetic tape volumes with or without labels. There are two types, either or both of which can be specified at system generation time in the SCHEDULR macro instruction.

Error statistics by volume (ESV) is intended primarily to be used with labeled volumes, but will handle an unlabeled volume if the serial number is given to the operating system. Statistics about the number of read or write errors and the system and unit on which the volume is located are recorded.

Error volume analysis (EVA) is intended primarily to be used for unlabeled or nonstandard labeled volumes. It monitors the number of read or write errors based on the limits provided at system generation time.

These options can be specified for all types of control programs.

### How to Use

Error statistics by volume (ESV) collects a set of statistics for each labeled tape volume whenever the volume is open. An unlabeled tape volume can be handled if the serial number has been supplied to the operating system.

If ESV=SMF is specified at system generation time, the statistics are accumulated on the system management facility (SMF) data sets SYS1.MANX and SYS1.MANY. ACCTRTN=SMF should be specified in the SCHEDULR macro instruction, but if it is not coded, it is assumed. If any subparameter for ACCTRTN other than SMF is specified, it is ignored and SMF is assumed. The TIMER keyword parameter is also required in the SUPRVSOR macro instruction. The IFHSTATR utility program is used to print the ESV records, record 21, from an SMF data set on magnetic tape. If SYS1.MANX is on tape, no transfer is required, but if the SMF data sets are on a direct-access device, you must dump them onto tape in order to be able to extract the ESV records. The SMF dump program, IFASMFDP, is used to transfer the data from SYS1.MANX and SYS1.MANY to tape.

If ESV=CON is specified or if ESV is not coded, an abridged version of the statistics is printed on the console. This occurs at end-of-volume or when the tape volume is closed.

You can provide your own recording routine. ESV=CON must be specified, or the keyword parameter can be omitted because the default is CON. The UCBS, in the proper format, are constructed at system generation time. You can provide your own access method, using SVC 91, specify your own record format, and select your own recording data set. If you use the record 21 format instead of your own, you can use the IFHSTATR utility to print out the statistics.

Error volume analysis (EVA) acts as a monitor about the number of read and write errors for unlabeled or nonstandard labeled tape volumes. You provide the maximum limits for read errors and/or write errors and, if the maximum is reached or exceeded, a message, IEA620I, is printed on the console.

## Where to Find Information

Further information about volume statistics functions can be found in:

- SVC routines, writing -- OS MFT Guide, OS MVT Guide
- System management facility (SMF) -- OS SMF, OS System Generation
- SMF data sets -- OS System Generation, OS SMF
- Tape labels -- OS Tape Labels
- Storage requirements -- OS Storage Estimates
- IFASMFDP utility program -- OS SMF
- IFHSTATR utility program -- OS Utilities

## What to Code When Generating a System

The following code is required to have the error recording routines included in the system:

```
SCHEDULR      ..., [ESV={CON  
                  {SMF}  
                  [,EVA=(n1,n2) ]
```

For MFT and MVT (including M65MP) if ESV=SMF then:

```
SCHEDULR      ...,ESV=SMF,  
              ACCTRTN=SMF  
  
SUPRVSOR      ...,TIMER=JOBSTEP
```

## Data Sets

This section contains the detailed information about the required and optional data sets for the generating operating system and the new or modified operating system. The contents and special notes, such as the need for cataloging and release level, are discussed for each required data set in the generating system. A table, listing the components, the distribution data sets, whether they are required or optional, and where they are placed, is also included.

The contents, type, specification considerations, and special notes are discussed for each system data set in the new or modified operating system.





## Required and Optional Data Sets for Generating Systems

This section contains detailed information about the required and optional data sets for the generating operating system.

## SYS1.GENLIB

This data set is required in any generating system.

### Contents

This data set contains the macro definitions of the system generation macro instructions used during Stage I.

### Special Notes

#### Cataloging:

This data set must be cataloged in the system catalog of the operating system being used as the generating system.

#### Release Level:

This data set, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and the various component libraries must all be of the same release level. They can be used only to generate an operating system of the same release level as themselves.

If the generating system is release 18 or earlier, the following must be performed before beginning Stage II:

- Copy the IEBCOPY utility program from SYS1.LINKLIB in the starter operating system to SYS1.LINKLIB in the generation system.
- Copy the module named IGG019C8 from SYS1.SVCLIB in the starter operating system to SYS1.SVCLIB in the generating system.

## SYS1.MODGEN, SYS1.MODGEN2

This data set is required in any generating system.

### Contents

This data set contains the macro definitions of the system generation macro instructions used during Stage II assemblies.

### Special Notes

#### Cataloging:

This data set must be cataloged in the system catalog of the operating system being used as the generating system.

#### Release Level:

This data set, SYS1.GENLIB, SYS1.TSOGEN, and the various component libraries must all be of the same release level. They can be used only to generate an operating system of the same release level as themselves.

If the generating system is not release 20, you must use Assembler F from the release 20 distribution package.

## **SYS1.TSOGEN**

This data set is required in any generating system that includes TSO.

### **Contents**

This library contains the macro definitions of the TSO system generation macro instructions used during Stage II assemblies.

### **Special Notes**

#### **Cataloging:**

This data set must be cataloged in the system catalog of the operating system being used as the generating system.

#### **Release Level:**

This data set, SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the various component libraries must all be of the same release level. They can be used only to generate an operating system of the same release level as themselves.

If the generating system is not release 20, you must run Assembler F from the release 20 distribution package.

## Component Data Sets

Some or all of these data sets can be selected.

### Contents

The members of these data sets are the load modules from which the components of an operating system are generated.

### Special Note

#### Required Use:

The following table shows the names of all the components that can be included in an operating system, and the data set from which the system generation program will take them. Also indicated is where the component will be found in the new or modified system. If "Optional" is indicated in the "Required" column, the component library data set need only be cataloged in the generating system when the component is to be generated in the new system.

Component	Component Libraries		
	Obtained From:	Required:	Placed In:
ALGOL	SYS1.AL531	Optional	SYS1.LINKLIB
ALGOL Library	SYS1.LM532	Optional	SYS1.ALGLIB
American National Standard COBOL	SYS1.CB545	Optional	SYS1.LINKLIB
American National Standard COBOL Library	SYS1.LM546	Optional	SYS1.COBLIB
Assembler F	SYS1.AS037	Optional	SYS1.LINKLIB
BDAM	SYS1.DM509	Optional	SYS1.SVCLIB SYS1.NUCLEUS
BTAM	SYS1.CQ513	Optional	SYS1.SVCLIB SYS1.NUCLEUS SYS1.TELCMLIB
COBOL E	SYS1.CO503	Optional	SYS1.LINKLIB
COBOL E Library	SYS1.COBLIB	Optional	SYS1.COBLIB
Control Program	SYS1.CI505	Required <sup>1</sup>	SYS1.NUCLEUS SYS1.SVCLIB SYS1.LINKLIB
Conversational Remote Job Entry	SYS1.RC551	Optional	SYS1.LINKL SYS1.TELCMLIB
Data Management (primary routines)	SYS1.DM508	Required <sup>1</sup>	SYS1.LINKLIB SYS1.NUCLEUS SYS1.SVCLIB
Data Set Utilities	SYS1.UT506	Required <sup>1</sup>	SYS1.LINKLIB
Emulators Integrated	SYS1.CI505	Optional	SYS1.SVCLIB SYS1.NUCLEUS
FORTTRAN G	SYS1.FO520	Optional	SYS1.LINKLIB
FORTTRAN H	SYS1.FO500	Optional	SYS1.LINKLIB
FORTTRAN Library	SYS1.LM501 SYS1.FORTLIB	Optional	SYS1.FORTLIB
FORTTRAN Syntax Checker	SYS1.FO550	Optional	SYS1.LINKLIB
Generalized Trace Facility	SYS1.CI505 SYS1.CI535	Required	SYS1.LINKLIB SYS1.NUCLEUS
Graphic Job Processor	SYS1.RC541	Optional	SYS1.SVCLIB SYS1.LINKLIB

Figure 76 (Part 1 of 3). Components and Generating System Data Sets

Component	Component Libraries		
	Obtained From:	Required:	Placed In:
Graphic Programming Services	SYS1.I0523	Optional	SYS1.LINKLIB SYS1.SVCLIB SYS1.NUCLEUS
Graphic Subroutine Package	SYS1.LM537	Optional	SYS1.LINKLIB SYS1.FORTLIB SYS1.PL1LIB SYS1.COBLIB
ISAM	SYS1.I0526	Optional	SYS1.SVCLIB SYS1.NUCLEUS
Linkage Editor F	SYS1.ED521	Optional	SYS1.LINKLIB
Loader Program	SYS1.LD547	Optional	SYS1.LINKLIB
Macro Library	SYS1.MACLIB SYS1.TSOMAC SYS1.TCAMMAC	Required Optional Optional	SYS1.MACLIB
Multiple Console Support	SYS1.CI505 SYS1.CI535 SYS1.CQ513	Optional	SYS1.NUCLEUS SYS1.SVCLIB SYS1.LINKLIB
MVT and M65MP	SYS1.CI535	Optional	SYS1.NUCLEUS SYS1.SVCLIB SYS1.LINKLIB
Online Test Executive Program	SYS1.DN533	Optional	SYS1.SVCLIB SYS1.LINKLIB SYS1.NUCLEUS
Parameter Library	SYS1.PARMLIB	Required	SYS1.PARMLIB
PL/1	SYS1.NL511	Optional	SYS1.LINKLIB
PL/1 Library	SYS1.LM512 SYS1.PL1LIB	Optional	SYS1.PL1LIB SYS1.LINKLIB
PL/1 Syntax Checker	SYS1.PL552	Optional	SYS1.LINKLIB

Figure 76 (Part 2 of 3). Components and Generating System Data Sets

Component	Component Libraries		
	Obtained From:	Required:	Placed In:
Procedure Library	SYS1.PROCLIB	Required	SYS1.PROCLIB
QTAM	SYS1.CQ519	Optional	SYS1.TELCMLIB SYS1.NUCLEUS SYS1.SVCLIB
Recovery Management	SYS1.DN539	Required <sup>1</sup>	SYS1.LINKLIB SYS1.SVCLIB
Remote Job Entry	SYS1.RC536	Optional	SYS1.LINKLIB SYS1.TELCMLIB
RPG	SYS1.RG038	Optional	SYS1.LINKLIB
Satellite Graphic Job Processor	SYS1.RC543	Optional	SYS1.LINKLIB
SER	SYS1.DN527	Required <sup>1</sup>	SYS1.LINKLIB SYS1.NUCLEUS
Service Aids	SYS1.DN554 SYS1.MACLIB	Optional	SYS1.LINKLIB SYS1.MACLIB
Sort/Merge	SYS1.SM023	Optional	SYS1.LINKLIB
Sort Library	SYS1.SORTLIB SYS1.SM023	Optional	SYS1.SORTLIB
System/360-1130 Data Transmission	SYS1.LM542	Optional	SYS1.LINKLIB SYS1.FORTLIB
TCAM	SYS1.CQ548	Optional	SYS1.SVCLIB SYS1.LINKLIB SYS1.NUCLEUS SYS1.TELCMLIB
Telecommunications Library	SYS1.RC536 SYS1.CQ513 SYS1.CQ519 SYS1.CQ548	Optional	SYS1.TELCMLIB
TSO	SYS1.CI555	Optional	SYS1.LINKLIB SYS1.SVCLIB SYS1.NUCLEUS
	SYS1.TSOGEN	Optional	
	SYS1.DUADS	Optional	SYS1.UADS
	SYS1.DHELP	Optional	SYS1.HELP
	SYS1.DCMDLIB	Optional	SYS1.CMDLIB

<sup>1</sup>Not required for Processor/Library generation

Figure 76 (Part 3 of 3). Components and Generating System Data Sets



## System Data Sets

This part of the section contains detailed information about all the system data sets that can be specified for a new or modified operating system during system generation. These data set descriptions are arranged in alphabetical order by the fully qualified data set names. The following information is supplied for each of the data sets:

- Whether the data set is required or optional
- What the data set contains
- Whether the data set is sequential or partitioned
- What you must know when you are specifying the data set for the new or modified system

In some cases, special notes about the use of the data set are provided. Figure 77 is a summary of the system data sets.

System Data Set	Required	Type	System		DCB Subparameters	Cataloged
			Residence	Allocation		
SYSCTLG	Yes	Seq.	Yes	Yes	None <sup>11</sup>	No
SYS1.NUCLEUS	Yes	PDS	Yes	No	None	Optional
SYS1.SVCLIB <sup>1</sup>	Yes	PDS	Yes	Yes	DSORG=POU, RECFM=U, BLKSIZE=1024 <sup>2</sup>	Recommended
SYS1.LOGREC <sup>3</sup>	Yes	Seq.	Yes	No	None	No
SYS1.LINKLIB <sup>1</sup>	Yes	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Yes
SYS1.PARMLIB	Yes	PDS	Optional	No	RECFM=F, BLKSIZE=80	Recommended <sup>5</sup>
SYS1.PROCLIB	Yes	PDS	Optional	Yes	RECFM=FB, LRECL=80, BLKSIZE=3360 <sup>6</sup>	Recommended
SYS1.SYSJOBQE <sup>12</sup>	Yes	Seq.	Optional	No	None	Recommended
SYS1.IMAGELIB	No	PDS	Optional	No	RECFM=U, BLKSIZE=1024 <sup>2</sup>	Yes
SYS1.MACLIB	No	PDS	Optional	Yes	RECFM=FB, LRECL=80, BLKSIZE=3360 <sup>6</sup>	Recommended
SYS1.SORTLIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.ALGLIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.COBLIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.PORTLIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.PL1LIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.TELCHLIB	No	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Recommended
SYS1.SYSVLOGX	No	Seq.	Optional	No	RECFM=VB, BLKSIZE=3625 <sup>4 7</sup>	Yes
SYS1.SYSVLOGY	No	Seq.	Optional	No	RECFM=VB, BLKSIZE=3625 <sup>4 7</sup>	Yes
SYS1.ROLLOUT <sup>8</sup>	No	Seq.	Optional	No	None	Yes
SYS1.ASRLIB <sup>3</sup>	No	Seq.	Yes	No	None	No
SYS1.ACCT	No	Seq.	Optional	No	RECFM=U, BLKSIZE=3625 <sup>4</sup>	No
SYS1.MANX	No	Seq.	Optional	No	None	Recommended
SYS1.MANY	No	Seq.	Optional	No	None	Recommended
SYS1.DCMLIB	No <sup>13</sup>	PDS	Optional	No	None	Yes
SYS1.DUMP <sup>9</sup>	No	Seq.	Optional	No	None	Yes
SYS1.CMDLIB	Yes <sup>10</sup>	PDS	Optional	Yes	RECFM=U, BLKSIZE=3625 <sup>4</sup>	Yes
SYS1.HELP	No	PDS	Optional	Yes	RECFM=FB, BLKSIZE=3360 <sup>6</sup>	Yes
SYS1.UADS	Yes <sup>10</sup>	PDS	Optional	Yes	DSORG=PO, RECFM=FB, LRECL=80, BLKSIZE=000	Yes
SYS1.BROADCAST	Yes <sup>10</sup>	Direct	Optional	No	None	Yes

- 1 Space should be allocated by cylinders.
- 2 BLKSIZE=3072 must be specified if the new system is to have the Machine Check Handler for Model 65 (MCH) option.
- 3 Space must not be allocated for this data set by the user.
- 4 BLKSIZE=3625 applies if the system data set resides on a 2311. BLKSIZE=20483 must be specified for a 2301, BLKSIZE=4892 for a 2303, BLKSIZE=14136 for a 2305-1, BLKSIZE=14660 for a 2305-2, BLKSIZE=7294 for a 2314, and BLKSIZE=13030 for a 3330.
- 5 SYS1.PARMLIB must be cataloged if it does not reside on the system-residence volume.
- 6 The value of BLKSIZE must be a multiple of 80 which is less than or equal to 3600 for a 2311, 20480 for a 2301, 14080 for a 2305-1, 14640 for a 2305-2, 4880 for a 2303, 7280 for a 2314, or 12960 for a 3330.
- 7 BLKSIZE must be greater than half the amount of storage allocated for the system log buffer by means of the WTLBFRS parameter of the SCHEDULR macro instruction.
- 8 Space need not be allocated for this data set by the user.
- 9 This data set can only reside on 2311, 2314, 2301, 2302, 2303, 2305, or 3330 direct-access devices or on any 2400 or 3400 series magnetic tape drive.
- 10 This data set is required in a TSO system.
- 11 Should not be specified. If the DSORG subparameter is specified, the catalog cannot be moved or copied by the IEHMOVE utility program.
- 12 SYS1.SYSJOBQE must have a contiguous primary space allocation.
- 13 Required only if transient DCM support or PFK command entry is specified.

**Note:** If linkage editor F is used in the generating system, the blocksize found in the DSCB for the data sets whose record form is specified as U (RECFM=U) will be the maximum blocksize for the device being used. This is not necessarily the size of the current record.

Figure 77. System Data Sets

## SYSCTLG

### Contents

This data set is required for all operating systems.

This data set contains pointers to all the cataloged data sets in an operating system.

#### Creation:

The initial entries are created by the system generation program.

### Type

This is a sequential data set.

### Specifying for the New System

#### Location:

This data set must be on the system-residence volume.

Only the index (SYS1.) of the system catalog must reside on the system-residence volume catalog.

#### DD Statement:

The standard format is used.

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

Secondary allocations can be specified.

#### Catalog Statement:

This statement must not be coded for this data set.

# SYS1.ACCT

This data set is required if user-written accounting routines are in the system.

**Contents** User-written accounting routines store accounting information in this data set.

**Type** This is a sequential data set.

**Specifying for the New System** Location:

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

DD Statement:

Space must be allocated.

Secondary extents cannot be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  7294 if on a 2314
                  13030 if on a 3330
```

Catalog Statement:

This statement must not be coded for this data set.

**Special Notes** Two Data Sets:

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

Required:

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

Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

# SYS1.ALGLIB

This data set is required when the ALGOL compiler is in the system.

## Contents

The members of this data set are ALGOL subroutines in load module form.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  7294 if on a 2314
                  13030 if on a 3330
```

### Catalog Statement:

You should catalog this data set.

## Special Note

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

## SYS1.ASRLIB

This data set is required only if the machine-check handler (MCH) is in the system.

### Contents

This data set contains the refreshable nucleus modules for the machine-check handler routines for the S/360 Models 65 and 85 and the S/370 Models 155 and 165.

### Type

This is a sequential data set.

### Specifying for the New System

#### Location:

This data set must be on the system-residence volume.

#### DD Statement:

This statement must not be coded for this data set.

#### Catalog Statement:

This statement must not be coded for this data set.

### Special Notes

#### Space Allocation and Cataloging:

The system generation program allocates space for and catalogs this data set on the system-residence volume during system generation.

#### Reinitialization:

This data set is reinitialized by NIP after each IPL.

# SYS1.BROADCAST

This data set is required for TSO systems.

Contents This data set stores two types of TSO messages.

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

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

Type This is a direct data set.

Specifying for the New or Modified System Location:

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

DD Statement:

The standard format is used.

Space must be allocated.

Secondary allocations cannot be specified.

Catalog Statement:

This data set should be cataloged.

Special Notes Initialization:

After system generation, the first use of the data set by:

- User SEND CP,
- Operator SEND,
- Account/SEND interface

will cause the data set to be formatted and initialized. The data set will not be reformatted until another system generation.

Information:

For further information on this data set, see OS TSO Command Language.

# SYS1.COMDLIB

This data set is required if you want a TSO system.

## Contents

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

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  4892 if on a 2303
                  7294 if on a 2314
                  13030 if on a 3330
```

### Catalog Statement:

You should catalog this data set.



## SYS1.COBLIB

This data set is required if the COBOL compiler is in the system.

### Contents

The members of this data set are COBOL subroutines in load module form.

### Type

This is a partitioned data set.

### Specifying for the New or Modified System

#### Location:

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

#### DD Statement:

The standard format is used.  
Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  4892 if on a 2303
                  7294 if on a 2314
                  13030 if on a 3330
```

#### Catalog Statement:

You should catalog this data set.

### Special Notes

#### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB will be the maximum blocksize for the device being used.

#### Generating System:

For COBOL E: SYS1.COBLIB must exist as a cataloged partitioned data set in the generating system and must be of the same release level as the SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2 and component libraries being used.

For American National Standard COBOL: The SYS1.CB545 and SYS1.LM546 component libraries must be cataloged in the generating system.

User-written Programs: If you want any of your own programs in the new or modified SYS1.COBLIB, you must add them after system generation is over. Be sure you allocate enough additional space in this data set for your own programs.

## SYS1.DCMLIB

This data set is required if the transient display module function is specified in the SCHEDULR or SECONSLE macro instructions.

### Contents

This data set is used to hold portions of the display control modules that are not resident in main storage.

### Type

This is a partitioned data set.

### Specifying for the New System

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

DD Statement: Space must be allocated for this data set. Secondary allocation cannot be made since the space must be contiguous.

Catalog Statement: You should catalog this data set.

# SYS1.DUMP

This data set is required if you want a core image dump recorded.

Contents This data set is used to contain one core image dump recorded by the ABEND or ABTERM routines.

Type This is a sequential data set.

Specifying for the New System Location:

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

The volume that contains this data set can be on:

- 2301 drum
- 2302 disk
- 2303 drum
- 2305-1 FHSF (fixed-head storage facility)
- 2305-2 FHSF
- 2311 disk
- 2314 direct-access storage or any
- 2400 series magnetic tape unit
- 3400 series magnetic tape unit
- 3330 direct-access storage

DD Statement:

This statement is used only if this data set is on a direct-access volume and you choose to allocate space in which case you must also write an EOF record as the first record in the data set.

The standard format is used.

Secondary allocations cannot be specified. If you do not allocate this data set and it is cataloged, NIP will allocate it for you and write the EOF record.

Catalog Statement:

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

Special Notes Magnetic Tape:

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

Information:

For further detailed information about core image dumps, see Guide to Reading OS System Dumps.

# SYS1.FORTLIB

This data set is optional.

## Contents

The members of this data set are FORTRAN subprograms in load module form.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  7294 if on a 2314
                  13030 if on a 3330
```

### Catalog Statement:

You should catalog this data set.

## Special Notes

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

### Generating System:

If you want to transfer your own programs from the generating system to the new or modified system, catalog the SYS1.FORTLIB with those programs in the generating system catalog. If there are no such programs to transfer, catalog only the appropriate component libraries in the generating system.

## SYS1.HELP

This data set is optional for TSO systems. It is required if the TSO HELP command is to be used.

### Contents

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

### Type

This is a partitioned data set.

### Specifying for the New or Modified System

#### Location:

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

#### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=FB,LRECL=80,  
BLKSIZE= a multiple of 80, less than or  
equal to:  
    80 if unblocked  
    3600 for a 2301  
    20480 for a 2301  
    4880 for a 2303  
    14080 for a 2305-1  
    14640 for a 2305-2  
    7280 for a 2314  
    12960 for a 3330
```

#### Catalog Statement:

You should catalog this data set.

### Special Note

#### Information:

For further information on this data set, see OS TSO Command Language.

# SYS1.IMAGELIB

This data set is required if a 1403 printer with the universal character set (UNVCHSET) feature or a 3211 printer is in the system.

## Contents

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

## Type

This is a partitioned data set.

## Specifying for the New System

### Location:

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

### DD Statement:

Space must be allocated.

No secondary allocation is allowed.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE=1024
                3072 if
MCH is specified for the Model 65.
```

### Catalog Statement:

This data set must be cataloged if a 1403 with the universal character set feature or a 3211 printer is part of the system.

## Special Note

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

# SYS1.LINKLIB

This data set is required in all operating systems.

## Contents

The members of this data set are programs and routines that are referred to by XCTL, ATTACH, LINK, or LOAD macro instructions. Nonresident operating system programs, such as the COBOL compiler, are contained in this data set. The S/360 Model 85 Machine-Check Handler Nucleus (IGFMCH10) and Initialization (IGFMCHF0) modules and S/370 Machine-Check Handler Nucleus (IGFMCHE0) and Initialization (IGFMCHF0) modules are also included.

This data set must contain an assembler-language processor with an alias of ASMBLR and linkage editor with an alias of IEWL. It must also include the utilities IEBCOPY, IEHLIST, IEHPRGM, IEHIOSUP, IEBUPDTE, and the IFCEREPO (EREP) and IFCDIP00 programs.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

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

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on 2311
                  20483 if on 2301
                  4892 if on 2303
                  14136 if on 2305-1
                  14660 if on 2305-2
                  7294 if on 2314
                  13030 if on 3330
```

### Catalog Statement:

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

## Special Notes

### Concatenation:

After system generation, up to sixteen data sets composed of up to sixteen extents each can be concatenated to this data set. The link library list LNKLST00 in SYS1.PARMLIB must be modified. The IEBUPDTE utility program can be used.

### IEBUPDAT:

If this system utility program is to be link-edited into this data set, you must allocate additional space for the utility program. See the "Including IEBUPDAT in the New System" section.

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.



## SYS1.LOGREC

	<p>This data set is required in all operating systems.</p>
Contents	<p>This data set is used to record statistical data about machine errors that is used by the IBM customer engineer.</p>
Type	<p>This is a sequential data set.</p>
Specifying for the New System	<p><u>Location:</u></p> <p>The system generation program initializes and allocates space for this data set on the system-residence volume.</p> <p>This data set must be on the system-residence volume.</p> <p><u>DD Statement:</u></p> <p>This statement must not be coded for this data set.</p> <p><u>Catalog Statement:</u></p> <p>This statement must not be coded for this data set.</p>
Special Notes	<p><u>Reinitialization:</u></p> <p>If this data set must be reinitialized after system generation, refer to the description of the IFCDIP00 utility program in <u>OS Service Aids</u>.</p> <p><u>Reallocation:</u></p> <p>The size of this data set can be increased or decreased after system generation by using IFCDIP00 to reallocate it. Refer to the publication <u>OS Service Aids</u> for a description of IFCDIP00.</p> <p><u>Message:</u></p> <p>A message is printed out during Stage II that provides information about this data set.</p>

# SYS1.MACLIB

This data set is optional in the new or modified system, but is required in the generating system.

## Contents

The members of this data set are the macro definitions for the system macro instructions used by the assembler-language processor.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=FB,LRECL=80,  
BLKSIZE= 3360 or a multiple of 80, which is  
          less than, or equal to:  
          80 if unblocked  
          3600 for a 2311  
          20480 for a 2301  
          14080 for a 2305-1  
          14640 for a 2305-2  
          7280 for a 2314  
          12960 for a 3330
```

### Catalog Statement:

You should catalog this data set.

## Special Notes

### Generating System:

This data set must be of the same release level as SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, SYS1.TSOGEN, and the component libraries being used.

If Assembler E is used in the generating system, this data set must be unblocked. This can be done by IEBCOPY if you specify in the SYSUT2 DD statement:

```
DCB=(RECFM=FB,BLKSIZE=80,LRECL=80)
```

## SYS1.MANX, SYS1.MANY

One or both of these data sets must be specified if the system management facility is in the system.

### Contents

One or both of these data sets contain the data collected by the system management facility (SMF) routines.

### Type

These are sequential data sets.

### Specifying for the New System

#### Location:

If a direct-access device is used, then both data sets must be defined. If magnetic tape is used, only SYS1.MANX can be defined.

One or both must reside on a permanently resident volume. It can be the system-residence volume.

#### DD Statement:

The standard format is used.

Secondary allocations, if they are coded, are ignored.

#### Catalog Statement:

You should catalog these data sets.

### Special Note

#### Magnetic Tape:

If magnetic tape is used, only one data set (SYS1.MANX) can be defined, but up to 20 volumes can be used each time the system is loaded.

## SYS1.NUCLEUS

This data set is required in all operating systems.

Contents                    This data set usually contains only one member, the resident portion (nucleus) of the control program.

Type                        This is a partitioned data set.

Specifying for the        Location:  
New or Modified System

This data set must be on the system-residence volume.

DD Statement:

The standard format is used.

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

There cannot be any secondary allocations.

Catalog Statement:

You should catalog this data set.

## SYS1.PARMLIB

This data set is required in all operating systems.

### Contents

The members of the parameter library are the RAM, BLDL, RSVC, and SMFDFLT system parameter lists used by the nucleus initialization program (NIP), the PRESRES list used by the master scheduler, and the LNKST00 list used to concatenate data sets to SYS1.LINKLIB. If SYS1.PARMLIB is copied from the starter operating system, it will contain the standard RAM list (IEAIGG00) and the optional RAM list (IEAIGG01).

### Type

This is a partitioned data set.

### Specifying for the New System

#### Location:

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

#### DD Statement:

Space must be allocated for this data set except when the standard IBM parameter list is not going to be used. In the latter case, a null allocation must be made.

Secondary allocations cannot be made because the space must be continuous.

The following DCB subparameters must be specified:

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

#### Catalog Statement:

This data set must be cataloged if it does not reside on the system-residence volume.

You should always catalog this data set.

# SYS1.PLILIB

This data set is required if the PL/I compiler is in the system.

## Contents

The members of this data set are PL/I subprograms in load module form.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14660 if on a 2305-2
                  7294 if on a 2314
                  13030 if on a 3330
```

### Catalog Statement:

You should catalog this data set.

## Special Notes

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

### Generating System:

If this data set is going to be in the new system, it must exist as a cataloged partitioned data set in the generating system catalog, and be of the same release level as SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the component libraries being used.

# SYS1.PROCLIB

This data set is required in all operating systems.

## Contents

The members of this data set include the cataloged procedures used to perform certain system functions, such as compile-link-edit-go.

## Type

This is a partitioned data set.

## Specifying for the New System

### Location:

This data set must be on a direct-access volume. It can be on the system-residence volume. (For MFT and MVT: The SCHEDULR PROCRES keyword parameter should not be specified if this data set is cataloged.)

### DD Statement:

Space must be allocated. A null allocation is not permitted.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=FB,LRECL=80,
BLKSIZE= 3360 or a multiple of 80, which is
          less than, or equal to:
           80 if unblocked
          3600 for a 2311
          20480 for a 2301
          4880 for a 2303
          14080 for a 2305-1
          14640 for a 2305-2
          7280 for a 2314
          12961 for a 3330
```

This data set must be blocked if the PROCLIB macro instruction is specified during system generation.

### Catalog Statement:

You should catalog this data set.

## Special Note

### SET Command:

If this data set is cataloged, or if it resides on the system-residence volume, you do not have to specify the PROC subparameter in the initial SET command during IPL for the new system. (For detailed information, see OS Operator's Reference.)

# SYS1.ROLLOUT

This data set is required if the MVT (including M65MP), rollout/rollin function is specified in the CTRLPROG macro instruction.

Contents

This data set is used to hold the information rolled out of main storage.

Type

This is a sequential data set.

Specifying for the New System

Location:

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

This data set can only reside on a 2301, 2305-1, 2305-2, 2311, 2314, 2321, or 3330.

DD Statement:

You do not have to specify a space allocation, but if you do, it must be one or more tracks.

The standard format is used.

Catalog Statement:

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

Special Note

IPL Time:

If you did not allocate space for this data set, the nucleus initialization program (NIP) will do so. The space will be allocated on the volume specified in the DD statement.

Failure:

If this data set was not cataloged or if not enough space was allocated, the rollout/rollin function will not be used.



## SYS1.SORTLIB

This data set is required if the sorting and merging functions are in the system.

### Contents

The members of this data set are the load modules from which the system produces a sort/merge program at execution time.

### Type

This is a partitioned data set.

### Specifying for the New or Modified System

#### Location:

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

#### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14640 if on a 2305-2
                  7294 if on a 2314
                  13030 if on a 3330
```

#### Catalog Statement:

You should catalog this data set.

### Special Notes

#### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB is the maximum blocksize for the device being used.

#### Generating System:

If this data set is going to be in the new system, it must exist as a cataloged partitioned data set in the generating system catalog and be of the same release level as SYS1.GENLIB, SYS1.MODGEN, SYS1.MODGEN2, and the component libraries being used.

# SYS1.SVCLIB

This data set is required in all operating systems.

## Contents

The members of this library are the nonresident SVC routines, the data management access methods, the system's standard error recovery (SER) routines, and the S/360 Model 85 and S/370 Machine-Check Handler nonresident modules.

## Type

This is a partitioned data set.

## Specifying for the New or Modified System

### Location:

This data set must be on the system-residence volume.

### DD Statement:

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

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

Secondary allocations can be specified.

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

The following DCB subparameters must be specified:

```
DSORG=POU,RECFM=U,BLKSIZE= 1024
                               3072 if
MCH is specified for Model 65
```

### Catalog Statement:

You should catalog this data set because some cataloged procedures use its name to refer to the system-residence volume.

## Special Note

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB will be the maximum blocksize for the device being used.

# SYS1.SYSJOBQE

This data set is required in all operating systems.

Contents This data set is used as a work area by the job scheduler.

Type This is a sequential data set.

Specifying for the New System Location:

This data set must be on a direct-access volume. It can be on the system-residence volume. The SCHEDULR JOBQRES keyword parameter should not be specified if this data set is cataloged.)

DD Statement:

For maximum efficiency, alternate tracks should not be used.

Space must be allocated for this data set, but no more than 1215 tracks can be allocated on a 2314 and no more than 745 tracks can be allocated on a 3330.

If this data set is date protected, operator intervention is required.

Secondary allocations cannot be specified.

Catalog Statement:

You should catalog this data set.

Special Notes

SET Command:

If this data set is cataloged, or if it resides on the system-residence volume, UNIT need not be specified in the job queue parameter during IPL when Q is being formatted.  
(For detailed information, see OS Operator's Reference.)

Allocation and Cataloging:

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

## SYS1.SYSVLOGX, SYS1.SYSVLOGY

These data sets are required if you want a system log in your system.

### Contents

These data sets contain system log data that consists of:

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

### Type

These are sequential data sets.

### Specifying for the New System

#### Location:

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

#### DD Statement:

The standard format is used.

Secondary allocations cannot be specified.

The following DCB subparameters must be specified:

```
RECFM=VB,BLKSIZE= 3625 if on a 2311
                   20483 if on a 2301
                   4892 if on a 2303
                   14136 if on a 2305-1
                   14660 if on a 2305-2
                   7294 if on a 2314
                   13030 if on a 3330
```

BLKSIZE must be greater than half the amount of storage allocated for the system log buffer by means of the WTLBFRS parameter of the SCHEDULR macro.

#### Catalog Statement:

This statement must be coded for these data sets.

#### Allocation and Cataloging:

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

# SYS1.TELCMLIB .

This data set is required if TCAM, BTAM and/or QTAM are in the system. It is also required for a TSO system.

## Contents

The members of this data set are telecommunications subroutines in load module form.

## Type

This is a partitioned data set.

## Specifying for the New System

### Location:

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

### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

```
RECFM=U,BLKSIZE= 3625 if on a 2311
                  20483 if on a 2301
                  4892 if on a 2303
                  14136 if on a 2305-1
                  14660 if on a 2305-1
                  7294 if on a 2314
                  13030 if on a 3330
```

### Catalog Statement:

You should catalog this data set.

## Special Note

### Blocksize:

If Linkage Editor F is used in the generating system, the blocksize in the DSCB will be the maximum blocksize for the device being used.

## SYS1.UADS

This data set is required for TSO systems.

### Contents

There are one or more members for each terminal user. Basically, the data set is a list of terminal users who are authorized to use TSO. It contains information about each of them.

### Type

This is a partitioned data set.

### Specifying for the New or Modified System

#### Location:

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

#### DD Statement:

The standard format is used.

Secondary allocations can be specified.

The following DCB subparameters must be specified:

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

#### Catalog Statement:

This data set should be cataloged.

### Special Note

#### Information:

For further detailed information about this data set, see OS TSO Command Language.

# System Generation Macro Instructions

This section contains the detailed information necessary to select and code the system generation macro instructions.

The conventions used to code the macro instructions, the text format, a summary of the macro instructions by group (machine configuration, control program, etc.) and type of generation, and a cross-reference table showing the dependencies among the macro instructions are included.

The macro instructions are alphabetically organized for ease of reference. A description, requirements, parameters, and coding hints and additional information are provided for each macro instruction.





## Conventions

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

### CODING MACRO INSTRUCTIONS

The rules for coding system generation macro instructions are those of the assembler language. The following paragraphs are a summary of these rules as stated in the publication, OS Assembler Language.

System generation macro instructions have the following standard format:

---

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

---

The 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. Unless otherwise indicated in the description of individual macro instructions, the name field of a system generation macro instruction is ignored during system generation.

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

The 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 keyword parameter consists of a keyword followed by an equal sign (=) and the keyword value. The keyword value must be a single value or a list of values; in the latter case, the values must be separated by commas and the list 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. Extensive comments may be written by using a series of cards with an asterisk in the first column of each card. A macro instruction that has no parameters cannot have comments.

A typical system generation macro instruction might appear as:

```
NAME OPERATION KEY1=value1,value2),KEY3=value,...
```

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

## Describing Macro Instructions

The following conventions are used in this publication to illustrate the format and coding of system generation macro instructions:

- Uppercase letters, numbers, and punctuation marks must be coded exactly as shown. Exceptions to this convention are brackets, [ ]; braces, { }; ellipses,...; and subscripts. These are never coded.
- Lowercase letters and words 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 the programmer's discretion. Conversely, the lack of brackets indicates that an item or group of items must be coded.
- Braces { } group related items.
- Stacked items enclosed in braces represent alternative items. Only one of the stacked items should be coded.
- If an alternative item is underlined, the operating system will automatically assume it is your choice if none of the items is coded.
- An ellipsis ... indicates that the preceding item or group of items can be coded two or more times in succession.

### **Macro Instructions Text Format**

The content of the new operating system is specified through system generation macro instructions. The macro instructions appear in alphabetical order and the description of each follows the format shown in Figure 78.

---

MACRO INSTRUCTION NAME

A description of the macro instruction, its function, and whether it is required or optional is provided.

Basic Requirements

All basic requirements for the use of the macro instruction are listed.

Nucleus Generation

Additional requirements for using the macro instruction during a nucleus generation, if applicable, are listed. (See Figure 79.)

Processor/Library Generation

Additional requirements for using the macro instruction during a Processor/Library generation, if applicable, are listed. (See Figure 79.)

I/O Device Generation

Additional requirements for using the macro instruction during an I/O Device generation, if applicable, are listed. (See Figure 79.)

---

Name	Operation	Operand
Format of the macro instruction		

---

**Name Field:** Description of the name field, if applicable. (See "Coding Macro Instructions" in the "Conventions" section.)

**Operand Field:** Each keyword parameter in the operand field is described. Any defaults or notes about each keyword parameter and/or its subparameters are provided.

Coding Hints and Additional Information

Any additional information that might help the system programmer is provided.

**Example:** One or more examples of valid ways of coding the macro instruction are supplied. These examples are not meant to be recommended ways of coding the macro instruction. They are just intended to illustrate the mechanics of coding a valid combination of parameters.

NAME OPERATION KEY1=value1,value2),KEY3=value,...

---

Figure 78. Format of Macro Instruction Descriptions

## Macro Instruction Summary

Not all system generation macro instructions are required for the system generation process. Figure 79 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, and the functions it specifies are not generated if it is included in the input deck. (The type of system generation can be specified in the GENERATE macro instruction.) Notes and guidelines about the various types of system generation are provided at the end of this section following Figure 89, which contains the cross-references between the macro instructions.

Figure 79 also shows which macro instructions can be issued more than once during a system generation process. All UNITNAME macro instructions having the same NAME value must appear together in the input deck. Each IOCTRL macro instruction must precede in the input deck to system generation those IODEVICE macro instructions that define devices attached to that control unit. All other system generation macro instructions, with the exception of GENERATE, can be issued in any order. The GENERATE macro instruction must be the last macro instruction in the input deck for the system generation process.

Dependencies among the parameters of a macro instruction are illustrated by the macro instruction format and, when necessary, by tables within each macro instruction description. Dependencies among macro instructions are stated in the descriptions of each macro instruction and summarized at the end of this section in Figure 89.

Several keywords in the system generation macro instructions request the specification of the unit name of a device, for example, UNIT=name in the FORTLIB macro instruction and RESNAME=name in the GENERATE macro instruction. The unit name of a device can be one of the following:

- Unit address, such as 192.
- Device types, such as 2311. (Device types are listed in Appendix A.)
- Group name, or name of a collection of devices, such as TAPE or SYSDA. (See UNITNAME macro instruction.)

It is recommended that device types be used, although all three types are equally valid.

Group	Macro Instruction	Operating System	Type of System Generation <sup>1</sup>		
			Nucleus	Processor/ Library	I/O
Machine Configuration	CENPROCS	Required	Required	Required	Required
	CHANNEL <sup>7</sup>	Required	Required	---	Required
	IOCONTRL <sup>7</sup>	Required	Required	---	Required
	IODEVICE <sup>7</sup>	Required	Required	---	Required
	IMAGELIB	Optional	---	---	Optional <sup>11</sup>
	UCS	Optional	---	---	Optional
	UNITNAME <sup>7</sup>	Optional	---	---	Optional
Control Program	CTRLPROG	Required	Required	---	Required
	SCHEDULR	Required	Required	Optional <sup>10</sup>	Required
	SUPRVSOR	Optional <sup>2</sup>	Optional <sup>2</sup>	Optional <sup>3</sup>	Optional <sup>2</sup>
	CKPTREST	Optional	---	---	---
	GJOBCTL	Optional	Optional	---	Optional
	PARTINS	Optional <sup>4</sup>	Optional <sup>4</sup>	---	---
	SECMODS	Optional	Optional	---	Optional
Data Management	SECONSLE <sup>7</sup>	Optional	Optional	---	Optional
	DATAMGT	Optional <sup>5</sup>	Optional <sup>5</sup>	---	Optional <sup>12</sup>
	GRAPHICS	Optional	Optional	Optional <sup>6</sup>	Optional
	PTOP	Optional	---	Optional	---
User-Written Routines	TELCMLIB	Optional	---	---	---
	LINKLIB	Optional	---	Optional	---
	RESMODS	Optional	Optional	---	---
	SVCLIB	Optional	---	Optional	---
Processing Programs and Libraries	SVCTABLE <sup>7</sup>	Optional	Optional	---	Optional
	ALGOL	Optional	---	Optional	---
	ALGLIB	Optional	---	Optional	---
	ASSEMBLR <sup>7</sup>	Optional	---	Optional	---
	MACLIB	Optional <sup>5</sup>	---	Optional <sup>5</sup>	---
	CHECKER	Optional	---	Optional	---
	CMDLIB <sup>8</sup>	Optional	---	Optional	---
	COBOL <sup>7</sup>	Optional	---	Optional	---
	COBLIB <sup>7</sup>	Optional	---	Optional	---
	EDIT <sup>8</sup>	Optional	---	Optional	---
	EDITOR <sup>7</sup>	Optional	---	Optional	---
	EMULATOR	Optional	Optional	---	Optional
	FORTTRAN <sup>7</sup>	Optional	---	Optional	---
	FORTLIB	Optional	---	Optional	---
	HELP <sup>8</sup>	Optional	---	Optional	---
	LOADER	Optional	---	Optional	---
	OUTPUT <sup>8</sup>	Optional	---	Optional	---
	PARMLIB	Optional	---	Optional	---
	PL1	Optional	---	Optional	---
	PL1LIB	Optional	---	Optional	---
	PROCLIB	Optional	---	Optional	---
RPG	Optional	---	Optional	---	
SORTMERG	Optional	---	Optional	---	
SORTLIB	Optional	---	Optional	---	
SYSUTILS	Optional	---	---	Optional	
TSOPTION	Optional	---	---	---	
UADS <sup>8</sup>	Optional	---	Optional	---	
GENERATE	GENERATE <sup>9</sup>	Required	Required	Required	Required
	GENTSO <sup>9</sup>	Required	Required	Required	Required

- <sup>1</sup> The type of system generation is specified with the GENTYPE parameter of the GENERATE or GENTSO macro instruction.
- <sup>2</sup> This macro instruction is required for MVT and M65MP.
- <sup>3</sup> This macro instruction is used during a Processor/Library generation only if, the PL1LIB macro instruction is used.
- <sup>4</sup> This macro instruction is used if, and only if, MPT is generated.
- <sup>5</sup> This macro instruction is required if CIRB is desired in the new system.
- <sup>6</sup> This macro instruction is used during a Processor/Library generation only if, the FORTLIB macro instruction and/or the COBLIB and/or the PL1LIB macro instructions and the GRAPHICS macro instruction had been used in the complete system generation.
- <sup>7</sup> These macro instructions can be used more than once in the input deck.
- <sup>8</sup> These macro instructions are used only for TSO systems.
- <sup>9</sup> GENTSO is the last system generation macro that can be used if a TSO system is being generated or modified. GENERATE is used for all other generations or modifications.
- <sup>10</sup> This macro instruction is required for a TSO processor generation.
- <sup>11</sup> This macro instruction is required for a 1403 printer with the UNVCHSET feature or a 3211 printer.
- <sup>12</sup> If this macro was used during the last complete system generation, it must be respecified during an I/O device generation. It cannot be specified if it was not used in the last complete system generation.

Figure 79. System Generation Macro Instructions



## ALGLIB

The ALGLIB macro instruction specifies the inclusion of the ALGOL subroutine library (SYS1.ALGLIB) in the new operating system. This macro instruction is optional.

### Basic Requirement

The system data set SYS1.ALGLIB must be allocated space in the new system. It may be cataloged in the new system, which can be done during the preparation for system generation.

### Processor/Library Generation

When an existing ALGOL library, SYS1.ALGLIB, is being replaced, the following must be done:

- The existing library must be scratched and uncataloged before the new one can be initialized.
- The member IHIERROR must be scratched from SYS1.LINKLIB.

---

Name	Operation	Operand
	ALGLIB	[UNIT=name [VOLNO=serial]]

---

**Default:** If no keyword parameters are coded, the system data set SYS1.ALGLIB is placed on the new system-residence volume.

#### UNIT=

specifies a direct-access device that will hold the volume on which the ALGOL subroutine library will be generated.

name

is the device type.

#### Notes:

- **Required:** If this keyword parameter is specified, VOLNO must be specified.
- **Device:** The direct-access device assigned to hold the new ALGOL library, SYS1.ALGLIB, must be a device that already exists in the generating system.

#### VOLNO=

specifies the serial number of the volume that is to contain the ALGOL subroutine library for the new system.

serial

is the serial number.

#### Notes:

- **Required:** If this keyword parameter is specified, UNIT must be specified.
- **Serial Number:** The one specified must be the same as the one used in the VOLUME keyword parameter of the DD statement when space was allocated for SYS1.ALGLIB.

ALGLIB

## Coding Hints and Additional Information

This example illustrates the use of the AGLIB macro instruction to specify the inclusion of the ALGOL subroutine library in the new operating system. The unit name is 2301. The volume serial number is 333777.

```
AGLIB UNIT=2301,VOLNO=333777
```



# ALGOL

The ALGOL macro instruction specifies the inclusion of the ALGOL compiler. This macro instruction is optional.

## Basic Requirements

- The ALGLIB macro instruction must be specified.
- In the CENPROCS macro instruction, INSTSET=SCNTF or INSTSET=UNIV must be specified.
- This macro instruction cannot be specified if MODEL=195 has been specified in the CENPROCS macro instruction.

## Processor/Library Generation

Sufficient space for the ALGOL compiler must be allocated in SYS1.LINKLIB.

If the ALGOL macro instruction is used to provide a replacement for an existing ALGOL compiler, then the following conditions must also exist.

Members IEX10, IEX11, IEX20, IEX21, IEX30, IEX40, IEX50, and IEX51 must be scratched from SYS1.LINKLIB. (The space that these members occupied can be recovered by use of the IEBCOPY utility program.)

---

Name	Operation	Operand
	ALGOL	[ SIZE=size ] [ PUNCH= { NODECK } { DECK } ] [ TYPERUN= { LOAD { NOLOAD } } ] [ SORCODE= { EBCDIC } { ISO } ] [ SORLIST= { SOURCE } { NOSOURCE } ] [ PRECISN= { SHORT } { LONG } ]

---

All the keyword parameters provide defaults for the various options that can affect an ALGOL compilation. These default options will be used if the corresponding values are omitted from the PARM field of the EXEC statement for the ALGOL compilation.

PRECISN=  
specifies the internal representation of real values.

SHORT  
specifies fullwords.

LONG  
specifies doublewords.

Default: If this keyword parameter is omitted, SHORT is assumed.

ALGOL

**PUNCH=**  
specifies the production of a punched deck of the object program.

**DECK**  
specifies that a punched deck is to be produced.

**NODECK**  
specifies that a punched deck is not to be produced.

Default: If this keyword parameter is omitted, NODECK is assumed.

**SIZE=**  
specifies the maximum number of bytes of main storage available to the ALGOL compiler.

**size**  
is an integer from 45056 to 999999.

Default: If this keyword parameter is omitted, 45056 is assumed.

**SORCODE=**  
specifies the character set used to keypunch the source programs to be compiled.

**EBCDIC**  
specifies the EBCDIC character set.

**ISO**  
specifies the standard 46-character set in BCD established by the International Standards Organization (ISO) for ALGOL.

Default: If this keyword parameter is omitted, EBCDIC is assumed.

**SORLIST=**  
specifies the production of a listing of the ALGOL source program and identifier table.

**SOURCE**  
specifies that the listing is to be produced.

**NOSOURCE**  
specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, SOURCE is assumed.

**TYPERRUN=**  
specifies the production of input to the linkage editor from the program being compiled.

**LOAD**  
specifies that the program is to be processed by the linkage editor after compilation.

**NOLOAD**  
specifies that the program is only to be compiled.

Default: If this keyword parameter is omitted, LOAD is assumed.

## Coding Hints and Additional Information

This example illustrates the use of the ALGOL macro instruction to specify the inclusion of the ALGOL compiler. Unless otherwise specified at compilation time, the compiler will use 90112 bytes of main storage and will accept source programs written in the standard 46-character set in BCD. A source program listing is produced, internal real values are represented in doublewords, and compiled source programs are processed by the linkage editor; a punched deck is not produced.

```
ALGOL  SIZE=90112,PRECISN=LONG,SORCODE=ISO
```

ALGOL



## ASSEMBLR

The ASSEMBLR macro instruction specifies the inclusion of the assembler language processor. This macro instruction is required for the first system generation and is optional after that.

### Basic Requirements

- Each operating system must have an assembler-language processor.
- For MVT (including M65MP) systems: If IBM-supplied cataloged procedures are going to be used and the F128 linkage editor is included in the system, the REGION parameter in the cataloged procedures with linkage editor steps must be changed. The alias for the linkage editor, IEWL, is given to the linkage editor included in the system, unless more than one is specified, in which case the alias is given to the largest. To change the parameter, use the IEBUPDTE utility program. Information about region sizes is provided in OS Storage Estimates.
- If Assembler E is used for system generations, then SYS1.GENLIB must be unblocked.

### Processor/Library Generation

Sufficient space for the assembler being included must have been allocated in SYS1.LINKLIB.

Name	Operation	Operand
	ASSEMBLR	DESIGN=F

DESIGN=

specifies the design level of the assembler-language processor to be included in the new system.

F

specifies the F-design level.

### Coding Hints and Additional Information

ASSEMBLR

When you are using this macro instruction, the following guidelines can be helpful.

- IEUASM is the name of Assembler F.
- The alias ASMBLR is given to the assembler-language processor in the system.

Example: The following example illustrates the use of the ASSEMBLR macro instruction to specify Assembler F.

```
ASSEMBLR DESIGN=F
```



# CENPROCS

The CENPROCS macro instruction describes the central processing unit and its physical features. This macro instruction is required.

## Basic Requirements

- For S/360 Model 40: The keyword parameter INSTSET must be specified for this model or a system generation error will occur.
- For S/360 Model 91: The universal instruction set has only the EDIT and EDMK decimal instructions. A Decimal Simulator Routine can be specified with the OPTIONS keyword parameter in the SUPRVSOR macro instruction.
- For S/360 Model 40: If the universal instruction set is specified, the storage protection feature is not included. It is not standard for this model.

## Nucleus, Processor/Library, and I/O Device Generation

The CENPROCS macro instruction must be coded with the same parameters and subparameters specified as those used in the last complete system generation.

Name	Operation	Operand
CENPROCS		[ MODEL= { 40 50 65 75 85 91 135 145 155 165 195 } ] [ STORAGE= { G GD GE GF GFD DH H HG I IH J JI K KJ KK L KL STD } ] [ INSTSET= { COMM SCNTF UNIV } ] [ FEATURE= (feature[ ,feature ] ) ]

CENPROCS

**FEATURE=**

specifies the optional features installed in the central processing unit. These values can be written in any order.

**PROTECT (For Model 40)**

specifies the storage protection feature.

**S/370 (For Model 195)**

specifies S/370 Model 195.

Notes:

- Program Support: The programming support for storage protection is specified by **OPTIONS=PROTECT** in the **SUPRVSOR** macro instruction.
- Required: This subparameter is specified only for Model 40. It is a standard feature on all other systems and need not be specified.

**INSTSET=**

specifies the instruction set available in the central processing unit.

**STD**

specifies the standard instruction set.

**COMM**

specifies the commercial instruction set. This is the standard set with the decimal feature.

**SCNTF**

specifies the scientific instruction set. This is the standard set with the floating-point feature.

**UNIV**

specifies the universal instruction set. This is the standard set with the decimal, floating-point, and storage protection features.

Note: If **TYPE=MVT** or **M65MP** is specified in the **CTRLPROG** macro

Default:

- If this keyword parameter is omitted for Model 40, a system generation error will occur.
- **UNIV** is the default value only for Models 50, 65, 75, 85, 91, 155, 165, and 195.
- **COMM** is the default value for Models 135 and 145.

Notes:

- Model 40: The universal instruction set does not have the storage protection feature.
- Model 91: The universal instruction set has only the **EDIT** and **EDMK** decimal instructions. (A Decimal Simulator routine for Model 91 may be specified with the **OPTIONS** keyword of the **SUPRVSOR** macro instruction.)
- Model 195: The universal instruction set should be specified. If any other instruction set is specified, **UNIV** will be assumed.



MODEL=

specifies the model of the central processing unit.

Default: If this keyword parameter is omitted, 40 is assumed.

Note:

- M65MP Systems: 65 must be specified.

STORAGE=

specifies the size of main storage.

<u>Value</u>	<u>Storage Size (in bytes)</u>
G	128K
GD	144K
GE	160K
GF	192K
GFD	208K
DH	240K
H	256K
HG	384K
I	512K
IH	768K
J	1024K
JI	1536K
K	2048K
KJ	3072K
KK	4096K
L	4096K
KL	6144K

Default: If this keyword parameter is omitted, G (or 128K) is assumed.

Notes:

- Storage Size Usage:

GD and DH	135 only
GE and GFD	145 only
HG	50/145/155 only

These storage sizes are invalid on other models.

- M65MP Systems: I, IH, J, or K only can be specified.
- Model 40: Storage size 192K (GF) can be specified only for this model.
- Model 91: Storage 1024K (J) or larger must be specified for this model.
- Model 135: Always specify a nominal storage size even if control storage has been expanded into main storage. Minimum main storage requirements remain the same as specified in OS Storage Estimates.
- S/370 Model 165: Only storage sizes I, J, JI, K, or KJ can be specified.
- S/360 Model 195 and S/370 Model 195: Only storage sizes J, K, KJ, or L can be specified.

CENPROCS

Coding Hints and Additional Information

The chart below shows the relationship between two of the parameters in this macro instruction.

<u>Parameter</u>	<u>Comment</u>	<u>Parameter</u>
MODEL	For 40 requires --	INSTSET

Example: In the following example, a CENPROCS macro instruction is used to describe a model 40 central processing unit with a main storage size of 128K bytes. The commercial instruction set is used.

CENPROCS MODEL=40,STORAGE=G,INSTSET=COMM

# CHANNEL

The CHANNEL macro instruction describes the channel characteristics. This macro instruction is required.

## Basic Requirements

- There must be a CHANNEL macro instruction for each channel of the installation's computing system.
- Burst devices cannot be attached to a multiplexor channel or to the multiplex portion of a high-speed multiplexor channel.

## Nucleus Generation

The CHANNEL macro instructions must be coded with the same keyword parameters and subparameters specified as those used in the last complete system generation.

## I/O Device Generation

- Each channel in the computing system must be specified. If the channel specifications are not changed, the same coding must be repeated.
- Channels can be added, deleted, or changed, but each channel in the new system must be specified.

---

Name	Operation	Operand
[name]	CHANNEL	ADDRESS=address TYPE= { SELECTOR MULTIPLEXOR HISPEEDMULTIPLEXOR BLKMPXR }

---

## Name Field

The name is used in system generation error messages (see Appendix D) to identify any CHANNEL macro instruction that produces an error.

ADDRESS=  
specifies the address of the channel.

address  
is a hexadecimal number from 0 to D.

TYPE=  
specifies the type of channel.

SELECTOR  
is the selector channel.

MULTIPLEXOR  
is the multiplexor channel.

HISPEEDMULTIPLEXOR  
is the 2870 multiplexor channel.

CHANNEL

BLKMPXR

is the 2880 block multiplexor channel or any other block multiplexor channel.

### Coding Hints and Additional Information

When you use this macro instruction, the following guideline can be helpful.

If nothing is coded in the name field, a sequential identification number is supplied by the assembler. This is done for any CHANNEL macro instruction that does not have a name. 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 while processing that macro instruction.

Example: The following example illustrates the use of the CHANNEL macro instruction to describe a multiplexor channel with address 0.

MPX CHANNEL ADDRESS=0,TYPE=MULTIPLEXOR

## CHECKER

The CHECKER macro instruction specifies the inclusion of the syntax checker for FORTRAN and/or PL/I language processors. This macro instruction is optional for CRJE and TSO systems.

### Basic Requirements

- If both types of syntax checkers are to be included, two macro instructions must be used -- one for each type.
- OPTIONS=CRJE or TSO must be specified in the SCHEDULR macro instruction.
- When TYPE=FORTRAN is coded, SIZE will be ignored.
- When TYPE=PL1 is coded, DESIGN will be ignored.

### Processor/Library Generation

OPTIONS=CRJE or TSO must have been specified in the SCHEDULR macro instruction in the last complete system generation.

---

Name	Operation	Operand
	CHECKER	For FORTRAN: [ DESIGN= ([ E ][ , ][ G ][ , ][ H ] ) ] TYPE=FORTRAN
		For PL/I: [ SIZE= ([ 16K ][ , ][ 20K ][ , ][ 27K ] ) ] TYPE=PL1

---

**DESIGN=** (For FORTRAN)  
specifies that one or more of the FORTRAN syntax checker design levels are to be included.

**E**  
specifies the E-design level.

**G**  
specifies the G-design level.

**H**  
specifies the H-design level.

**Default:** When TYPE=FORTRAN and this keyword parameter is not coded, design-level G is assumed.

#### Note:

The design level specified should agree with the design level specified in the FORTRAN macro instruction.

CHECKER

**SIZE= (For PL/I)**

specifies that one or more versions of the PL/I syntax checker are to be included. Only one checker version can be used in a TSO system. If **OPTIONS=TSO** is specified in the **SCHEDULR** macro instruction and more than one version of the PL/I syntax checker is included, the TSO system will use the 20K version if specified; otherwise, it will use the 27K version.

**16K**

specifies the syntax checker for restricted checking. It is fully resident in 16K bytes of main storage.

**20K**

specifies the syntax checker for thorough checking that is partially dynamic in 20K bytes of main storage.

**27K**

specifies the syntax checker for thorough checking that is fully resident in 27K bytes of main storage.

**Default:** When **TYPE=PL1** and **SIZE** is not coded, the 20K form of the syntax checker is assumed.

**TYPE=**

specifies the type of syntax checker to be included.

**FORTTRAN**

specifies the FORTRAN syntax checker.

**PL1**

specifies the PL/I syntax checker.

**Note:**

**Required:** This keyword parameter can have only one subparameter. To specify both forms of the syntax checker, two macro instructions must be coded.

Coding Hints and Additional Information

**Example:** This example illustrates the use of the **CHECKER** macro instruction to select the FORTRAN syntax checking routines. The G-design level will be included in the new system.

```
CHECKER    TYPE=FORTRAN
```

## CKPTREST

The CKPTREST macro instruction makes it possible for you to modify the standard list of ABEND cards. This macro instruction is optional.

### Basic Requirements

- Checkpoint/restart is a standard feature in all systems.
- Step restart can always be used based on the specifications in the RD parameter in the EXEC statement.
- If the CHKPT macro instruction is used in any program or the RESTART keyword parameter is used in a JOB statement, the RESIDNT keyword parameter of the SUPRVSOR macro instruction must be specified -- the subparameter ACSMETH for MFT systems and the subparameter RENTCODE for MVT and M65MP systems.

---

Name	Operation	Operand
	CKPTREST	[ ,ELIGBLE=(code[,code]...) ] [ NOTELIG=(code[,code]...) ]

---

### ELIGBLE=

specifies the user defined ABEND codes that are to be eligible for automatic restart.

code

is a decimal integer from 0 to 4095.

### Notes:

- Maximum: A maximum of ten values may be specified even though none of the standard system ABEND codes has been made not eligible. For further information, see OS Supervisor Services and Macro Instructions.
- PL/I F Compiler: To force or cancel an automatic restart of a job step using the IHERES routine, the user completion code 4092 must be specified in the ELIGBLE parameter. The RD parameter must also be specified in the JOB or EXEC statement with R or RNC as the subparameter.

### NOTELIG=code

specifies which of the operating system ABEND codes in the standard list below are not eligible for automatic restart. These codes may be listed in any order.

001, 031, 033, 03A, 0A3, 0B0, 0F3,  
100, 106, 113, 117, 137,  
20A, 213, 214, 217, 2F3,  
313, 314, 317, 32D,  
413, 414, 417, 422,  
513, 514,  
613, 614, 626, 637,  
700, 714, 717, 737,  
806, 813, 837,  
906, 913, 926, 937,  
A14,  
B14, B37,  
C13,  
E37

CKPTREST

Default: If this parameter is omitted, all the ABEND codes listed above are eligible for automatic restart.

#### Coding Hints and Additional Information

More detailed information about ABEND codes can be found in the OS Messages and Codes and OS Supervisor Services and Macro Instructions.

Example: This example illustrates the use of the CKPTREST macro instruction to select ABEND codes for automatic restart. The operating system ABEND codes of 03A, 626, and 317 are not to be eligible for automatic restart, and the user-defined codes of 95 and 122 are to be eligible.

```
CKPTREST NOTELIG=(03A,626,317),ELIGBLE=(95,122)
```



## CMDLIB

The CMDLIB macro instruction specifies the inclusion of the TSO Command Processing Programs Library (SYS1.CMDLIB) in the new operating system. This macro instruction is optional and valid only in a complete or processor system generation of a TSO system.

### Basic Requirement

The CMDLIB macro instruction must be specified, if the OUTPUT and/or EDIT macro instructions are required.

### Processor/Library Generation

- The CMDLIB macro instruction can be specified during a Processor/Library generation if it wasn't specified during the complete system generation.
- The macro must not be respecified if the values selected for the parameter of the OUTPUT and EDIT macros are to be changed during a Processor/Library generation.

---

Name	Operation	Operand
	CMDLIB	

---

Example: This example illustrates the use of the CMDLIB macro instruction to specify the inclusion of the TSO Command Processing Programs Library, SYS1.CMDLIB, in the new operating system.

CMDLIB

CMDLIB



# COBLIB

The COBLIB macro instruction specifies the inclusion of the COBOL subroutine library (SYS1.COBLIB) in the new operating system. This macro instruction is optional.

## Basic Requirements

- Space must be allocated for SYS1.COBLIB. The data set may be cataloged in the new system.
- A combined subroutine library, one that has E-design level and/or American National Standard COBOL, can be generated. A COBLIB macro instruction must be coded for each one that is desired.
- If a combined subroutine library is desired, the UNIT and VOLNO keyword parameters should be the same in each COBLIB macro instruction. If they are not, the specification in the last COBLIB macro instruction will determine where the library will be placed.

## Processor/Library Generation

- If a combined subroutine library is desired, the COBOL macro instruction must be specified.
- If a combined subroutine library is going to replace an existing SYS1.COBLIB, there must be enough space for the new modules in the data set.

---

Name	Operation	Operand
	COBLIB	DESIGN={E} {U} [UNIT=name VOLNO=serial]

---

DESIGN=  
specifies the design level of the subroutine library.

E  
specifies design-level E.

U  
specifies American National Standard COBOL, X3.23-1968.

### Note:

- Model 91 or 195: Design-level E cannot be specified.

COBLIB

**UNIT=**

specifies the direct-access device that will hold the volume that will contain the COBOL library.

name  
is the device type.

**Default:** If the UNIT and VOLNO keyword parameters are omitted, the COBOL subroutine library is placed on the new system-residence volume.

**Notes:**

- **Device:** The direct-access device specified must exist in the generating system.
- **Required:** If this keyword parameter is coded, VOLNO must also be specified.

**VOLNO=**

specifies the volume that is to contain the new COBOL subroutine library.

serial  
is the serial number.

**Default:** If the UNIT and VOLNO keyword parameters are omitted, the COBOL subroutine library is placed on the new system-residence volume.

**Notes:**

- **Serial number:** The value specified must be the same as the value given to the VOLUME keyword of the DD statement used to allocate space for SYS1.COBLIB during the preparation for system generation.
- **Required:** If this keyword parameter is coded, UNIT must also be specified.

**Coding Hints and Additional Information**

This example illustrates the use of the COBLIB macro instruction to specify the inclusion of the American National Standard COBOL subroutine library in the new operating system. SYS1.COBLIB is to reside on the new system-residence volume.

COBLIB DESIGN=U

# COBOL

The COBOL macro instruction specifies the inclusion of the COBOL compiler. This macro instruction is optional.

## Basic Requirements

- INSTSET=COMM or UNIV must be specified in the CENPROCS macro instruction. The commercial or the universal instruction set is required for COBOL compilations and executions.
- INSTSET=UNIV must be specified in the CENPROCS macro instruction if either floating-point literals are used at compilation time, or exponentiation to a noninteger power or a floating-point number is used at object time.
- The COBLIB macro instruction must be specified.
- This macro instruction must be coded once for each design level.
- For MVT (including M65MP) Systems: If IBM-supplied cataloged procedures are going to be used and the F128 linkage editor is included in the system, the REGION parameter in the cataloged procedures with linkage editor steps must be changed. The alias for the linkage editor, IEWL, is given to the linkage editor included in the system, unless more than one is specified, in which case the alias is given to the largest. To change the parameter, use the IEBUPDTE utility program. Information about region sizes is provided in OS Storage Estimates.

## Processor/Library Generation

- Sufficient space must have been allocated for the new compiler in SYS1.LINKLIB.
- The same design level as a previously generated COBOL compiler can be specified.
- If a different design level of the COBOL compiler is specified, the COBLIB macro instruction must also be specified. This must be done to have a combined subroutine library generated.

---

Name	Operation	Operand
	COBOL	For COBOL E
		DESIGN=E
		[ MSGLEV= {FLAGW} {FLAGE} ]
		[ LINECNT=lines ]
		[ DATAMAP= {DMAP } {NODMAP} ]
		[ PROCMAP= {PMAP } {NOPMAP} ]
		[ DISPCHK= {DISPCK } {NODISPCK} ]
		[ BUFSIZE=number ]
		[ EDIT= {REGED } {INVED} ]

---

COBOL

---

For American National Standard COBOL:

```
DESIGN=U
[ MSGLEV= {FLAGW}
           {FLAGE} ]
[ LINECNT=lines ]
[ SIZE=size ]
[ BUF=number ]
[ SORLIST= {SOURCE }
           {NOSOURCE} ]
[ PUNCH= {NODECK}
         {DECK} ]
[ TYPERUN= {LOAD }
           {NOLOAD} ]
[ SEQCHK= {SEQ }
         {NOSEQ} ]
[ SPACE= {SPACE1 }
         {SPACE2 }
         {SPACE3} ]
[ LITCHAR= {APOST }
          {QUOTE } ]
[ REFLIST= {NOXREF }
          {XREF } ]
[ CNDENSE= {NOCLIST }
          {CLIST } ]
[ SUPPRES= {NOSUPMAP }
          {SUPMAP } ]
[ DATAMAP= {NODMAP }
          {DMAP } ]
[ PROCMAP= {NOPMAP }
          {PMAP } ]
[ TRNCATE= {NOTRUNC }
          {TRUNC } ]
```

---

DESIGN=

specifies the design level of the COBOL compiler.

E

specifies design-level E.

U

specifies American National Standard COBOL, X3.23-1968.

Note:

Model 91 or 195: Design-level E cannot be specified if MODEL=91 or 195 is specified in the CENPROCS macro instruction.

The following keyword parameters provide the default options for the various options that can affect a COBOL compilation. These default options will be used if the corresponding values are omitted from the PARM field of the EXEC statement for the COBOL compilation.

BUF= (For American National Standard)  
specifies the number of bytes of main storage to be used for buffer allocation by the American National Standard COBOL compiler.

number  
is an integer from 2768 to 99999.

Defaults:

- If BUF is omitted and SIZE is specified, the value of BUF is calculated as:

$$\frac{SIZE-81920}{4} + 2768$$

- If both BUF and SIZE are omitted, a value of 2768 is assumed for BUF.

Note:

This value must be included in the value given to the SIZE keyword parameter.

BUFSIZE= (For E)  
specifies the size, in bytes, of each of the six work buffers used during a COBOL compilation.

number  
is a number from:  
180 to 32000 for magnetic tape,  
180 to 3600 for volumes on 2311 Disk Storage drives, or  
180 to 20000 for volumes on 2301 Drum Storage drives.

Default: If this parameter is omitted, a value of 180 is assumed.

Notes:

- Size: The maximum size is an object time option, not checked during system generation.
- Formula: The following formula can be used as a guide to determine the maximum value that can be specified to optimize the allocation of available storage for the data-name table and work buffers. (Any remainder should be ignored.)

$$\text{number} = \frac{M-30000 - [(13+L) (N)]}{6}$$

number  
is the size of each work buffer. If the result is less than 180, 180 must be specified.

M  
is the size (in bytes) of main storage. This value should correspond to the size specified in the CENPROCS macro instruction.

L  
is the length of the average data name.

N  
is the number of data names.

COBOL

CNDENSE= (For American National Standard)

CLIST

specifies that a condensed listing will be produced. A condensed listing contains, in the procedure portion, the source card numbers and the first generated instruction for each verb.

Note: Restriction - If CLIST is specified, PROCMAP=PMAP cannot be specified.

NOCLIST

specifies that no condensed listing will be produced.

Default: If this keyword parameter is omitted, NOCLIST is assumed.

DATAMAP= (For E or American National Standard COBOL)

specifies the production of a listing of the data names and their addresses either relative to load point for the working storage section or relative to the record addresses for the file or linkage sections.

DMAP

specifies that a listing is to be produced.

NODMAP

specifies that the listing is not to be produced.

Default: The default option is DMAP for E level COBOL and NODMAP for American National Standard COBOL.

DISPCHK= (For E)

specifies the generation of object code which determines whether a field to be displayed exceeds the record length of the device on which it is to be written.

DISPCK

specifies that a check is to be made.

NODISPCK

specifies that no check is required.

Default: If this keyword parameter is omitted, DISPCK is assumed.

EDIT= (For E)

specifies the editing function to be used by the compiler.

REGED

specifies that the standard monetary editing function will be used.

INVED

specifies that the inverted monetary editing function will be used.

Default: If this keyword parameter is omitted, REGED is assumed.

LINECNT= (For all design levels)

specifies the number of lines to be printed on each page of the COBOL compiler output listing.

lines

is a two-digit integer from 10 to 99.

Default: If this keyword parameter is omitted, a value of 60 is assumed.



**LITCHAR=** (For American National Standard)  
identifies the character that will be used to delineate literals. It will also be used by the compiler to delineate figurative constants.

**APOST**  
specifies that the apostrophe (') will be used.

**QUOTE**  
specifies that the double quote (") will be used.

Default: If this keyword parameter is omitted, APOST is assumed.

**MSGLEV=** (For all design levels)  
specifies the type of compilation error messages to be printed.

**FLAGW**  
specifies that all warning and error messages are to be printed.

**FLAGE**  
specifies that warning messages are not to be printed.

Default: If this keyword parameter is omitted, FLAGW is assumed.

**PROCMAP=** (For E or American National Standard COBOL)  
specifies the production of a listing of the generated instructions for each statement in the Procedure Division.

**PMAP**  
specifies that the listing is to be produced.

Note:

Restriction: If PMAP is specified, CONDENSE=CLIST cannot be specified.

**NOPMAP**  
specifies that the listing is not to be produced.

Default: The default option is PMAP for E level COBOL and NOPMAP for American National Standard COBOL.

**PUNCH=** (For American National Standard COBOL)  
specifies the production of a punched deck of the object program.

**DECK**  
specifies that a punched deck is to be produced.

**NODECK**  
specifies that a punched deck is not to be produced.

Default: If this keyword parameter is omitted, NODECK is assumed.

COBOL

REFLIST= (For American National Standard)

XREF  
specifies that a cross listing will be produced.

NOXREF  
specifies that no cross listing will be produced.

Default: If this keyword parameter is omitted, NOXREF is assumed.

SEQCHK= (For American National Standard)

specifies the checking of the source program card sequence numbers.

SEQ  
specifies that the source program card sequence numbers are to be checked.

NOSEQ  
specifies that the source program card sequence numbers are not to be checked.

Default: If this keyword parameter is omitted, SEQ is assumed.

SIZE= (For American National Standard)

specifies the number of bytes of main storage available to the American National Standard COBOL compilers.

size  
is an integer from 81920 to 9999999.

Default: If this keyword parameter is omitted, a value of 81920 is assumed.

Note:

For further information on this parameter, refer to OS American National Standard COBOL Programmer's Guide.

SORLIST= (For American National Standard)

specifies the production of a listing of the COBOL source program.

SOURCE  
specifies that the listing is to be produced.

NOSOURCE  
specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, SOURCE is assumed.

SPACE= (For American National Standard)  
specifies the line spacing on the listing obtained when the SOURCE and/or  
MAP options are specified.

SPACE1  
specifies single spacing.

SPACE2  
specifies double spacing.

SPACE3  
specifies triple spacing.

Default: If this keyword parameter is omitted, SPACE1 is assumed.

SUPPRES= (For American National Standard)

NOSUPMAP  
specifies that the object code listing, the object deck, and the load  
module, if the E-level message is generated by the compiler, are to  
be produced.

SUPMAP  
specifies that the above items are to be suppressed.

Default: If this keyword parameter is omitted, NOSUPMAP is assumed.

TRNCATE= (For American National Standard)  
identifies the treatment of the computational (binary) items during a move  
operation.

TRUNC  
specifies that the item is moved to a receiving field according to  
the specification in the PICTURE clause.

NOTRUNC  
specifies that the item is moved to a receiving field according to  
the size of the field in storage (halfword, fullword, etc.)

Default: If this keyword parameter is omitted, NOTRUNC is assumed.

TYPERRUN= (For American National Standard)  
specifies the production of input to the linkage editor from the program  
being compiled.

LOAD  
specifies that the program is to be processed by the linkage editor  
after compilation.

NOLOAD  
specifies that the program is to be compiled only.

Default: If this keyword parameter is omitted, LOAD is assumed.

COBOL

## Coding Hints and Additional Information

The following guidelines can be helpful.

- BUF and SIZE: The value for BUF must be included when the value for SIZE is determined.
- PROC=PMAP and CNDENSE=CLIST: These two are mutually exclusive.

Example: The following example illustrates the use of the COBOL macro instruction to specify an E-design-level COBOL compiler. The number of lines to be printed in each compiler output listing is 55. Listings of data names and their addresses and listings of the generated instructions for each statement in the Procedure Division are to be produced. All warning and error messages are to be printed. The generation of object code to determine the length of fields to be displayed is not required. The size of each of the six work buffers used during a COBOL compilation is 5708. The standard monetary editing function will be used.

The formula used to compute the BUFSIZE value is as follows:

$$\frac{65536-30000-[(13+10)(56)]}{6} = 5708$$

where the main-storage size specified is 64K, the length of the average data name is 10, and the number of data names is 56.

COBOL DESIGN=E,LINECNT=55,DISPCHK=NODISPCK,BUFSIZE=5708

The CTRLPROG macro instruction specifies the control program options. This macro instruction is required.

Basic Requirements

- Models 91 or 195: MVT must be specified if MODEL=91 or 195 is specified in the CENPROCS macro instruction.
- S/370 Models 145, 155, or 165: MFT or MVT must be specified if MODEL=145, 155, or 165 is specified in the CENPROCS macro instruction.
- S/370 Model 135: MFT must be specified if MODEL=135 is specified in the CENPROCS macro instruction.

Nucleus Generation

- The type of control program specified must be the same as that specified in the last complete system generation. Each nucleus of an operating system must support the same control program, because the resident portion of the control program (nucleus) must be compatible with the nonresident portion.
- The subparameters for the following keyword parameters can be changed: TMSLICE, QSPACE, and ADDTRAN. The other keyword parameters must be coded the same as for the last complete system generation.

I/O Device Generation

The other keyword parameters must be the same as those specified in the last complete system generation.

Name	Operation	Operand
	CTRLPROG	<u>For MFT</u>
		TYPE=MFT MAXIO=number [ HIARCHY= { EXCLUDE } { INCLUDE } ] [ OVERLAY= { BASIC ADVANCED } ] [ FETCH= { STD } { PCI } ] [ SYSQUE=size ] [ TMSLICE=(Pn-Pm,SLC-time) ]

CTRLPROG

Name	Operation	Operand
	CTRLPROG (Cont.)	<p><u>For MVT</u></p> <p>TYPE=MVT  MAXIO=number  [ HIARCHY= {<u>EXCLUDE</u>}            {<u>INCLUDE</u>} ]  [ TMSLICE=(priority,SLC-time            [,priority,SLC-time]...) ]  [ OPTIONS=( [ ROLLOUT ] ) ]  [ QSPACE=number ]  [ ADDTRAN=number ]  [ OVERLAY=<u>ASYCHRON</u> ]</p> <p>If HIARCHY=INCLUDE is specified:  [ FETCH=<u>PCI</u> ]</p> <p><u>For M65MP</u></p> <p>TYPE=M65MP  [ HIARCHY=<u>EXCLUDE</u> ]  [ TMSLICE=(priority,SLC-time            [,priority,SLC-time]...) ]  MAXIO=number  [ OPTIONS=( [ ROLLOUT ] ) ]  [ QSPACE=number ]  [ ADDTRAN=number ]  [ OVERLAY=<u>ASYCHRON</u> ]</p>

ADDTRAN= (For MVT including M65MP)  
specifies additional pairs of transient areas to a maximum of 32.

number  
is the number of areas.

FETCH=  
specifies the type of program fetch to be used.

STD (For MFT)  
specifies standard fetch.

PCI (For MFT or MVT)  
specifies the use of Program Controlled Interrupt during fetching  
of a program into storage.

Default: If HIARCHY=INCLUDE is specified, FETCH=PCI is assumed.

HIARCHY= (For all types)  
specifies storage hierarchies for the system.

EXCLUDE  
specifies that the system is not to have storage hierarchies.

INCLUDE  
specifies that the storage hierarchy facility is to be in the new system.

Default: If this keyword parameter is omitted, EXCLUDE is assumed.

Notes:

- MFT: Hierarchies for partitions are defined in the PARTITNS macro instruction.
- MVT: The hierarchy structure will be maintained even though there may not be a 2361 Core Storage unit on the system.
- Restriction: HIARCHY=INCLUDE cannot be specified if TYPE=M65MP, or if 91, 145, 155, 165, 195, or M85 is specified in the CENPROCS macro instruction.

If HIARCHY=INCLUDE is specified, main storage cannot be specified for more than 1024K.

MAXIO= (For all types)  
specifies the maximum number of I/O operations that can be simultaneously processed by the new operating system. This number is the sum of those I/O operations that can be executed simultaneously and those that can be queued while not being executed.

number  
is a decimal number.

Notes:

- Channel Programs: This number limits the maximum number of channel programs that can be started when using access methods or graphic programming services.
- Formula: A recommended minimum value for this parameter is 1 for each teleprocessing or unit record device, 1 for every four direct-access devices, and 1 for every two graphics or tape devices. The value specified should be at least equal to the maximum number of channel programs that may be active at any one time.

OPTIONS=

ROLLOUT (For MVT including M65MP)  
specifies the inclusion of the rollout/rollin function.

Note:

- The rollout/rollin function requires that SYS1.ROLLOUT be cataloged in the new system before IPL.

CTRLPROC

OVERLAY=

specifies the overlay supervisor options.

BASIC (For MFT)

specifies synchronous overlay without exclusive call checking.

ADVANCED (For MFT)

specifies synchronous overlay with error checking for invalid SEGWT instructions.

ASYNCHRON (For MVT including M65MP)

specifies the asynchronous overlay supervisor for MVT.

Defaults:

- For MFT: If this keyword parameter is omitted, BASIC is assumed.
- For MVT (including M65MP): If this keyword parameter is omitted, ASYNCHRON is assumed.

QSPACE= (For MVT including M65MP)

specifies the number of 2048-byte blocks required for the system queue area.

number

is a decimal number.

Default: If this keyword parameter is omitted, a value of 10 is assumed.

Notes:

- Changes: The value specified can be changed at IPL time if OPTION=COMM is specified in the SUPRVSOR macro instruction. Refer to OS Storage Estimates for information on what value to specify for this keyword parameter.
- SMF: If SMF has been specified in the ACCTRTN keyword parameter of the SCHEDULR macro instruction, then additional system queue area must be added. See OS Storage Estimates for the formula to use to calculate the system queue area size.

SYSQUE= (For MFT)

specifies the size of the system queue area (SQA) in bytes.

size

is a decimal number.

Note: The number must be in double word multiples. See OS Storage Estimates for the formula for the system queue area.

Defaults:

- If MCS is not specified and this keyword parameter is omitted, a value of 1600 is assumed.
- If CONOPTS=MCS is specified in the SCHEDULR macro instruction and this keyword parameter is omitted, or if less than 2600 is coded, then 2600 is assumed.

Notes:

- MCS: If CONOPTS=MCS is specified, additional space will be required in the system queue area.



- MFT Systems With Subtasking: The number of subtasks that can be handled is partially determined by the size of the system queue area.
- SMF: If SMF has been specified in the ACCTRTN keyword parameter of the SCHEDULR macro instruction, then additional system queue area must be added.

TMSLICE= (For MFT)

specifies time-slicing for the jobs scheduled into a continuous group of partitions.

Pn-Pm

specifies the continuous group of partitions.

n

is the lowest partition number.

m

is the highest partition number.

Note:

Required: The partition numbers used must also be specified in the PARTITNS macro instruction.

SLC-time

is a decimal integer from 20 to 9999 that specifies the maximum number of milliseconds for which each ready task in the group of partitions is to have control of the CPU during one pass through the group.

Notes:

- Changes: The values specified can be changed when the scheduler is made ready after IPL if OPTIONS=COMM is specified in the SUPRVSOR macro instruction.
- Job Class: To ensure that all jobs of a given class are time-sliced, that job class should be specified only for partitions in the group defined by Pn-Pm. The classes of job that can be scheduled into each partition are specified in the PARTITNS macro instruction.

TMSLICE= (For MVT including M65MP)

specifies time-slicing for all tasks that have a given priority number.

Priority

is a decimal integer from 0 to 13 that specifies the priority number.

SLC-time

is a decimal integer from 20 to 9999 that specifies the maximum number of milliseconds for which each ready task of the priority specified is to have control of the CPU during one pass through the group.

CTRLPROG

Notes:

- Positional: These two subparameters must be written in the order shown.
- Time-Slicing: Can be specified for each of the 14 priorities.
- Required: When this keyword parameter is specified, TIMER=INTERVAL or JOBSTEP must be specified in the SUPRVSOR macro instruction.

TYPE=  
specifies the type of control program.

MFT  
specifies multiprogramming with a fixed number of tasks.

Note: The PARTITNS macro instruction must be specified.

MVT  
specifies multiprogramming with a variable number of tasks.

Notes:

- Required: This type of control program can operate only on a S/360 Model 40 or larger central processing unit that has at least 256K bytes of main storage. The universal instruction set is also required. (See the CENPROCS macro instruction.)
- Model 91: If MODEL=91 is specified in the CENPROCS macro instruction, this subparameter must be specified
- Model 195: If MODEL=195 is specified in the CENPROCS macro instruction, this subparameter must be specified.

M65MP  
specifies the version of MVT that controls multiprocessing and uses two S/360 Model 65s.

Note:

Required: This type of control program can operate only on a Model 65 multiprocessing system (in the multisystem mode or partitioned mode) that has at least 512K bytes of main storage. INSTSET=UNIV must be specified in the CENPROCS macro instruction.

Default: If this keyword parameter is omitted, MVT is assumed.

Coding Hints and Additional Information

The chart below shows the relationship between some parameters in this macro instruction.

<u>Parameter</u>	<u>Condition</u>	<u>Parameter</u>
HIARCHY=INCLUDE	assumes	FETCH=PCI
HIARCHY=INCLUDE	invalid with	TYPE=M65MP

Examples: The following example illustrates the use of the CTRLPROG macro instruction to specify multiprogramming with a variable number of tasks. The maximum number of I/O operations that can be processed simultaneously is 30. The asynchronous overlay supervisor, service aids, and PCI fetch are assumed. Fifteen 2K blocks are specified for the system queue area.

There will be a total of six transient areas: the original pair plus two additional pairs. The rollout/rollin function is not included. All tasks having a priority of 10 can have control of the CPU for a maximum of 512 milliseconds at a time; those with a priority of 7 can have control for a maximum of 256 milliseconds at a time.

```
CTRLPROG  TYPE=MVT,MAXIO=30,QSPACE=15,ADDTRAN=2,  
          TMSLICE=(10,SLC-512,7,SLC-256)
```

# DATAMGT

The DATAMGT macro instruction allows optional access methods. This macro instruction is optional, except when SVC2B (CIRB) is to be specified to be used in the system; then it is required.

## Basic Requirements

- If the track overflow feature is to be used, SVC2B must be included in the system.
- The basic sequential access method (BSAM) and the basic partitioned access method (BPAM) are provided automatically with all types of control programs.
- BTAM or QTAM: If either or both access methods are specified, TIMER=INTERVAL or JOBSTEP must be specified in the SUPRVSOR macro instruction. The generation of SYS1.TELCMLIB is specified with the TELCMLIB macro instruction.
- TCAM: TCAM and BDAM must be specified. TIMER=INTERVAL and WAIT=MULTIPLE must be specified and if the control program is MPT, OPTIONS=ATTACH must also be specified in the SUPRVSOR macro instruction. SYS1.TELCMLIB must be specified with the TELCMLIB macro instruction.
- CIRB: If SVC2B (CIRB) is desired in the new system, BDAM, ISAM, or BTAM must be specified.
- CRJE: If conversational remote job entry (CRJE) is to be used in the new system, BTAM must be specified.
- RJE: If remote job entry (RJE) is to be used in the new system, BTAM and BDAM must be specified.
- TSO: BDAM and TCAM will be assumed if they are not specified.

## Nucleus Generation

If this macro instruction was used during the last complete system generation, it must be respecified during this type of generation. The same parameters and operands must be coded.

## I/O Device Generation

- If this macro instruction was used during the last complete system generation, it must be respecified during this type of generation. The same parameters and operands must be coded.
- If this macro instruction was not used during the last complete system generation, it cannot be coded during this type of generation.

DATAMGT

---

Name	Operation	Operand
	DATAMGT	ACSMETH= (method[ ,method ]...)

---

**ACSMETH=**  
specifies the optional access methods to be included.

**BDAM**  
is the basic direct access method (BDAM) and routines for creating a direct access data set.

**ISAM**  
is the queued and basic index sequential access methods (QISAM and BISAM).

**BTAM**  
is the basic telecommunications access method (BTAM).

**QTAM**  
is the queued telecommunications access method (QTAM).

**TCAM**  
is the telecommunications access method (TCAM).

Note:

Coding: These values may be listed in any order.

Coding Hints and Additional Information

The following example illustrates the use of the DATAMGT macro instruction to specify that the basic direct-access method is to be included in the new operating system.

DATAMGT ACSMETH=BDAM

## DCMLIB

The DCMLIB macro instruction specifies the inclusion of the display control module library (SYS1.DCMLIB) in the new operating system. This macro instruction is optional.

### Basic Requirements

- Space must be allocated for SYS1.DCMLIB
- SYS1.DCMLIB must be cataloged in the new system during system generation.

### I/O Device Generation

- SYS1.DCMLIB is required if transient DCM support has been specified for operator display consoles.

---

Name	Operation	Operand
	DCMLIB	[UNIT=name VOLNO=serial]

---

#### UNIT=

specifies a direct-access device in the generating system on which the DCM library is to be generated.

name

is the device type

#### VOLNO=

specifies the volume that is to contain the DCM library.

serial

is the serial number

Default: If the UNIT and VOLNO parameters are omitted, the DCM library is placed on the LINKVOL volume as specified in the GENERATE macro.

#### Notes:

- Required: VOLNO and UNIT must be coded.
- Value: The amount of storage allocated to SYS1.DCMLIB is specified by the SPACE parameter of the DCMLIB DD statement during preparation for system generation.

DCMLIB

### Coding Hints and Additional Information

This example illustrates the use of the DCMLIB macro instruction to include the display control module library in the new operating system. The unit name is 2311; the volume serial number is 003475.

```
DCMLIB UNIT=2311,VOLNO=003475
```



## EDIT

The EDIT macro instruction specifies the physical characteristics and processing attributes of the various types of data sets (see Figure 80) you want 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 processor or prompter to which the EDIT data set will be supplied when the RUN subcommand of EDIT is issued. This macro is optional and valid only in a complete or processor system generation of a TSO system.

### Basic Requirement

The CMDLIB macro instruction must be specified if the EDIT macro instruction is specified.

### Processor/Library Generation

CMDLIB must be specified during a Processor/Library generation if it was not specified during the complete generation.

---

Name	Operation	Operand
	EDIT	DSTYPE=(name[ ,name]...) BLOCK=([ n],[ n],...) FORMAT=( { FIXED { VAR { FXDONLY } , ... ) FIXED=([ d-m],[ d-m],...) VAR=([ d-m],[ d-m],...) CONVERT=( { CAPS { ASIS { CAPSONLY } , ... ) CHECKER=([ name],[ name],...) USEREXT=([ name],[ name],...) PRMPTER=([ name],[ name],...) USERSRC=( { DATASET } , [ ... ] { INCORE

---

**Note:** Commas are required to denote missing operands on all keywords except DSTYPE. Defaults for omitted operands are listed in Figure 80.

#### DSTYPE=

specifies the IBM standard or user-defined data set type (see Figure 80) that are to have their attributes modified or defined for use by the TSO EDIT command.

#### name

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 is permitted here. Up to ten user-defined data set types, in addition to any of the standard data set types, can be specified up to a total of eighteen. (See Figure 80.)

EDIT

**BLOCK=**

specifies the default block length, in bytes, for the new data sets created by the TSO EDIT command.

**n**

specifies the default blocksize for any data set of the type specified by the associated NAME field of the DSTYPE operand. The value of n is an integer from 1 to 32760.

**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 for data sets of the type specified by the associated NAME field of the DSTYPE operand.

**VAR**

specifies that variable record format is the default for data sets of the type specified by the NAME field of the DSTYPE operand.

**FXDONLY**

specifies that only fixed record format is allowable for data sets of the type specified by the associated NAME field of the DSTYPE operand. Specifying the VAR operand for this data set type is considered invalid.

**FIXED=**

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

**d**

specifies the default logical-record length for any fixed format data set of the type specified by the NAME field of the DSTYPE operand. The value of d is an integer from 1 to 255, and the associated n value, if specified, with the BLOCK operand must be a multiple of this value. If variable-record format is the default for the data set type, and if this keyword value is being used to define the maximum fixed-record length, a value of 0 should be coded for d.

**-**

is a hyphen, and must be coded exactly as shown.

**m**

specifies the maximum logical record length for any fixed format data set of the type specified by the associated NAME field of the DSTYPE operand. The value of m is an integer from 1 to 255. The associated N value if specified with the BLOCK operand, must be a multiple of this value, and the m value must be greater than or equal to the D value specified.



**VAR=**

specifies the default and maximum allowable length, in bytes, of logical records with variable format for new data sets created by the TSO EDIT command. This operand should not be coded if FXDONLY is specified with the FORMAT operand for the associated data set type.

**d**

specifies the default logical-record length for any variable format data set of the type specified by the associated name field of the DSTYPE operand. The value of d is an integer from 5 to 255, and must be less than or equal to n-4 of the associated N value specified with the BLOCK operand.

**-**

is a hyphen, and must be coded exactly as shown.

**m**

specifies the maximum logical record length for any variable format data set of the type specified by the associated NAME field of the DSTYPE operand. The value of m is an integer from 5 to 255, and must be less than or equal to n-4 of the associated N value specified with the BLOCK operand and greater than or equal to the d value specified.

**CONVERT=**

specifies whether upper case conversion of characters is desired for the TSO EDIT command input data.

**CAPS**

specifies that the CAPS operand (on the TSO EDIT command), which requests upper case conversion of data, is to be the default for data sets of the type specified by the associated NAME field of the DSTYPE operand.

**ASIS**

specifies that the ASIS operand on the TSO EDIT command, which requests that the data remain as entered (for example, upper and lower case), is to be the default for data sets of the type specified by the associated NAME field of the DSTYPE operand.

**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.

**CHECKER=**

specifies the installation-supplied processor that is to be used by the TSO EDIT command to check lines in the data set for proper syntax.

**name**

is the 1- to 8-character module name which the TSO EDIT command will use to invoke the CHECKER for data sets of the type specified by the associated NAME field of the DSTYPE operand.

**USEREXT=**

specifies the installation-supplied user exit that is to be invoked by the TSO EDIT command to interpret the parameters specified in the subfield of the data set type operand of the TSO EDIT command. The encoded parameters will be passed as part of the parameter list to the specified processor in the CHECKER operand. This operand is invalid for any of the standard data set types.

**name**

is the 1 to 8 character exit name for data sets of the type specified by the associated NAME field of the DSTYPE operand.

EDIT

PRMPTER=

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

name

is the 1- to 8-character prompter name for data sets of the type specified by the associated NAME field of the DSTYPE operand.

USERSRC=

specifies the type of source input acceptable to the processor specified in the PRMPTER operand under the TSO EDIT command.

DATASET

specifies that the only allowable source of input to the processor specified in the PRMPTER operand is a sequential data set.

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 a sequential data set.

Data Set Type	Blocksize	Record Format	Default LRECL		Maximum LRECL		Data Conversion	Checker Name	Prompter Name	Prompter Input
			F	V	F	V				
Standard Data Set Types										
PLIF	400	FXDONLY	80	0	100	0	CAPONLY	PLIFSCAN	-----	DATASET
FORTE	400	FXDONLY	80	0	80	0	CAPONLY	IPDSNEXC	-----	DATASET
FORTG	400	FXDONLY	80	0	80	0	CAPONLY	IPDSNEXC	-----	DATASET
FORTH	400	FXDONLY	80	0	80	0	CAPONLY	IPDSNEXC	-----	DATASET
ASM	1680	FXDONLY	80	0	80	0	CAPONLY	-----	ASM*	DATASET
TEXT	1680	VAR	0	255	255	255	ASIS	-----	-----	DATASET
DATA	1680	FIXED	80	0	255	255	CAPS	-----	-----	DATASET
CLIST	1680	VAR	0	255	255	255	CAPONLY	-----	-----	DATASET
CNTL	1680	FXDONLY	80	0	80	0	CAPONLY	-----	-----	DATASET
COBOL	400	FXDONLY	80	0	80	0	CAPONLY	-----	COBOL*	DATASET
PLI	400	VAR	0	104	100	104	CAPS	PLISCAN	PLI	DATASET
IPLI	1680	VAR	0	120	120	120	CAPONLY	IKJNC211*	IKJNC211*	DATASET
BASIC	1680	VAR	0	120	120	120	CAPONLY	IKJNC211*	IKJNC211*	DATASET
GOPORT	1680	VAR	0	255	255	255	CAPS	IPDSNEXC	GOPORT*	INCORE
FORTGI	400	FXDONLY	80	0	80	---	CAPONLY	IPDSNEXC	PORT*	DATASET
Any User Defined Data Set Types										
----	1680	FIXED	80	0	255	255	CAPS	-----	-----	-----
*IBM Program Product.										
--Null Value										

Figure 80. Attribute Defaults for Operand Omitted on the EDIT Macro

## Coding Hints and Additional Information

Following are two examples that illustrate the use of the EDIT macro.

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

In this example, 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 80). Also the standard data-set type, DATA, will have a default blocksize of 2400 instead of 1680 (see Figure 80). All other standard data-set types will use the default values in Figure 80.

```
EDIT DSTYPE=(USER01,USER02),        X
      BLOCK=(,400),                 X
      FORMAT=(,VAR),                 X
      VAR=(,104-104),               X
      FIXED=(,0-100)
```

In this example, two new data-set-types, USER01 and USER02, are defined for the TSO EDIT command. Data-set-type, USER01, will use all the system defaults for its attributes (see Figure 80).

Data-set-type, USER02, will have a blocksize of 400. Its record format default will be variable with fixed allowable. Its variable format default and maximum logical record length will be 104. Since fixed record format is allowable, its fixed format maximum logical record length will be 100.

Note that 0 is coded for the fixed-format default logical-record length, since variable format is specified as the format default. All standard data-set types will use the default values in Figure 80.

EDIT



## EDITOR

The EDITOR macro instruction specifies the inclusion of the linkage editor. This macro instruction is required for the first system generation and is optional after that.

### Basic Requirements

- Each operating system must have a linkage editor.
- This macro instruction must be coded once for each design level that is to be included in the new system.
- The alias IEWL, used by the cataloged procedures, and the alias LINKEDIT, used by the supervisor-assisted linkages, are given to the linkage editor chosen. If more than one is included, the alias names are given to the largest linkage editor chosen.
- MVT (including M65MP) System: If the F128 linkage editor is specified and IBM-supplied cataloged procedures are going to be used, the REGION parameter in the cataloged procedures that have linkage editor steps must be changed. The IEBUPDTE utility can be used to do this. Information about region sizes is provided in OS Storage Estimates.

### Processor/Library Generation

- Additional design levels of the linkage editor can be included.
- Sufficient space must have been allocated for any additional linkage editors in SYS1.LINKLIB.
- The aliases IEWL and LINKEDIT are assigned to the largest linkage editor in the modified system after this generation. The aliases can be reassigned after the generation is over by using the IEBUPDTE system utility program.
- The SIZE parameter for an F-design-level linkage editor can be changed by respecification of the same linkage editor during a Processor/Library generation.

EDITOR

---

Name	Operation	Operand
	EDITOR	DESIGN= { F44 } { F88 } { F128 }
		[ SIZE=(size <sup>1</sup> ,size <sup>2</sup> ) ]

---

**DESIGN=**

specifies the design level of the linkage editor to be included.

**F44**

specifies the F-design-level linkage editor that operates in 44K bytes of main storage.

**F88**

specifies the F-design-level linkage editor that operates in 88K bytes of main storage.

**F128**

specifies the F-design-level linkage editor that operates in 128K bytes of main storage.

**SIZE=** (For all design levels)

specifies the default options at link-edit time for the maximum number of bytes of main storage available to the linkage editor and to its corresponding TEST buffer. The values specified must be of the form nnnnK, where nnnn is an integer and K represents 1024 bytes.

**size1**

is a value from the design level of the linkage editor (44K, 88K, or 128K) to 9999K. It specifies the storage available to the linkage editor.

Default: If this subparameter is omitted, a value of 90K is assumed for the F44 or F88 linkage editors, or of 21K for the F128 linkage editor.

Note:

Required: If this subparameter is specified, it must be coded first.

**size2**

is a value from 6K to 100K. It specifies the storage available to the TEXT buffer.

Default: If this subparameter is omitted, 16K is assumed for the F44 linkage editor, 8K for the F88 linkage editor, or 18K for the F128 linkage editor.

Notes:

- Required: If both subparameters are specified, this one must be coded last. If only this one is specified, it must be preceded by a comma.
- Formula: The following conditions must also exist:

$$\text{Value} = \text{size2} < \text{size1} - N$$

N is 38K for F44  
is 44K for F88  
is 66K for F128

Coding Hints and Additional Information

The following example illustrates the use of the EDITOR macro instruction to specify the F-design-level linkage editor that operates in 88K bytes of main storage. There are 200K bytes available to the linkage editor and 8K available to the TEXT buffer.

EDITOR DESIGN=F88,SIZE=200K

# EMULATOR

The EMULATOR macro instruction specifies the inclusion of certain support routines for integrated emulators in the new operating system. This macro instruction is optional.

## Basic Requirements

- TIMER=JOBSTEP or INTERVAL must be specified in the SUPRVSOR macro instruction.
- This macro instruction must be specified during a complete generation before an emulator can be assembled and link-edited into a system.
- This macro instruction must not be coded for the M65MP system.
- Enough room must be allocated in SYS1.LINKLIB for the emulator program that will be generated later unless the emulator will reside in another library that will be specified during emulation generation.

## Nucleus Generation

This macro instruction must be specified in a nucleus generation if it was specified for the last complete generation.

## I/O Device Generation

This macro instruction must be specified in an I/O device generation if it was specified for the last complete generation.

---

Name	Operation	Operand
	EMULATOR	

---

EMULATOR

## Coding Hints and Additional Information

Specific requirements for emulators on the S/360 Model 85 and S/370 Model 165 include:

- MODEL=85 or 165 must be specified in the CENPROCS macro instruction.
- TYPE=MFT or MVT must be specified for the CTRLPROG macro Instruction.
- The OPTIONS operand of the SUPRVSOR macro instruction must include IDENTIFY and ATTACH for the MFT systems.

Specific requirements for emulators on the S/370 Models 135, 145, and 155 include:

- MODEL=135, 145 or 155 must be specified in the CENPROCS macro instruction.
- TYPE=MFT or MVT (Models 145 and 155 only) must be specified in the CTRLPROG macro instruction. For the Model 135 only, TYPE=MFT can be specified.
- If disk units are supported, ACSMETH=BDAM must be coded in the DATAMGT macro instruction.
- If the 7074 emulator for the Model 155 is to be generated, the OPTIONS operand of the SUPRVSOR macro instruction must include IDENTIFY and ATTACH for the MFT systems.

This macro will cause one or two special SVCs to be included in the new system:

- SVC 88 for S/370 Models 135,145, 155, and 165
- SVC 88 and 89 for S/360 Model 85

Detailed information about incorporating integrated emulators into the system is provided in:

709/7090/7094 II Emulator Program for System/360 Model 85

Emulating the IBM 1401, 1440 and 1460 on the IBM System/370 Models 145 and 155 using OS/360

Emulating the IBM 1410 and 7010 on the IBM System/370 Models 145 and 155 using OS/360

Emulating the IBM 7074 on the IBM System/370 Models 155 and 165 using OS/360

Emulating the IBM 7094 on the IBM Models 85 and 165 using OS/360

Emulating the IBM 7080 on the IBM System/370 Model 165 using OS/360

Example:

This example illustrates the use of the EMULATOR macro instruction to include support for integrated emulators.

EMULATOR



# FORTLIB

The FORTLIB macro instruction specifies the inclusion of the FORTRAN subroutine library (SYS1.FORTLIB) in the new operating system. This macro instruction is optional.

## Basic Requirements

- Space must be allocated for SYS1.FORTLIB. The data set may be cataloged in the new system during the preparation for system generation.
- Only one design level of SYS1.FORTLIB can be generated during a system generation.
- The G- and H-design-level libraries support programs compiled with any design level of the FORTRAN compiler.
- The subroutines in any SYS1.FORTLIB can be used by any operating system program.
- FORTRAN G and H cataloged procedures assume that the logical I/O units 06, 05, and 07 have been specified for OBJERR, ONLNRD, and ONLNPCH, respectively. If these have not been specified in the OBJERR, ONLNRD, and ONLNPCH parameters, then the cataloged procedures must be modified as described in OS FORTRAN IV (G and H) Programmer's Guide.
- MVT (including M65MP) Systems: If IBM-supplied cataloged procedures are going to be used and the F128 linkage editor is included in the system, the REGION parameter in the cataloged procedures with link-edit steps must be changed. The alias for the linkage editor, IEWL, is given to the linkage editor included in the system, unless more than one is specified, in which case the alias is given to the largest. To change the parameter, use the IEBUGDTE utility program. Information about region sizes is provided in OS Storage Estimates.

## Processor/Library Generation

- If the generating system is not the one being modified and the existing SYS1.FORTLIB contains user-written modules or GSP:

The catalog of the generating system must point to the existing SYS1.FORTLIB.

The new SYS1.FORTLIB can be cataloged.

If the new SYS1.FORTLIB is to be placed on the volume occupied by the existing SYS1.FORTLIB, the existing one must be moved and recataloged. The space can be recovered by use of the IEBCOPY utility program.

When the generation is over, the existing SYS1.FORTLIB must be scratched and uncataloged.

- If the generating system is not the one being modified and the existing SYS1.FORTLIB does not contain any user-written modules or GSP:

The existing SYS1.FORTLIB must be scratched and uncataloged.

The new SYS1.FORTLIB must be cataloged in the generating system and the receiving system.

FORTLIB

- If the generating system is the one being modified:

The existing SYS1.FORTLIB may be recataloged and moved to another volume.

The new SYS1.FORTLIB must be allocated space but it cannot be cataloged.

---

Name	Operation	Operand
	FORTLIB	For Design Level E
		[DESIGN=E] [UNIT=name] VOLNO=serial [ UNTABLE=number ] [ OBJERR=unit ]
		For Design Level G and H
		DESIGN= {G  H  } [ UNIT=name ] [ VOLNO=serial ] [ OBJERR=unit ] [ ONLNRD=unit ] [ ONLNPCH=unit ] [ BOUNDRY= {ALIGN NOALIGN} ] [ OPTERR= {EXCLUDE INCLUDE} ] [ ADDNTRY=number ]

---

**ADDNTRY=** (For G or H)  
specifies the number of option table entries to be generated in addition to the required number.

**number**  
is a decimal number from 0 to 598.

Default: If this parameter is omitted, a value of 0 is assumed.

Notes:

- **Size:** The option table has an 8-byte entry for each IBM-designated error condition. An additional 8-byte entry should be added for each installation error condition.
- **Required:** This parameter is valid if and only if OPTERR=INCLUDE is specified.
- For further information on option table entries, refer to the publication OS FORTRAN IV (G and H) Programmer's Guide.

**BOUNDRY=** (For G or H)  
specifies the inclusion of the execution time boundary alignment routine in SYS1.LINKLIB.

**ALIGN**  
specifies that the routine is provided.

**NOALIGN**  
specifies that the routine is not provided.

Default: If this keyword parameter is omitted, ALIGN is assumed.

Note:

Model 91: NOALIGN must be specified if MODEL=91 is coded in the CENPROCS macro instruction.

**DESIGN=**  
specifies the design level of the FORTRAN subroutine library.

**E**  
is the E-design level.

**G**  
is the G-design level.

**H**  
is the H-design level.

Note:

Model 91: Design-level E must not be specified if MODEL=91 is specified in the CENPROCS Macro Instruction

**OBJERR=** (For all design levels)  
specifies which FORTRAN logical I/O unit is to be used for object time error messages and FORTRAN dumps.

**unit**  
is a two-digit decimal number.

Default: If this keyword parameter is omitted, a value of 06 is assumed.

Notes:

- Value: The number cannot exceed the value given to UNTABLE, and cannot be the same as the value given to ONLNRD or to ONLNPCH.
- Recommended: It is recommended that when using the E-design-level library, OBJERR=03 be specified to avoid the allocation of an additional output device.

**ONLNPCH=** (For G or H)  
specifies which FORTRAN logical I/O unit is to be used when the PUNCH (online) statement is encountered.

**unit**  
is a two-digit decimal number.

Default: If this keyword parameter is omitted, a value of 07 is assumed.

Notes:

- Value: The number cannot exceed the value given to UNTABLE, and cannot be the same as the value given to OBJERR or to ONLNRD.

FORTLIB

- Cataloged Procedures: The FORTRAN G and H cataloged procedures assume logical I/O units 06, 05, and 07 as the OBJERR, ONLNRD, and ONLNPC units, respectively. If a different value is given to any of those parameters, the cataloged procedures must be modified, as described in OS FORTRAN IV (G and H) Programmer's Guide.

ONLNRD= (For G or H)  
specifies which FORTRAN logical I/O unit is to be used when the READ (online) statement is encountered in a FORTRAN source program.

unit  
is a two-digit decimal number.

Default: If the ONLNRD parameter is omitted, a value of 05 is assumed.

Note:

- Value: The number must not exceed the value given to UNTABLE, and cannot be the same as the value given to OBJERR or to ONLNPC.

OPTERR= (For G or H)  
specifies the type of object-time error message to be generated.

EXCLUDE  
specifies minimum error service and the termination of the job if an object-time error is detected.

INCLUDE  
specifies complete error service and the possibility that a job may continue after an object-time error is detected (dynamic and default control is provided for each error condition).

Default: If this keyword parameter is omitted, EXCLUDE is assumed.

UNIT= (For all design levels)  
specifies the direct-access device that will hold the volume on which SYS1.FORTLIB will be generated.

name  
is the device type of a direct-access device that exists in the generating system.

- Default: if the UNIT and VOLNO keyword parameters are omitted, the FORTRAN subroutine library is placed on the new system-residence volume.

Note:

- Required: If UNIT is specified, VOLNO must be coded.

VOLNO= (For all design levels)  
specifies the volume that is to contain SYS1.FORTLIB.

serial  
is the serial number of the volume.

- Default: If the UNIT and VOLNO keyword parameters are omitted, the FORTRAN subroutine library is placed on the new system-residence volume.

Notes:

- Value: The value specified must be the same as the value given to the VOLUME keyword of the DD statement used to allocate space for SYS1.FORTLIB during the preparation for system generation.
- Required: If VOLNO is specified, UNIT must be coded.

UNTABLE= (For all design levels)  
specifies the number of FORTRAN logical I/O units to be used at object time.

number  
is a two-digit integer from 08 to 99.

Default: If this parameter is omitted, a value of 08 is assumed.

Note:

Value: This number does not necessarily correspond to the number of I/O devices in the installation.

Coding Hints and Additional Information

The graphic subroutine package (GSP) for FORTRAN IV is specified by the GRAPHICS macro instruction.

Detailed information about FORTRAN IV library subroutines can be found in the following publications:

OS FORTRAN IV Library

OS FORTRAN IV Library: Mathematical and Service Subprograms

The chart below shows the relationship between the parameters of this macro instruction.

<u>Parameter</u>	<u>Comment</u>	<u>Parameter</u>
I/O unit specified in:		I/O unit specified in:
OBJERR must be	≤	UNTABLE
	≠	ONLNRD
	≠	ONLNPCH
ONLNRD must be	≤	UNTABLE
	≠	OBJERR
	≠	ONLNPCH
ONLNPCH must be	≤	UNTABLE
	≠	OBJERR
	≠	ONLNRD
IF OPTERR=INCLUDE	then	ADDNTRY can be coded.

FORTLIB

Example: The following example illustrates the use of the FORTLIB macro instruction to specify the inclusion of the FORTRAN G subroutine library in the new operating system. The unit name is 2301. The volume serial number is 333555. Thirty-two logical units are to be used by the object time load modules. The third unit is to be used for error messages and FORTRAN dumps. Minimum error services are to be provided.

FORTLIB UNIT=2301,VOLNO=333555,UNTABLE=32,OBJERR=03,DESIGN=G

# FORTRAN

The FORTRAN macro instruction specifies the inclusion of the FORTRAN compiler. This macro instruction is optional.

## Basic Requirements

- INSTSET=SCNTF or UNIV must be specified in the CENPROCS macro instruction. The scientific or the universal instruction set is required for FORTRAN compilation as well as executions.
- This macro instruction must be specified once for each design level to be generated.
- MVT (including M65MP) Systems: If IBM-supplied cataloged procedures are to be used and the F128 linkage editor is included in the system, the REGION parameter in the cataloged procedures with link-edit steps must be changed. This alias for the linkage editor, IEWL, is given to the linkage editor included in the system, unless more than one is specified, in which case the alias is given to the largest. To change the parameter, use the IEBUPDTE utility program. Information about region sizes is provided in OS Storage Estimates.

## Processor/Library Generation

- Sufficient space for the new FORTRAN compiler must have been allocated in SYS1.LINKLIB.
- A previously generated FORTRAN compiler can be respecified.
- In order to regain space occupied by previously generated compilers, use the IEBCOPY utility program.

---

Name	Operation	Operand
	FORTRAN	For FORTRAN G
		DESIGN=G
		[ PUNCH= [NODECK] DECK ]
		[ SORLIST= [SOURCE NOSOURCE ] ]
		[ STORMAP= [NOMAP] MAP ]
		[ OBJPROG= [LOAD NOLOAD ] ]
		[ SORCODE= [EBCDIC] BCD ]
		[ LINECNT=number ]
		[ OBJLIST= [NOLIST] LIST ]
		[ OBJID= [NOID] ID ]

---

FORTRAN

Name	Operation	Operand
	FORTRAN (Cont.)	For FORTRAN H  DESIGN=H [ PUNCH= [ NODECK ] [ DECK ] ] [ SORLIST= [ SOURCE ] [ NOSOURCE ] ] [ STORMAP= [ NOMAP ] [ MAP ] ] [ OBJPROG= [ LOAD ] [ NOLOAD ] ] [ SORCODE= [ EBCDIC ] [ BCD ] ] [ LINECNT=number ] [ SIZE=size ] [ OBJLIST= [ NOLIST ] [ LIST ] ] [ OBJID= [ NOID ] [ ID ] ] [ OPT= [ 0 ] [ 1 ] [ 2 ] ] [ SOREDIT= [ NOEDIT ] [ EDIT ] ] [ SORXREF= [ NOXREF ] [ XREF ] ]

DESIGN=  
specifies the design level of the FORTRAN compiler.

G  
is the G-design level.

H  
is the H-design level.

Note:

FORTRAN H: The FORTRAN H compiler can operate only with a central processing unit that has at least 256K bytes of main storage. (See the CENPROCS macro instruction.)

The following keyword parameters provide the default options for the various options that can affect a FORTRAN compilation. Default options are the options that are assumed if the corresponding values of the PARM keyword are omitted from an EXEC statement in a FORTRAN compilation.

LINECNT= (For G or H)  
specifies the number of lines to be printed on each page of the FORTRAN output listing.

number  
is a two-digit integer from 01 to 99.

Default: If this keyword parameter is omitted, a value of 50 is assumed.

OBJID= (For G or H)  
specifies the assignment of internal statement numbers to calls and function references.



ID specifies that statement numbers are to be assigned.

NOID specifies that statement numbers are not to be assigned.

Default: If this keyword parameter is omitted, NOID is assumed.

OBJLIST= (For G or H)  
specifies the production of a pseudo-assembly listing of the direct program.

LIST specifies that the listing is to be produced.

NOLIST specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, NOLIST is assumed.

OBJPROG= (For G or H)  
specifies the production of input to the linkage editor from the program being compiled.

LOAD specifies that the source program is to be processed by the linkage editor after compilation.

NOLOAD specifies that the source program is only to be compiled.

Default: If this keyword parameter is omitted, LOAD is assumed.

OPT= (For H)  
specifies how the optimization of the execution time of the object modules produced by the FORTRAN H compiler is to be done.

0 specifies that the object module is not to be optimized.

1 specifies that it is to receive full register assignment and basic program optimization.

2 specifies that it is to receive full register assignment and complete program optimization.

Default: If this keyword parameter is omitted, 0 is assumed.

PUNCH= (For G or H)  
specifies the production of a punched deck of the object program.

DECK specifies that a punched deck is to be produced.

NODECK specifies that a punched deck is not to be produced.

Default: If this keyword parameter is omitted, NODECK is assumed.

SIZE= (For H)

specifies the amount of main storage available to the FORTRAN H compiler, exclusive of the storage required for buffers and system use.

size

is an integer from 153600 to 9999999, or it may be specified as nnnnK, where nnnn is an integer from 150 to 9999 and K represents 1024 bytes.

Default: If this keyword parameter is omitted, a value of 204800 is assumed.

Notes:

- **Use:** The value specified determines the size of three compiler tables whose size cannot be specified at execution time. The size of all the dictionary and text areas for the compiler is determined at execution time.
- **Value:** The value specified should be the largest amount of main storage that would ever be needed by the FORTRAN H compiler, exclusive of the storage required for buffers and system use.
- **Information:** For further information about this keyword parameter, see OS FORTRAN IV (G and H) Programmer's Guide.

SORCODE= (For G or H)

specifies the character set used to keypunch the source programs to be compiled.

BCD

specifies the BCD character set.

EBCDIC

specifies the EBCDIC character set.

Default: If this keyword parameter is omitted, EBCDIC is assumed.

SOREEDIT= (For H)

specifies the production of a structured source listing in the data set defined by the SYSPRINT DD statement.

EDIT

specifies that the listing is to be produced.

Note:

Required: If this subparameter is specified, OPT=2 must be specified.

NOEDIT

specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, NOEDIT is assumed.

Note:

- **Use:** This listing indicates the loop structure and the logical continuity of the source program.

SORLIST= (For G or H)  
specifies the production of a listing of the FORTRAN source program.

SOURCE  
specifies that the listing is to be produced.

NOSOURCE  
specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, SOURCE is assumed.

SORXREF= (For H)  
specifies the production of a cross-reference listing in the data set defined by the SYSPRINT DD statement.

XREF  
specifies that the listing is to be produced.

NOXREF  
specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, NOXREF is assumed.

STORMAP= (For G or H)  
specifies the production of a map showing the relative locations of variables, constants, etc., in the source program.

MAP  
specifies that the map is to be produced.

NOMAP  
specifies that the map is not to be produced.

Default: If this keyword parameter is omitted, NOMAP is assumed.

#### Coding Hints and Additional Information

Syntax Checker: The syntax checking routines for this compiler can be included in the system by use of the CHECKER macro instruction.

Chart: The chart below shows the relationship between the parameters of this macro instruction.

<u>Parameter</u>	<u>Comment</u>	<u>Parameter</u>
If SOREDIT=EDIT	then must be coded	OPT=2

Example: The following example illustrates the use of the FORTRAN macro instruction to specify a G-design level FORTRAN compiler. The BCD character set is to be the default character set option at compilation time. Unless otherwise specified at compilation time, a FORTRAN source program listing is to be produced, and compiled source programs are to be processed by the linkage editor. Punched decks and source program maps of variables, constants, etc., are not to be produced.

FORTRAN DESIGN=G,SORCODE=BCD

## GENERATE

The GENERATE macro instruction specifies the data sets, volumes, and I/O devices required for the system generation process, the system generation output options, and the type of generation being performed. This macro instruction is required for non-TSO generations; it must never be used in a TSO generation.

### Basic Requirements

- The GENERATE macro instruction must be the last system generation macro instruction in your input deck and must immediately be followed by an assembler END statement.
- The sequential data sets specified in the UT1SDS, UT2SDS, UT3SDS, and UT4SDS keyword parameters must be cataloged in the generating system.
- The data set specified in the UT3SDS keyword parameter is used by the linkage editor and must be on a direct-access device.
- The data specified by the UT4SDS keyword parameter is used by the IEBCOPY utility program and must reside on a direct-access volume.
- If the data sets specified in the UT1SDS and UT2SDS keyword parameters are to reside on magnetic tape, standard labels (SL) or no labels (NL) must be specified when the data sets are defined. (See "Input Deck For System Generation" in the "Preparation for System Generation" section.)
- The data set specified in the OBJPDS keyword parameter must be a cataloged partitioned data set in the generating system.
- If UTDISP=DELETE is specified, the data sets specified in the UT1SDS and UT2SDS keyword parameters are uncataloged and deleted after the Stage II assemblies, and the data set specified in the UT3SDS and UT4SDS keyword parameters is uncataloged and deleted after the Stage II IEBCOPY steps. (See "The Job Stream" section.)
- The UTDISP keyword parameter refers only to the disposition of the four sequential data sets specified in the UT1SDS, UT2SDS, UT3SDS, and UT4SDS keyword parameters after they are used during Stage II. It does not refer to the DISP keyword parameter of the DD statements used for these data sets in the input deck for system generation.

### Complete Operating System Generation

- The serial number specified in the RESVOL keyword parameter cannot be the serial number of the system-residence volume of the generating system.
- The serial number specified in the LNKVOL keyword parameter cannot be the serial number of the SYS1.LINKLIB of the generating system.

GENERATE

## Nucleus Generation

- The machine configuration and type of control program cannot be changed.
- Additional resident functions cannot be made transient.
- Transient functions can be made resident.
- For the RESNAME, RESVOL, and RESTYPE keyword parameters, the unit name, serial number, and device type must be the same as the system-residence volume of the system being modified, which could be the generating system.
- There must be enough space on the system-residence volume for the new member.
- For the LNKNAME, LNKVOL, and LBMAINT keyword parameters, the unit name, serial number, and size must be the same as those of the SYS1.LINKLIB of the system being modified.

## Processor/Library Generation

- There must be sufficient space on the data sets for the new processor(s) and/or libraries.
- For the RESNAME, RESVOL, and RESTYPE keyword parameters, the unit name, serial number, and device type must be the same as those of the system-residence volume of the system being modified, which could be the generating system.
- For the LNKNAME, LNKVOL, and LBMAINT keyword parameters, the unit name, serial number, and size must be the same as those of the SYS1.LINKLIB of the system being modified, which could be the generating system.
- Members may be added to existing libraries if sufficient space is available. If not, then a new data set must be created or the old one must be reallocated. (See the "Reallocating Data Sets" section.)

## I/O Device Generation

- The system being modified cannot be the generating system.
- The serial number specified in the RESVOL keyword parameter cannot be the volume serial number of the system-residence volume of the generating system.
- The serial number specified in the LNKVOL keyword parameter cannot be the serial number of the SYS1.LINKLIB of the generating system.

Name	Operation	Operand
	GENERATE	<u>For Complete or Processor</u>

```

[ GENTYPE= { ALL
             PROCESSOR } ]
OBJPDS=SYS1.name
UT1SDS=(SYS1.name {,SL})
                  {,NL})
UT2SDS=(SYS1.name {,SL})
                  {,NL})
UT3SDS=SYS1.name
UT4SDS=SYS1.name
[ UTDISP= { KEEP
           UNCATLG } ]
[ RESTYPE= { 2301
            2303
            2305-1
            2305-2
            2311
            2314
            3330 } ]
RESNAME=name
RESVOL=serial
[ LNKNAME=name ]
[ LNKVOL=serial ]
[ LBMAINT= { F }
           { E } ]
[ ASMPRT= { OFF }
          { ON } ]
[ LEPRT= (option[ ,option ] ) ]
[ INDEX= { SYS1 }
         { name } ]
[ SYOUTDV= (name,n) ]
[ DIRDATA= { CATALOG }
           { VTOC }
           { PDS } ]
[ JOBSTMT= { NOTSUPPLIED }
           { SUPPLIED } ]

```

For Nucleus

```

GENTYPE=(NUCLEUS,n)
OBJPDS=SYS1.name
UT1SDS=(SYS1.name {,SL})
                  {,NL})
UT2SDS=(SYS1.name {,SL})
                  {,NL})
UT3SDS=SYS1.name
UT4SDS=SYS1.name
[ UTDISP= { KEEP
           UNCATLG } ]
[ RESTYPE= { 2301
            2303
            2305-1
            2305-2
            2311
            2314
            3330 } ]
RESNAME=name

```

GENERATE

Name	Operation	Operand
	GENERATE (Cont.)	<p><u>For Nucleus (continued)</u></p> <p>RESVOL=serial  [ LNKNAME=name ]  [ LNKVOL=serial ]  [ LBMAINT= { F }  { E } ]  [ ASMPRT= { OFF }  { ON } ]  [ LEPRT= (option[ ,option ] ) ]  [ INDEX= { SYS1 }  { name } ]  [ SYOUTDV= (name,n) ]  [ JOBSTMT= { NOTSUPPLIED }  { SUPPLIED } ]</p> <p><u>For I/O Device</u></p> <p>GENTYPE= (IO,n)  OBJPDS=SYS1.name  UT1SDS= (SYS1.name { ,SL }  { ,NL } )  UT2SDS= (SYS1.name { ,SL }  { ,NL } )  UT3SDS=SYS1.name  UT4SDS=SYS1.name  [ UTDISP= { KEEP  UNCATLG }  { DELETE } ]  [ RETYPE= { 2301  2303  2305-1  2305-2  2311  2314  3330 } ]  RESNAME=name  RESVOL=serial  [ LNKNAME=name ]  [ LNKVOL=serial ]  [ LBMAINT= { F }  { E } ]  [ ASMPRT= { OFF }  { ON } ]  [ LEPRT= (option[ ,option ] ) ]  [ INDEX= { SYS1 }  { name } ]  [ SYOUTDV= (name,n) ]  [ DIRDATA= { CATALOG }  { VTOC }  { PDS } ]  [ LOGREC= { NOPRINT }  { PRINT } ]  [ JOBSTMT= { NOTSUPPLIED }  { SUPPLIED } ]</p>



ASMPRT= (For all types)

specifies whether assembly listings are to be produced for the modules assembled during system generation.

ON

specifies that assembly listings are to be generated.

OFF

specifies that assembly listings are not to be generated.

Default: If this keyword parameter is omitted, OFF is assumed.

DIRDATA= (For Complete, Processor, or I/O Device)

specifies the system directory data for the new system is to be printed during system generation as one of the following:

For ALL or PROCESSOR generations:

CATALOG

specifies that the catalog of the new system is to be printed.

VTOC

specifies that the volume table of contents (VTOC) of each volume in the new system is to be printed. The catalog is also to be printed.

PDS

specifies that the directories of all partitioned data sets in the new system are to be printed. The VTOCs and the catalog are also to be printed.

For I/O Generation:

CATALOG

specifies that the system catalog will be printed unchanged.

VTOC

specifies that the system catalog and the volume table of contents of the volume where SYS1.LINKLIB and SYS1.SVCLIB reside will be printed.

PDS

specifies that the contents of SYS1.LINKLIB and SYS1.SVCLIB will be printed. The functions of VTOC will also be provided.

Default: If this keyword parameter is omitted, no system directory data is printed.

GENTYPE=

specifies the type of system generation. (See Figure 79 in the "System Generation Macro Instructions" section.)

ALL

specifies that a complete system is to be generated.

NUCLEUS,n

specifies that only a nucleus is to be generated.

n

is a decimal number from 1 to 9 that will be used to identify a new member of SYS1.NUCLEUS. The member name will be IEANUC0n.

Value: The value 1 is always assigned to the nucleus generated when ALL is specified. Therefore, if 1 is specified in this subparameter, the new nucleus generated replaces the nucleus generated during a complete system generation.

GENERATE

**PROCESSOR**

specifies that only language processors and libraries are to be generated.

**IO,n**

specifies that the machine configuration will be modified.

**n**

is a decimal number from 1 to 9.

Note:

Value: This number is the last character of the name of the nucleus to be modified. For example, if the value 1 is specified, the primary nucleus will be modified.

Default: If this keyword parameter is omitted, ALL is assumed.

**INDEX= (For all types)**

specifies the qualifier for the new system data sets that will be created during system generation.

**SYS1**

is a qualifier.

**name**

is a qualifier that can be from one to eight alphameric characters; the first one must be alphabetic.

Note:

The new system data sets that can be defined with the user-assigned qualifier are: ALGLIB, ASRLIB, COBLIB, FORTLIB, LINKLIB, LOGREC, MACLIB, NUCLEUS, PARMLIB, PL1LIB, PROCLIB, SORTLIB, SVCLIB, TELCMLIB, and IMAGELIB.

Default: If this keyword parameter is omitted, SYS1 is assumed.

**JOBSTMT= (For all types)**

specifies whether your own JOB statement for Stage II of the system generation process will be put in the system generation input deck.

**NOTSUPPLIED**

specifies that your own JOB statement will not be in the input deck.

**SUPPLIED**

specifies that you will supply in the input deck a JOB statement to be used for Stage II.

Defaults:

- If this keyword parameter is omitted, NOTSUPPLIED is assumed.
- If this keyword parameter is omitted or NOTSUPPLIED is coded, the following JOB statement will be used:

```
//SYSGEN JOB 1,'SYSTEM GENERATION'
```

Notes:

- Required Placement: Your JOB statement must be before the GENERATE macro instruction but after any global SET statements. You must place a REPRO statement immediately preceding the JOB statement. For each continuation line for the JOB statement, there must also be a REPRO statement preceding each continuation statement. (For detailed information about the REPRO statement, see OS Assembler Language.)
- DD Statement: When SUPPLIED is specified, then the SYSIN DD statement that precedes the system generation macro instructions in the input deck must be //SYSIN DD DATA. A //SYSIN DD \* statement cannot be used, because it causes the JOB statement following the REPRO statement to be scheduled as a new job.

LBMAINT= (For all types)

specifies the size of the load modules in the SYS1.LINKLIB and SYS1.FORTLIB being generated.

E  
specifies 1024-byte load modules.

Note:

If E is specified, any linkage editor can be used to maintain the new system.

F  
specifies larger load modules.

Note:

Required: If F-is specified, an F-design level linkage editor must be used to maintain the new system.

Default: If this keyword parameter is omitted, F is assumed.

LEPRT= (For all types)

specifies linkage editor print options as one or two of the following values. The values included in braces { } are mutually exclusive.

LIST  
specifies that the list of control statements is to be in card-image format.

MAP  
specifies that a module map will be produced.

XREF  
specifies that a cross-reference table will be produced. (XREF includes the MAP option).

Default: If this keyword parameter is omitted, only linkage editor error messages, if any, are printed. For a more detailed description of these options, see the publication OS Linkage Editor and Loader.

GENERATE

LNKNAME= (For all types)  
specifies a direct-access device in the generating system.

name  
is the device type.

Default: If LNKNAME and LNKVOL are omitted, the link library is placed on the new system-residence volume.

Notes:

- Use: The volume on this device will contain the link library.
- Coding: If LNKNAME is specified, then LNKVOL must also be coded.

LNKVOL= (For all types)  
specifies the volume that will contain the link library.

serial  
is the serial number.

Default: If LNKNAME and LNKVOL are omitted, the link library is placed on the new system-residence volume.

Notes:

- Required: The value specified must be the same as the value given to the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.LINKLIB during the preparation for system generation.
- Coding: If LNKVOL is specified, then LNKNAME must also be coded.

LOGREC= (For I/O Device)  
specifies whether or not the contents of the data set SYS1.LOGREC will be printed before it is scratched during an I/O device generation.

NOPRINT  
specifies that the contents of SYS1.LOGREC will not be printed.

Note: If this subparameter is specified, you must print the SYS1.LOGREC data set to save any necessary information.

PRINT  
specifies that the contents of SYS1.LOGREC of the old system will be printed.

Note: This subparameter can be used only if the generating system is of Release 17 level or later.

Default: If this keyword parameter is omitted, NOPRINT is assumed.

OBJPDS= (For all types)  
specifies the partitioned data set to be used for the storage of object modules assembled during system generation.

SYS1.name  
is the lowest-category simple name of the partitioned data set.

Note:

This data set must exist as a cataloged partitioned data set in the generating system.

RESNAME= (For all types)  
specifies a direct-access device in the generating system.

name  
is the unit address, device type, or group name for the new system-residence volume.

Note:

The volume on this device will contain the new system-residence volume.

RESTYPE= (For all types)  
specifies the device type of the new system-residence device as 2311, 2301, 2303, 2305-1, 2305-2, 2314, or 3330.

Default: If this keyword parameter is omitted, 2311 is assumed.

RESVOL= (For all types)  
specifies the serial number of the new system-residence volume.

serial  
is the serial number.

Note:

Required: The value specified must be the same as the value given to the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.NUCLEUS during the preparation for system generation.

SYOUTDV= (For MFT or MVT including M65MP)  
specifies the direct-access devices to be used for Stage II output data sets (SYSPRINT) during a generation. Device types are described in Appendix A.

name  
is the device type.

n  
specifies the number of volumes that may be occupied by the data set.

Default: If this keyword parameter is omitted, SYSOUT=A is assumed.

Note:

A generation under MFT or MVT (including M65MP) will need about 2500 tracks on a 2311.

GENERATE

UTDISP= (For all types)

specifies disposition information for the four sequential data sets used during system generation.

KEEP

specifies that the data sets are to remain cataloged in the generating system. The space they occupy is to remain allocated to them.

UNCATLG

specifies that catalog entries for the data sets are to be removed from the index structure. Those data sets on direct-access devices remain tabulated in the VTOC.

DELETE

specifies that the data sets are to be uncataloged and the space allocated to them is to be released.

Default: If this keyword parameter is omitted, KEEP is assumed.

UT1SDS=

UT2SDS= (For all types)

UT3SDS=

UT4SDS=

these parameters specify the names of the sequential data sets to be used during system generation by the assembler, linkage editor, and utilities.

Default: For UT1SDS and UT2SDS, SL is assumed.

Note:

These data sets must exist as cataloged data sets in the generating system.

#### Coding Hints and Additional Information

The following example illustrates the use of the GENERATE macro instruction to specify the generation of an operating system. The sequential data sets named SYS1.UTIL1, SYS1.UTIL2, SYS1.UTIL3, and SYS1.UTIL4 are used during system generation by the assembler, linkage editor, and utilities. SYS1.UTIL3 and SYS1.UTIL4 reside on a direct-access volume. SYS1.UTIL1, SYS1.UTIL2, SYS1.UTIL3, and SYS1.UTIL4 are to remain cataloged and are to be allocated space in the generating system after Stage II.

SYS1.OBJMOD is the name of the partitioned data set to be used for the storage of load modules assembled during system generation. The unit name of the new system-residence device is 190, the device type is 2311, and the serial number of the system-residence volume is SYSTEM. Assembly listings, linkage editor printed output, and system directory data are not to be produced. The link library is to be placed on the system-residence volume. The size of the load modules in the new SYS1.LINKLIB is to be 1024 bytes.

```
GENERATE UT1SDS=SYS1.UTIL1,UT2SDS=SYS1.UTIL2,UT3SDS=SYS1.UTIL3,  
         UT4SDS=SYS1.UTIL4,OBJPDS=SYS1.OBJMOD,RESNAME=190,  
         RESTYPE=2311,RESVOL=SYSTEM,LBMAINT=E
```

## GENTSO

The GENTSO macro instruction specifies the data sets, volume, and I/O devices required for a TSO system generation. It also specifies the system generation output options and the type of generation being performed.

This macro instruction must be used instead of the GENERATE macro instruction when a TSO system is generated or modified. See the "Time Sharing Option (TSO)" discussion in the "Program Options Descriptions" section for TSO coding requirements.

The macro instruction has the same parameters and requirements as the GENERATE macro instruction. In addition, it has two parameters that GENERATE does not have:

---

Name	Operation	Operand
	GENTSO	[CMDNAME=name] [CMDVOL=serial] See GENERATE for the other parameters.

---

**CMDNAME=** (For Complete or Processor)  
specifies a direct-access device in the generating system. The volume of this device will contain the SYS1.CMDLIB.

name  
is the device type.

**Default:** If the CMDLIB macro instruction is specified and CMDNAME and CMDVOL are not specified, SYS1.CMDLIB is placed on the new system-residence volume.

### Notes:

- **Required:** Both CMDNAME and CMDVOL keyword parameters must be coded if either is specified.
- The CMDLIB macro must be specified if CMDNAME and CMDVOL are specified.

**CMDVOL=** (For Complete or Processor)  
specifies the volume that is to contain the SYS1.CMDLIB data set.

serial  
is a serial number.

**Default and Notes:** See CMDNAME.

GENTSO

C



## GJOBCTL

The GJOBCTL macro instruction specifies the inclusion of the graphic job processor (GJP) and/or the satellite graphic job processor (SGJP). GJP enables jobs to be defined and initiated from IBM 2250 Display Units. SGJP enables jobs to be defined and initiated from remote 1130/2250 graphic subsystems. This macro instruction is optional.

### Basic Requirements

- The GRAPHICS macro instruction must be specified.
- A set of cataloged procedures is required for GJP and SGJP. These cataloged procedures can be added to SYS1.PROCLIB before or after system generation by use of the IEBUPDTE utility program. The cataloged procedures required and the method for adding them to either library are described in OS MFT Guide and OS MVT Guide.
- All the keyword parameters can be specified for any type.

### I/O Device Generation

- This macro instruction cannot be specified if it was not specified in the last complete system generation.
- This macro instruction may be specified to change the addresses specified in the UNITS parameter. The addresses originally generated will remain unchanged if this macro instruction is not specified.

---

Name	Operation	Operand
	GJOBCTL	$\left[ \begin{array}{l} \text{TYPE} = \left\{ \begin{array}{l} \text{GJP} \\ \text{SGJP} \end{array} \right\} \\ \text{UNITS} = (\text{address}[, \text{address}] \dots) \\ \text{SYSOUT} = (\text{classname}^1, \text{classname}^2, \text{classname}^3, \\ \qquad \qquad \qquad \text{classname}^4) \\ \text{CLASS} = (\text{jobclass}^1, \text{jobclass}^2[, \text{jobclass}^3]) \\ \text{PRI} = (\text{priority}^1, \text{priority}^2) \\ [\text{REGION} = ([\text{number}^1][, \text{number}^2][, \text{number}^3]) ] \\ [\text{ALLOC} = ([\text{number}^1][, \text{number}^2]) ] \\ [\text{UNITREF} = \text{name}] \\ [\text{LINEREF} = \text{name}] \\ [\text{LINE} = \text{number}] \\ [\text{DAPOOL} = \text{name}] \end{array} \right]$

---

GJOBCTL

### ALLOC=

specifies the number of tracks for the primary and secondary allocations for a new data set (allocated if the SYSTEM ALLOCATE option is specified in a DESCRIBE DATA frame).

#### number1

is an integer from 1 to 9999. It is the first positional subparameter, and specifies the number of tracks for primary allocation.

Default: If this subparameter is omitted, a value of 20 is assumed.

number2

is an integer from 1 to 9999. It is the second positional subparameter, and specifies the number of tracks for secondary allocation.

Default: If this subparameter is omitted, a value of 5 is assumed.

CLASS=

specifies the job classes to be used by GJP or SGJP and by jobs defined by GJP or SGJP operations.

jobclass1

is the first positional subparameter. It specifies the job class for GJP or SGJP and for foreground jobs.

jobclass2

is the second positional subparameter. It specifies the job class of background jobs.

jobclass3

is the third positional subparameter. It specifies the job class of the initial processor in SGJP.

Note:

MFT: This subparameter can be specified only if TYPE=MFT is specified in the CTRLPROG macro instruction.

Note:

Value: Each value must be a letter from A through O.

DAPOOL=

specifies the name of a collection of direct-access devices that can be used for data sets if the user specifies the DIRECT ACCESS POOL option in a DESCRIBE DATA frame.

name

is a value that must be the same as the value given to the NAME keyword parameter of a UNITNAME macro instruction that defines a collection of direct-access devices.

Default: If this keyword parameter is omitted, a value of SYSDA is assumed.

LINE=

specifies the maximum number of lines to be printed on each page of the PRINTED RECORD output listing.

number

is an integer from 10 to 99.

Default: If this keyword parameter is omitted, a value of 50 is assumed.

LINEREF=

specifies the default option for the ddname that will be placed in an internally generated DD statement for an 1130/2250 subsystem when the user omits the SUBSYSTEM REFERENCE parameter during job definition.

name

is a value that cannot exceed eight alphameric characters, the first of which must be alphabetic.

Default: If this keyword parameter is omitted, the name FT99F001 is assumed.

Note:

LINEREF need be specified only if SGJP is specified with the TYPE keyword parameter.

PRI=

specifies the priorities to be assigned to jobs defined by GJP or SGJP operations.

priority1

is the first positional subparameter. It specifies the priority of foreground jobs.

priority2

is the second positional subparameter. It specifies the priority of background jobs.

Note:

Value: Each value specified must be an integer from 0 to 13.

REGION=

specifies the number of 1024-byte blocks of main storage required for the regions for GJP or SGJP and for jobs defined by GJP or SGJP operations.

number1

is a decimal number from 1 to 999 that specifies the region size for GJP or SGJP.

Note: This must be the first positional parameter.

number2

is a decimal number from 1 to 999 that specifies the region size foreground jobs.

Default: If this subparameter is omitted, the value specified in the PARM keyword parameter of the EXEC statement for the reader/interpreter procedure used for the job is assumed.

Note: This must be the second positional parameter.

number3

is a decimal number from 1 to 999 that specifies the region size of background jobs.

Default: If this subparameter is omitted, the value specified in the PARM keyword parameter of the EXEC statement for the reader/interpreter procedure used for the job is assumed.

Note: This must be the third positional parameter.

Notes:

- Value: Each value specified must be an integer from 1 to 999.
- MVT (including M65MP): The REGION parameter need be specified only if TYPE=MVT or M65MP is specified in the CTRLPROG macro instruction.

GJOBCTL

**SYSOUT=**

specifies the four classnames to be used for system output from jobs defined by GJP or SGJP operations.

**classname1**

is the first positional subparameter. It specifies the MSGCLASS for foreground jobs only. (This classname must not be used by any SYSOUT writer or other jobs while GJP or SGJP is in operation.)

**classname2**

is the second positional subparameter. It specifies the MSGCLASS for background jobs only. (This classname can be used for any other purpose.)

**classname3**

is the third positional subparameter. It specifies the print output class. (This classname can be used for print output by other jobs.)

**classname4**

is the fourth positional subparameter. It specifies the punch output class. (This classname can be used for punch output by other jobs.)

Note:

- **Value:** Each value specified must be a letter from A to Z or a number from 0 through 9.

**TYPE=**

specifies the type of graphic job control to be included as one of the following:

<u>Value</u>	<u>Processor</u>
GJP	The graphic job processor (GJP) only is to be included.
SGJP	The satellite graphic job processor (SGJP) only is to be included.
GJP,SGJP	Both GJP and SGJP are to be included.

Note:

SGJP requires BTAM and processor-to-processor data transmission modules. BTAM is specified with the DATAMGT macro instruction; processor-to-processor modules are specified with the PTOP macro instruction.

**UNITREF=**

specifies the default option for the dname that will be placed in an internally generated DD statement for a GJP foreground job when you omit the DISPLAY UNIT REFERENCE parameter in the SPECIFY JOB STEP frame.

**name**

is a value that cannot exceed eight alphanumeric characters; the first character must be alphabetic.

Default: If this keyword parameter is omitted, the name SYSDSPLY is assumed.

Note:

**Required:** UNITREF need only be specified if GJP is specified with the TYPE keyword parameter.

**UNITS=**

specifies a 2250 display unit that is to be used for GJP operations and/or 1130/2250 subsystems to be used for SGJP operations.

address

is the address of a 2250 display unit.

**Notes:**

- **Required:** Four or five data sets must be cataloged and allocated space in the new system for each address specified with the UNITS keyword parameter. This function can be performed before or after system generation using the IEHPROGM utility program. The data sets and their allocations are described in OS MFT Guide and OS MVT Guide.
- **Maximum:** A maximum of 15 addresses can be specified. Each address value must be the same as that specified for a device or telecommunications line in an IODEVICE macro instruction.

**Coding Hints and Additional Information**

The following example illustrates the use of the GJOBCTL macro instruction to specify the inclusion of the graphic job processor in an MVT operating system. The addresses of the 2250 units used are 106 and 206. X is the MSGCLASS for all foreground jobs, A is the MSGCLASS for all background jobs, A is the print class, and B is the punch class. Jobclass G is specified for each graphic job processor and each foreground job, and jobclass C is specified for background jobs.

Priority 13 is assigned to foreground jobs and priority 5 to background jobs. The number of 1024-byte blocks of main storage required for each region are: 60 for each graphic job processor region, 60 for each foreground job region, and 50 for each background job region. NEW data sets will be allocated 20 tracks for primary allocation and 5 tracks for secondary allocation. Internally generated DD statements are to have the SYSDSPPLY ddname. The collection of devices named SYSDA is to be used for direct-access data sets. A maximum of 50 lines are to be printed on each page of the output listing.

```
GJOBCTL  TYPE=GJP,UNITS=(106,206),SYSOUT=(X,A,A,B),CLA,SS=(G,C),
          REGION=(,60,50),PRI=(13,5)
```

GJOBCTL



# GRAPHICS

The GRAPHICS macro instruction specifies the inclusion of graphic programming services. This macro instruction is optional.

## Basic Requirements

- This macro instruction must be specified if the GJOBCTL macro instruction is specified.
- If the 2250 Display Unit, Model 3 is specified in the SCHEDULR or SECONSLE macro instruction as an operator console, this macro instruction must be included in the input deck.

## Nucleus Generation

- This macro instruction is included if the nucleus is to support graphic programming services.
- If the new nucleus is to support the same set of graphic programming services that was included in the system during the last complete generation, then this macro instruction must be coded exactly the way it was specified in the last complete system generation.

## Processor/Library Generation

- This macro instruction is included only if SYS1.FORTLIB (FORTLIB macro instruction) or SYS1.PL1LIB (PL1LIB macro instruction) are being replaced and graphic programming services was included in the last complete system generation.
- Only GSP=INCLUDE has to be coded, if it is required, because the PORRTNS parameter is ignored.

## I/O Device Generation

- The graphic programming services (GPS) cannot be added during this type of generation.
- If graphics were specified during the last complete system generation, then they must be specified during this type of generation with exactly the same parameters and subparameters.

GRAPHICS

Name	Operation	Operand
	GRAPHICS	$\left[ \begin{array}{l} \text{PORRTNS} = \left\{ \begin{array}{l} \text{INCLUDE} \\ \text{EXCLUDE} \end{array} \right\} \\ \text{GSP} = \left\{ \begin{array}{l} \text{EXCLUDE} \\ \text{INCLUDE} \end{array} \right\} \end{array} \right]$

**GSP=**

specifies the inclusion of the FORTRAN IV graphic subroutine package (GSP) in SYS1.LINKLIB and SYS1.FORTLIB and/or SYS1.PL1LIB and/or SYS1.COBLIB.

**INCLUDE**

specifies that GSP is to be included.

**EXCLUDE**

specifies that GSP is not to be included.

Default: If this keyword parameter is omitted, EXCLUDE is assumed.

**PORRTNS=**

specifies the inclusion of problem-oriented routines (PORs) in SYS1.LINKLIB.

**INCLUDE**

specifies that PORs are to be included.

**EXCLUDE**

specifies that PORs are not to be included.

Default: If this keyword parameter is omitted, INCLUDE is assumed.

Coding Hints and Additional Information

The information necessary to use the graphic programming services can be found in the following publications about the Operating System:

OS GSP for FORTRAN IV, COBOL, and PL/1

OS GPS for IBM 2250 Display Unit

OS GPS for IBM 2260 Display Station (Local Attachment)

OS GPS for IBM 2280 and 2282 Display Units

Example: The following example illustrates the use of the GRAPHICS macro instruction to specify the inclusion of graphic programming services in the new operating system. Problem-oriented routines are to be included in SYS1.LINKLIB. GSP is not to be included.

GRAPHICS



# HELP

The HELP macro instruction specifies the inclusion of the TSO SYS1.HELP data set in the new operating system. This macro instruction is optional and valid only in a complete or processor generation of a TSO system.

## Basic Requirement

- Space must be allocated for SYS1.HELP. The data set may be cataloged in the new system.

## Processor/Library Generation

- This macro instruction can be used to add or replace a SYS1.HELP data set. If an existing data set is to be replaced, it must be scratched and uncataloged before the new one is added.

---

Name	Operation	Operand
	HELP	[ UNIT=name VOLNO=serial ]

---

Default: If the UNIT and VOLNO keyword parameters are omitted, SYS1.HELP is placed on the new system-residence volume.

### UNIT=

specifies a direct-access device in the generating system. The volume on this device will contain the SYS1.HELP data set.

name

is the device type.

### Note:

Required: If this keyword parameter is specified, VOLNO must also be coded.

### VOLNO=

specifies the volume that is to contain the SYS1.HELP data set.

serial

is a serial number.

### Note:

Required: If this keyword parameter is specified, UNIT must also be coded.

## Coding Hints and Additional Information

This example illustrates the use of the HELP macro instruction to specify the inclusion of the TSO SYS1.HELP data set in the new operating system. The unit name is 2301 and the volume serial number is 43367.

```
HELP UNIT=2301,VOLNO=43367
```

HELP



# IMAGELIB

The IMAGELIB macro instruction specifies the inclusion of the image library (SYS1.IMAGELIB) in the new operating system.

## Basic Requirements

- This macro instruction is required if UNIT=1403 and FEATURE=UNVCHSET or if UNIT=3211 is specified in an IODEVICE macro instruction.
- Space must be allocated for SYS1.IMAGELIB.
- SYS1.IMAGELIB must be a cataloged data set.

## I/O Device Generation

- This macro instruction is required if UNIT=1403 and FEATURE=UNVCHSET or if UNIT=3211 is specified in an IODEVICE macro instruction.

---

Name	Operation	Operand
	IMAGELIB	[UNIT=name VOLNO=serial]

---

**Default:** If the UNIT and VOLNO keyword parameters are omitted, the image library is placed on the new system-residence volume.

### UNIT=

specifies a direct-access device in the generating system that has the volume on which the image library is to be generated.

name

is the device type.

### Note:

**Required:** If this keyword parameter is specified, VOLNO must also be specified.

### VOLNO=

specifies the volume that is to contain the image library.

serial

is the serial number.

IMAGELIB

### Notes:

- **Value:** The value specified must be the same as the value specified in the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.IMAGELIB.
- **Required:** If this keyword parameter is specified, UNIT must also be coded.

## Coding Hints and Additional Information

This example illustrates the use of the IMAGELIB macro to specify the inclusion of the parameter library in the operating system to be generated. The unit name is 2314. The volume serial number is 333666.

```
IMAGELIB UNIT=2314,VOLNO=333666
```



# IOCONTRL

The IOCONTRL macro instruction describes a control unit and its operating system requirements. This macro instruction is required.

## Basic Requirements

- An IOCONTRL macro instruction is required for each control unit (listed in Figure 81) in the computing system.
- Valid combinations for the UNIT, MODEL, and FEATURE keywords are shown in Figure 81, which follows the example for this macro instruction.
- A maximum of forty IOCONTRL macro instructions can be specified during a system generation. If more are required, see Appendix B for the procedure to be followed.

## Nucleus Generation

- Each IOCONTRL macro instruction specified during the last complete generation must be respecified and followed by the appropriate IODEVICE macro instruction(s).

## I/O Device Generation

- All I/O control units that are to be in the new system must be specified with this macro instruction and followed by the appropriate IODEVICE macro instruction(s).
- The control units specified may be additions, deletions, changes, or the same as in the last complete system generation.

---

Name	Operation	Operand
[name]	IOCONTRL	UNIT=unit ADDRESS=address [ MODEL=model ] [ FEATURE=(feature[ ,feature ]...) ]  For UNIT=2840 Only:  [ EXPBFR=number ]

---

IOCONTRL

## Name Field:

name  
is used in system generation error messages (see Appendix D) to identify any IOCONTRL macro instruction that produces an error.

Operand Field:

ADDRESS= (For all units)  
specifies the address of the control unit.

address  
is two hexadecimal digits with a value from 00 to DF.

Notes:

- Value: This value normally corresponds to the two high-order digits of the addresses of the devices attached to the control unit. If the high-order digits of the device addresses differ, the lowest value must be used. For example, if the addresses of the devices attached to the control unit are 00E and 010, the value given to the ADDRESS keyword of the IOCONTRL macro instruction must be 00.
- Required: There must be only one IOCONTRL macro instruction for each control unit. The value given to the address must be the lowest address of the control unit. The other addresses must not be specified. Duplicate control unit addresses must not be specified for a 2848 graphic control unit. Duplicate control unit addresses may be specified for a 2840 graphic control unit attached to a selector subchannel. However, only one duplicate address is allowed, and the addresses of the devices attached to the control unit with the duplicate address might be eight or higher (for example, 3D8, 3D9, 3DA, etc.).

EXPBFR= (For 2840)  
specifies that programs written for a 2250 Model 1, not using the DCB operand GTYPE=BASIC, can be used with a 2250 Model 2 or 3 attached to a 2840.

number  
is an integer from 1 to 8192.

Default: If this keyword parameter is omitted, a value of 4096 is assumed.

Note:

The value specified is the amount of buffer space, in bytes, required by the programs.

**FEATURE=** (For any unit)

specifies the optional features that are present on the control unit. These values can be written in any order. Features enclosed in braces { } are mutually exclusive.

feature

is one of the following values:

<u>Value</u>	<u>Feature</u>
ABSLTVEC	Absolute vector control on 2840 Model 1
{ADSTORAG}	Additional storage on 2841
{2-CHANSW}	2-channel switch on 2841
ALTCTRL	Devices on a tape control unit attached to a matrix switch can be accessed through a different physical control unit. This can be specified only for an M65MP system.
BUFFER8K	Additional 8K buffer on 2840 Model 1
COLBNRY	Column binary on 2821 Model 1, 4, or 5
DATACONV	Data conversion 2403, 2404, 2803, or 2804
LINEADDR	Line addressing on 2848
NODESCUR	Non-destructive cursor on 2848
RECOFLO	Record overflow on 2841
SHARED	All drives attached to 2820 or 2841 can be shared by two CPUs. SHARED requires 2-CHANSW.
2-CHANSW	2-channel switch on 2820
7-TRACK	7-track compatibility on 2403, 2404, 3803, 2804, or 3803
16-DRIVE	16-drive addressing on 2403 or 2803
800BPI	800 BPI on Model 4, 5, or 6 of 2403 or 2415, or on Model 2 of 2803 or 2804, 3410, or on Model 1 of 3803

Notes:

- **M65MP:** 2-CHANSW must be specified if TYPE=M65MP is specified in the CTRLPROG macro instruction.
- If a 3803 control unit is specified, 7-track and 800BPI cannot be specified together.
- If SHARED is specified, alternative channel addressing (see the OPTCHAN parameter of the IODEVICE macro instruction) cannot be specified for any of the direct-access devices attached to the 2820 or the 2841.

**MODEL=** (For some units)

specifies the model, if any, of the control unit.

model

is a value that can be 1, 2, 3, 4, 5, 6, 21, 22, or N1.

**UNIT=** (For all units)

specifies the control unit number.

unit

is a number that can be 1051, 2403, 2404, 2415, 2701, 2702, 2703, 2803, 2804, 2820, 2821, 2822, 2840, 2841, 2848, 3410, 3803, 3811, or 3272.

Notes:

- The IBM 1052 printer keyboard is attached to Model 40 or above through an IBM 1052 adapter and not through a control unit. Thus, an IOCONTRL macro instruction is not needed in this case.

IOCONTRL

- 2816 or 2844 is implied through the specification of the OPTCHAN parameter in the IODEVICE macro instruction and need not be specified in any macro instruction.
- If a 2260 is attached to a 2701 through a 2848, the 2848 is implied and must not be specified.
- If a 2821 is attached to a multiplexor channel, the CE Panel Mode Switch of the 2821 should be set in the "2, 4 byte mode." It must not be set in the burst mode.
- A 3411 is a 3410 tape drive and control unit. To use a 3411, specify UNIT=3410.
- Only model 1 devices (3277, 3284, 3286) may be connected to a 3272 model 1 control unit. Either model 1 or model 2 devices or both may be connected to a 3272 model 2.
- If a 2702 is specified, a dummy device should be specified for line 0 so that your Program Systems Representative can run diagnostics programs.

#### Coding Hints and Additional Information

If no name is entered, the macro instruction supplies sequential identification numbers to the IOCTRL macro instructions in the same order in which these macro instructions are introduced in the input stream. These numbers are used instead of names for identification purposes. For example, if the name is omitted from the sixth IOCTRL macro instruction in the input stream, the name UNIT#6 is supplied in each diagnostic message resulting from an error encountered in the macro instruction.

UNIT	MODEL	FEATURE
1051	1 or N1	
	1, 2, or 3	ALTCTRL <sup>1</sup> DATA CONV 7-TRACK 16-DRIVE
2403	4, 5, or 6	ALTCTRL <sup>1</sup> DATA CONV 7-TRACK 16-DRIVE 800BPI
2404	1, 2, or 3	ALTCTRL <sup>1</sup> DATA CONV 7-TRACK
	1, 2, or 3	ALTCTRL <sup>1</sup> DATA CONV 7-TRACK
2415	4, 5, or 6	ALTCTRL <sup>1</sup> DATA CONV 7-TRACK 800BPI <sup>2</sup>

Figure 81 (Part 1 of 3). Keyword Values for IOCTRL Macro Instruction



UNIT	MODEL	FEATURE
2701		
2702		
2703		
	1	ALTCTRL <sup>1</sup> DATACONV 7-TRACK 16-DRIVE
2803		
	2	ALTCTRL <sup>1</sup> DATACONV 7-TRACK 16-DRIVE 800BPI
	3	DATACONV 7-TRACK 16-DRIVE
	1	ALTCTRL <sup>1</sup> DATACONV 7-TRACK
2804		
	2	ALTCTRL <sup>1</sup> DATACONV 7-TRACK
	3	ALTCTRL <sup>1</sup> DATACONV 7-TRACK
2820		SHARED 2-CHANSW
2821	1, 4, or 5 2, or 3	COLBNRY
2822		
2840	1	BUFFER8K ABSLTVEC
	2	
2841		ADSTORAG or 2-CHANSW RECOFLO SHARED

IOCONTRL

Figure 81 (Part 2 of 3). Keyword Values for IOCONTRL Macro Instruction

UNIT	MODEL	FEATURE
2848	1, 2, 3, 21, or 22	LINEADDR NODESCUR
3410	1, 2, or 3	800BPI
3803	1	ALTCTRL <sup>1</sup> 7-TRACK 800BPI
3811		
3272	1, or 2	

<sup>1</sup>ALTCTRL can be specified only for an M65MP system.

<sup>2</sup>Dual density capability is provided when 800BPI is specified, therefore, dual density should not be specified in the IODEVICE macro instruction.

Figure 81 (Part 3 of 3). Keyword Values for IOCONTRL Macro Instruction

Example: The following example illustrates the use of the IOCONTRL macro instruction to describe an IBM 2821 control unit, Model 4, with the column binary feature. The address of the control unit is 05.

```
IOCONTRL UNIT=2821,MODEL=4,ADDRESS=05,FEATURE=COLBNRY
```

## IODEVICE

The IODEVICE macro instruction describes the characteristics of an input/output device and its operating system requirements. This macro instruction is required.

### Basic Requirements

- Each uniquely addressable input/output device in the system must be specified in an IODEVICE macro instruction.
- In the input deck for system generation, each IOCTRL macro instruction must be immediately followed by the IODEVICE macro instructions that define devices attached to that control unit.
- Figure 82 shows the valid combinations of values for the UNIT, MODEL, FEATURE, and ADAPTER parameters.
- For telecommunication devices: There must be one IODEVICE macro instruction for each telecommunication line. The IODEVICE macro instruction applies to a telecommunication line, not to a terminal device.
- For telecommunication terminals: All terminals on a line must be of the same type with the same features. The type of terminal is used to identify the line in the UNIT parameter.
- Unit address: 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.
- Device type: During system generation, device types are also provided in every operating system for each type of device specified by the UNIT parameter of an IODEVICE macro instruction. Device types are described in Appendix A.
- Group names: A device or a collection of devices can be assigned a group name by use of the UNITNAME macro instruction.
- A maximum of 96 IODEVICE macro instructions can be specified during a system generation. If more devices are desired, a special procedure, which can be used under certain conditions, allows up to 768 devices to be specified. (See Appendix B for a description of the procedure.) The exact maximum number of devices allowed for a particular system depends on several factors, which are described in detail in Appendix B.
- Burst devices cannot be specified for multiplexor channels or the multiplex portion of a high-speed multiplexor channel. Burst devices are: 2250, 2301, 2302, 2303, 2305, 2311, 2314, 2321, 2401, 2402, 2403, 2404, 2415, 2420, 3330, 3410, and 3420.

IODEVICE

### Nucleus Generation

The same IODEVICE macro instructions used during the last complete generation must be respecified without any changes.

### I/O Device Generation

- All I/O devices that are to be in the new system must be specified.
- There can be additions, deletions, changes, or the same specifications as those in the last complete generation.

Name	Operation	Operand
[ name ]	IODEVICE	UNIT={ unit } { DUMMY } ADDRESS={ address ( address, no. of units ) } [ MODEL=model ] [ FEATURE=( feature[ , feature ]... ) ] [ IOQUEUE={ FIFO PRIORITY } ] [ ERRTAB=nnn ] [ DEVTYPE=type ]  For UNIT=2301, 2302, 2303, 2305, 2311, 2314, 2321, 2401, 2402, 2403, 2404, 2420, 3330, or 3420 Only:  [ OPTCHAN=( address[ , address ]... ) ]  For UNIT=2250 (models 2 or 3), 2280, or 2282 Only:  [ NUMSECT=number ]  For Telecommunications Lines Only:  ADAPTER=adapter [ SETADDR=value ] [ OBRCNT=number ]

Name Field:

name  
 is used in system generation error messages (see Appendix D) to identify any IODEVICE macro instruction that produces an error.

Operand Field:

UNIT= (For any device)  
 specifies the device type of the device.

unit  
 is a value that can be 1030, 1050, 1050X, 1052, 1053, 1060, 1130, 1275, 1285, 1287, 1288, 1403, 1419, 1442, 1443, 2020, 2250, 2260, 2265, 2280, 2282, 2301, 2302, 2303, 2305, 2311, 2314, 2321, 2401, 2402, 2403, 2404, 2415, 2420, 2495, 2501, 2520, 2540R, 2540P, 2671, 2740, 2740C, 2740X, 2741C, 2741P, 2780, 2955, 3066, 3210, 3211, 3215, 3277, 3284, 3286, 3330, 3410, 3420, 3505, 3525, 5450, 7770, 115A, 83B3, BSC1, BSC2, BSC3, S360, TWX<sup>1</sup> or WTTA<sup>1</sup>.

<sup>1</sup>Terminals that are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM-supplied products or programs may have on such terminals.

Notes:

- Telecommunications Lines: UNIT specifies the device type that is connected to the telecommunications line, or the type of binary synchronous line configuration: nonswitched point-to-point (BSC1), switched point-to-point (BSC2), or nonswitched multipoint (BSC3).

Binary synchronous may also be specified by UNIT=2780, 1130, 2220, or S360.

- Restriction: Burst devices cannot be specified for a multiplexor channel or multiplex portion of high-speed multiplexor channel.
- 1275: This device is available only through World Trade branch offices.
- 1285, 1287, or 1288: The error routines for these devices can be loaded and made resident at IPL time by specification of RESIDNT=ERP in the SUPRVSOR macro instruction.
- 1050X: Refers to the 1050 with the time-out suppression feature.
- 2020 or S360: These refer to a remote S/360 Model 20 or a remote S/360 attached to the channel through a 2701 or 2703 control unit. For the purposes of system generation, this remote S/360 Model 20 or remote S/360 is considered to be a telecommunications line, and must be defined as such with an IODEVICE macro instruction.
- 2250: This is the Model 195 Integrated CRT operator's console. Model 195 must be specified in the CENPROCS macro instruction, CONOPTS=MCS in the SCHEDULR macro instruction, and MODEL=1, FEATURE=(ALKYB2250,CHARGNTR,BUFFER4K or BUFFER8K...) must be specified in the IODEVICE macro instruction.
- 2250 or 2260 Restriction: When the 2250 or 2260 is specified as an operator console, then CONOPTS=MCS must be coded in the SCHEDULR macro instruction and the GRAPHICS macro instruction must be coded.
- 2305 (Models 1 and 2) Fixed-Head Disk Storage and 3330 Disk Storage Drive: If you specify either the 2305 or the 3330, specify RESIDNT=ACSMETH in the SUPRVSOR macro instruction. Also add IGG019C4, IGG019FN, IGG019FP, IGG019EK to the standard RAM list, IEAIGG00. IGG019C0, which is already on the standard list, is also required.
- 2305 Restrictions: The ADDRESS parameter must indicate only one unit (although eight unit control blocks) will be generated, for example, UNIT=2305,ADDRESS=130 not ADDRESS=(130,8).

The unit position of the value specified for ADDRESS= must be either 0 or 8, for example, ADDRESS=130 or ADDRESS=138.

After a 2305 has been specified, the next succeeding device address must be at least 8 higher, for example, UNIT=2305,ADDRESS=130, UNIT=2311,ADDRESS=138.

It must be 8 higher because of the eight UCBS generated which have the addresses 130-137.

The 2305 must always be addressed by the low-order address (the address assigned during system generation).

IODEVIC

- 2319 (2314): The 2319 drives are functionally equivalent to the 2314 drives. The default value for the 2319 is 8 drives. If you use less than 8 drives, you must specify the number of drives (see ADDRESS= in this section). To use a 2319, specify UNIT=2314: If you have a 2319 and you want to specify UNIT=2319, include the UNITNAME macro with a name of 2319. Appendix E contains a description of the 2319.
- 2495 Tape Cartridge Reader: This device can be specified on the multiplexor channel for Models 30, 40, and 50, or on a control unit position on the 2870 multiplexor channel interface for Models 65 and 75.
- 2540R and 2540P: These are the same IBM 2540 card read punch. Each function is specified in a separate IODEVICE macro instruction.
- 2596 (1442 N1): The 2596 card read punch is functionally equivalent to the 1442 N1 card read punch. If a 2596 is being used, the following must be specified: UNIT=1442; ADDRESS=(2596 device address); and MODEL=N1. The FEATURE=CARDIMAGE parameter may not be specified. For a detailed description of the 2596, see Appendix E.
- 2740C and 2740X: 2740C refers to a 2740 using correspondence code (the inhibit command is automatically provided); 2740X refers to a 2740 using the PTTC code (the inhibit command is automatically provided). One of these devices must be specified if 2740 devices are to be used for CRJE.
- 2741C and 2741P: 2741C refers to a 2741 using correspondence code; 2741P refers to a 2741 using PTTC code.
- 3066: This is the S/370 Model 165 Integrated CRT Operator's Console. UNIT and ADDRESS are the only operands that can be specified. MODEL=165 must be specified in the CENPROCS macro must be specified instruction, CONOPTS=MCS must be specified in the SCHEDULR macro instruction, and the GRAPHICS macro instruction must be coded.
- 5450: This is the Model 85 Integrated CRT Operator's Console. UNIT and ADDRESS are the only operands that can be specified. MODEL=85 must be specified in the CENPROCS macro instruction, CONOPTS=MCS must be specified in the SCHEDULR macro instruction, and the GRAPHICS macro instruction must be coded.
- BSC1: This refers to a nonswitched point-to-point binary synchronous line. This is specified no matter which types of remote BSC stations, supported by BTAM for this type of line, are connected to the line.
- BSC2: This refers to a switched point-to-point binary synchronous line. This is specified no matter which types of remote BSC stations, supported by BTAM for this type of line, communicate with the central computer over the line. (The central computer refers to the system for which this IODEVICE macro instruction is being written.)
- BSC3: This refers to a nonswitched multipoint binary synchronous line. This is specified no matter which types of remote BSC stations, supported by BTAM for this type of line, are connected to the line.
- 2955: ONLNTEST is assumed when you specify UNIT=2955 and do not specify ONLNTEST in the SUPRVSOR macro.
- TWX: This refers to the teletype Model 33 or 35.

**DUMMY**

specifies an unsupported device.

**Notes:**

- **UCB:** A 32-byte UCB with all its standard fields is generated.
- **Support Routines:** You must provide your own I/O support routines for the device.
- **Unit Address:** Unit addresses are not generated for unsupported devices. If they are desired, they can be generated by the UNITNAME macro instruction. The unit address must be generated to make it possible to refer to the device with job control language statements. (See the examples following the description of the UNITNAME macro instruction.)

**ADAPTER=** (For telecommunications lines)

specifies the terminal control or transmission adapter used to connect a telecommunications I/O device to a transmission control unit.

**BSCA**

specifies that an IBM Binary Synchronous Terminal Adapter Type II attaches an 1130, 1800, 2020, 2715, 2770, 2780, 2972, S/360, or S/360 Model 20 to a 2701, or IBM Binary Synchronous Terminal Control Type II attaches an 1130, 1800, 2020, 2715, 2770, 2780, 2972, S/360, or S/360 Model 20 to a 2703.

**IBM1**

specifies that an IBM Terminal Adapter Type I attaches a 1050, 1060, 2740, 2740C, 2740X, 2741, 2741C, or 2741P to a 2701, or IBM Terminal Control Type I attaches a 1050, 1060, 2740, 2740C, 2740X, 2741, 2741C, or 2741P to a 2702 or 2703.

**IBM2**

specifies that an IBM Terminal Adaptor Type II attaches a 1030 to a 2701, or IBM Terminal Control Type II attaches a 1030 to a 2702 or 2703.

**IBM3**

specifies that an IBM Terminal Adapter Type III attaches a 2848/2260 to a 2701.

**IBMT**

specifies that an IBM Telegraph Adapter attaches a 1050 to a 2701, or IBM Terminal Control Type I and a Telegraph Line Adapter attaches a 1050 to a 2702 or 2703.

**TELE1**

specifies that a Telegraph Adapter Type I attaches a 115A or 83B3 to a 2701, or Telegraph Terminal Control Type I attaches a 115A or 83B3 to a 2702 or 2703.

**TELE2**

specifies that a Telegraph Adapter Type II attaches a TWX to 2701, or Telegraph Terminal Control Type II attaches a TWX to a 2702 or 2703.

**TELEW**

specifies that a World Trade Telegraph Adapter (WTTA) attaches a World Trade telegraph terminal to a 2701, 2702, or 2703.

IODEVICE

ADDRESS= (For any device)  
specifies the address or addresses of devices or telecommunication lines.

address

is the unit address of three hexadecimal digits from 000 to DFF.

Notes:

- Required: If "no. of units" is specified, "address" must be the lowest address of the group of devices being specified.
- UCB: For each device address that is assembled, a UCB is created.
- Value: The high-order digit is the address of the channel (specified in the CHANNEL macro instruction) to which the device is attached. For example, if the unit address of the device is 190, the address of the channel is 1.
- 1275 and 1419: The address specified must be the address of the primary control unit and must be an even number. The unit control blocks (UCB) for both the primary and secondary control units will be generated by the one IODEVICE macro instruction.

no. of units

is a value from 1 to 255 indicating the number of consecutively addressed devices of the same type attached to the same control unit.

Default: If this subparameter is omitted, a value of 1 is assumed for all devices except the 2314 and 3330 which have a default value of 8.

Notes:

- This subparameter is not valid for the 2305.
- Use: The value is the number of units to be used and the total number of sequential addresses to be assembled. For example, if ADDRESS=(190,5) is specified, the device addresses 190, 191, 192, 193, and 194 would be assembled.

Note:

1275 and 1419: There must be only one IODEVICE macro instruction for each 1275 or 1419. Two addresses are generated by the one IODEVICE macro instruction.

DEVTYPE= (For any device)

specifies any additional characteristics of the device.

type

is a value of hexadecimal characters.

Notes:

- This parameter should be specified if UNIT=DUMMY is specified.
- Use: For further information about this keyword parameter, refer to OS MFT Guide or OS MVT Guide, and to the description of the UCB in OS System Control Blocks.



ERRTAB= (For any device)

specifies that an error routine, other than its standard error routine is to be used for the device. Either an IBM-supplied or a user-written routine may be specified.

nnn

is a decimal number from 000 to 254.

Notes:

- IBM error routines: These have the values 000 through 219 and 230 through 254.
- User-written routines: These can have the values 220 through 229.
- Error Routine Name: This decimal number (nnn) is used as the suffix of the name IGE00 under which the error routine is contained in SYS1.SVCLIB.
- This parameter should be specified if UNIT=DUMMY is specified.

FEATURE= (For any device)

specifies the optional features that are present on the device. These values can be written in any order. Features enclosed in braces { } are mutually exclusive.

feature

is one of the following values.

<u>Value</u>	<u>Feature</u>
ABSLTVEC	Absolute vector control on 2250 Model 1
ALKYB2250	Alphameric keyboard on 2250
{ ALKYB2260 }	Alphameric keyboard on 2260
{ DEKYB2260 }	Alphameric-numeric insert keyboard
{ NMKYB2260 }	Numeric keyboard on 2260
ALTCTRL	Drives on a 2314 can be accessed through an alternate control unit. This can only be specified for an M65MP system.
ASCACHAR	ASCII A character generator on 3277
ASCBCHAR	ASCII B character generator on 3277
DOCHAR	Domestic character generator on 3277, 3284, or 3286
FRCHAR	French character generator on 3277, 3284, or 3286
GRCHAR	German character generator on 3277, 3284, or 3286
KACHAR	Katakana character generator on 3277, 3284, or 3286
UKCHAR	United Kingdom character generator on 3277, 3284, or 3286
ASKY3277	ASCII typewriter keyboard on 3277
DEKY3277	Data entry keyboard on 3277
EBKY3277	EBCDIC typewriter keyboard on 3277
OCKY3277	Operator console keyboard on 3277
AUDALRM	Audible alarm on 3277
AUTOANSR	Automatic answering capability for 1050, 1130, 2020, 2740, 2780, S360, or TWX. This is assumed if BSC2 is specified.
AUTOCALL	Automatic calling feature for 1050, 1130, 2020, 2740, 2780, S360, or TWX. This is assumed if BSC2 is specified.
AUTOPOLL	Automatic polling feature for 1030, 1050, 1060, 1130, 2020, 2740, or 2780. If AUTOPOLL is specified for 2740, SCONTROL must also be specified. This is assumed if BSC3 is specified.
{ BUFFER4K }	4096-byte buffer storage on 2250 Model 1
{ BUFFER8K }	8192-byte buffer storage on 2250 Model 1
CARDIMAGE	Card image on 1442, 2501, 2520, 3505, or 3525
CHARGNTR	Character generator on 2250 Model 1

IODEVICE

<u>Value</u>	<u>Feature</u>
CHECKING	VRC/LRC checking on 2740
DESIGNFEAT	Graphic design feature on 2250 Model 1
DUALCODE	Decode and dual code feature for 1130, 2020, 2780, BSC1, BSC2, BSC3, or S360
DUALCOMM	Dual communication interface for 1130, 2020, 2780, BSC1, BSC2, BSC3, or S360
DUALDENS	Dual density on Model 4, 5, or 6 of 2401, 2402, or 2403 on Model 2 or 3 of 3410 or on Model 3, 5, or 7 of 3420.
INTERRUPT	Indicates the 2740 is a Model 1 with RPQ#S30031 added to allow improved performance when the 2740 is being used as an MCS console.
KB66KEY	66-key keyboard on 3277
KB70KEY	70-key keyboard on 3277
KB78KEY	78-key keyboard on 3277
KB81KEY	81-key keyboard on 3277
LIGHTPEN	Light pen detect on 2250 Model 1 or 2
MAGCDRD	Magnetic card reader adapter on 3277
MDECOMPAT	Mode compatibility on Model 1, 2, or 3 of 2401, 2402, or 2403
MULTILINE	3525 prints up to 25 lines per card.
NUMLOCK	Numeric lock feature on 3277
OIU	2760 Optical Image Unit; must be used with a 2740 that has checking specified.
{OPPAN1}	First operator control panel on 2250 Model 1
{OPPAN2}	First and second operator control panel on 2250 Model 1
PRGMKYBD	Programmed function keyboard on 2250
READWRITE	Simultaneous reading and writing on 2401, or 2402
{SCONTROL}	Station control feature on 2740. SCONTROL may not be specified if AUTOANSR or AUTOCALL is specified.
{XCONTROL}	Transmit control feature on 2740. XCONTROL requires AUTOANSR, AUTOCALL or both.
SELCHSET	Selective character set on 1443
SELPEN	Selector pen on 3277
SHARED	The 2305, 2314, and 3330 can be shared by two or more CPUs. For the 2314, SHARED requires the presence of either 2-CHANSW or a 2844 control unit, or both. (The presence of a 2844 is implied by FEATURE=SHARED and must not be specified elsewhere.) The 2305 and 3330 require a 2-CHANSW but not a 2844 control unit.
TWOLINE	3525 prints one or two lines per card.
UNVCHSET	Universal character set on 1403 Model 2, 3, or N1 (see the UCS macro instruction)
2-CHANSW	2-channel switch on 2305, 2314, and 3330
{7-TRACK}	7-track head on 2401, 2402, 2403, 2404, 2415, or 3420
{9-TRACK}	9-track head on 2401, 2402, 2403, 2404, 2415, 3410, or 3420
24ADDPOS	24 additional print positions on 1443

Notes:

- 2260: The FEATURE keyword is ignored if specified for a 2260 attached to a 2701 control unit by a 2848.
- 2400: Either 7-TRACK or 9-TRACK must be specified when the value of the UNIT keyword is 2401, 2402, 2403, 2404, or 2415.
- 3400: Either 7-TRACK or 9-TRACK must be specified when the value of the UNIT KEYWORD IS 3420.
- 3410: 9-TRACK must be specified.
- 2740 with INTERRUPT Feature: Must have a 2702 or 2703 with the feature #8200 or a 2701 with RPQ#858492.

- 2740C or 2740X: CHECKING must be specified for these devices.
- 3277, 3284, or 3286: If a character generator is not specified, DOCHAR is assumed.
- 3277: A keyboard is optional; no more than one keyboard may be specified. If KACHAR is specified as the character generator, the keyboard must be specified using either KB70KEY and DEKY3277 or KB81KEY and EBKY3277.
- AUTOPOLL: When this is specified for the telecommunications line, neither AUTOCALL nor AUTOANSR can be specified for that line. AUTOPOLL is valid only if the control unit is either a 2702, 2703, or a 2701 with an IBM Binary Synchronous Terminal Adapter Type II (BSCA).
- DUALDENS: When this is specified, the device type names 2400 and 2400-3 are generated in addition to 2400-4. If DUALDENS is specified for a 3410 Model 2 or 3 or a 3420 tape unit, the following device type names are generated: 3400-4, 3400-3, 2400-4, 2400-3, and 2400.
- DUALCODE and DUALCOMM: These can be specified only if the telecommunications line is attached to a 2701.
- OIU and SCONTROL or XCONTROL: These are mutually exclusive.
- READWRITE: When this is used, one OPTCHAN must be specified.

IOREQUE=

specifies the type of I/O request queuing to be provided by the supervisor for the device.

FIFO

specifies first-in-first-out queuing.

PRIORITY

specifies queuing according to task priority.

ORDERED

specifies queuing according to a cylinder address.

Note:

This subparameter may be specified only for the 2311, 2314, or 2321. If this subparameter is specified for any other device, FIFO is substituted.

MODEL= (For any device)

specifies the model number, if any, of the device.

model

is a value that can be 1, 2, 3, 4, 5, 6, 7, 8, B1, B2, B3, N1, or N2.

IODEVICE

Note:

- Required: This keyword parameter must be specified if the unit has a model number (see Figure 82).

NUMSECT= (For 2250, 2280, or 2282)

specifies the number of 256-byte buffer sections in the 2840 control unit that are guaranteed to be available to the 2250 (Model 2 or 3), 2280, or 2282. These buffer sections can only be used by the device being specified in this IODEVICE macro instruction.

number

is a value from 1 to a maximum value that is computed as follows:

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

A is the size of the 2840 buffer.

B is the number of devices attached to the 2840.

Defaults:

- If this keyword parameter is omitted, the device uses the sections not guaranteed to other devices; however, there must always be at least one section available (guaranteed or not) for assignment to each device.
- If the 2250 Model 3 is specified as an operator console and this keyword parameter is omitted, a value of 16 is assumed.

Notes:

- 2250: If the 2250 Model 3 is specified as an operator console, the number specified must be greater than 15.
- Limit: The total amount of buffer sections guaranteed to the devices attached to a 2840 must not exceed the number of sections in the buffer of that 2840.
- Available: All sections not guaranteed to a device are available to all devices attached to the 2840.
- Information: Note that the assignment of guaranteed sections limits the number of available consecutive sections. For further information, see OS GPS for IBM 2250 Display Unit.

OBRCNT= (For 2715 only)

specifies the number of area stations connected to the 2715 on the communications line defined by this IODEVICE macro instruction.

number

is a number from 0 to 800.

Notes:

The cumulative total of all values specified in OBRCNT keyword parameters cannot exceed 800 during a system generation. If 800 is exceeded, 800 will be assumed.

OPTCHAN= (For 2301, 2302, 2303, 2305, 2311, 2314, 2321, 2401, 2402, 2403, 2404, 2420, 3330, or 3420)

specifies the alternate channels by which any of the above devices may be addressed.

address

is a value that must be greater than the high-order digit of the value in the ADDRESS keyword parameter, except for a selector subchannel.

A selector subchannel must be specified as two characters. The first must be the channel address of the 2870, and the second must be D, E, or F.

Notes:

- 2301, 2302, 2303, 2305, 2311, 2321, or 3330: Only one alternate channel may be specified. If this keyword parameter is specified for the 2302, 2303, 2311, or 2321 devices, then 2-CHANSW is required on the 2841 control unit. The 2305 and 3330 also require a 2-CHANSW, but not with a 2841 control unit.
- 2314: This requires the two-channel switch feature, a 2844 switching unit, or both. In these cases, the presence of the 2844 is implied by specification of the OPTCHAN parameter and must not be specified elsewhere. The two-channel switch for the 2314 is specified with the FEATURE keyword.
- 2314, 2401, 2402, 2403, 2404, 2420, or 3420: A maximum of three alternate channels can be specified. The addresses of alternate channels can be written.
- Magnetic tape: If a magnetic tape drive has the READWRITE feature, only one alternate channel must be specified. A magnetic tape drive attached to a 2403 or 2803 control unit requires a 2816 in order to have alternate channel addressing.
- Maximum: There is a maximum of 128 optional channel paths.
- Required: There must be no more than one IODEVICE macro instruction for each I/O device, regardless of the number of alternate addresses given to the device. For example, if the primary address of a device is 181, and if it can also be addressed through channels 2, 3, and 4, there must not be separate IODEVICE macro instructions that define the address of the device as 281, 381, or 481.

The primary address of the device, that is, the one with the lowest channel address, must be specified in the ADDRESS keyword. The other channel addresses must be specified with the OPTCHAN keyword. In this example, the macro instruction for the device must contain the parameters ADDRESS=181 and OPTCHAN=(2,3,4).

- Required for Alternate Channel: Each value specified is the address of an alternate channel as specified in the CHANNEL macro instruction.
- SHARED: OPTCHAN can be specified for a 2314 when SHARED has also been specified, but cannot be specified for a 2305 or 3330.

SETADDR= (For telecommunication lines)

specifies the set address (SAD) command to be issued for a telecommunications line attached to a 2702 or 2703 control unit.

value

is one of the following:

<u>Value</u>	<u>Command</u>
0	SADZERO
1	SADONE
2	SADTWO
3	SADTHREE

Note:

2702: This keyword parameter must be specified if the device is attached to a 2702 control unit.

IODEVICE

## Coding Hints and Additional Information

If no name is entered, the macro instruction supplies sequential identification numbers to the IODEVICE macro instructions in the same order in which these macro instructions are introduced into the input stream. These numbers are used instead of names for identification purposes. For example, if the name is omitted from the eleventh IODEVICE macro instruction in the input stream, the name DEV#11 is supplied in each diagnostic message that results from an error encountered in the macro instruction.

---

UNIT	MODEL	FEATURE	ADAPTER
1030		AUTOPOLL	IBM2
1050 <sup>1</sup>		AUTOANSR AUTOCALL AUTOPOLL	IBM1 or IBMT
1050X		AUTOANSR AUTOCALL	IBM1
1052 <sup>2</sup>	5, 6, 7, or 8		
1053	4		
1060		AUTOPOLL	IBM1
1130 <sup>1</sup>		AUTOANSR AUTOCALL AUTOPOLL DUALCODE DUALCOMM	BSCA
1275			
1285			
1287			
1288			
1403 <sup>3</sup>	2, 3, or N1 7	UNVCHSET	
1419			
1442	N1 or N2	CARDIMAGE	
1443	N1	SELCHSET 24ADDPOS	
2020		AUTOANSR AUTOCALL AUTOPOLL DUALCODE DUALCOMM	BSCA

---

Figure 82 (Part 1 of 6). Keyword Values for the IODEVICE Macro Instruction

UNIT	MODEL	FEATURE	ADAPTER
2250	1 <sup>4</sup> , 1 <sup>9</sup>	ABSLTVEC ALKYB2250 BUFFER4K or BUFFER8K CHARGNTR DESIGNFEAT LIGHTPEN OPPAN1 or OPPAN2 PRGMKYBD	
	2	ALKYB2250 LIGHTPEN PRGMKYBD	
	3 <sup>5</sup> , 1 <sup>9</sup>	ALKYB2250 PRGMKYBD	
2260 <sup>6</sup>	1 or 2	ALKYB2260 or NMKYB2260 or DEKYB2260	IBM3 <sup>7</sup>
2265 <sup>15</sup>			IBM3 <sup>7</sup>
2280			
2282			
2301			
2302	3 or 4		
2303			
2305	1 or 2	2-CHANSW, SHARED	
2311			
2314		ALTCTRL <sup>8</sup> 2-CHANSW SHARED	
2321			
2401	1, 2, or 3	READWRITE 7-TRACK or 9-TRACK MDECOMPAT	
	4, 5, or 6	READWRITE 9-TRACK DUALDENS	
	8	READWRITE 7-track	

IODEVICE

Figure 82 (Part 2 of 6). Keyword Values for the IODEVICE Macro Instruction

UNIT	MODEL	FEATURE	ADAPTER
2402	1, 2, or 3	READWRITE 7-TRACK or 9-TRACK MDECOMPAT	
	4, 5, or 6	READWRITE 9-TRACK DUALDENS	
2403	1, 2, or 3	7-TRACK or 9-TRACK MDECOMPAT	
	4, 5, or 6	7-TRACK or 9-TRACK DUALDENS	
2404 <sup>9</sup>	1, 2, or 3	7-TRACK or 9-TRACK	
2415	1, 2, 3, 4, 5, or 6	7-TRACK or 9-TRACK	
2420			
2495			
2501	B1 or B2	CARDIMAGE	
2520	B1, B2, or B3	CARDIMAGE	
2540R	1		
2540P	1		
2671	1		
2740 <sup>1 10</sup>		AUTOANSR AUTOCALL AUTOPOLL CHECKING <sup>11</sup> INTERRUPT <sup>16</sup> SCONTROL <sup>11</sup> or XCONTROL <sup>11</sup> OIU <sup>11</sup>	IBM 1
2740C		AUTOANSR CHECKING <sup>12</sup>	IBM 1
2740X		AUTOANSR AUTOCALL CHECKING <sup>12</sup>	IBM 1
2741C		AUTOANSR	IBM 1
2741P		AUTOANSR	IBM 1

Figure 82 (Part 3 of 6). Keyword Values for the IODEVICE Macro Instruction



UNIT	MODEL	FEATURE	ADAPTER
2780 <sup>1</sup>		AUTOANSR AUTOCALL AUTOPOLL DUALCODE DUALCOMM	BSCA
2955			
3066 <sup>17, 19</sup>			
3210 <sup>18</sup>			
3211			
3215 <sup>18</sup>			
3277	1 or 2	ASCACHAR or ASCBCHAR or DOCHAR or FRCHAR or GRCHAR or KACHAR or UKCHAR ASKY3277 or DEKY3277 or EBKY3277 or OCKY3277 AUDALRM KB66KEY or KB70KEY or KB78KEY or KB81KEY MAGCDRD NUMLOCK SELPEN	
3284	1 or 2	DOCHAR or FRCHAR or GRCHAR or KACHAR or UKCHAR	
3286	1 or 2	DOCHAR or FRCHAR or GRCHAR or KACHAR or UKCHAR	
3330		2-CHANSW, SHARED	
3410	1, 2, or 3	9-TRACK DUALDENS	
3420	3, 5, 7	7-TRACK or 9-TRACK DUALDENS	

IODEVICE

Figure 82 (Part 4 of 6). Keyword Values for the IODEVICE Macro Instruction

UNIT	MODEL	FEATURE	ADAPTER
3505		CARDIMAGE	
3525		CARDIMAGE MULTILINE or TWOLINE	
5450 <sup>19</sup>			
7770	3		
115A			TELE 1
83B3			TELE1
BSC 1	Any BSC station on a nonswitched point-to-point line.	DUALCODE DUALCOMM	BSCA
BSC2 <sup>13</sup>	Any BSC station on a switched point-to-point line.	AUTOCALL AUTOANSR DUALCODE DUALCOMM	BSCA
BSC3 <sup>14</sup>	Any BSC station on a nonswitched multipoint line.	AUTOPOLL DUALCODE DUALCOMM	BSCA
S360		AUTOANSR AUTOCALL DUALCODE DUALCOMM	BSCA
TWX		AUTOANSR AUTOCALL	TELE2
WTTA			TELEW

Figure 82 (Part 5 of 6). Keyword Values for the IODEVICE Macro Instruction

- <sup>1</sup>AUTOPOLL cannot be specified if either AUTOANSR or AUTOCALL (or both) is specified.
- <sup>2</sup>A 2150 used to connect a 1052 is addressed as the 1052, and may not be specified.
- <sup>3</sup>A 1404 printer is supported only as a continuous-form printer, and must be specified as a 1403 Model 2.
- <sup>4</sup>If the 2250 Model 1 is specified as a console device, it must have the alphameric keyboard, character generator, and the 4K or 8K buffer. CONOPTS=MCS must be specified in the SCHEDULR macro instruction.
- <sup>5</sup>If the 2250 Model 3 is specified as a console device, it must have the alphameric keyboard. CONOPTS=MCS must be specified in the SCHEDULR macro instruction and the GRAPHICS macro instruction must be coded.
- <sup>6</sup>If the 2260 Model 1 is specified as a console device, it must have the alphameric keyboard and the 2848 must be a Model 3 that has the line addressing feature. CONOPTS=MCS must be specified in the SCHEDULR macro instruction.
- <sup>7</sup>For only the 2848/2260 or 2845/2265.
- <sup>8</sup>ALTCTRL can be specified only for an M65MP system.
- <sup>9</sup>The READWRITE feature is implicit for the 2404 and must not be specified.
- <sup>10</sup>If CONOPTS=MCS is specified in the SCHEDULR macro instruction and the 2740 is specified as a console device, the 2740 station control feature cannot be used, the 2740 must have the record checking feature, the 2740 can be connected only to the 2701, 2702, or 2703, and the communications line must be nonswitchable. Only one 2740 per communications line can be specified as a console device.
- <sup>11</sup>CHECKING must be specified if OIU is specified; SCONTROL and XCONTROL cannot be specified if OIU is specified.
- <sup>12</sup>CHECKING must be specified.
- <sup>13</sup>AUTOCALL and AUTOANSR are assumed.
- <sup>14</sup>AUTOPOLL is assumed.
- <sup>15</sup>2845/2265 display control and display station.
- <sup>16</sup>If RPQ#S30031 is present, FEATURE=INTERRUPT may be specified. This offers improved performance in applications where contention for the console is high.
- <sup>17</sup>Only for S/370 Model 165.
- <sup>18</sup>Only for S/370 Models 135, 145 or 155.
- <sup>19</sup>CONOPTS=MCS must be specified in the SCHEDULR macro instruction, and the GRAPHICS macro instruction must be coded.

Figure 82 (Part 6 of 6). Keyword Values for the IODEVICE Macro Instruction

Examples: The following example illustrates the use of the IODEVICE macro instruction to describe an IBM 1404 Model 2 printer. The address of the device is 20E.

```
PRINTER2 IODEVICE UNIT=1403,MODEL=2,ADDRESS=20E
```

IODEVICE

The following example illustrates the use of the IODEVICE macro instruction to describe an IBM 2401 Model 3, magnetic tape drive with a 9-track head. The address of the device is 181. This device can be addressed alternatively through channel 2.

```
TAPE1 IODEVICE UNIT=2401,ADDRESS=181,OPTCHAN=2,MODEL=3,F  
      EATURE=9-TRACK
```



## LINKLIB

You can use the LINKLIB macro instruction to add your own routines, in load module form, to the link library (SYS1.LINKLIB) for the new system. This macro instruction is optional.

### Basic Requirements

- The load modules that are going to be included in SYS1.LINKLIB must be members of a partitioned data set.
- The data set that contains the load modules must be cataloged in the generating system as SYS1.name.

### Processor/Library Generation

- Sufficient space must have been allocated in SYS1.LINKLIB for your own routines.

---

Name	Operation	Operand
	LINKLIB	PDS=SYS1.name MEMBERS=(name[,name]...)

---

#### PDS=

specifies the partitioned data set that contains the routines to be added.

#### SYS1.name

The name cannot exceed eight alphameric characters. The first character must be alphabetic.

#### MEMBERS=

specifies the member to be added.

#### name

cannot exceed eight alphameric characters. The first character must be alphabetic.

#### Note:

- **Maximum:** A maximum of 20 routines may be included in the link library during one system generation process.

### Coding Hints and Additional Information

The following example illustrates the use of the LINKLIB macro instruction to add the routines PAYROLL, COMPILER, and MULT to the link library. These routines are members of the SYS1.USER partitioned data set.

```
LINKLIB PDS=SYS1.USER, MEMBERS=(PAYROLL,COMPILER,MULT)
```

LINKLIB



# LOADER

This macro instruction specifies that the loader processing program is to be included. This instruction is optional.

## Basic Requirements

- This macro instruction can appear only once in the input deck.

## Processor/Library Generation

- The loader program can be added to or deleted from the system.
- If it was specified in an earlier generation, it does not have to be respecified to keep it in the new system.

---

Name	Operation	Operand
	LOADER	[LIN=in ddname] [LIB=lib ddname] [PRINT=print ddname] [SIZE=size] [PARM=(option[ ,option]...)]

---

### LIN=

in ddname

is the ddname of the primary input data set that contains the input to the loader program.

Default: If this keyword parameter is omitted, the default is SYSLIN.

### LIB=

lib ddname

is the ddname of the library that is searched to resolve external references.

Default: If this keyword parameter is omitted, the default is SYSLIB.

### PARM=

specifies the options that are used by the loader program. The options can be listed in any order.

#### PRINT

specifies that the diagnostic messages and the map of external references are written to the data set specified by the PRINT parameter.

#### NOPRINT

specifies that neither diagnostic messages nor the map of external references are written.

Default: If NOPRINT is omitted, PRINT is the default value.

LOADER

Notes:

Required: PRINT must be specified if you want the map of external references.

NOMAP

specifies that the map of external references is not written.

MAP

specifies that the map of external references is written to the data set specified by the PRINT parameter.

Default: If MAP is omitted, NOMAP is the default value.

Notes:

- Restriction: MAP cannot be specified if NOPRINT has been coded.
- Required: PRINT must be specified if MAP is specified.

NOLET

specifies that execution of the loaded program is not attempted when a level 2 severity error occurs during loading.

LET

specifies that execution of the loaded program is attempted when a level 2 severity error occurred during loading.

Default: If LET is omitted, NOLET is the default value.

CALL

specifies that the partitioned data set specified in the LIB parameter is searched for any unresolved external references that remain after the loaded program has been processed.

NOCALL

specifies that the partitioned data set specified in the LIB parameter is not searched.

Default: If NOCALL is omitted, CALL is the default value.

Note:

Required: If NOCALL is specified, NORES must be specified.

RES

specifies that the resident modules are searched to resolve any external references that remain after the loaded program has been processed and before the SYSLIB defined data set is searched.

NORES

specifies that the resident modules are not searched.

Default: If NORES is omitted, RES is the default value.

Notes:

- Required: CALL must be specified if RES is specified.
- NORES cannot be specified if CALL has been specified.

Default: If this keyword parameter is omitted, the default values are: PRINT, NOMAP, NOLET, CALL, RES.



PRINT=

print ddname

is the ddname of the data set used for the map of external references and the diagnostic messages.

Default: If this keyword parameter is omitted, the default is SYSLOUT.

SIZE=

is the amount of dynamic main storage required by the loader program to hold its own buffers, its own tables, and the problem program.

size

is a decimal number of the form nnnnK, that can be from 2K to 8192K. (K represents 1024 bytes.)

Default: If this keyword parameter is omitted, the default is 100K.

#### Coding Hints and Additional Information

Detailed information about the loader program can be found in OS Linkage Editor and Loader.

Information that can be used to plan for the loader program is provided below.

Dynamic: The loader program will be loaded into the dynamic portion of main storage

Resident: You can make the loader program resident in the reenterable load module area for MFT or in the link pack area for MVT (including M65MP) systems by:

1. Adding either one or both loader program load module names to the RAM list in the SYS1.PARMLIB.
2. Putting either one or both loader program load module names in a separate list and specifying their inclusion at NIP time. To make it possible to do this, the COMM option in the OPTIONS parameter of the SUPRVSOR macro instruction must be specified. When the message IEA101A requests 'SPECIFY SYSTEM PARAMETERS', the operator uses the RAM=aa form in the reply.

Example: The following example illustrates the use of the LOADER macro instruction to specify the inclusion of the loader program in the new operating system. The data set SYSLOUT is used to hold diagnostic messages and the map of external references. The dynamic storage available to the loader program is 50K. The map of external references is to be placed on SYSLOUT. Execution is not attempted if there is a severity-level-2 error during loading. SYSLIB will be searched for any unresolved references after the loaded program has been processed. The resident modules will not be searched to resolve any external references.

LOADER PRINT=SYSLOUT,SIZE=50K,PARM=(MAP,NORES)

LOADER



# MACLIB

The MACLIB macro instruction specifies the inclusion of the macro library (SYS1.MACLIB) in the new operating system. This macro instruction is optional, except when SVC2B (CIRB) is desired in the new system -- then it is required.

## Basic Requirements

- Space must be allocated for SYS1.MACLIB. It may be cataloged in the new system during the preparation for system generation.
- This macro instruction may not be used if the volume that contains SYS1.MACLIB of the generating system and the volume that is to contain the new SYS1.MACLIB cannot be mounted at the same time as the IEBCOPY step executed after the Stage II assemblies. (See "The Job Stream" section.) SYS1.MACLIB may be included in the new system after system generation with the IEBCOPY utility program.
- If this macro instruction is omitted, the IMDSADMP program are not included. However, the IMDSADMP service aid program may also be retrieved from SYS1.MACLIB after system generation, as described in OS Service Aids.

## Processor/Library Generation

- This macro instruction can be used if SYS1.MACLIB has not been added to the system either during a previous generation or with the IEBCOPY utility program.

---

Name	Operation	Operand
	MACLIB	[ UNIT=name VOLNO=serial [ EXCLUDE=(option[,option]...) ]

---

### EXCLUDE

specifies the groups of macro instructions to be excluded from the new SYS1.MACLIB has one or more of the following values.

### BTAM

specifies that the macro instructions used for BTAM are to be excluded.

### QTAM

specifies that the macro instructions used for QTAM are to be excluded.

### TCAM

specifies that the macro instructions used for TCAM are to be excluded.

### TSO

specifies that the macro instructions used for TSO are to be excluded.

### GPS

specifies that the macro instructions used for graphics programming services (GPS) are to be excluded.

### OCR

specifies that the macro instructions used for optical readers are to be excluded.

MACLIB

Note:

Coding: These values can be listed in any order.

UNIT=

specifies a direct-access device in the generating system that will contain the volume on which the macro library will be generated.

name  
is the device type.

Default: If the UNIT and VOLNO keyword parameters are omitted, the macro library is placed on the new system-residence volume.

Note:

Required: If this keyword parameter is specified, VOLNO must be coded.

VOLNO=

specifies the volume that is to contain the macro library.

serial  
is the serial number.

Default: If the UNIT and VOLNO keyword parameters are omitted, the macro library is placed on the new system-residence volume.

Notes:

- Value: The value specified must be the same as the value given to the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.MACLIB during the preparation for system generation.
- Required: If this keyword parameter is specified, UNIT must be coded.

Coding Hints and Additional Information

This example illustrates the use of the MACLIB macro instruction to specify the inclusion of the macro library in the new operating system. The unit name is 2311. The volume serial number is 003475. The macro instructions used for graphic programming services are to be excluded from the new SYS1.MACLIB.

MACLIB UNIT=2311,VOLNO=003475,EXCLUDE=GPS

# OUTPUT

The OUTPUT macro instruction specifies the inclusion of the class defaults for the OUTPUT command in the new operating system. This macro instruction is optional and valid only in a complete operating system or processor generation of a TSO system.

## Basic Requirement

The CMDLIB macro instruction must be specified if the OUTPUT macro instruction is specified.

## Processor/Library Generation

The CMDLIB macro instruction must be specified during a processor/library generation if it was not specified during the complete operating system generation.

---

Name	Operation	Operand
	OUTPUT	CLASS= ([ classname1 ][ ,classname2 ])

---

**CLASS=**  
specifies class defaults for the TSO OUTPUT command.

**classname1**  
is the background message class default.

**classname2**  
is the print output class default.

**Default:** The default value for either class is X.

### Note:

**Value:** Each value specified must be a letter from A to Z or a number from 0 to 9.

## Coding Hints and Additional Information

This example illustrates the use of the OUTPUT macro instruction to specify the class defaults for the TSO OUTPUT command. X is the background message class default and A is the print output class default.

```
OUTPUT CLASS=(,A)
```

OUTPUT



# PARMLIB

The PARMLIB macro instruction specifies the inclusion of the parameter library (SYS1.PARMLIB) in the new operating system. This macro instruction is optional.

## Basic Requirements

- Space must be allocated for SYS1.PARMLIB.
- If SYS1.PARMLIB is not to reside on the system volume, it must be cataloged in the new system during the preparation for system generation.
- If this macro instruction is not specified, a null data set must be defined for SYS1.PARMLIB in the new system. (See "Input Deck for Initialization" in the "Preparation for System Generation" section.)

## Processor/Library Generation

- If a null SYS1.PARMLIB is being replaced, it must be scratched and uncataloged before the new SYS1.PARMLIB is initialized during the preparation for generation.

---

Name	Operation	Operand
	PARMLIB	[UNIT=name VOLNO=serial]

---

**Default:** If the UNIT and VOLNO keyword parameters are omitted, the parameter library is placed on the new system-residence volume.

### UNIT=

specifies a direct-access device in the generating system that will contain the volume on which the parameter library will be generated.

name

is the device type.

### Note:

**Required:** If this keyword parameter is specified, VOLNO must also be coded.

### VOLNO=

specifies the volume that is to contain the parameter library.

serial

is a serial number.

### Notes:

- **Value:** The value specified must be the same as the value specified in the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.PARMLIB.
- **Required:** If this keyword parameter is specified, UNIT must also be coded.

PARMLIB

## Coding Hints and Additional Information

This example illustrates the use of the PARMLIB macro instruction to specify the inclusion of the parameter library in the operating system to be generated. The unit name is 2311. The volume serial number is RFS111.

```
PARMLIB UNIT=2311,VOLNO=RFS111
```



## PARTITNS

The PARTITNS macro instruction specifies the number of partitions and the class and size of each partition in an MFT system. This macro instruction is required and used only if TYPE=MFT is specified in the CTRLPROG macro instruction.

### Note:

The Generalized Trace Facility (GTF) component of the control program requires a partition when started. If one partition or two partitions and TCAM are requested, an additional inactive (0K size) partition is generated.

### Nucleus Generation

This macro instruction must be used to specify either the same or different partitions than those specified for the first MFT nucleus.

---

Name	Operation	Operand
	PARTITNS	operand[,operand]...

---

### For Systems Without Hierarchies

Each operand must be written in the following format for MFT without storage hierarchies:

Pn(C-class,S-nK)  
S

Uppercase letters, parentheses, and hyphens must be written exactly as shown.

Pn  
specifies the partition number.

n  
is an integer from 0 to 51.

### Note:

- Partitions must be numbered sequentially starting with 0, but they may be specified in any order.

C-class  
specifies the function of the partition.

class  
is one of the following values:

C-R  
specifies a resident reader partition.

Note: A maximum of 3 resident reader partitions may be specified.

PARTITNS

C-W specifies a resident writer partition.

Note: A maximum of 36 resident writer partitions may be specified.

C-xxx specifies a problem-program partition.

xxx is one, two, or three alphabetic characters from A through O that indicate the job classes that can use the partition.

Notes:

- The order in which these characters are written indicates the order in which the job classes are to be scheduled for the partition.
- At least one, and not more than fifteen, problem-program partitions may be specified.

S-nK specifies the size of the partition in bytes.

nK must be an expression from 8K to xK, where xK is the size of main storage minus the size of the nucleus. (K represents 1024 bytes.)

Notes:

- All resident reader partitions and at least one problem-program partition must be at least the size of the scheduler.
- The size of each resident writer partition must be at least 10K bytes.
- If OPTIONS=PROTECT is specified in the SUPRVSOR macro instruction, n should be a multiple of 2 for all partitions. In this case, (n+1) is assumed at IPL time for each n that is not a multiple of 2.
- MFT With Subtasking: When ATTACH with the subtasking capability is selected, the partition size must be large enough to hold all the tasks and all the subtasks that could exist at one time within a partition.

For Systems With Hierarchies

Each operand must be written in one of the following formats if Hierarchy=INCLUDE is specified in the CTRLPROG macro instruction:

Pn(C-class,H0-nk)	for partitions in hierarchy 0
Pn(C-class,H1-hK)	for partitions in hierarchy 1
Pn(C-class,H0-nK,H1-nK)	for partitions consisting of two segments (one in each hierarchy)

Uppercase letters, parentheses, hyphens, and numbers (0 and 1) must be written exactly as shown.

If there is no IBM 2361 Core Storage unit on the system and hierarchy 1 is used, you will have to redefine the partitions at IPL time.

In using storage hierarchies on a Model 50, if reader or writer partitions are defined in hierarchy 1 and programs containing CCWs are loaded in hierarchy 1, overrun will occur. This will degrade the performance or result in an unrecoverable I/O error.

**Pn**  
specifies the partition number.

**n**  
is an integer from 0 to 51.

Note: Partitions may be specified in any order as long as every partition in the sequence is included.

**C-class**  
specifies the function of the partition.

**class**  
is one of the following values:

**C-R**  
specifies a resident reader partition.

Note: A maximum of 3 resident reader partitions may be specified.

**C-W**  
specifies a resident writer partition.

Note: A maximum of 36 resident writer partitions may be specified.

**C-xxx**  
specifies a problem-program partition.

**xxx**  
is one, two, or three alphabetic characters from A through O that indicate the job classes that can use the partition.

Notes:

- The order in which these characters are written indicates the order in which the job classes are to be scheduled for the partition.
- At least one, and not more than fifteen, problem-program partitions may be specified.

**H0-nK** (for partitions in hierarchy 0), **H1-nK** (for partitions in hierarchy 1),  
**H0-nK, H1-nK** (for segmented partitions)

specifies the size of the partition in bytes.

**nK**  
must be an expression from 8K to xK, where xK is the size of main storage minus the size of the nucleus. (K represents 1024 bytes.)

Default: n should be a multiple of 2; if it is not, (n+1) is assumed at IPL time.

PARTITNS

Notes:

- All resident reader partitions and at least one problem-program partition must be at least the size of the scheduler.
- The size of each resident writer partition must be at least 10K bytes.
- Resident reader and resident writer partitions must not be segmented.

Coding Hints and Additional Information

This example illustrates the use of the PARTITNS macro instruction to generate a four-partition system. Partition 0 is a 30720-byte problem program partition. Jobs that specify CLASS=B on their JOB cards are scheduled into partition 0. If no CLASS=B jobs are pending, CLASS=C jobs are scheduled. If neither CLASS=B nor CLASS=C jobs are pending, then CLASS=A jobs are scheduled into partition 0. Partition 2 is a 10240-byte resident writer partition. Partition 1 is a 30720-byte resident reader partition (partition 1 need not be specified before partition 2). Partition 3 is a 36864-byte problem-program partition. Only jobs that specify CLASS=D are scheduled into partition 3.

```
PARTITNS  P0 (C-BCA,S-30K) , P2 (C-W,S-10K) , P1 (C-R,S-30K) ,  
          P3 (C-D,S-36K)
```

The following example illustrates the use of the PARTITNS macro instruction to generate an eight-partition system with LCS. Partition 0 is a 47104-byte problem-program partition in hierarchy 0 that supports job classes A, D, and E. Partition 1 is a 47107-byte resident reader partition in hierarchy 0. Partition 4 is a 10240-byte resident writer partition in hierarchy 0. Partition 3 is a segmented problem-program partition with 20480 bytes in hierarchy 0 and 102400 bytes in hierarchy 1. Partition 3 supports job classes D, and F. Partition 5 is a 20480-byte resident reader partition in hierarchy 0. Partition 6 is a 14336-byte resident writer partition in hierarchy 0. Partition 7 is a 204800-byte problem-program partition in hierarchy 0 that supports job classes C and A. Partition 2 is a segmented problem-program partition with 10240 bytes in hierarchy 0 and 51200 bytes in hierarchy 1. Partition 2 supports job class M.

```
PARTITNS  P0 (C-ADE,H0-46K) , P1 (C-R,H0-46K) ,  
          P4 (C-W,H0-10K) , P3 (C-DF,H0-20K,H1-100K) ,  
          P5 (C-R,H0-20K) , P6 (C-W,H0-14K) ,  
          P7 (C-CA,H0-200K) , P2 (C-M,H0-10K,H1-50K)
```

# PL1

The PL1 macro instruction specifies the inclusion of the PL/I compiler. This macro instruction is optional.

## Basic Requirements

- PL1LIB macro instruction must be specified.
- INSTSET=UNIV must be specified in the CENPROCS macro instruction.
- If the object programs contain WAIT statements with multiple arguments, then WAIT=MULTIPLE must be specified in the SUPRVSOR macro instruction.
- If an automatic restart of a job step is to be forced or canceled by using the IHERES routine, the user completion code 4092 must be specified in the ELIGBLE keyword parameter of the CKPTREST macro instruction. The RD parameter must also be specified in the JOB or EXEC statement with R or RNC as the subparameter.

## Processor/Library Generation

- There must be sufficient space for the new PL/I compiler in SYS1.LINKLIB.
- The PL1LIB macro instruction must be specified.
- The PL/I compiler being added must be of the same release level as the generating system.

If the compilation time defaults are going to be changed, then these additional conditions must exist:

- The member IEMAP must be scratched from SYS1.LINKLIB. (To regain the space occupied by IEMAP, use the IEBCOPY utility program.)
- The PL1LIB macro instruction must be the same as that used during the last generation, unless SYS1.PL1LIB is redefined.

Name	Operation	Operand
	PL1	DESIGN=F [ SIZE=size ] [ CHARSET= [ CHAR60 ] [ CHAR48 ] ] [ PUNCH= [ NODECK ] [ DECK ] ] [ TYPERUN= [ LOAD ] [ NOLOAD ] ] [ SORCODE= [ EBCDIC ] [ BCD ] ] [ OBJLIST= [ NOLIST ] [ LIST ] ] [ MSGLEV= [ FLAGW ] [ FLAG ] [ FLAGS ] ] [ SYNTAXLV= [ SYNCHKT ] [ SYNCHKE ] [ SYNCHKS ] ]

PL1

Name	Operation	Operand
	PL1 (continued)	$\left[ \begin{array}{l} \text{OPT} = \left\{ \begin{array}{l} 1 \\ 2 \\ 0 \end{array} \right\} \\ \text{SORLIST} = \left\{ \begin{array}{l} \text{SOURCE} \\ \text{NOSOURCE} \end{array} \right\} \\ \text{EXTLIST} = \left\{ \begin{array}{l} \text{NOEXTREF} \\ \text{EXTREF} \end{array} \right\} \\ \text{ATRLIST} = \left\{ \begin{array}{l} \text{NOATR} \\ \text{ATR} \end{array} \right\} \\ \text{REFLIST} = \left\{ \begin{array}{l} \text{NOXREF} \\ \text{XREF} \end{array} \right\} \\ \text{SORMGIN} = (m, n) \\ \text{LINECNT} = \text{number} \\ \text{CMPTIME} = \left\{ \begin{array}{l} \text{NOMACRO} \\ \text{MACRO} \end{array} \right\} \\ \text{MACLIST} = \left\{ \begin{array}{l} \text{SOURCE2} \\ \text{NOSOURCE2} \end{array} \right\} \\ \text{MCPUNCH} = \left\{ \begin{array}{l} \text{NOMACDECK} \\ \text{MACDECK} \end{array} \right\} \\ \text{COMPILE} = \left\{ \begin{array}{l} \text{COMP} \\ \text{NOCOMP} \end{array} \right\} \\ \text{STMDIAG} = \left\{ \begin{array}{l} \text{NOSTMT} \\ \text{STMT} \end{array} \right\} \\ \text{DICTYPE} = \left\{ \begin{array}{l} \text{NOEXTDIC} \\ \text{EXTDIC} \end{array} \right\} \\ \text{OPRINT} = \left\{ \begin{array}{l} \text{OP} \\ \text{NOOP} \end{array} \right\} \\ \text{LEVCNT} = \left\{ \begin{array}{l} \text{NONEST} \\ \text{NEST} \end{array} \right\} \\ \text{OBJCODE} = \left\{ \begin{array}{l} \text{OBJOUT} \\ \text{OBJIN} \end{array} \right\} \\ \text{PAGECTI} = \text{column} \\ \text{DELETE} = (\text{item}[, \text{item}] \dots) \end{array} \right.$

**CHARSET=**

specifies the number of characters in the character set used to write the source program to be compiled.

**CHAR60**

specifies a character set with 60 characters.

**CHAR48**

specifies a character set with 48 characters.

Default: If this keyboard parameter is omitted, CHAR60 is assumed.

**DESIGN=**

specifies the F-design level of the PL/I compiler.

**SIZE=**

specifies the maximum number of bytes of main storage available to the PL/I compiler at compilation time.

**size**

is an integer from 45056 to 999999.

Default: If this keyword parameter is omitted, a value of 999999 is assumed.

Notes:

- Use: For further information on this keyword parameter, refer to OS PL/I (F) Programmer's Guide.
- Recommended: It is recommended that SIZE=999999 not be used if the storage available is 49152 bytes or less. In this case, the compiler may calculate that there is not enough storage available when, in fact, compilation is still possible. To avoid this situation, an actual value must be given to SIZE.
- Maximum: If 999999 is specified, all available main storage is to be used by the PL/I compiler.

The following keyword parameters provide the setting of defaults for the various options that can affect a PL/I compilation. Defaults are the options that are assumed if the corresponding values of the PARM keyword parameters are omitted from an EXEC statement in a PL/I compilation. The DELETE keyword parameter specifies a list of values that cannot be used as values of the PARM keyword parameter.

ATRLIST=

specifies the production of a listing for each identifier, giving the identifier with full qualification, the statement number declaring the identifier, and a list of attributes pertaining to the identifier.

ATR

specifies that a listing is to be produced.

NOATR

specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, NOATR is assumed.

CMPTIME=

specifies whether there is to be processing during compilation.

MACRO

specifies that compilation-time processing is required.

NOMACRO

specifies that compilation-time processing is not required.

Default: If this keyword parameter is omitted, NOMACRO is assumed.

COMPILE=

specifies that compilation is to proceed after the compilation-time processor has been used.

COMP

specifies that compilation is required.

NOCOMP

specifies that compilation is not required.

Default: If this keyword parameter is omitted, COMP is assumed.

**DICTYPE=**

specifies that an extended capacity dictionary is to be used even at the expense of compilation time.

**EXTDIC**

specifies that a extended dictionary is to be used.

**NOEXTDIC**

specifies that a normal dictionary is to be used.

Default: If this keyword parameter is omitted, NOEXTDIC is assumed.

**EXTLIST=**

specifies the production of a listing of all external data, external entries, and files.

**EXTREF**

specifies that a listing is to be produced.

**NOEXTREF**

specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, NOEXTREF is assumed.

**LEV CNT=**

specifies the production of a scope level and DO-nest depth count with the source listing.

**NEST**

specifies that a listing is to be produced.

**NONEST**

specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, NONEST is assumed.

**LINECNT=**

specifies the maximum number of lines to be printed in each page of a PL/I compiler output listing.

**number**

is an integer from 10 to 99.

Default: If this keyword parameter is omitted, a value of 50 is assumed.

**MACLIST=**

specifies the listing of the input to the compilation-time processor.

**SOURCE2**

specifies that a listing is to be produced.

**NOSOURCE2**

specifies that the listing is not to be produced.

Default: If this keyword parameter is omitted, SOURCE2 is assumed.



**MCPUNCH=**

specifies the production of a punched deck of the compilation-time processor output.

**MACDECK**

specifies that a deck is to be produced.

**NOMACDECK**

specifies that a deck is not to be produced.

Default: If this keyword parameter is omitted, NOMACDECK is assumed.

**MSGLEV=**

specifies the type of compilation messages to be printed.

**FLAGW**

specifies that warning messages, error messages, and severe error messages are to be printed.

**FLAGE**

specifies that only error messages and severe error messages are to be printed.

**FLAGS**

specifies that only severe error messages are to be printed.

Default: If this keyword parameter is omitted, FLAGW is assumed.

**OBJCODE=**

specifies the inclusion of additional code and a module in the object program to handle imprecise interrupts and to provide complete support for running the object program on a S/360 Model 91 or 195.

**OBJIN**

specifies that the additional code and module are to be included.

**OBJOUT**

specifies that the additional code and module are not to be included.

Default: If this keyword parameter is omitted, OBJOUT is assumed.

**OBJLIST=**

specifies the production of a listing of the object program.

**LIST**

specifies that a listing is to be produced.

**NOLIST**

specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, NOLIST is assumed.

**OPRINT=**

specifies the production of a listing of compiler options.

**OP**

specifies that this listing is to be produced.

**NOOP**

specifies that it is not to be produced.

Default: If this keyword parameter is omitted, OP is assumed.

**OPT=**

specifies optimization of execution time of the object program produced by the compiler.

0

specifies that the object program is to be optimized unless it increases the object-time storage requirements.

1

specifies that the object program is to be optimized even at the expense of the object-time storage requirements.

2

specifies the same function as OPT=1 and that the object code for DO groups, subscripts, and array expressions will be optimized even though it may severely affect the compilation time.

Default: If this keyword parameter is omitted, 1 is assumed.

**PAGECTL=column**

specifies the number of the column in which a control character is to be specified.

Default: If this keyword parameter is omitted, no page control is provided.

**Note:**

The number must be outside the scope of the values in the SORMGIN keyword parameter.

**PUNCH=**

specifies the production of a punched deck of the object program.

**DECK**

specifies that a punched deck is to be produced.

**NODECK**

specifies that a punched deck is not to be produced.

Default: If this keyword parameter is omitted, NODECK is assumed.

**REFLIST=**

specifies the production of a listing for each identifier, including the identifier with full qualification, the number of the statement that declares the identifier, and a list of all statements in which reference is made to the identifier.

**XREF**

specifies that a listing is to be produced.

**NOXREF**

specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, NOXREF is assumed.

**SORCODE=**  
specifies the character set used to keypunch the source programs to be compiled.

**BCD**  
specifies the BCD character set.

**EBCDIC**  
specifies the EBCDIC character set.

Default: If this keyword parameter is omitted, EBCDIC is assumed.

**SORLIST=**  
specifies the production of a printed listing of the PL/I source program.

**SOURCE**  
specifies that a listing of the source text is to be produced.

**NOSOURCE**  
specifies that a listing is not to be produced.

Default: If this keyword parameter is omitted, SOURCE is assumed.

**SORMGIN=**  
specifies the margins for scanning of the source statements.

**m**  
specifies the beginning margin

**n**  
specifies the end margin. If the source statement input to the compiler is from the system input stream (that is, if it follows a DD\* statement), the condition  $2 \leq m \leq n \leq 100$  must hold.

Default: If this keyword parameter is omitted, a value of 2 is assumed for m and a value of 72 is assumed for n.

**STMDIAG=**  
specifies the contents of diagnostic messages printed during execution of the compiled source program.

**STMT**  
specifies that the messages are to contain source program statement numbers.

**NOSTMT**  
specifies that the messages are not to contain source program statement numbers.

Default: If this keyword parameter is omitted, NOSTMT is assumed.

**Note:**

Offsets from PL/I entry points are included in the messages in both cases.

**SYNTAXLV=**  
specifies the condition for compilation termination if errors occur during syntax checking.

**SYNCHKE**  
specifies that termination will occur after compilation if an error of severity ERROR or above is found.

**SYNCHKS**

specifies that termination will occur after compilation if an error of severity SEVERE or above is found.

**SYNCHKT**

specifies that termination will occur immediately if an error of severity TERMINATION is found. No special message will be printed.

Default: If this keyword parameter is omitted, SYNCHKT is assumed.

**TYPERRUN=**

specifies the production of input to the linkage editor from the program being compiled.

**LOAD**

specifies that the program is to be processed by the linkage editor after compilation.

**NLOAD**

specifies that the source program is to be compiled only.

Default: If this keyword parameter is omitted, LOAD is assumed.

**DELETE=**

specifies that the keyword values or keywords in the value list cannot be used at compilation time in the PARM field of the EXEC statement. The following values can be specified; each has been described above:

ATR	LINECNT	NLOAD	OPT
BCD	LIST	NOMACDECK	PAGECTL
CHAR48	LOAD	NOMACRO	SIZE
CHAR60	MACDECK	NONEST	SORMGIN
COMP	MACRO	NOOP	SOURCE
DECK	NEST	NOSOURCE	SOURCE2
EBCDIC	NOATR	NOSOURCE2	STMT
EXTDIC	NODECK	NOSTMT	SYNCHKE
EXTREF	NOCOMP	NOXREF	SYNCHKS
FLAGE	NOEXTDIC	OBJIN	SYNCHKT
FLAGS	NOEXTREF	OBJOUT	XREF
FLAGW	NOLIST	OP	

Coding Hints and Additional Information

Syntax Checker: The syntax checking routines for this compiler can be included in the new or modified system by use of the CHECKER macro instruction.

Example: The following example illustrates the use of the PL1 macro instruction to specify an F-design-level PL/I compiler that operates in 56320 bytes of main storage. The compiled source programs are to be processed by the linkage editor unless otherwise specified at compilation time. All warning and error messages are to be printed, the EBCDIC character set is used to punch the source programs, the character set used to write the source programs has 60 characters, and the execution time of the object programs is not to be optimized.

The default options for scanning the source statements are 2 and 72, and there are 50 lines per printed page.

The following keywords and keyword values cannot be used at compilation time in the PARM field of the EXEC statement: CHAR48, BCD, FLAGE, FLAGS, OPT, LINECNT, and SORMGIN. Unless otherwise specified at compilation time, listings of the source text are to be produced. The following, however, are not produced:

- A punched deck of the object program
- Listings of the object program
- A listing of external data, entries, and files
- Listings of identifiers and their attributes
- Numbers
- Qualifications
- References

Unless specified at compilation time, compilation-time processing is not required, and the diagnostic messages are not to contain source program statement numbers.

```
PL1 DESIGN=F, TYPERUN=LOAD, SIZE=56320,  
DELETE=(CHAR48, BCD, FLAGE, FLAGS, OPT, LINECNT, SORMGIN)
```

PL1



## PL1LIB

The PL1LIB macro instruction specifies the inclusion of the PL/I subroutine library (SYS1.PL1LIB) in the new operating system. This macro instruction is optional unless the PL1 macro instruction is specified -- then it is required.

### Basic Requirements

- Space must be allocated for SYS1.PL1LIB.
- If the object programs contain WAIT statements with multiple arguments, the WAIT=MULTIPLE must be specified in the SUPRVSOR macro instruction.
- SYS1.PL1LIB may be cataloged in the new system during the preparation for system generation.
- The shared-library feature can be used only with an operating system that is of Release 18 level or later.

### Processor/Library Generation

- The SUPRVSOR macro instruction used during the last complete system generation must be used with all the parameters coded exactly the same way.
- The PL1 macro instruction does not have to be used.
- The PL/I library being added must be of the same level as the generating system.

If the PL1LIB macro instruction is used to respecify an existing SYS1.PL1LIB, then these additional guidelines must be used.

- The existing SYS1.PL1LIB should be scratched and uncataloged before the new SYS1.PL1LIB is initialized. (To regain the space occupied by the old SYS1.PL1LIB, use the IEBCOPY utility program.)
- If SYS1.PL1LIB is respecified for a previously generated PL/I compiler, then the generating system must be of the same level as the system used to generate the existing PL/I compiler.

PL1LIB

Name	Operation	Operand
	PL1LIB	<pre>[ UNIT=name   VOLNO=serial   LIBFCNS= [REAL             COMPLEX ] ]   MODES= ( [NOTK   [ ,operand ] [ ,operand ] ] )            [TASK ] ]</pre>
		<p>If MODES is specified, then the following parameters can be used:</p> <pre>[ STORG=(operand[ ,operand]...) ] [ CONVS=(operand[ ,operand]...) ] [ STRGS=(operand[ ,operand]...) ] [ ARRAY=(operand[ ,operand]...) ] [ MATHS=(operand[ ,operand]...) ] [ STRIO=(operand[ ,operand]...) ] [ RECIO=(operand[ ,operand]...) ]</pre>

**UNIT=**

specifies a direct-access device in the generating system that will contain the volume on which the PL/I subroutine library will be generated.

name

is the device type.

**Default:** If the UNIT and VOLNO parameters are omitted, the PL/I subroutine library is placed on the new system-residence volume.

**Note:**

**Required:** If this keyword parameter is specified, VOLNO must be coded.

**VOLNO=**

specifies the volume that is to contain the PL/I subroutine library.

serial

is the serial number.

**Default:** If the UNIT and VOLNO parameters are omitted, the PL/I subroutine library is placed on the new system-residence volume.

**Notes:**

- **Value:** The value specified must be the same as the value given to the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.PL1LIB during the preparation for system generation.
- **Required:** If this keyword parameter is specified, UNIT must be coded.



**LIBFCNS=**

specifies the inclusion of complex object-time functions in SYS1.PL1LIB.

**REAL**

specifies that the complex functions are not to be included.

**COMPLEX**

specifies that the complex functions are to be included.

Default: If this keyword parameter is omitted, REAL is assumed.

**MODES=**

specifies that the PL/I shared library feature is to be included in the new system.

**NOTK**

specifies that the form of the modules included will be for tasks that are not multitasking.

**TASK**

specifies that the form of the modules included will be for tasks that are multitasking.

Default: If this keyword parameter or both NOTK and TASK are omitted, NOTK is assumed.

Notes:

- These two subparameters are mutually exclusive.
- The subparameter that is selected to be coded must appear first in the parameter statement.
- A shared library for a multitasking program cannot be used by a program that is not multitasking. The reverse is also true: a shared library for a program that is not multitasking cannot be used by a multitasking program.

**REAL**

specifies that only modules that deal with real expressions will be made resident.

**CMPX**

specifies that only modules that deal with complex expressions will be made resident.

Default: If this keyword parameter or both REAL and CMPX are omitted, REAL is assumed.

PL1LIB

Notes:

- It is recommended that when CMPX is specified that REAL also be specified, since many of the CMPX modules use real modules.
- Either REAL or CMPX, or both, must be specified if the shared-library feature is desired.

Default: If this keyword parameter is omitted, NOTK and REAL are assumed.

Notes:

COMM must be specified in the OPTIONS parameter and RENTCODE must be specified in the RESIDNT parameter of the SUPRVSOR macro instruction.

When the MODES keyword parameter is specified, the following seven keyword parameters can be used.

**ARRAY=**

specifies which array-handling modules are made resident.

**BASIC**

specifies the modules that perform all array functions on noninterleaved arrays.

**LEAF**

specifies the modules that perform all array functions on interleaved arrays.

**Note:**

Each group is modified to agree with the REAL or CMPX subparameter specified in the MODES parameter.

**CONVS=**

specifies which conversion package modules are made resident.

**BASIC**

specifies the most commonly used conversion modules.

**EDIT**

specifies the conversion modules associated with the edit stream I/O conversions.

**BIT**

specifies the bit string conversion modules.

**CHAR**

specifies the character string conversion modules.

**PICT**

specifies the modules commonly used in converting variables with the PICTURE attribute.

**STERL**

specifies the modules used when performing sterling conversions.

**OPT1**

specifies the conversion modules selected when OPT=1 or OPT=2 has been specified in the PL1 macro instruction.

**MATHS=**

specifies which modules associated with the arithmetic operators and built-in functions will be made resident.

**BASIC**

specifies the modules that are concerned with arithmetic operators, including the function forms of operators.

**SHORT**

specifies the modules associated with the short-precision form of the arithmetic built-in functions.

LONG

specifies the modules associated with the long-precision form of the arithmetic built-in functions.

Note:

Each group is modified to agree with the REAL or CMPX subparameter in the MODES parameter.

RECIO=

specifies which modules that are common to all modes of record-oriented I/O are made resident.

BASIC

specifies the modules that are common to the initialization of any record-oriented I/O statement.

WAIT

specifies the modules that handle the record-oriented I/O wait option.

Note:

Each group is modified to agree with the NOTK or TASK subparameter specified in the MODES parameter.

STORG=

specifies that some storage-management modules are made resident.

ERR

specifies that the error-handling modules are made resident. This includes all ON codes.

LISTP

specifies that all list processing modules are made resident.

Note:

The basic storage-management modules are automatically included if the MODES=TASK parameter is specified.

STRGS=

specifies which string-handling modules are made resident.

BIT

specifies the bit-string-handling modules.

CHAR

specifies the character-string-handling modules.

STR

specifies only the STRING built-in function and pseudo variable modules.

STRIO=

specifies which stream-oriented I/O modules are made resident.

DATA

specifies the modules that handle data-directed stream I/O.

LIST specifies the modules that handle list-directed stream I/O.

EDIT specifies the modules that handle edit-directed stream I/O.

Note:

Each group is modified to agree with the NOTK or TASK subparameter specified in the MODES parameter.

Coding Hints and Additional Information

Detailed information can be found in the following publications:

OS PL/I (F) Subroutine Library

OS GSP for FORTRAN IV, COBOL, and PL/I

Shared-Library Feature: The following guidelines should be used when specifying the shared library.

- MODES -- This keyword parameter must be specified if the shared-library feature is desired. NOTK and REAL are the default values. NOTK and TASK are mutually exclusive positional subparameters for MODES. The selected one must be coded first.
- STORG -- This keyword parameter can be used only if MODES is specified.
- CONVS -- This keyword parameter can be used only if MODES is specified. BASIC is dependent on whether REAL or CMPX is specified.
- STRGS -- This keyword parameter can be used only if MODES is specified.
- STRIO -- This keyword parameter can be used only if MODES is specified. The subparameters depend on whether NOTK or TASK is specified.
- RECIO -- This keyword parameter can be used only if MODES is specified. The subparameters depend on whether NOTK or TASK is specified.
- ARRAY -- This keyword parameter can be used only if MODES is specified. The subparameters depend on whether REAL or CMPX is specified.
- MATHS -- This keyword parameter can be used only if MODES is specified. The subparameters depend on whether REAL or CMPX is specified.

---

If MODES= (any combination of subparameters)

Then any of the following keyword parameters and associated subparameters can be used:

STORG	with	ERR	LISTP	
CONVS	with	EDIT	BIT	CHAR
		PICT	STERL	OPT1
STRGS	with	BIT	CHAR	STR

---

Figure 83 (Part 1 of 3). Summary of Coding Dependencies for the Shared-Library Feature

---

In addition to the combination above or below:

If MODES= NOTK or TASK

Then any of the following keyword parameters and associated subparameters can be used.

STRIO	with	DATA	EDIT	LIST
RECIO	with	BASIC	WAIT	

---

In addition to the combination above:

If MODES= REAL or CMPX

Then any of the following keyword parameters and associated subparameters can be used:

MATHS	with	BASIC	SHORT	LONG
CONVS	with	BASIC		
ARRAY	with	BASIC	LEAF	

---

Figure 83 (Part 2 of 2). Summary of Coding Dependencies for the Shared-Library Feature

Example: The following example illustrates the use of the PL1LIB macro instruction to specify the inclusion of the PL/I subroutine library in the new operating system. Complex object time functions are not to be included. This library is to be placed on the new system-residence volume.

PL1LIB

PL1LIB



# PROCLIB

The PROCLIB macro instruction specifies the inclusion of the procedure library (SYS1.PROCLIB) in the new operating system. This macro instruction is optional.

## Basic Requirements

- The SYS1.PROCLIB data set must have space allocated for it. It may be cataloged in the new system during the preparation for system generation.
- If IBM-supplied cataloged procedures are going to be used, the group names used by these procedures must be defined by UNITNAME macro instructions.
- This macro instruction can be used with an MVT generating system to include the IBM-distributed SYS1.PROCLIB if a user-assigned qualifier is specified in the INDEX parameter of the GENERATE macro instruction.
- If this macro instruction is not coded, a null data set must be defined and available to the new system.
- The CRJE macro instruction is required to support conversational remote job entry facility.

## Processor/Library Generation

- If a null SYS1.PROCLIB is going to be replaced, it must be scratched and uncataloged before the new SYS1.PROCLIB is initialized.

---

Name	Operation	Operand
	PROCLIB	[UNIT=name VOLNO=serial]

---

Default: If the keyword parameters UNIT and VOLNO are not coded, the procedure library for the new system is placed on the system-residence volume.

### UNIT=

specifies the direct-access device that will contain the volume on which the procedure library will be generated.

name

### Note:

Required: If this keyword parameter is coded, VOLNO must be coded.

### VOLNO=

specifies the serial number of the volume that is to contain the procedure library for the new system.

serial

is the serial number.

PROCLIB

Notes:

- Required: If this keyword parameter is coded, UNIT must be coded.
- Value: The value specified must be the same as that specified in the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.PROCLIB.

Coding Hints and Additional Information

This example illustrates the use of the PROCLIB macro instruction to specify the inclusion of the procedure library in the new operating system. The unit name is 2311. The volume serial number is 909090.

```
PROCLIB UNIT=2311,VOLNO=909090
```



# PTOP

The PTOP macro instruction specifies the inclusion of processor-to-processor support for transmitting data between the operating system and an IBM 1130 Disk Monitor System. This macro instruction is optional.

## Basic Requirements

- UNIT=1130 and ADAPTER=BSCA must be specified in the IODEVICE macro instruction.
- ACSMETH=BTAM must be specified in the DATAMGT macro instruction.
- SYS1.TELCMLIB must be specified with the TELCMLIB macro instruction.

## Processor/Library Generation

- UNIT=1130 and ADAPTER=BSCA in the IODEVICE macro instruction, ACSMETH=BTAM in the DATAMGT macro instruction, and the TELCMLIB macro instruction must have been specified during a complete generation.
- The FORTLIB macro instruction must be used to indicate where SYS1.FORTLIB resides if LIB=FORTLIB is specified and whether or not SYS1.FORTLIB was specified during an earlier generation, or is not on the system-residence volume.

---

Name	Operation	Operand
	PTOP	[ LIB= {FORTLIB} ] {LINKLIB}

---

### LIB=

specifies the library that will contain the System/360-1130 data transmission processor-to-processor routines.

### FORTLIB

specifies that the data transmission routines are placed in SYS1.FORTLIB.

Note: The FORTLIB macro instruction must also be specified.

### LINKLIB

specifies that the data transmission routines are placed in SYS1.LINKLIB.

Default: If this keyword parameter is omitted, FORTLIB is assumed.

### Note:

The PTOP macro instruction places the System/360-1130 data transmission modules into libraries (SYS1.FORTLIB and SYS1.LINKLIB) from which your calls to data transmission and data conversion subroutines can be resolved most efficiently.

PTOP

## Coding Hints and Additional Information

The following example illustrates the use of the PTOP macro instruction to specify inclusion of processor-to-processor support in the new operating system. The System/360-1130 data transmission routines will be placed in SYS1.FORTLIB.

```
PTOP
```

The second example provides another illustration of the use of the PTOP macro instruction specifying the inclusion of processor-to-processor support in the new operating system. The System/360-1130 data transmission routines will be placed in SYS1.LINKLIB.

```
PTOP      LIB=LINKLIB
```

# RESMODS

The RESMODS macro instruction adds user-written routines, in load module form, to the nucleus library (SYS1.NUCLEUS) to be generated. This macro instruction is optional.

## Basic Requirements

- The load modules to be included must be members of a partitioned data set.
- The data set that contains the load modules must be cataloged in the generating system as SYS1.name.

## Nucleus Generation

- Sufficient space must have been allocated to SYS1.NUCLEUS during the preparation for a complete generation.

---

Name	Operation	Operand
	RESMODS	PDS=SYS1.name MEMBERS=(name[ ,name]...)

---

### PDS=

specifies the partitioned data set that contains the load modules to be included.

### SYS1.name

is one to eight alphameric characters, the first of which is alphabetic.

### MEMBERS=

specifies the simple names of the members to be included.

### name

is one to eight alphameric characters, the first of which is alphabetic.

## Notes:

- **Maximum:** A maximum of ten load modules can be included in the nucleus.
- **SVC Routines:** If resident SVC routines are being included, each load module can contain more than one SVC routine. The type, number, and SVRB extended save area of each of the resident SVC routines to be included must be specified in the SVCTABLE macro instruction. (For further information on writing your own SVC routines, refer to OS MPT Guide or OS MVT Guide.)

RESMODS

## Coding Hints and Additional Information

This example illustrates the use of the RESMODS macro instruction to include the CONTROL and IORTN load modules in the nucleus library. These modules are members of the SYS1.NEW partitioned data set.

```
RESMODS PDS=SYS1.NEW, MEMBERS=(CONTROL,IORTN)
```



## RPG

The RPG macro instruction specifies the inclusion of the report program generator (RPG) language processor. This macro instruction is optional.

### Processor/Library Generation

- Sufficient space must have been allocated for the RPG processor in SYS1.LINKLIB.

---

Name	Operation	Operand
	RPG	

---

Restriction: The operand field must be left blank.

### Coding Hints and Additional Information

IESRPG is the name and RPG is the alias of the RPG language processor.

Example: The following example illustrates the use of the RPG macro instruction to specify the RPG language processor.

RPG

RPG



# SCHEDULR

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

## Nucleus Generation

- This macro instruction must be specified.
- Only WTOBFRS and REPLY can be changed.
- All other parameters must be respecified exactly as they were during the last complete generation.

## Processor/Library Generation:

- This macro instruction must be specified only if the TSO option is specified.
- All keyword parameters and their subparameters must be specified in exactly the same way as they were during the last complete operating system generation.

## I/O Device Generation

- This macro instruction must be specified.
- The CONSOLE and ALTCONS parameters can be changed. The primary, master, and alternate console addresses may be changed. No other parameters can be changed and must be respecified in exactly the same way as they were during the last complete generation.
- The addresses specified in the STARTR, STARTW, PROCRES, and JOBQRES parameters must be the same as those specified in the last complete generation, but they can be changed at IPL time.

---

Name	Operation	Operand
	SCHEDULR	<u>For TYPE=MFT</u>

```
TYPE=MFT
[ OPTIONS= ( [ LOG ] [ , [ CRJE ] [ , [ RJE ] ]
            [ NOLOG ] ) ]
[ STARTR= (A-address [ , V-serial ] [ , D-dsname ] ) ]
[ STARTW= (A-address [ , V-serial ] [ , D-dsname ] ) ]
[ ACCTRTN= { NOTSUPPLIED
            BASIC
            SMF
            } ]
[ VL MOUNT=AVR ]
[ TAVR= { 800
          200
          556
        } ]
[ DESIGN= { 30K
            44K
          } ]
[ STARTI= { MANUAL
           AUTO
         } ]
[ WTOBFRS=number ]
[ REPLY=number ]
```

SCHEDULR

Name                    Operation                    Operand  
SCHEDULR                    For TYPE=MFT (Continued)  
(Cont.)

[ PROCRES=address ]  
[ JOBQRES=address ]  
[ JOBQFMT=number ]  
[ JOBQLMT=number ]  
[ JOBQTMT=number ]  
[ ESV= { CON  
         NO  
         SMF } ]  
[ EVA= { NO  
         {n<sup>1</sup>, n<sup>2</sup>} } ]  
[ JOBQWTP=number ]  
[ WTLCLASS=classname ]  
[ WTLBFRS=number ]  
[ MINPART=number ]  
[ CONOPTS= { NOMCS  
              { MCS { NOEXIT } }  
              { EXIT } } ]  
If CONOPTS=NOMCS is specified:  
CONSOLE= { address  
          { (I-address, 0-address) } }  
[ ALTCONS= { address  
            { (I-address, 0-address) } } ]  
If CONOPTS=MCS is specified:  
CONSOLE= { address  
          { (I-address, 0-address) } }  
ALTCONS= { address  
          { (I-address, 0-address) } }  
[ ROUTCDE= { 1  
             { (routine code[, routing code]...) } } ]  
[ OLDWTOR= (routing code[, routing code]...) ]  
[ AREA= (nn[, nn... ] ) ]  
[ HARDCPY= ( { SYSLOG  
             address  
             { ALL  
             {, (routing code[, routing code]...)}  
             { NOCMDS  
             {, INCMDS  
             {, STCMDS  
             { CMDS } } ]  
[ OPSTRAN=nn ]  
[ PFK=nn ]

For TYPE=MVT

TYPE=MVT  
[ OPTIONS= ( { LOG } [ , ] [ CRJE  
                  { NOLOG }  
                  [ , ] [ RJE ] [ , ] [ TSO ] ) ]  
[ STARTR= (A-address[, V-serial] [ , D-dsname ] ) ]  
[ STARTW= (A-address[, V-serial] [ , D-dsname ] ) ]  
[ ACCTRTN= { NOTSUPPLIED  
            BASIC  
            SMF } ]



Name	Operation	Operand
	SCHEDULR (Cont.)	For TYPE=MVT (Continued)

```

[ VLMOUNT=AVR ]
[ TAVR= { 800 }
         { 200 }
         { 556 } ]
[ STARTI= { MANUAL }
           { AUTO } ]
[ WTOBFRS=number ]
[ REPLY=number ]
[ PROCRES=address ]
[ JOBQRES=address ]
[ JOBQFMT=number ]
[ JOBQLMT=number ]
[ JOBQTMT=number ]
[ SUBMITQ=maxno ]
[ ESV= { CON }
        { NO }
        { SMF } ]
[ EVA= { NO }
        { (n1, n2) } ]
[ JOBQWTP=number ]
[ WTLCLSS=classname ]
[ WTLBFRS=number ]
[ INITQBF=number ]
[ MINPART=number ]
[ CONOPTS= { NOMCS }
            { (MCS { , NOEXIT } ) }
            { , EXIT } ]
If CONOPTS=NOMCS is specified:
CONSOLE= { address }
          { (I-address, O-address) }
[ ALTCONS= { address }
            { (I-address, O-address) } ]
If CONOPTS=MCS is specified:
CONSOLE= { address }
          { (I-address, O-address) }
ALTCONS= { address }
          { (I-address, O-address) }
[ ROUTCDE= { 1 }
            { (routing code[ , routing code]...) } ]
[ OLDWTOR= (routing code[ , routing code]...) ]
[ AREA= (nn[ , nn... ] ) ]
[ HARDCPY= { { SYSLOG }
             { address }
             { , ALL }
             { , (routing code[ , routing code]...) }
             { , NOCMDS }
             { , INCMDS }
             { , STCMDS }
             { , CMDS } } ]
[ OPSTRAN=nn ]
[ PFK=nn ]

```

SCHEDULR

Name	Operation	Operand
	SCHEDULR (Cont.)	<p>For TYPE=M65MP</p> <p>TYPE=M65MP</p> <p>[ OPTIONS= ( [ { LOG } ] [ , ] [ CRJE ] [ , ] [ RJE ] [ TSO ] ) ]</p> <p>[ NOLOG ]</p> <p>[ STARTR= ( A-address [ , V-serial ] [ , D-dsname ] ) ]</p> <p>[ STARTW= ( A-address [ , V-serial ] [ , D-dsname ] ) ]</p> <p>[ ACCTRTN= { NOTSUPPLIED } ]</p> <p>[ BASIC ]</p> <p>[ SMF ]</p> <p>[ VLMOUNT=AVR ]</p> <p>[ TAVR= { 800 } ]</p> <p>[ 200 ]</p> <p>[ 556 ]</p> <p>[ STARTI= { MANUAL } ]</p> <p>[ AUTO ]</p> <p>[ WTOBFRS=number ]</p> <p>[ REPLY=number ]</p> <p>[ PROCRES=address ]</p> <p>[ JOBQRES=address ]</p> <p>[ JOBQFMT=number ]</p> <p>[ JOBQLMT=number ]</p> <p>[ JOBQTMT=number ]</p> <p>[ ESV= { CON } ]</p> <p>[ NO ]</p> <p>[ SMF ]</p> <p>[ EVA= { NO } ]</p> <p>[ (n1, n2) ]</p> <p>[ JOBQWTP=number ]</p> <p>[ WTLCLSS=classname ]</p> <p>[ WTLBFRS=number ]</p> <p>[ INITQBF=number ]</p> <p>[ MINPART=number ]</p> <p>[ CONOPTS= { NOMCS } ]</p> <p>[ { MCS { , NOEXIT } } ]</p> <p>[ { , EXIT } ]</p> <p>If CONOPTS=NOMCS is specified:</p> <p>CONSOLE= ( { address } )</p> <p>{ I-address, 0-address }</p> <p>{ , address }</p> <p>{ , I-address, 0-address } )</p> <p>[ ALTCONS= ( { address } ) ]</p> <p>{ I-address, 0-address }</p> <p>{ , address }</p> <p>{ , I-address, 0-address } )</p> <p>If CONOPTS=MCS is specified:</p> <p>CONSOLE= { address }</p> <p>{ (I-address, 0-address) }</p> <p>ALTCONS= { address }</p> <p>{ (I-address, 0-address) }</p>

Name	Operation	Operand
	SCHEDULR (Cont.)	For TYPE=M65MP (Continued) <pre> [ROUTCDE={1   (routing code[,routing code]...)}] [OLDWTOR=(routing code[,routing code]...)] [AREA=(nn[,nn...])] [HARDCPY={ {SYSLOG             {address             {,ALL             {, (routing code[,routing code]...)}             {,NOCMDS             {,INCMDS             {,STCMDS             {,CMDS             } } [OPSTRAN=nn] [PFK=nn] </pre>

**AREA=**  
specifies the dimensions of the display areas to be set aside for status displays on the display console screen specified in the CONSOLE= operand.

**nn**  
is a decimal number equal to the number of display screen lines to be in the display area. The minimum specification is four lines; the maximum specification is the number of lines in the message area of the defined display area. Each nn defines one display area of the size indicated. The first nn defines the bottom-most display area on the screen (the bottom lines of the message area). Subsequent nn's define areas stacked above the bottom-most area working toward the top of the screen.

**Default:** If this parameter is not specified when a display console is specified in the CONSOLE= operand, the system defaults one display area for that console. The size of the display area depends on the type of console specified. Display area default sizes for the various types of display consoles are as follows:

<u>Console</u>	<u>Number of Lines</u>
2250	14
2260*	8
Consoles for Models 85, 165, 91, or 195	14

\*There is no display area default when this console has been specified as an output-only console.

Output-only and limited-capability display consoles may have defined display areas only if they have been specified for status displays (SECONSLE macro, USE=SD).

**Note:**

The AREA parameter reserves 36 bytes of storage in the nucleus for each area defined. The areas defined become the default during each IPL. Areas may also be added by the operator after IPL. The storage required for these additional areas (32 bytes per area) is obtained dynamically from the system queue area (SQA). The AREA parameter should be used to define the areas to be used in a normal case, not the possible maximum.

SCHEDULR

ACCTRTN=

specifies whether you are supplying an accounting routine.

NOTSUPPLIED (For all types)

specifies that you do not intend to provide an accounting routine.

BASIC (For all types)

specifies that you are supplying accounting routines.

Notes:

These routines should be placed in SYS1.CI505 before the system generation process is started. (For further information on the accounting routines, refer to the appropriate section in the OS MFT Guide or OS MVT Guide.)

SMF (For all types)

specifies that the system management facilities (SMF) routines are to be included.

Default: If ESV=SMF has been specified, this subparameter becomes the default for ACCTRN.

Notes:

- MFT: BSAM modules (IGC019BA, IGC019BB, IGC019CD) must be resident if SMF data sets are on direct-access devices. When SMF recording is on tape, BSAM modules (IGC019BA, IGC019BB, IGC019CC) must be made resident.
- Required: For MFT or MVT, TIMER=JOBSTEP must be specified in the SUPRVSOR macro instruction if this option is specified.
- User-written routines: These can be added to SMF by means of the exits provided. These routines should be placed in SYS1.CI505 before the system generation process is started. If user-written routines are added to the system for the IEFUJI, IEFUSI, and/or IEFACTRT exits, the sizes of these modules must be added to the size specified in the MINPART keyword parameter. (For further information on the exit routines, refer to the appropriate section in the OS MFT Guide or OS MVT Guide.)
- Data Sets: If this subparameter is specified, both SMF data sets must be allocated space and cataloged in the new system (if they are on direct-access devices).

Defaults:

- If this keyword parameter is omitted and ESV=SMF is not specified, NOTSUPPLIED is assumed.
- If ESV=SMF is specified, ACCTRTN=SMF is assumed.

Notes:

- MFT: TIMER=JOBSTEP must also be specified in the SUPRVSOR macro instruction if the job-step CPU timing feature of MFT is desired.
- Included: Specifying this keyword parameter results in the inclusion of the module that is the access to the SYS1.ACCT data set.

ALTCONS= (For all types)  
specifies that a device is to be used as an alternate console.

address  
is the unit address of an alternate console.

Note:

MCS: If the multiple console support (MCS) option is specified, this device must have 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.

Notes:

- Required: This keyword parameter must be specified if MCS is included in the new system.
- Acceptable Devices: Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. The devices that can be used are listed in Figure 84.
- MCS: A device specified as a part of a composite console cannot be specified in any other way, that is, as a single device or as a part of a composite console with a different companion device. However, the same combination can be repeated more than once.

CONOPTS= (For all types)  
specifies the console options desired.

NOMCS  
specifies that MCS is not desired.

Default: If this keyword parameter is omitted, this subparameter and NOEXIT are assumed.

MCS  
specifies that MCS is desired.

Notes:

- Required: If MCS is specified, the device specified in the ALTCONS parameter of this macro instruction must also be specified in the CONSOLE parameter of a SECONSLE macro instruction.
- Optional: If MCS is specified, then the following subparameters may be used.

EXIT  
specifies that a user exit is desired before the routing code of the write-to-operator (WTO and WTOR) message is used to send it to the appropriate consoles.

NOEXIT  
specifies that no user exit is desired.

Default: If this keyword parameter is omitted, NOMCS and NOEXIT are assumed.

SCHEDULR

CONSOLE= (For all types)  
specifies the primary console, or, if the multiple console support (MCS) option is specified, the master console.

address  
is the unit address. If MCS is specified, the device must have 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.

Notes:

- Acceptable Devices: Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. The devices that can be used are listed in Figure 84.
- MCS: Any device specified as part of a composite console cannot be specified in any other way, that is, as a single device or as a part of a composite console with a different companion device. However, the same combination can be repeated more than once.

DESIGN= (For MFT)  
specifies the design level (in bytes) of the MFT job scheduler.

EVA= (For all types)  
specifies the use of and threshold values for error volume analysis (EVA).

NO  
specifies that EVA will not be used.

n<sup>1</sup>, n<sup>2</sup>  
are integers from 1 to 255.

n<sup>1</sup>  
is the number of temporary read errors that cause an ESV message to be printed on the console.

n<sup>2</sup>  
is the number of temporary write errors that cause an ESV message to be printed on the console.

Default: If this keyword parameter is omitted, EVA is not included in the operating system.

ESV=  
specifies the destination of volume error statistics records.

NO (For all types)  
specifies that the records will not be constructed.

CON (For all types)  
specifies that an abridged version of the records will be constructed and put on the console.

SMF (For all types)  
specifies that the records will be constructed and written on the  
SMF data sets, SYS1.MANX or SYS1.MANY.

Notes:

- Requirements: ACCTRTN=SMF should be specified. If it is not specified, it is assumed.
- MFT and MVT Systems: TIMER=JOBSTEP must be specified in the SUPRVSOR macro instruction.

Default: If this parameter keyword is omitted, ESV=CON is assumed.

HARDCPY= (For all types)  
specifies that a hard-copy log will be used to record operator commands,  
system commands and responses, and write-to-operator (WTO and WTOR)  
messages.

SYSLOG  
specifies that the data that is supposed to go to the hard-copy log  
will now go to the system log.

Notes:

- If this subparameter is specified, it must be changed temporarily at IPL time to the address of an operator console that has output capability. For a general description, see the "Program Options Descriptions" section. For detailed operating instructions, see OS Operator's Reference.
- Use: The data for the hard-copy log is handled in the same way as the data for the system log. (See WTLBFRS parameter description.)

address  
is the unit address of a console with output that is used as the hard-copy log. (See Figure 84 for the devices that can be used.)

Notes:

- Restriction: A graphic device cannot be specified as the hard-copy log.
- Address: The unit address used must be that specified for the device in an IODEVICE macro instruction.
- Required: This device must also be specified either in the CONSOLE parameter of the SCHEDULR macro instruction or in a SECONSLE macro instruction.

SCHEDULR

ALL  
specifies that all write-to-operator (WTO and WTOR) messages are to be put in the hard-copy log.

routing code  
is a number from 1 to 16 that designates the routing code that the hard-copy log is authorized to receive.

Note:

For detailed information about routing codes, see OS Supervisor Services and Macro Instructions.

**NOCMDS**

no operator or system commands or responses are desired. This option is ignored if the hard-copy log is mandatory.

**INCMDS**

the operator and system commands and responses (but not status displays) are to be written on the hard-copy log.

**STCMDS**

the operator and system commands, responses, and status displays (except time-interval updated status displays) are to be written on the hard-copy log.

**CMDS**

the operator and system commands, responses and status displays (static and time-interval updated) are to be written on the hard-copy log.

Default: CMDS is assumed if the parameter is omitted or if NOCMD has been specified in a system for which the hard-copy log is mandatory.

Default: If this keyword parameter is omitted, ALL and NOCMDS is assumed.

Notes:

- Required: This keyword parameter can be used only if CONOPTS=MCS has been specified.
- Graphic Console: When a graphic console is active or if there is more than one active console, a hard-copy log will be required during system initialization or job execution time.
- Coding: The subparameters for HARDCPY are positional and must be coded in the sequence shown in the macro instruction format provided at the beginning of the macro instruction description. 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.

**INITQBF=** (For MVT including M65MP)

specifies the number of buffers in main storage that will hold logical tracks from SYS1.SYSJOBQE.

**number**

is an integer from 0 to 255 that is the number of 1024-byte buffers to be set up in main storage. Use the following formula:

$$\text{INITQBF} = \frac{88+37 \cdot N+L(8+176 \cdot N)}{1024}$$

**N**

is the number of 176-byte records in a logical track of SYS1.SYSJOBQE. This value is specified in the keyword parameter JOBQFMT.

**L**

is the number of logical tracks to be in main storage at any one time.

Note: Round off the value to the next highest multiple of 1024.



**Notes:**

- For further information about specifying the system job queue, see OS Storage Estimates.
- Track Stacking: The method used to handle the data brought in from SYS1.SYSJOBQE is called track stacking. Track stacking permits one or more logical tracks for a particular job to reside temporarily in main storage as an ordered series.
- Change Value: This number can be changed by the operator at IPL time if OPTIONS=COMM is specified in the SUPRVSOR macro instruction.

**JOBQFMT=** (For all types)

specifies the size of each logical track for SYS1.SYSJOBQE.

number

is an integer from 10 to 255 that is the total number of 176-byte records in each logical track.

Default: If this keyword parameter is omitted, a value of 12 is assumed.

**Notes:**

- Change Value: This value can be changed by the operator at IPL time if OPTIONS=COMM is specified in the SUPRVSOR macro instruction. (For detailed information about specifying a value for this parameter, see OS MFT Guide or OS MVT Guide.)
- SMF: If SMF is specified in the ACCTRTN keyword parameter, then the value must be increased by two additional messages per job and per step.

**JOBQLMT=** (For all types)

specifies the number of 176-byte records in SYS1.SYSJOBQE to be reserved for each initiator started.

number

is an integer less than or equal to 9999.

Default: If this keyword parameter is omitted, a value of 60 is assumed.

**Notes:**

- Change Value: The value can be changed at IPL time if OPTIONS=COMM is specified in the SUPRVSOR macro instruction. Refer to OS MFT Guide or OS MVT Guide information about how to specify a value for this parameter.
- Size: A job is terminated if the number of records required to initiate it exceeds the value specified. It is recommended that the value specified for JOBQLMT be a multiple of the value given to JOBQFMT.

**JOBQRES=** (For all types)

specifies the device on which SYS1.SYSJOBQE resides. (This keyword parameter is not used if SYS1.SYSJOBQE was cataloged.)

address

is the address of the device.

Default: If this keyword parameter is omitted and SYS1.SYSJOBQE was not cataloged, the address of the system-residence device is assumed.

SCHEDULR

Notes:

- Value: The address must be the same as that specified for the device in the IODEVICE macro instruction.
- Change Value: This address can be changed at IPL time.

JOBQTMT= (For all types)

specifies the number of 176-byte records in SYS1.SYSJOBQE to be reserved for the termination of jobs that require more records for initiation than those specified in JOBQLMT.

number

is an integer less than or equal to 9999.

Default: If this keyword parameter is omitted, a value of 60 is assumed.

Note:

Change Value: This number can be changed at IPL time if OPTIONS=COMM is specified in the SUPRVSOR macro instruction. Refer to OS MFT Guide or OS MVT Guide for information about how to specify a value for this parameter.

JOBQWTP= (For all types)

specifies the number of 161-byte SYSJOBQE records that the write-to-programmer routine can use for messages during a job.

number

is a number from 0 to 20.

Default: If this keyword parameter is omitted, a value of 2 is assumed.

Notes:

- An additional two records are reserved specifically for system messages. These two records are used after the specified limit is reached during a job. (See OS MFT Guide or OS MVT Guide for information about specifying the value.)
- MFT and MVT: This value can be changed at IPL time if OPTIONS=COMM is specified in the SUPRVSOR macro instruction. (See OS Operator's Reference and OS Messages and Codes for information about changing values at IPL time.)

MINPART= (For all types)

specifies the number of 1024-byte blocks of main storage required for the minimum size region or partition in which to initiate a job.

number

MFT: It must be a decimal number equal to or greater than the value specified in the DESIGN keyword parameter.

MVT including M65MP: It must be a decimal number equal to or greater than 52 + the value of INITQBF.

Defaults:

- MFT: If the number is less than the value in DESIGN, then the value in the DESIGN keyword parameter is used and a warning message is issued. If the DESIGN keyword parameter is omitted, the default value for DESIGN is used.

- MVT including M65MP: If this keyword parameter is omitted and the keyword INITQBF is omitted, then a value of 52 is assumed for MINPART. If this keyword is omitted and the keyword INITQBF is specified, then MINPART is given the value of 52 plus the value of INITQBF.

Notes:

- Value Change: This number can be changed at IPL time.
- SMF: If your own routines for the IEFUJI, IEFUSI, and/or IEFACRT exits are going to be included in the system, the size(s) of these routines must be added to the value specified in this keyword parameter.

OLDWTOR= (For all types)  
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.

routing code  
is a number from 1 to 16.

Default: If this keyword parameter is omitted, the master console receives all WTO and WTOR messages that do not have routing and descriptor codes.

Notes:

- For detailed information about the descriptor and routing codes, see OS Supervisor Services and Macro Instructions.
- Required: This keyword parameter can be used only if CONOPTS=MCS has been specified.
- Specifying a routing code of 11 will not make the messages into WTP messages to the system message class. Default routing codes apply only to consoles.

OPSTRAN= (For all types)  
specifies the transient Display Control Module (DCM) group to which the display (CRT) console, specified in the CONSOLE= parameter, is assigned. OPSTRAN applies only to display consoles.

nn  
is a decimal number from 1 to 32 selected by the user.

Notes:

- Required: Multiple Console Support (MCS).
- If the OPSTRAN=nn parameter is not specified, each Display Control Module (DCM) must be resident in main storage at all times.

If the OPSTRAN=nn parameter is specified, up to four consoles may be assigned to a transient group.

Display Control Modules (DCMs) are control blocks that contain system information required for display (CRT) consoles. Each display console has its own DCM. For further information about DCMs, see OS MFT Supervisor Logic or OS MVT Supervisor Logic.

- The use of transient DCMs can reduce resident storage requirements associated with display consoles. Any type of display console can operate from a transient area, but the master console, its alternate, and other high message consoles perform with greater speed if they are not assigned to a transient area. Transient DCMs work to greatest advantage for consoles with lower than average message activity.

SCHEDULR

OPTIONS=

specifies job scheduler and system log options. These values may be listed in any order.

CRJE (For all types)

specifies inclusion of conversational remote job entry (CRJE).

Notes:

- If this subparameter is specified, BTAM must be specified in the DATAMGT macro instruction.
- For detailed information about CRJE, see OS CRJE System Programmer's Guide. For a general description see the "Program Options Descriptions" section in this publication.

LOG (For all types)

specifies that a system log is desired.

Notes:

- Required: If this subparameter is specified, the data sets SYS1.SYSVLOGX and SYS1.SYSVLOGY must be cataloged in the new system.
- MFT: If this subparameter is specified for an MFT system, BSAM modules (IGC019BA, IGC019BB, IGC019CD) must be resident in the system.

NOLOG (For all types)

specifies that no system log is desired.

Note: This subparameter cannot be specified if HARDCPY=SYSLOG has been specified.

RJE (For all types)

specifies the inclusion of remote job entry.

Note: If RJE is specified, BDAM or BTAM must be specified with the DATAMGT macro instruction. (For further information on this parameter, refer to the publication OS RJE. For a general description, see the "Program Options Descriptions" section in this publication.)

TSO (For MVT including M65MP)

specifies the inclusion of the OS Time-Sharing Option. TYPE=MVT or TYPE=M65MP must be specified.

Default: If this keyword parameter is omitted or NOLOG is not specified, LOG is assumed.

Note: If this keyword parameter is specified, a comma must not precede the first or only subparameter.

PROCRES= (For all types)

specifies the device on which SYS1.PROCLIB resides. (This keyword parameter is not used if SYS1.PROCLIB was cataloged.)

address

is the address of the device.

Default: If this keyword parameter is omitted and SYS1.PROCLIB was not cataloged, the address of the system-residence device is assumed.

Notes:

- Value: The address must be the same as that specified for the device in the IODEVICE macro instruction.
- Change Value: This address can be changed at IPL time if OPTIONS=COMM has been specified in the SUPRVSOR macro instruction.

PFK= (For all types)

specifies that the console is to have programmed-function-keyboard (PFK) command entry and/or light pen command entry.

nn

is a decimal number from 1 to 12 indicating the number of PFK keys or key numbers to be associated with operator commands.

Notes:

- Required: Multiple Console Support (MCS).
- PFK command entry applies only to display (CRT) consoles. The specified console must have either a PFK or, light pen attached. The number indicated by nn specifies:

the number of PFK keys that the operator can associate with commands after IPL, or

the number of light-pen-detectable numerical indicators in the PFK line of the screen that the operator can associate with commands after IPL.

If the specified console has both a PFK and a light pen, both methods of command entry will be made available when the PFK macro is specified. For further information about PFK and light-pen command entry, refer to OS Operator's Guide for Display Consoles.

REPLY= (For all types)

number

is the number of reply queue elements to be used by the WTOR routines. Each reply queue element is 24 bytes.

Defaults:

- If this keyword parameter is omitted, the following values are assumed:

<u>With storage size:</u>	<u>For MFT</u>	<u>For MVT (including M65MP)</u>
G	1	---
GD	1	---
GE	1	---
GF	2	5
GFD	2	5
DH	1	---
H	5	10
HG	5	10
I	10	20
IH	10	20
J or larger	15	20

SCHEDULR

- If the REPLY parameter is omitted and OPTIONS=ONLNTEST is specified in the SUPRVSOR macro instruction and storage size equals G, a value of 1 is assumed.
- If the EMULATOR macro instruction is specified and this keyword parameter is omitted or the value specified is less than 2, then two reply queue elements (RQE) are assigned.

Note:

Emulator: For each additional emulator generated, one additional reply queue element must be specified. This is not done if only one emulator is executing at a time.

ROUTCDE= (For all types)

specifies which routing codes the master console is authorized to receive.

routing code

is a number from 1 to 16.

Default: If this keyword parameter is omitted, only routing codes 1 and 2 are recognized by the master console.

Notes:

- Required: This keyword parameter can only be used if CONOPTS=MCS has been specified.
- Routing codes 1 and 2 are always assigned automatically. (For detailed information about the routing codes, see OS Supervisor Services and Macro Instructions.)

STARTI= (For all types)

specifies whether a START INIT (ALL) command for MFT, or a START INIT command for MVT (including M65MP) is to be executed automatically each time the new operating system is loaded into main storage after IPL.

AUTO

specifies that the command is executed automatically.

MANUAL

specifies that the command is not executed automatically.

STARTR= (For all types)

specifies that a START RDR command is executed automatically each time the new operating system is loaded into main storage after IPL.

A-address

is the unit address of the I/O device to be started.

Note: This address value must be the same as that specified for the device in the IODEVICE macro instruction.

V-serial

is the serial number of the labeled volume associated with the device.

D-dsname

is the data set name associated with the device to be started. The dsname cannot exceed eight characters.

**STARTW= (For all types)**

specifies that a START WTR command is executed automatically each time the new operating system is loaded into main storage after IPL.

**A-address**

is the unit address of the device to be started.

**Note:** This address value must be the same as that specified for the device in the IODEVICE macro instruction.

**V-serial**

is the serial number of the labeled volume associated with the device.

**D-dsname**

is the name of the data set associated with the device to be started. The dsname cannot exceed eight characters.

**SUBMITQ= (for TSO)**

specifies the maximum number of logical tracks to be reserved for TSO foreground-initiated background (FIB) jobs in the system job queue.

**maxno**

is an integer from 0 to 9999.

**Default:** If this keyword parameter is omitted, 50 logical tracks are assumed.

**TAVR= (For all types)**

specifies the standard density for 7-track magnetic tape volumes used with automatic volume recognition (AVR).

**Default:** If this keyword parameter is omitted, 800 BPI is assumed.

**Note:**

This keyword parameter may be specified only if VL MOUNT=AVR has been coded.

**TYPE=**

specifies the type of job scheduler.

**MFT**

specifies the multiprogramming with a fixed number of tasks job scheduler. It must not be specified for TSO systems.

**MVT**

specifies the multiprogramming with a variable number of tasks job scheduler. MVT may be specified for TSO systems.

**M65MP**

specifies the version of MVT that controls multiprocessing and uses two model 65s. M65MP may be specified for TSO systems.

SCHEDULR

**Note:**

**Required:** The value given to this keyword parameter must be the same as that specified for the TYPE keyword parameter in the CTRLPROG macro instruction.

**VL MOUNT= (For all types)**

specifies the volume mounting procedures.

**AVR**

specifies automatic volume recognition.

WTLBFRS= (For all types)

specifies the size of the buffer area that is used as temporary storage for write-to-log (WTL) messages that will be written on SYS1.SYSVLOGX or SYS1.SYSVLOGY.

number

is an integer 0, 4, or greater.

Notes:

- Size: The size of the area is determined by multiplying the number supplied by 148; 148 represents the maximum size of a WTL message. No matter what value is specified, the size of the area is not greater than twice the track size of the device type for which SYS1.SYSVLOGX and SYS1.SYSVLOGY are cataloged. If these data sets reside on different device types, the smaller track size will be used in the calculation.
- If 0 is specified or this keyword parameter is omitted and OPTIONS=NOLOG is specified, all messages are written on the primary console, or the master console if the multiple console support (MCS) option is specified.
- MCS: When MCS is specified in the CONOPTS keyword parameter and SYSLOG is specified in the HARDCPY keyword parameter, this number must be 4 or greater.
- Error: If 0, 1, 2, or 3 is coded and OPTIONS=LOG is specified, 4 will be substituted.
- Maximum: The maximum number of buffers allowed is 255.
- Optimum Number: Although each system has different system log requirements, an optimum number of buffers for best log performance is from 10 to 15 buffers.

Default: If this keyword parameter is omitted and OPTIONS=LOG is specified, a value of 4 is assumed.

Note:

- Use: When half of the buffer area is filled, a check is made to see if it can be written on either data set. If both data sets are unavailable, the contents of the buffer are held until one of the data sets is available. If all the buffers become full, any additional WTL messages are converted to write-to-operator (WTO) messages. Then they are written on the primary console (or the master console if the MCS option is specified) and on the hard-copy log if there is one.

WTLCLSS= (For all types)

specifies the classname to be used as a default for SYSOUT for write-to-log (WTL) messages.

classname

is a letter from A through Z, or a number from 0 through 9. (If A is specified, SYSOUT and WTL messages will be interspersed.)

Default: If this keyword parameter is omitted, L is assumed.

WTOBFRS= (For all types)

specifies the number of buffers to be used by the write-to-operator (WTO) routines.

number

is an integer 0 or greater.



Defaults:

- If MCS is not specified and this keyword parameter is omitted, the following values are assumed.

<u>With storage size:</u>	<u>For MFT</u>	<u>For MVT Including M65MP</u>
G	1	---
GE	1	---
GF	5	10
GFD	5	10
H	10	20
HG	10	20
I	20	40
IH	20	40
J or larger	30	40

- If CONOPTS=MCS is specified and the value is less than two buffers per console, then the specified value is ignored and two buffers per console are assigned. If this keyword parameter is omitted, two buffers are assumed for each operator's console in the system.
- If the EMULATOR macro instruction is specified and this keyword parameter is omitted or the value specified is less than 2, then two buffers are assigned.
- If the WTOBFRS parameter is omitted and OPTION=ONLNTEST is specified in the SUPRVSOR macro instruction and storage size equals G, a value of 3 is assumed.

Notes:

- Size: Each buffer will be 144-bytes long in a system that does not have MCS; each buffer will be 148-bytes long if the system has MCS.
- This number should be at least equal to: three times the number of initiators expected to be active at any one time plus the maximum number of jobs that may be in the system at any one time.
- MFT: If ONLNTEST is specified in the OPTIONS keyword parameter of the SUPRVSOR macro instruction, a minimum of three buffers is required.
- For each additional emulator generated, you must specify one additional buffer.
- You must specify additional buffers when display areas are defined for display console screens and/or when the MONITOR ACTIVE facility (applies only to display consoles) is used. You may use the following algorithm to calculate the number of buffers required:

$$(28 \times X) - Y = \text{number of required buffers}$$

2

where:

X = total number of defined display areas

Y = total number of lines in all display areas planned for use by nondynamic status displays

SCHEDULR

Coding Hints and Additional Information

The keyword parameters that are concerned with and affect SYS1.SYSJOBQE are:

- INITQB -- amount of storage to hold logical tracks
- JOBQFMT -- size of logical track
- JOBQLMT -- number of records reserved for each initiator
- JOBQRES -- device for SYSJOBQE
- JOBQTMT -- number of records reserved for termination
- SUBMITQ -- (TSO) number of tracks reserved for foreground initiated background (FIB) jobs

The chart below shows the relationships between parameters in this macro instruction.

<u>Parameter</u>	<u>Comment</u>	<u>Parameter</u>
HARDCPY=SYSLOG	Cannot specify or allow to default if:	OPTIONS=NOLOG
MINPART	If omitted value depends on:	DESIGN
SUBMITQ=maxno	Can be specified only if:	OPTIONS=TSO
WTLBFRS	Value 4 or greater if:	CONOPTS=MCS and HARDCOPY=SYSLOG

Subparameter	Unit address of:	Notes
address	1052 Models 5, 6, or 7	Only Model 7 can be used with MCS
	2150	
	2250 Model 1 with alphameric keyboard, character generator, and 4K or 8K buffer	Can be used only with MCS; light pen is optional
	Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and the IODEVICE macro instruction -- NUMSECT parameter.)	Can be used only with MCS; light pen is optional
	2260 Model (local attachment) with alphameric keyboard with a 2848 Model 3 that has the line addressing feature	Can be used only with MCS
	2740 Model 1 with record checking feature or with optional interrupt feature. Only one per communication line	Can be used only with MCS

Figure 84 (Part 1 of 4). Devices for Which Unit Addresses can be Specified in CONSOLE and ALTCONS Parameters of SCHEDULR

Subparameter	Unit address of:	Notes
	3066 System Console	Can be used only with MCS for a S/370 Model 165
	3210 Console Printer Keyboard	Can be used only for S/370 Models 135, 145, or 155
	3215 Console Printer Keyboard	Can be used only for S/370 Models 135, 145, or 155
	3277 Model 2 with keyboard	Can be used only with MCS; selector pen is optional
	3505 Reader	
	3525 Card Punch	Must have reader feature
	5450 Console for Model 85	Can be used only with MCS
I-address	1052 Models 5, 6, or 7	Only Model 7 can be used with MCS
	1442 Model N1	
	2150	
	2250 Model 1 with alphameric keyboard, character generator, and 4K or 8K buffer	Can be used only with MCS; light pen is optional
	Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and the IODEVICE macro instruction -- NUMSECT parameter.)	Can be used only with MCS; light pen is optional
	2260 Model (local attachment) with alphameric keyboard with a 2848 Model 3 that has the line addressing feature.	Can be used only with MCS
	2501	
	2520	
	2540 Reader	

SCHEDULR

Figure 84 (Part 2 of 4). Devices for Which Unit Addresses can be Specified in CONSOLE and ALTCONS Parameters of SCHEDULR

Subparameter	Unit Address of:	Notes
	2740 Model 1 with record checking feature or with optional interrupt feature. Only one per communication line	Can be used only with MCS
	3066 System Console	Can be used only with MCS for a S/370 Model 165
	3210 Console Printer Keyboard	Can be used only for S/370 Models 135, 145, or 155
	3215 Console Printer Keyboard	Can be used only for S/370 Models 135, 145 or 155
	3277 Model 2 with keyboard	Can be used only with MCS; selector pen is optional
	3505 Card Reader	
	3525 Card Punch	Must have reader feature
	5450 Console for Model 85	Can be used only with MCS
O-address or address (in HARDCPY)	1052 Models 5, 6, or 7	Only Model 7 can be used with MCS
	1403	
	1443	
	2150	
	2250 Model 1 with alphameric keyboard, character generator and 4K or 8K buffer	Can be used only with MCS; light pen is optional
	Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and the IODEVICE macro instruction -- NUMSECT parameter.)	Can be used only with MCS; light pen is optional

Figure 84 (Part 3 of 4). Devices for Which Unit Addresses can be Specified in CONSOLE and ALTCONS Parameters of SCHEDULEP

Subparameter	Unit Address of:	Notes
2260	Model (local attachment) with alphameric keyboard with a 2848 Model 3 that has the line addressing feature	Can be used only with MCS
2740	Model 1 with record checking feature or with optional interrupt feature. Only one per communication line	Can be used only with MCS
3066		Can be used only with MCS for a S/370 Model 165. Cannot be used in the HARDCPY parameter
3210		Can be used only for S/370 Models 135, 145, or 155
3211		
3215		Can be used only for S/370 Models 135, 145, or 155
3277	Model 2	Can be used only with MCS; cannot be used with HARDCPY parameter
5450	Console for Model 85	Can be used only with MCS

Figure 84 (Part 4 of 4). Devices for Which Unit Addresses can be Specified in CONSOLE and ALTCONS Parameters of SCHEDULR

SCHEDULR

#### SCHEDULR

#### Examples:

The following example illustrates the use of the SCHEDULR macro instruction to specify the MVT job scheduler required for multiprogramming with a variable number of tasks (MVT). The address of the primary console is 01A. A composite console is used as an alternative console; its input address is C0C and its output address is 00D. An accounting routine is not supplied. START RDR and START WTR commands are executed automatically after IPL. The devices to be started are assigned addresses 00E and 00F. WTO routines use 75 buffers and WTOR routines use 25 reply-queue elements.

A START INIT command is executed automatically after IPL. The classname for WTL messages is L. A maximum of 20 buffers are used for WTL messages. SYS1.SYSJOBQE and SYS1.PROCLIB are located on the system-residence device. The format of SYS1.SYSJOBQE will be twenty 176-byte records for each logical track. Eighty 176-byte records (4 logical tracks) are reserved for each initiator started, and eighty 176-byte records are reserved for the termination of jobs that require more than 80 records for initiation. Fifteen 1024-byte buffers are requested for SYS1.SYSJOBQE. Sixty-seven 1024-byte blocks are required to process a job (fifty-two plus fifteen INITQBF buffers). The master console is a display console, the screen has one display area, 10 lines in length. The console has a program function keyboard (PFK) and twelve PFK keys are designated for later association with operator commands.

```
SCHEDULR TYPE=MVT,CONSOLE=01A,ALTCONS=(I-00C,0-00D),
          STARTR=A-00E,STARTW=A-00F,WTOBFRS=75,REPLY=25,
          STARTI=AUTO,WTLBFRS=20,JOBQFMT=20,JOBQLMT=80,
          JOBQTMT=80,INITQBF=15,MINPART=67,AREA=10,PFK=12
```

The following example illustrates the use of the SCHEDULR macro instruction to specify the MVT job scheduler required for multiprogramming with a variable number of tasks (MVT). The multiple console support option is specified in the CONOPTS parameter and the system log is specified in the OPTIONS parameter. The WTOBFRS parameter specifies that 80 buffers are used by the WTO routines, and the REPLY parameter specifies that 18 reply-queue elements are used by the WTOR routines.

The master console is a composite console whose input address is 00C and whose output address is 01B. The alternate console is a device with input and output capabilities whose address is 00A. The ROUTCDE parameter indicates that the master console or its alternate accepts routing codes with number 1, 2, 4, 5 or 9. The OLDWTOR parameter indicates that numbers 1, 2, 3, 6 and 9 are assigned to all WTO and WTOR messages that do not already have routing and descriptor codes. The HARDCPY parameter indicates that a device with address 00E is used to record operator commands. The omission of the second positional parameter indicates that no write-to-operator messages are put on the hard-copy log. The third positional parameter in the HARDCPY parameter specifies that operator and system commands and responses are written on the hard-copy log.

```
SCHEDULR TYPE=MVT,OPTIONS=LOG,WTOBFRS=80,
          REPLY=18,CONOPTS=MCS,
          CONSOLE=(I-00C,0-01B),ALTCONS=00A,
          ROUTCDE=(1,2,4,5,9),OLDWTOR=(1,2,3,6,9),
          HARDCPY=(00E,,CMDS)
```

# SECMODS

The SECMODS macro instruction specifies the type of system error recovery desired for each secondary CPU. This macro instruction is optional.

## Basic Requirements

- OPTIONS=COMM must be specified in the SUPRVSOR macro instruction.
- This macro instruction cannot be specified if TYPE=M65MP is specified in the CTRLPROG macro instruction, or if 85 or 195 is specified in the CENPROCS macro instruction.
- S/360 Models 85, 91, or 195 can be used only as a primary CPU.

## Nucleus Generation

- This macro instruction is used so that the new nucleus supports the same secondary CPUs as those supported by the primary nucleus.
- The same parameters that were used during the last complete generation must be respecified without any changes.

## I/O Device Generation

- This macro instruction must be specified if it was specified during the last complete generation.
- The same keyword parameters and their subparameters that were used during the last complete generation must be respecified without any changes.
- If this macro instruction was not specified during the last complete generation, it cannot be specified.

---

Name	Operation	Operand
	SECMODS	operand[,operand]...

---

SECMODS

Each operand must be written in the following format:

model[ ([ SIZE-storage ],[ SER-option] ) ]

Uppercase letters, parentheses, and hyphens must be written exactly as shown.

model

specifies the model of the secondary CPU as one of the following: 40, 50, 65, or 75 (for these models the primary CPU is S/360); ALL, 135, 145, 155, or 165 (for these models, the primary CPU is S/370).

Notes:

- A maximum of five secondary CPUs can be specified.
- A maximum of three secondary CPUs can be specified with a S/360 primary CPU.
- ALL is specified alone to designate all S/360 CPUs other than the one specified in the CENPROCS macro. It can only be used with a S/370 primary CPU.

storage

specifies the size of main storage as one of the following values:

<u>Value</u>	<u>Storage Size</u>
G	128K
GF	192K
H	256K
HG	384K
I	512K
IH	768K
J	1024K
K	2048K
L	4096K

Notes:

- The configuration must be able to execute in the storage size specified in this parameter.
- Storage size 192K (GF) can be specified only for the S/360 Model 40.
- Storage size 384K (HG) can be specified only for the S/360 Model 50 and S/370 Model 155.

option

specifies the system error-recovery option as either SER0 or SER1.

Default: If this parameter is omitted, the value shown in Figure 85 is assumed for the secondary CPU model and storage size.

Coding Hints and Additional Information

The following facts may be helpful.

- A secondary CPU is one other than the primary CPU specified in the CENPROCS macro instruction on which the new system may run.
- The system error-recovery options for the primary CPU are specified with the SER parameter of the SUPRVSOR macro instruction.



Storage	S/360 Models					S/370 Models		
	40	50	65	75	135	145	155	165
G	SER0	SER0				MCH		
GF	SER1				MCH			
H	SER1	SER1	SER1	SER1		MCH	MCH	MCH
HG		SER1					MCH	
I		SER1	SER1	SER1		MCH	MCH	MCH
IH		SER1	SER1	SER1			MCH	MCH
J			SER1	SER1			MCH	MCH
K							MCH	MCH
L							MCH	MCH

Figure 85. SER Default Values for Secondary CPUs

Example: This example illustrates the use of the SECMODS macro instruction to provide SER support for two secondary CPUs. It is assumed that a Model G40 was specified with the CENPROCS macro instruction. The first operand specifies SER1 for Model H50 (SER1 is assumed). The second operand specifies SER0 for Model H40.

```
SECMODS 50(SIZE-H),40(SIZE-H,SER-SER0)
```



## SECONSLE

The SECONSLE macro instruction specifies a secondary console when the multiple console option (MCS) is specified in the CONOPTS keyword parameter of the SCHEDULR macro instruction for the new system. This macro instruction is required when MCS is specified.

### Basic Requirements

- The CONSOLE keyword parameter of this macro instruction must be used to specify the alternate console for the master console that is specified in the ALTCONS keyword parameter of the SCHEDULR macro instruction.
- A maximum of 31 secondary consoles can be specified. If more are coded, the system generation process is terminated.
- Each device specified in an ALTCONS keyword parameter must also be specified in a CONSOLE keyword parameter of a SECONSLE macro instruction or the console parameter of the SCHEDULR macro instruction.

### Nucleus Generation

- This macro instruction must be specified if it was specified in the last complete operating system generation. The same keyword parameters and subparameters must be coded.
- If this macro instruction was not used in the last complete operating system generation, it cannot be specified.

### I/O Device Generation

- This macro instruction must be specified if it was specified in the last complete operating system generation.
- This macro instruction can be used to add, delete, or change the secondary console specifications.
- If this macro instruction was not used in the last complete generation, it cannot be specified.

SECONSLE

Name	Operation	Operand
	SECONSLE	CONSOLE= { address 0-address [,USE= SD] [MS] (I-address,0-address) } [ALTCONS= { address 0-address (I-address,0-address) }] [OPSTRAN=nn] [PFK=nn] [ROUTCDE= { ALL ( routing code[, routing code]... ) }] [VALDCMD= (command code[, command code]... )] [AREA= (nn[, nn... ] ) ]

**AREA=**

specifies the dimensions of the display areas to be set aside for status displays on the display console screen specified in the **CONSOLE=** operand.

**nn**

is a decimal number equal to the number of display screen lines to be in the display area. The minimum specification is four lines; the maximum specification is the number of lines in the message area of the defined display area. Each **nn** defines one display area of the size indicated. The first **nn** defines the bottom-most display area on the screen (the bottom lines of the message area). Subsequent **nn**'s define areas stacked above the bottom-most area working toward the top of the screen.

**Default:** If this parameter is not specified when a display console is specified in the **CONSOLE=** operand, the system defaults one display area for that console. The size of the display area depends on the type of console specified. Display area default sizes for the various types of display consoles are as follows:

<u>Console</u>	<u>Number of Lines</u>
2250	14
2260*	8
Consoles for Models 85, 165, 91, or 195	14

\*There is no display area default when this console has been specified as an output-only console.

Output-only and limited-capability display consoles may have defined display areas only if they have been specified for status displays (**SECONSOLE** macro, **USE=SD**).

**Note:** The **AREA** parameter reserves 36 bytes of storage space in the nucleus for each area defined. The areas defined here become the default during each IPL. Areas may also be added by the operator after IPL. Storage required for these additional areas (32 bytes per area) is obtained dynamically from the system queue area (**SQA**). The **AREA** parameter should be used to define the areas to be used in a normal case, not the possible maximum.

**ALTCONS=**

specifies the address or addresses of the alternate console.

**address**

is the unit address of an alternate console device with input and output capabilities.

**O-address (only)**

is the unit address of a console with output capability.

**I-address**

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

**O-address**

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

**Default:** If this keyword parameter is omitted, the master console specified in the **CONSOLE** keyword parameter of the **SCHEDULR** macro instruction is assigned as the alternate console.

Notes:

- Required: The device specified must also be specified in the CONSOLE keyword parameter of either the SCHEDULR or a SECONSLE macro instruction.
- Address: Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. For a list of the devices that can be used, see Figure 86.
- Restriction: A console with only output capability cannot be specified as the alternate console for a console with input and output capabilities.
- Composite Console: A device specified as a part of a composite console cannot be specified in any other way, that is, as a single device or as a part of a composite console with a different companion device. However, the same combination can be specified more than once.

CONSOLE=

specifies the address or addresses of the secondary console.

address

is the unit address of the secondary console device with input and output capabilities.

O-address (alone)

is the unit address of a console with output capability.

USE=

defines the intended use of a 2260 or 3277 display (CRT) console.

SD

indicates that the specified console is to be used as an output-only console for status displays.

MS

indicates that the specified console is to be used as an output-only console to display operator messages.

Default: If "O-address" is specified for a 2260 or 3277 display console, and the USE= parameter is omitted, the USE=SD parameter is assumed and the console is designated for status displays. If "address" is specified for a display console, and USE= is omitted, the console is designated as an input/output (full capability) console.

I-address

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

O-address

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

SECONSLE

Notes:

- Required: Each unit address used must be the same as that specified for the device in an IODEVICE macro instruction. For a list of the devices that can be used see Figure 86.
- Composite Console: A device specified as a part of a composite console cannot be specified in any other way, that is, as a single device or as a part of a composite console with a different companion device. However, the same combination can be specified more than once.

- Output-only Display Consoles: A 2260 or 3277 display console without input capability should be designated as "O-address," and SD or MS according to its purpose. A 2260 or 3277 display console with input and output capability can be designated by either "address" or "O-address" and SD or MS; however, specifying "O-address" disables the console's input capability. If "address" and USE=SD or MS are specified for a 2260 or 3277 console with input and output capability, the console is initially placed in output-only mode. After IPL, the operator can alter the console for either input/output or output-only use. He can also respecify SD or MS. All display consoles, other than the 2260 or 3277, should be specified by "address," and the USE= parameter should be omitted.

OPSTRAN= (For all types)

specifies the transient display control module (OCM) group to which the display (CRT) console, specified in the CONSOLE= parameter, is assigned. OPSTRAN applies only to display consoles.

nn

is a decimal number from 1 to 32 selected by the user.

Notes:

- Required: Multiple Console Support (MCS).
- If the OPSTRAN=nn parameter is not specified, each Display Control Module (DCM) must be resident in main storage at all times. If the OPSTRAN=nn is specified, up to four consoles may be assigned to a transient group.

Display Control Modules (DCMs) are control blocks that contain system information required for display (CRT) consoles. Each display console has its own DCM. For further information about DCMs, see OS MFT Supervisor Logic or OS MVT Supervisor Logic.

- The use of transient DCMs can reduce resident storage requirements associated with display consoles. Any type of display console can operate from a transient area, but a master console, its alternate, and other high message consoles perform with greater speed if they are not assigned to a transient area. Transient DCMs work to greatest advantage for consoles with lower than average message activity.

PFK=

specifies that the console is to have programmed-function-keyboard (PFK) command entry and/or light pen command entry.

nn

is a decimal number from 1 to 12 indicating the number of PFK keys or key numbers to be associated with operator commands.

Notes:

- Required: Multiple Console Support (MCS).
- PFK command entry applies only to display (CRT) consoles. The specified console must have either a PFK or a light pen attached. The number indicated by nn specifies:

the number of PFK keys that the operator can associate with commands after IPL, or

the number of light-pen-detectable numerical indicators in the PFK line of the screen that the operator can associate with commands after IPL.

If the specified console has both a PFK and a light pen, both methods of command entry will be made available when the PFK macro is specified. For further information about PFK and light pen command entry, refer to OS Operator's Guide for Display Consoles.

**ROUTCDE=**  
specifies which routing codes the console receives.

**ALL**  
specifies that all routing codes are desired.

routing code  
a number from 1 to 16 that designates which routing code is recognized and accepted by this console.

**Default:** If this keyword parameter is omitted, no routing codes are assigned to this console.

**Note:**

**Information:** For detailed information about the routing codes, see OS Supervisor Services and Macro Instructions.

**VALDCMD=**  
specifies which commands can be entered from this console.

command code  
is a number from 1 to 3. (0 is always assumed.)

**Notes:**

- **Value:** One or more numbers may be specified that indicate which command groups can be entered from this console. (For detail information about command code groups, see OS Operator's Reference.)
- **Restriction:** This parameter cannot be specified if the console only has output capability.
- **Use:** The commands DISPLAY, LOG, REPLY, SHOW, BRDCST, and MSG can be used at all consoles with input capability.

Although DISPLAY and MONITOR can be used at all consoles with input capability to terminate a display of STATUS, JOBNAMEs, SPACE, DSNAME, or SESS with the STOP command, the command issuing the DISPLAY or MONITOR command must also have Group 1 command authority.

Coding Hints and Additional Information

The following example illustrates the use of the SECONSLE macro instruction to specify a secondary console whose address is 01F and whose alternate console is a composite console with addresses 00C and 00D. The secondary console can receive all routing codes as specified by the ROUTCDE parameter and can enter commands 1, 2, and 3 as specified by the VALDCMD parameter. The secondary console is a display console assigned to transient group number 05. The console includes a PFK. Support for command entry for 10 of its keys is requested by means of the PFK= operand.

```
SECONSLE      CONSOLE=01F,ALTCONS=(I-00C,O-00E),  
              ROUTCDE=ALL,VALDCMD=(1,2,3),OPSTRAN=05,PFK=10
```

The following example illustrates the use of the SECONSLE macro instruction to specify a secondary console with output-only capability. The address of the console is 00E, and the address of the alternate console is 00B (the alternate console also has output-only capability). The secondary console is a display console. It is used to display system status displays, as specified by the USE= parameter. The console's screen is divided into two display areas -the bottom-most area is seven lines long, and the top area is four lines long. These display area specifications are made by means of the AREA= parameter.

SECONSLE

The console is assigned to transient group number 05, as specified by the OPSTRAN parameter.

SECONSLE            CONSOLE=0-00E,USE=SD,ALTCONS=0-00B,AREA=(7,4),  
OPSTRAN=05

Subparameter	Unit	Address of:	Maximum Number	Notes
address	1052	Model 7	31	
	2250	Model 1 with alphameric keyboard, character generator and 4K or 8K buffer	31	Light pen is optional
		Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and IODEVICE macro instruction -- NUMSECT parameter.)		Light pen is optional
	2260	Model 1 Local attachment with alphameric keyboard with the 2848 Model 3 with the line-addressing feature	31	
	2740	Model 1 with record-checking feature or with optional-interrupt feature	31	Only one per communications line
	3066			Can only be used with MCS for a S/370 Model 165
	3210			Can only be used for S/370 Models 135, 145, and 155
	3215			Can only be used for S/370 Models 135, 145, and 155
	3277	Model 2 with keyboard	31	Selector pen is optional
	5450	Console for Model 85	11	
O-address (only)	1403		31	Maximum of 30 if all are O-address only consoles
	1443		31	
	2260	Model 1 Local attachment on a 2848 Model 3 with line-addressing feature (alphameric keyboard is not required)	31	
	3211			
	3277	Model 1	31	
	3277	Model 2	31	

Figure 86 (Part 1 of 3). Devices for Which Unit Addresses can be Specified in CONSOLE and ALTCONS Parameters of SECONSLE



Subparameter	Unit	Address of:	Maximum Number	Notes
	3284	Models 1 and 2	31	
	3286	Models 1 and 2	31	
I-address (composite)	1052	Model 7	31	
	1442	Model N1	31	
	2250	Model 1 with alphameric keyboard, character generator, and 4K or 8K buffer	31	Light pen is optional
		Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and IODEVICE macro instruction -- NUMSECT parameter.)		Light pen is optional
	2260	Model 1 Local attachment with alphameric keyboard with the 2848 Model 3 with the line-addressing feature	31	
	2501		31	
	2520		31	
	2540	Reader	31	
	2740	Model 1 with record-checking feature or with optional-interrupt feature	31	Only one per communications line
	3066			Can only be used with MCS for a S/370 Model 165
	3210			Can only be used for S/370 Models 135, 145, and 155
	3215			Can only be used for S/370 Models 135, 145, and 155
	3277	Model 2 with keyboard	31	Selector pen is optional
	3505	Card Reader	31	
	3525	Card Punch	31	Must have reader feature
	5450	Console for Model 85	14	
O-address (composite)	1052	Model 7	31	
	1403		31	Maximum of 30 if all are O-address only consoles
	1443		31	

SECONSL

Figure 86 (Part 2 of 3). Devices for Which Unit Addresses Can Be Specified in CONSOLE and ALTCONS Parameters of SECONSL

Subparameter	Unit	Address of:	Maximum Number	Notes
	3211			
	2250	Model 1 with alphameric keyboard, character generator, and 4K or 8K buffer	31	Light pen is optional
		Model 3 with alphameric keyboard (Requires the GRAPHICS macro instruction and IODEVICE macro instruction -- NUMSECT parameter.)		Light pen is optional
	2260	Model 1 Local attachment with alphameric keyboard with the 2848 Model 3 with the line-addressing feature	31	
	2740	Model 1 with record-checking feature or with optional-interrupt feature	31	Only one per communications line
	3066			Can only be used with MCS for a S/370 Model 165
	3210			Can only be used for S/370 Models 135, 145, and 155
	3215			Can only be used for S/370 Models 135, 145, and 155
	3277	Model 2	31	
	5450	Console for Model 85	1 <sup>1</sup>	

<sup>1</sup> Subtract 1 from maximum number if this device type was specified as a master console.

Figure 86 (Part 3 of 3). Devices for Which Unit Addresses Can Be Specified in CONSOLE and ALTCONS Parameters of SECONSLE

# SORTLIB

The SORTLIB macro instruction specifies the inclusion of the sort/merge subroutine library (SYS1.SORTLIB) in the new operating system. This macro instruction is optional.

## Basic Requirements

- SYS1.SORTLIB must be allocated space in the new system. It may be cataloged in the new system during preparation for system generation.
- The SORTMERG macro instruction must be specified.

## Processor/Library Generation

The following condition must exist to replace an existing SYS1.SORTLIB.

- The existing SYS1.SORTLIB must be scratched and uncataloged before the new SYS1.SORTLIB is initialized during preparation for this generation. In order to regain the space that was occupied by the SYS1.SORTLIB that was scratched, use the IEBCOPY utility program.

---

Name	Operation	Operand
	SORTLIB	[ UNIT=name ] [ VOLNO=serial ]

---

**Default:** If the keyword parameters are omitted, the sort subroutine library is placed on the new system-residence volume.

### **UNIT=**

specifies a direct-access device in the generating system that will contain the volume on which the sort subroutine library will be generated.

name

is the device type.

### **Note:**

**Required:** If this keyword parameter is specified, VOLNO must be coded also.

### **VOLNO=**

specifies the volume that is to contain the sort subroutine library.

serial

is the serial number.

### **Notes:**

- **Required:** If this keyword parameter is specified, UNIT also must be coded.
- **Value:** The value specified must be the same as the value given to the **VOLUME** keyword of the DD statement used to allocate space for SYS1.SORTLIB during preparation for system generation.

SORTLIB

## Coding Hints and Addition

This example illustrates the use of the SORTLIB macro instruction to specify the inclusion of the sort subroutine library in the new operating system. The unit name is 2311. The volume serial number is 654321.

```
SORTLIB UNIT=2311,VOLNO=654321
```

# SORTMERG

The SORTMERG macro instruction specifies the inclusion of all or selected sorting and/or merging functions in the new operating system. This macro instruction is optional.

## Basic Requirements

- The SORTLIB macro instruction must be specified.
- If user-written exit routines that require link-editing at execution time are going to be included, the EDITOR macro instruction must be specified.
- Any function that has not been specified in this macro instruction must not be coded in the sort/merge control cards at execution time. For example, if the sorting of only fixed-length records is specified during system generation, the sorting of variable-length records must not be requested at sort/merge execution time. The sort/merge job would be terminated because the programs for sorting variable-length records were not in the system.

## Processor/Library Generation

- The SORTLIB macro instruction must be specified.
- If user-written exit routines that require link-editing at execution time are going to be included, a linkage editor must be in the system. If the system already has a linkage editor, the EDITOR macro instruction does not have to be specified.
- There must be enough space in SYS1.LINKLIB for the new sort/merge processor.

If this macro instruction is being used to replace an existing sort/merge processor, then the following conditions must also exist:

- Members IERRCB, IERRCM, IERRCZ, and IERRC000 (alias SORT) must be scratched from SYS1.LINKLIB. The space that was occupied by these members can be recovered by using the IEBCOPY utility program.
- The existing SYS1.SORTLIB must be scratched and uncataloged before initializing the new SYS1.SORTLIB unless the INDEX keyword parameter is specified in the GENERATE macro instruction and the new SORTLIB is defined with a user-supplied qualifier.

SORTMERG

Name	Operation	Operand
	SORTMERG	<pre> [ SIZE=size ] [ SORTOPT=FULLIB ] [ CNTLFLD= ([ SINGLE ][ , MULTIPLE ] ) ] [ RECTYPE= ([ VAR ][ , FIXED ][ , LONG ] ) ] [ SORTDEV= (device[ , device ]...) ] [ MERGE=MERGONLY ] [ MESSAGE= ( { PRINTER } { , ALL } )             { CONSOLE } { , CRITICAL } ) ] [ SORTOPT=MODPRGM ] </pre>

CNTLFLD=  
specifies the control field requirements for sorting and/or merging.

SINGLE  
specifies single control fields.

MULTIPLE  
specifies multiple control fields.

Note:

Required: One or both values must be coded.

MERGE=  
specifies which routines of the sort/merge processor can be executed independently.

MERGOONLY  
specifies the merge routines only.

Note:

If this keyword parameter is omitted and FULLIB is not specified for the SORTOPT parameter, then the message routines cannot be executed independently.

MESSAGE=  
specifies the I/O device on which sort/merge messages are to be printed and the type of messages to be produced.

PRINTER  
specifies that the messages are to be printed on a printer.

CONSOLE  
specifies that the messages are to be printed on a console.

ALL  
specifies that all the messages are to be printed.

CRITICAL  
specifies that only serious diagnostic messages are to be printed.

Default: If this parameter is omitted, no messages are printed during a sorting or merging operation.

Notes:

- Required: The values included in braces are mutually exclusive.
- Execution Time: This parameter can be changed at execution time. (See OS Sort/Merge for more information.)

RECTYPE=  
specifies the type and length of records to be sorted or merged.

VAR  
specifies variable-length records.

FIXED  
specifies fixed-length records.

LONG  
specifies records longer than 256 bytes.

Notes:

- VAR or FIXED: At least one of these must be specified. You can specify both.
- Spanned Records: If spanned records (VRE) are going to be sorted or merged, then VAR must be specified.

SIZE=

SIZE

specifies as a positive decimal integer the amount of main storage to be used by sort/merge. If this parameter is omitted a value of 12,000 is assumed. The value assigned may be overridden at execution time by the CORE= parameter passed to the program.

Default: If this keyword parameter is omitted, 12,000 is assumed.

Notes:

- Use: The amount specified is used for sorting only and does not include the space required for the data management functions.
- Execution Time: The amount can be changed at execution time. For further information, see OS Sort/Merge.

The sort/merge program can operate in 12,000 bytes for very short records, but a larger value is recommended for efficient execution. In any case where records over 50 bytes are to be sorted you should add at least whichever is the greater of

Input           BLKSIZE + 4 x LRECL,  
and Output       BLKSIZE + M x LRECL,

where M=(number of intermediate work areas to be used +1) /2 for tape sorts and M=number of intermediate work areas to be used for disk sorts.

A large storage size allows sort/merge to use more buffers and better sorting techniques, and to sort more records at a time in storage, thus greatly reducing the time taken for sorting. (For further information refer to OS Sort/Merge.)

The maximum value that can be specified for size is the difference between the total amount of main storage available and the amount required for data management routines.

- Formula: Use the following formula to determine the value of SIZE:

$$SIZE = A - 24N - Y - S - C$$

A is the total amount of main storage available for execution. The maximum amount is the number of bytes specified in the CENPROCS macro instruction minus the number of bytes required for the nucleus minus, if sort/merge is called by another program, the number of bytes occupied by other programs.

N is the maximum number of DD statements to be used in any of the sort/merge programs.

Y is a constant value of 1500 if any messages are to be written on SYSOUT. Otherwise, the value of Y is 0.

**SORTMERG**

- S is a constant with a value of 3500 if the merge-only function is not to be used. If the merge-only function is to be used, the value of S is 5000 when spanned records are not used, or 5500 when spanned records are used.
- C is the amount of main storage required by checkpoint/restart. To calculate this value, see OS Storage Estimates. If checkpoint/restart is not to be used, the value is 0.

**SORTDEV=**

specifies the device(s) used for sorting or merging.

**device**

is a value that can be 2301, 2305-1 or -2, 2311, 2314, or 2400.

Notes:

- Device: The value 2400 stands for 2401, 2402, 2403, 2404, 2415, 2420, 3410, and 3420.
- Use: One or more values can be coded.

**SORTOPT=**

specifies the options that you can select.

**FULLIB**

specifies that all the sort/merge functions are to be included in the new operating system.

**MODPRGM**

specifies that user-written modification programs will be included at sort/merge execution time.

Notes:

- FULLIB: If this subparameter is coded, then RECTYPE, SORTDEV, CNTLFLD, MERGE, MESSAGE, and SORTOPT=MODPRGM are ignored.
- Assumes: This subparameter assumes CONSOLE and ALL for the MESSAGE parameter.

Coding Hints and Additional Information

The following guidelines can be useful.

- The sort/merge program operates in 15,500 bytes of main storage, 12,000 bytes of which are actually used for sorting. However the amount 12,000 can be increased by specifying more than 12,000 in the SIZE parameter.
- SORTOPT=FULLIB assumes CONSOLE and ALL for the MESSAGE parameter.

Example: The following example illustrates the use of the SORTMERG macro instruction to specify the use of fixed-length records, single control fields, and IBM 2311 Disk Storage Drives. The merge functions are to be included. The maximum amount of main storage to be used for sorting is 12,000 bytes.

SORTMERG RECTYPE=FIXED,CNTLFLD=SINGLE,SORTDEV=2311,MERGE=MERGONLY



# SUPRVSOR

The SUPRVSOR macro instruction specifies the task supervisor options. This macro instruction is optional for MFT. It is required for MVT or M65MP.

## Basic Requirements

For assistance in choosing valid task supervisor options for MFT, MVT, and M65MP, refer to Figure 88 in the "Coding Hints and Additional Information" section at the end of this macro instruction description.

## Nucleus Generation

- The values specified in the following keyword parameters can be changed during this generation.

The operands for WAIT and TRACE can be changed.

For RESIDNT: Only the following keyword subparameters can be added to or deleted from the parameter statement: ACSMETH, RENTCODE, ERP, and TRSVC.

For OPTIONS: Only the following keyword subparameters can be added to or deleted from the parameter statement: TRSVCTBL, PROTECT, VALIDCHK, and COMM.

- All the other keyword parameters and their subparameters must be specified in exactly the same way as they were during the last complete operating system generation.

## Processor/Library Generation

- This macro instruction must be specified if PL1LIB macro instruction is used.
- All the keyword parameters and their subparameters must be specified in exactly the same way as they were during the last complete system generation.

## I/O Device Generation

- This macro instruction must be specified with the same keyword parameters and subparameters that were specified during the last complete operating system generation.
- If this macro instruction was not specified in the last complete operating system generation, it cannot be specified in this generation.

SUPRVSOR

Name	Operation	Operand
	SUPRVSOR	<pre> [ ASCII= { EXCLUDE            INCRS            INCTAN } ] [ RESIDNT=(function[,function]...) ] [ OPTIONS=(option[,option]...) ] [ ALTSYS=ddd ] [ TIMER= { TIME            INTERVAL            JOBSTEP } ] [ TRACE=number ] [ SER= { SER0         SER1         MCH } ] </pre>

**ALTSYS=** (For all control programs)  
specifies the alternate system-residence device to be used by the dynamic device reconfiguration (DDR) system residence support.

**ddd**  
is the unit address of a device of the same type as the system-residence device.

Note:

**Required:** This keyword parameter must be specified if **OPTIONS=DDRSYS** is coded.

**ASCII=** (For all control programs)  
specifies the inclusion or exclusion of the ASCII translate routine.

**EXCLUDE**  
specifies that the routine will not be included in the system.

**INCRS**  
specifies that the routine will be included in the SYS1.NUCLEUS data set.

**INCTAN**  
specifies that the routine will be included in the SYS1.SVCLIB data set.

Note:

**EXCLUDE** is the default value for the ASCII parameter.

**OPTIONS=**  
specifies task supervisor options as one or more of the following values. These values may be listed in any order.

**APR** (For all control programs)  
specifies that alternate path retry (APR) will be included.

**Note:** This subparameter does not have to be specified for the M65MP systems as it is automatically included.

**ATTACH** (For MFT only)  
specifies that the ATTACH function with the subtasking capability is to be included in the new operating system.

**Note:** This subparameter is required to support CRJE.

**Default:** If this subparameter is omitted, the ATTACH function without the subtasking capability will be included in the new operating system.

CCH (For all control programs)  
specifies that the channel-check handler be included.

Notes:

- S/360 Models 65, M65MP, 75, or 91: CCH must be specified if this option is to be included.
- S/360 Models 85 and 195 and S/370: CCH is automatically included.

COMM (For all control programs)  
specifies that there will be communication with the operator at IPL time for the purpose of changing certain system generation options.

Notes:

- This option must be specified if a BLDL list for SYS1.SVCLIB is desired.
- The subparameters BLDLTAB, ACSMETH, RENTCODE, ERP, and TRSVC can be deleted or modified at IPL time only if the COMM option was included at system generation.
- SCHEDULR Macro Instruction: The value in any of the following keyword parameters can be changed at IPL time: INITQBF, JOBQFMT, JOBQLMT, JOBQRES, JOBQMT, JOBQWTP, MINPART, PROCRES, STARTR, and STARTW.
- CTRLPROG Macro Instruction: The value in the keyword parameter QSPACE can be changed at IPL time.
- Required: This subparameter must be specified if telecommunication devices are specified for an M65MP system.
- Information: The communications procedure is described in OS Operator's Reference.

DDR (For all control programs)  
specifies that dynamic device reconfiguration (DDR) will be included.

Default: If this subparameter is omitted and DDRSYS and/or DDRNSL are specified, this option is assumed and included in the new system.

Note: This subparameter does not have to be specified for M65MP systems because it is automatically included.

DDRNSL (For all control programs)  
specifies that dynamic device reconfiguration for magnetic tape volumes with nonstandard labels will be included.

SUPRVSOR

Notes:

- This subparameter must be specified if there are nonstandard labels on any volumes in the system and DDR is desired.
- A nonstandard label routine must be supplied with the name NSLREPOS and placed in SYS1.MODGEN before the system generation process is started. (See "Program Options Descriptions" for further information.)

DDRSYS (For all control programs)  
specifies that the dynamic device reconfiguration for system-residence  
volumes will be included.

Note: If this subparameter is specified, then ALTSYS must also be  
specified.

DECSIM (For MVT)  
specifies that a decimal simulator routine for the S/360 Model 91  
will be included.

Notes:

- Use: DECSIM should be specified if COBOL or PL/I is to be included  
in the system, or if decimal arithmetic instructions are to be used  
with the assembler language.
- Restriction: This subparameter should not be specified for the S/360  
Model 195.

IDENTIFY (For MFT)  
specifies that the IDENTIFY function is to be included.

Notes:

- Required: If IDENTIFY is specified as a value of the RESIDENT keyword  
parameter, it need not be specified as a value of this keyword  
parameter.
- MVT (including M65MP): This subparameter does not have to be specified  
because it is assumed for MVT (including M65MP) systems.

NODAV (For all control programs)  
specifies that the code will not be included to verify the volume  
serial number of any 2311 or 2314 volumes (mounted after IPL time)  
that have not been specified by a system mount request.

Note: This subparameter is ignored if DDR, DDRSYS, or DDRNSL are specified.  
(The volume serial number verification is required when DDR is in the  
system.)

ONLNTEST (For all control programs)  
specifies that the online test (OLTEP) system function will be  
included. This function allows the running of I/O device tests under  
the operating system as a job step.

Notes:

- Obtaining: The online tests are contained in their own data set,  
and must be obtained from the IBM Branch Office representative.
- MFT: At least three buffers must be specified in the WTOBFRS parameter  
of the SCHEDULR macro instruction for online test.
- TIMER=INTERVAL or JOBSTEP in the SUPRVSOR macro instruction must be  
specified.
- Console Command: If you want to call OLTEP by a console command,  
the cataloged procedure OLTEP must be added to SYS1.PROCLIB.  
Additional space must be allocated in SYS1.PROCLIB during the  
preparation for system generation. (For detailed information, see  
OS OLTEP.)

PROTECT (For all control programs)  
specifies that the protect function routine will be included when  
the protect feature is part of the central processing unit.

Notes:

- MFT: PROTECT includes the VALIDCHK option.
- MVT (including M65MP): This value must be specified for MVT including M65MP systems.

RDE (For all control programs)  
specifies the inclusion of the reliability data extractor in the new  
system. RDE measures the reliability of the new system. For further  
information, see OS RDE Guide.

RER (For all control programs)  
specifies that the reduced error recovery procedure for magnetic tape  
will be used if requested with the OPTCD parameter of a DD statement  
of the DCB macro instruction.

Default: If this subparameter is omitted, all requests for reduced error  
recovery will be ignored.

Note:

Use: Reduced error recovery is used only for special applications, as  
described in the OS MFT Guide or OS MVT Guide.

TRSVCTBL (For MFT)  
specifies that a table containing the relative track addresses of  
all transient SVCs is to be stored in the resident portion of the  
control program.

Note:

- MVT (including M65MP): This subparameter does not have to be specified  
because it is assumed for MVT (including M65MP) systems.

VALIDCHK (For MFT)  
specifies that the WAIT, POST, and GETMAIN/FREEMAIN modules are to  
contain extra validity checking to determine whether addresses are  
located within proper boundaries.

Note:

Use: The validity checking for WAIT also checks for the number of events.

SUPRVSOR

Note:

For MVT (including M65MP): IDENTIFY TRSVCTBL are assumed and PROTECT must  
be specified.

RESIDNT=

specifies that one or more of the following functions, normally executed from the transient area, are to be included in the resident portion of the control program. These values can be listed in any order.

ACSMETH (For MFT)

specifies that access method modules are to be loaded and made part of the nucleus at IPL time.

Notes:

- ABEND/ABDUMP: In MFT, the BSAM modules IGG019BA, IGG019BB, and the BSAM device-dependent EOB module's are required to provide dumps of main storage during abnormal termination. These modules should be made resident using this subparameter, or by specifying the RAM= parameter at IPL time. This avoids bypassing the dump facility when there is insufficient space in the failing task partition where these modules must be loaded.
- Checkpoint/Restart: This subparameter must be specified for MFT systems if this facility is going to be used.
- MFT-MCS systems: If a card reader or a printer is specified as a console this subparameter must be specified.
- SMF: BSAM modules (IGC019BA, IGC019BB) must be resident in MFT to use the SMF option. If SMF is recording on a direct access device IGC019CD must also be resident; if on tape, IGC019CC must be resident.
- System Log: BSAM modules (IGG019BA, IGG019BB, IGG019CD) must be resident in MFT to use the log facility. To include them, this subparameter must be specified.

ATTACH (For all control programs)

specifies that the ATTACH function is to be included in the nucleus.

Note: The type of ATTACH made resident in MFT depends upon the subparameter of OPTIONS=.

BLDLTAB (For all control programs)

specifies that the directory entries for selected SYS1.SVCLIB or SYS1.LINKLIB modules are to be made resident at IPL time.

ERP (For all control programs)

specifies that the error recovery procedures for the 1285, 1287, and 1288 optical character readers are loaded and made resident at IPL time.

Note:

Limit: These device-dependent routines only apply to the 1285, 1287, and 1288 optical character readers. For a list of common error recovery routines that are loaded with the device dependent routines and made resident, see the OS Storage Estimates manual.

EXTRACT (For all control programs)

specifies that the EXTRACT function is to be included in the nucleus.

IDENTIFY (For all control programs)

specifies that the IDENTIFY function is to be included in the nucleus.

RENTCODE (For all control programs)  
specifies either of the following:

MFT: Any reenterable load modules from SYS1.LINKLIB or the loader program modules can be made resident.

MVT or M65MP: Any module in SYS1.LINKLIB or SYS1.SVCLIB can be made resident at IPL time. This value must be specified if checkpoint/restart is to be used in an MVT (including M65MP) system.

SPIE (For all control programs)  
specifies that the SPIE function is to be included in the nucleus.

TR SVC (For all control programs)  
specifies types 3 and 4 SVC modules are to be loaded and made resident at IPL time.

Note: OPTIONS=TR SVCTBL must be specified for MFT.

Notes:

- ATTACH, EXTRACT, IDENTIFY, and SPIE: These are assumed if TYPE=MVT (including M65MP) is specified in the CTRLPROG macro instruction.
- Use: The use of the option listed above is discussed in the OS MFT Guide and OS MVT Guide.
- Information: Some of the values specify functions that are to be made part of the nucleus during system generation; other values specify functions that are made resident at IPL time.

SER= (For all control programs)  
specifies the type of system error recovery desired for the CPU model specified in the CENPROCS macro instruction. One of two levels of system environment recording (SER0 or SER1) or machine-check handler (MCH) may be specified.

Default: If this keyword parameter is omitted, SEREP, SER0, SER1, or MCH will be supplied based on the CPU model and storage size specified in CENPROCS macro instruction. Figure 87 shows which default will be used.

Notes:

- Secondary CPUs: The nucleus generated may operate on a CPU other than that specified in the CENPROCS macro instruction. The SER option for each additional CPU must be specified with the SECMODS macro instruction.
- SYS1.ASRLIB: If MCH is specified and refreshability is to be supported, the system generation process allocates space to and catalogs the SYS1.ASRLIB system data set on the new system-residence volume.
- Model 91 or 195: SER1 must be specified.
- Model 40, 50, or 75: Only SER0 or SER1 can be specified.
- Model 65 With MFT or MVT: MCH can be specified and an F-design-level linkage editor must be specified.
- Model 65MP: MCH is assumed and an F-design-level linkage editor must be specified.
- Model 85 with MFT or MVT: All MCH modules on SYS1.SVCLIB must be link edited with the downward compatible (DC) attribute.

SUPRVSOR

- S/360 Model 85 and S/370 models with MFT or MVT: MCH is assumed.

	S/360 Models						S/370 Models				
Storage	40	50	65	65MP 75	85	91	195	135	145	155	165
DH								MCH			
G	SER0	SER0								MCH	
GD								MCH			
GE									MCH		
GF	SER1							MCH	MCH		
GFD									MCH		
H	SER1	SER1	SER1		SER1				MCH	MCH	
HG		SER1							MCH	MCH	
I		SER1	SER1	MCH	SER1	MCH			MCH	MCH	MCH
IH			SER1	MCH	SER1				MCH	MCH	MCH
J or larger			SER1	MCH	SER1	MCH	SER1	SER1		MCH	MCH

Figure 87. SER Defaults for CPUs

**TIMER=**

specifies the inclusion of the timer function when the timer feature is part of the central processing unit.

**TIME (For MFT)**

specifies the ability to request date plus time of day in various units of measurement.

**INTERVAL (For all control programs)**

provides the same functions as TIME, plus changing the time at midnight, and the ability to request, check, and cancel intervals of time.

**JOBSTEP (For all control programs)**

provides the same functions as INTERVAL, plus the timing of each job step and enforcement of job step time limits.

Default: If this keyword parameter is omitted, only the date will be provided.

Notes:

- Time-stamping: If proper time-stamping is expected on console messages, this keyword parameter must be included.
- SMF: If ACCTRTN=SMF or ESV=SMF is coded in the SCHEDULR macro instruction, TIMER=JOBSTEP must be specified.
- OLTEP: If OPTIONS=ONLYTEST is specified, either INTERVAL or JOBSTEP must be specified.
- Emulator: If the EMULATOR macro instruction is specified, TIMER=INTERVAL or JOBSTEP must be specified.

**TRACE=(For all control programs)**

specifies the inclusion of an optional trace table.

number

is the number of entries in the table.

**Note:**

Information: See the OS MFT Guide and OS MVT Guide for a description of the trace table.



Coding Hints and Additional Information

Figure 88 shows the relationship between parameters in this macro instruction.

<u>Parameter</u>	<u>Comment</u>	<u>Parameter</u>
RESIDNT=IDENTIFY	Don't need to code:	OPTIONS=IDENTIFY
RESIDNT=TR SVC	Then must specify (if MFT):	OPTIONS=TR SVCTBL
OPTIONS=COMM	These can be changed at IPL time:	BLDLTAB, ACSMETH, RENTCODE, TR SVC

Examples: The following example illustrates the use of the SUPRVSOR macro instruction to specify task supervisor options for multiprogramming with a fixed number of tasks (MFT). A table containing the relative track addresses of transient SVCs is to be stored in the resident portion of the control program. The IDENTIFY and the ATTACH functions are to be included in the resident portion of the control program. There are 100 entries in the trace table.

```
SUPRVSOR OPTIONS=TR SVCTBL,RESIDNT=(IDENTIFY,ATTACH),TRACE=100
```

The following example illustrates the use of the SUPRVSOR macro instruction to specify task supervisor options for multiprogramming with a variable number of tasks (MVT). It is assumed that ATTACH, EXTRACT, IDENTIFY, and SPIE are to be made resident. Resident types 3 and 4 SVC routines are to be loaded and made resident at IPL time. The IDENTIFY function and a table of the relative track addresses of all transient SVCs are assumed. The protect function is required. The operator may change TR SVC, MINPART, INITQBF, and QSPACE, but not RENTCODE and BLDLTAB. Timing of each job step is selected. There are 150 entries in the trace table. SER1 is to be used.

```
SUPRVSOR RESIDNT=TR SVC,TIMER=JOBSTEP,TRACE=150,SER=SER1,
OPTIONS=(PROTECT,COMM)
```

SUPRVSOR

Keyword	Value	MFT	MVT	M65MP
RESIDNT	ATTACH	Optional	Assumed	Assumed
	EXTRACT	Optional	Assumed	Assumed
	IDENTIFY	Optional	Assumed	Assumed
	SPIE	Optional	Assumed	Assumed
	BLDLTAB	Optional	Optional	Optional
	ACSMETH	Optional <sup>1</sup>	--	--
	RENTCODE	Optional	Optional <sup>1</sup>	Optional <sup>1</sup>
	TR SVC	Optional	Optional	Optional
OPTIONS	ERP	Optional	Optional	Optional
	ATTACH	Optional	Invalid	Invalid
	IDENTIFY	Optional	Assumed	Assumed
	TRSVCTBL	Optional <sup>2</sup>	Assumed	Assumed
	PROTECT	Optional	Required	Required
	VALIDCHK	Optional	Invalid	Invalid
	COMM	Optional	Optional	Optional
	ONLNTEST	Optional	Optional	Optional
	CCH	Optional <sup>7</sup>	Optional <sup>7</sup>	Required
	DECSIM	Invalid	Optional	Invalid
	RER	Optional	Optional	Optional
	APR	Optional	Optional	Assumed
	DDR	Optional	Optional	Assumed
	DDRSYS	Optional	Optional	Optional
DDRNSL	Optional	Optional	Optional	
RDE	Optional <sup>8</sup>	Optional <sup>8</sup>	Invalid	
ALTSYS6	Address	Optional	Optional	Optional
TIMER	TIME	Optional	Invalid	Invalid
	INTERVAL	Optional	Optional <sup>5</sup>	Optional <sup>5</sup>
	JOBSTEP	Optional <sup>4</sup>	Optional <sup>8</sup>	Assumed
TRACE	Number	Optional	Optional	Optional
SER	SERO	Optional	Optional	Invalid
	SER1	Optional	Optional	Invalid
	MCH	Optional <sup>3</sup>	Optional <sup>3</sup>	Assumed

<sup>1</sup> Required if checkpoint/restart facility of Checkpoint/Restart is to be used.

<sup>2</sup> Required if TR SVC is specified in the RESIDNT keyword.

<sup>3</sup> This option is a standard feature of the S/360 Model 85 and S/370 Models.

<sup>4</sup> Required if job/step CPU timing feature is desired.

<sup>5</sup> Either INTERVAL or JOBSTEP must be specified for MVT or M65MP.

<sup>6</sup> Must be specified if DDRSYS is coded in the OPTIONS keyword.

<sup>7</sup> This option is automatically included for S/360 Models 85 and 195 and S/370 Models.

<sup>8</sup> This option is valid for S/360 Model 195 and certain S/370 CPUs only. See the OS RDE Guide for a complete explanation of RDE.

Figure 88. SUPRVSOR Macro Instruction Values for MFT, MVT, and M65MP

## SVCLIB

The SVCLIB macro instruction is used to add your transient routines, in load module form, to the SVC library (SYS1.SVCLIB) during system generation. This macro instruction is optional.

### Basic Requirements

- The load modules to be included must be members of a partitioned data set. The data set must be cataloged as SYS1.name in the generating system.
- The number, type, and SVRB extended save area of each SVC routine to be added must be specified in the SVCTABLE macro instruction.

### Processor/Library Generation

- Sufficient space for your routines must have been allocated in SYS1.SVCLIB.
- The number, type, and SVRB extended save area of each transient SVC must have been specified with the SVCTABLE macro instruction during the last complete operating system generation or nucleus generation.

Name	Operation	Operand
	SVCLIB	PDS=SYS1.name MEMBERS=(name[,name]...)

PDS=

SYS1.name  
specifies the name of the partitioned data set that contains the routines to be added.

### Note:

The value of name cannot exceed eight alphameric characters, the first of which must be alphabetic.

MEMBERS=

name  
is the name of the member to be added. The name of each transient SVC routine must be of the form:

IGC<sup>ss</sup>nnn

ss  
is the number of the load module minus 1. For example, the second load module has a value of 01. The value of ss is always 00 for type 3 SVC routines.

nnn  
is an SVC number. It must be a signed decimal integer (such as 242=24B) if the routine is called directly by SVC.

SVCLIB

Notes:

- Nonstandard Label Routines: The names of nonstandard label routines must conform to the proper naming conventions described in the OS Tape Labels.
- Maximum: A maximum of 255 characters may be used in an operand field of any macro instruction.
- User-Written: User-written SVC routines are described in the OS MFT Guide and OS MVT Guide.

Coding Hints and Additional Information

The following example illustrates the use of the SVCLIB macro instruction to add the routines named IGC0025E, IGC0025D, IGC0025C, and IGC0025B to the SVC library. These routines are members of the SYS1.USERSVC partitioned data set, and are each called directly by an SVC.

```
SVCLIB PDS=SYS1.USERSVC, MEMBERS=(IGC0025E, IGC0025D, IGC0025C, IGC0025B)
```

## SVCTABLE

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

### Basic Requirements

- Type 1 and 2: For each type 1 or type 2 SVC, there should be a corresponding module specified in the RESMODS macro instruction: one module may contain more than one resident SVC routine.
- Type 3: For each type 3 SVC, there should be a corresponding module specified in the SVCLIB macro instruction: each module may contain only one transient SVC routine.
- Type 4: For each type 4 SVC, there should be one corresponding module specified in the SVCLIB macro instruction for each load module of the SVC routine.
- Information: For further information on user-written SVC routines, refer to the OS MFT Guide or OS MVT Guide.

### Nucleus Generation

- There should be an entry for each type 1 or 2 SVC specified in the RESMODS macro instruction.
- There should be an entry for each previously generated type 3 or 4 SVC (using the SVCLIB macro instruction) that is to be supported by the new nucleus.

### I/O Device Generation

- If this macro instruction was specified in the last complete generation, it must be specified with the same operands.
- If this macro instruction was not used in the last complete generation, it cannot be specified in this generation.

Name	Operation	Operand
	SVCTABLE	operand[,operand]...

SVCTABLE

Each operand must be written in the following format:

SVC-nnn-Ta-Sb

Uppercase letters and hyphens must be written exactly as shown.

nnn

specifies the SVC number as a decimal integer.

### Notes:

- Numeric Assignment: You must assign unique numbers to your SVC routines, and should assign them in descending order starting with 255 and ending with 200 to avoid conflict with the numbers assigned to IBM-written SVC routines.
- Maximum Value: The highest SVC number that may be assigned is 255.

a  
is 1, 2, 3, or 4.

Note:

Purpose: The number specifies the type of SVC.

b  
is a decimal number that specifies the number of doublewords in the extended save area of the SVRB associated with the SVC routine

Notes:

- Purpose: The decimal number indicates the number of doublewords by which the SVRB is to be extended.
- Required Values: A type 1 SVC must have a value of 0. Types 2, 3, and 4 can have a value of from 0 to 6.

Coding Hints and Additional Information

The following example illustrates the use of the SVCTABLE macro instruction to specify that four user-written SVCs are to be added to the new operating system.

```
SVCTABLE SVC-255-T4-S5,SVC-254-T2-S3,SVC-253-T3-S1,SVC-252-T1-S0
```

# SYSUTILS

The SYSUTILS macro instruction specifies the amount of main storage available to all the system and data set utilities except IEBCOPY. This macro instruction is optional.

## Basic Requirements

- System and data set utilities are generated with every operating system. They operate in 15K bytes of main storage unless this macro instruction is used to specify a larger amount. The IEBCOPY utility program requires 30K, but the program will obtain the storage it needs.

## I/O Device Generation

- If this macro instruction was specified in the last complete system generation, it must be specified with the same size in the I/O device generation.
- If this macro instruction was not used in the last complete operating system generation, it cannot be specified.

---

Name	Operation	Operand
	SYSUTILS	[ SIZE=size ]

---

### SIZE=

specifies the amount of main storage, in bytes, available to the system and data set utilities.

### size

is a decimal number of from 15360 to 999999, or it may be of the form nnnnK, where nnnn is an integer of from 15 to 9999 and K represents 1024 bytes.

Default: If this keyword parameter is omitted, a value of 15360 is assumed.

## Coding Hints and Additional Information

The following example illustrates the use of the SYSUTILS macro instruction to specify that there are 44K bytes of main storage available to the system and data set utilities.

```
SYSUTILS SIZE=44K
```

SYSUTILS





## TELCMLIB

The TELCMLIB macro instruction specifies the inclusion of the telecommunications subroutine library in the new operating system. This macro instruction is optional.

### Basic Requirement

- SYS1.TELCMLIB must have space allocated for it in the new system. It may be cataloged.
- TSO: This macro must be specified if a TSO system is to be generated.

---

Name	Operation	Operand
	TELCMLIB	[UNIT=name VOLNO=serial]

---

Default: If these keyword parameters are omitted, the telecommunications subroutine library is placed on the new system-residence volume.

#### UNIT=

specifies a direct-access device in the generating system that will contain the volume on which the telecommunications subroutine library will be generated.

name

is the device type.

#### Note:

If this keyword parameter is specified, VOLNO must be coded also.

#### VOLNO=

specifies the volume that is to contain the telecommunications subroutine library.

serial

is the serial number.

#### Notes:

- Value: The value specified must be the same as the value given to the VOLUME keyword parameter of the DD statement used to allocate space for SYS1.TELCMLIB during preparation for system generation.
- Required: If this keyword parameter is specified, UNIT must also be coded.

### Coding Hints and Additional Information

This example illustrates the use of the TELCMLIB macro instruction to specify the inclusion of the telecommunications subroutine library in the new operating system. SYS1.TELCMLIB is to reside on the new system-residence volume.

TELCMLIB

TELCMLIB



## TSOPTION

The TSOPTION macro instruction specifies the inclusion of several time-sharing options in the new operating system. This macro is valid only in a complete operating system generation of a TSO system.

---

Name	Operation	Operand
	TSOPTION	[ LOGTIME=time] [ LOGLINE=integer] [ BCLMT=integer]

---

### LOGTIME=

specifies length of elapsed time in seconds, during LOGON, before a message is written to the terminal.

time  
is an integer from 1 to 16,777,215

Default: 30 seconds.

### LOGLINE=

specifies the number of syntactically valid lines that may be entered before an attempt to LOGON is automatically canceled.

integer  
is an integer from 1 to 32,760

Default: 10 lines.

### BCLMT=

specifies the number of 130-byte records that will be set aside for your broadcast messages in the SYS1.BROADCAST data set.

integer  
is an integer from 1 to 1000.

Default: 100 records.

### Coding Hints and Additional Information

This example illustrates the use of the TSOPTION macro instruction to specify the time-sharing options. The new system will have an elapsed time of 500 seconds, ten lines, and 200 broadcast records.

```
TSOPTION LOGTIME=500,BCLMT=200
```

TSOPTION



# UADS

The UADS macro instruction specifies the inclusion of the TSO SYS1.UADS data set in the new operating system. This macro instruction is valid only in a complete operating system or processor generation of a TSO system.

## Basic Requirement

Space must be allocated for SYS1.UADS. The data set may be cataloged in the new system.

## Processor/Library Generation

This macro instruction can be used to add to or replace a SYS1.UADS data set. If an existing data set is to be replaced, it must be scratched and uncataloged before the new one is added.

---

Name	Operation	Operand
	UADS	[UNIT=name VOLNO=serial]

---

Default: If the UNIT and VOLNO keyword parameters are omitted, SYS1.UADS is placed on the new system-residence volume.

UNIT=  
specifies a direct-access device in the generating system. The volume on this device will contain the SYS1.UADS data set.

name  
is the device type.

### Note:

Required: If this keyword parameter is specified, VOLNO must also be coded.

VOLNO=  
specifies the volume that is to contain the SYS1.UADS data set.

serial  
is a serial number.

### Note:

Required: If this keyword parameter is specified, UNIT must also be coded.

## Coding Hints and Additional Information

This example illustrates the use of the UADS macro instruction to specify the inclusion of the TSO SYS1.UADS data set in the new operating system. The unit name is 2301. The volume serial number is 10454.

```
UADS UNIT=2301,VOLNO=10454
```

UADS



# UCS

The UCS macro instruction specifies the IBM standard character set images for a printer with the universal character set (UCS) feature. This macro instruction is optional.

## Basic Requirements

- This macro instruction can be specified only if UNIT=3211 or if FEATURE=UNVCHSET and UNIT=1403 have been specified in the IODEVICE macro instruction.
- If the UCS macro instruction is omitted, IBM standard character set images are not included in the system. Therefore, if you have a printer with UCS, you must add your own character set images to the system as described in the OS MFT Guide and OS MVT Guide.

## I/O Device Generation

- This macro instruction can be specified to add UCS support for the printer if it was not supplied in the last complete system generation.
- It does not have to be respecified if it has already been specified in a previous generation.

---

Name	Operation	Operand
	UCS	UNIT= {1403} {3211} IMAGE= (image[,image]...) DEFAULT=(image[,image...] )

---

### IMAGE=

specifies the IBM character set images to be included as one or more of the following values. These values may be listed in any order.

Value	IBM Standard Character Set Image
1403	3211
AN	All Arrangement A, standard EBCDIC CHARACTER SET, \$* CHARACTERS
HN	H11 Arrangement H, EBCDIC character set for FORTRAN and COBOL, 48 characters
	G11 ASCII character set
PCAN	Preferred character set, arrangement A
PCHN	Preferred character set, arrangement H
PN	P11 PL/I character set
ONC	PL/I preferred character set for commercial applications
ON	PL/I preferred character set for scientific applications
RN	Character set for commercial applications of FORTRAN and COBOL
TN	T11 Text printing, scientific 120 graphics

UCS

SN Preferred character set for text printing  
XN High-speed alphanumeric character set for 1403 Model 2  
YN High-speed alphanumeric character set for 1403 Model 3 or N1

**DEFAULT=**

specifies the character set images to be used as default options when a job does not specify a character set image through its job control language statements.

image

The values are the same as for the IMAGE keyword.

**UNIT=**

specifies the printer to be used for character set images designated by the IMAGE parameter.

Default: 1403 is assumed if this keyword parameter is omitted.

Notes:

- Required: Any value specified with the DEFAULT keyword parameter must also be specified with the IMAGE keyword.
- Recommendation: It is recommended that all character set images that can produce valid results as default options be specified.
- UCS Macro Use: The UCS macro can be used only once to include the 1403 and once to include the 3211.

Coding Hints and Additional Information

The following example illustrates the use of the UCS macro instruction to specify that the character set images AN, PN, and TN are to be included in the new system. The images AN and PN are designated as default options.

```
UCS IMAGE=(AN,PN,TN),DEFAULT=(AN,PN),UNIT=1403
```



## UNITNAME

The UNITNAME macro instruction is used to name a group of I/O devices. This macro instruction is optional.

### Basic Requirements

- This macro instruction is required to name each group of I/O devices in the system, except for device types. (See Appendix A.)
- All UNITNAME macro instructions having the same NAME value must appear consecutively in the input stream.
- If the IBM-supplied cataloged procedures in SYS1.PROCLIB are used, this macro instruction must be used to assign certain names to groups of I/O devices. These names will be used by the IBM-supplied cataloged procedures to specify the I/O devices required. The names required are:

SYSSQ      For magnetic tape and/or direct-access devices

SYSDA      For direct-access devices only

SYSCP      For card punch

- The only combination of unlike device types permitted in a group is magnetic tape and direct-access devices.
- A maximum of 50 uniquely named groups can be specified for a system. If more are required, see Appendix B for the procedure to follow.
- A maximum of 255 addresses can appear in one group. If more addresses must be listed for a particular name, another UNITNAME macro instruction is coded using the same name.
- A maximum of 255 characters can be used in the operand of any macro instruction.

### I/O Device Generation

- This macro instruction may be used to name a new group of I/O devices or rename an existing group of devices.
- If the group names are to remain the same, they must be respecified.

UNITNAME

---

Name	Operation	Operand
	UNITNAME	UNIT= { (address[ , address ]... [ , (address, n) ]... ) } { (address, n) } NAME=name

---

#### NAME=

specifies the name to be given to a group of I/O devices.

#### name

is a value from 1 to 8 characters. Each can be alphameric, national, or one of the two special characters, slash (/) and hyphen (-).

**UNIT=**

specifies a group of I/O devices that will be recognized by the name assigned.

**address**

is the unit address of an I/O device to be included in the group.

**Note:**

**Required:** When n is specified, this address must be the lowest number of the sequential addresses being specified.

**n**

is the number of units and the total of sequential addresses to be assigned. The value can be a number from 1 to 255.

**Notes:**

- **Same Addresses:** The addresses used must be the same as those specified in the IODEVICE macro instruction for those devices.
- **Parentheses Required:** If the form (address,n) is used as the only subparameter of the macro instruction, double parentheses must be used. For example, UNIT=((180,4)) would create a group of four devices that have the addresses 180, 181, 182, and 183.
- **Unlike Device Types:** The only combination of unlike device types permitted in a group is magnetic tape and direct access devices.

Coding Hints and Additional Information

**2420 Magnetic Tape Drives:** To select these drives specifically, establish them as a group by assigning a group name and specifying their unit addresses.

**Maximum Number of Devices Formula:** The maximum number of devices that can be included in groups is determined by the following formula:

$$510 - N = \text{Maximum number of devices}$$

N is the number of uniquely named groups. The maximum value of N is 50.

For example, if there are 40 groups, a maximum of 470 devices can be distributed among those groups.

**SYSDA and SYSSQ:** It is recommended that 2321 addresses not be included in these groups because some processors (COBOL E and sort/merge) do not support intermediate work data sets on 2321 volumes.

It is recommended that a group named SYSOUT be defined for intermediate system output data sets. The devices in the SYSOUT group should be a subset of the devices in the SYSDA group. The group name SYSOUT should be defined as the default value in the reader cataloged procedure.

## Examples

The following example illustrates the use of the UNITNAME macro instruction to assign the name TAPE to the devices located at 180, 181, 182, 183, 190, 191, 290, and 390.

```
UNITNAME NAME=TAPE,UNIT=((180,4),290,390,(190,2))
```

The following 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 with an IODEVICE macro instruction).

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

UNITNAME



## Cross-References Between the System Generation MACRO Instructions

Many of the keywords in the system generation macro instructions require keywords in other macro instructions to specify a particular function. The relationships between the keywords of the system generation macro instructions is shown in Figure 89.

Dependencies between the keywords within a macro instruction are not shown because they are illustrated by the macro instruction format and, when necessary, by tables within each macro instruction description.

### Note:

In the table below (All) has been used to indicate that the entire macro instruction (all the parameters) is affected by the macro instruction referred to.

Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to
ALGOL	(All)	ALGLIB CENPROCS	(All) INSTSET MODEL
ASSEMBLR	(All)	CENPROCS	MODEL
CENPROCS	INSTSET	SUPRVSOR	OPTIONS
	FEATURE	SUPRVSOR	OPTIONS
	MODEL	SUPRVSOR EMULATOR	OPTIONS (All)
CHECKER	(All)	SCHEDULR	OPTIONS
	DESIGN	FORTTRAN	DESIGN
CKPTREST	(All)	SUPRVOSR	RESIDNT
	ELIGBLE	PL1	(All)
COBLIB	(All)	COBOL	(All)
	DESIGN	CENPROCS GRPAHICS	MODEL GSP
COBOL	(All)	CENPROCS COBLIB	INSTSET (All)
	BUFSIZE	CENPROCS	STORAGE
	DESIGN	CENPROCS	MODEL

Figure 89 (Part 1 of 6). Cross-References Between Macro Instructions

Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to
CTRLPROG	HIARCHY	PARTITNS	(All)
	QSPACE	SCHEDULR SUPRVSOR	ACCTRIN OPTION
	SYSQUE	SCHEDULR	ACCTRIN CONOPTS
	TMSLICE	PARTITNS SUPRVSOR	(All) OPTIONS TIMER
	TYPE	CENPROCS EMULATOR  PARTITNS	INSTSET (All) STORAGE (All)
DATAMGT	ACSMETH	CTRLPROG SUPRVSOR TELCMLIB EMULATOR	TYPE TIMER (All) (All)
EDITOR	DESIGN	SUPRVSOR	SER
EDIT	(All)	CMDLIB	(All)
EMULATOR	(All)	SCHEDULR	REPLY WTOBFRS
		SUPRVSOR	TIMER OPTIONS
		CENPROCS CTRLPROG	MODEL TYPE
		DATAMGT	ACSMETH
FORTLIB	(All)	GRAPHICS	GSP
	BOUNDRY	CENPROCS	MODEL
	DESIGN	CENPROCS	MODEL
FORTRAN	(All)	CENPROCS	INSTSET
	DESIGN	CENPROCS	MODEL STORAGE
GJOBCTL	(All)	CTRLPROG	TYPE
	CLASS	CTRLPROG	TYPE
	DAPOOL	UNITNAME	(All)
	REGION	CTRLPROG	TYPE
	TYPE	DATAMGT PTOP	ACSMETH (All)
	UNITS	IODEVICE	UNIT

Figure 89 (Part 2 of 6). Cross-References Between Macro Instructions

Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to
GRAPHICS	(All)	FORTLIB	(All)
		COBLIB	(All)
		PL1LIB	(All)
		SECONSOLE	ALTCONS CONSOLE ALTCONS CONSOLE
IOCONTRL	(All)	IODEVICE	(All)
		FEATURE	IODEVICE OPTCHAN
		UNIT	IODEVICE OPTCHAN
IODEVICE	(All)	IOCONTRL	(All)
		UNITNAME	(All)
		ADDRESS	CHANNEL (All)
		FEATURE	CTRLPROG TYPE
		OPTCHAN	CHANNEL (All)
		UNIT	CENPROCS MODEL
			SCHEDULR UNITNAME CONOPTS (All)
LOADER	(All)	SUPRVSOR	OPTIONS
OUTPUT	(All)	CMDLIB	(All)
PARTITNS	(All)	CTRLPROG	TYPE HIARCHY
		SnnK	SUPRVSOR OPTIONS
PL1	(All)	CENPROCS	INSTSET
		CKPTREST	ELIGBLE
		PL1LIB	(All)
		SUPRVSOR	WAIT
PL1LIB	(All)	PL1	(All)
		SUPRVSOR	(All)
		CONVS	PL1 OPT
		MODES	SUPRVSOR OPTIONS OPT
PROCLIB	(All)	GENERATE UNITNAME	INDEX (All)
PTOP	(All)	IODEVICE	UNIT ADAPTER
		DATAMGT	ACSMETH
		TELCMLIB	(All)

Figure 89 (Part 3 of 6). Cross-References Between Macro Instructions

Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to
RESMODS	(All)	SVCTABLE	(All)
SCHEDULR	ACCTRTN	SUPRVSOR	TIMER
	ALTCONS	IODEVICE	ADDRESS
	CONSOLE	IODEVICE	ADDRESS
	ESV	SUPRVSOR	TIMER
	HARDCPY	IODEVICE SECONSLE	ADDRESS CONSOLE
	INITQBF	SUPRVSOR	OPTIONS
	JOBQFMT	SUPRVSOR	OPTIONS
	JOBQLMT	SUPRVSOR	OPTIONS
	JOBQRES	IODEVICE SUPRVSOR	ADDRESS OPTIONS
	JOBQTMT	SUPRVSOR	OPTIONS
	JOBQWTP	SUPRVSOR	OPTIONS
	MINPART	SUPRVSOR	OPTIONS
	OPTIONS	DATAMGT GENTSO	ACSMETH (All)
	PROCESS	IODEVICE SUPRVSOR	ADDRESS OPTIONS
	REPLY	EMULATOR SUPRVSOR	(All) OPTIONS
	STARTER	IODEVICE	ADDRESS
	STARTW	IODEVICE	ADDRESS
	TYPE	CTRLPROG	TYPE
	WTOBFRS	EMULATOR SUPRVSOR	(All) OPTIONS
	SECMODS	(All)	CENPROCS CTRLPROG SUPRVSOR
(storage)		CENPROCS	STORAGE

Figure 89 (Part 4 of 6). Cross-References Between Macro Instructions



Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to	
SECONSLE	(All)	CTRLPROG	TYPE	
		SCHEDULR	ALTCONS	
			CONOPTS	
	ALTCONS	IODEVICE	ADDRESS	
		SCHEDULR	CONSOLE	
	CONSOLE	IODEVICE	ADDRESS	
SORTLIB	(All)	SORTMERG	(All)	
SORTMERG	(All)	EDITOR	(All)	
		GENERATE	INDEX	
		GENTSO	INDEX	
		SORTLIB	(All)	
	SIZE	CENPROCS	STORAGE	
SUPRVSOR	(All)	PL1LIB	(All)	
		OPTIONS	CENPROCS	MODEL
			COBOL	(All)
			CTRLPROG	QSPACE
				TYPE
			PL1	(All)
			RPG	(All)
			SCHEDULR	INITQBF
				MINPART
				JOBQFMT
	JOBQLMT			
	JOBQRES			
	JOBQTMT			
	JOBQWTP			
	PROCRES			
	STARTR			
	STARTW			
	WTOBFRS			
		(All)		
	EMULATOR			
RESIDNT		CTRLPROG	TYPE	
		IOCONTRL	FEATURE	
		IODEVICE	FEATURE	
SER		CENPROCS	MODEL	
			STORAGE	
		SECMODS	option	
TIMER		EMULATOR	(All)	
		SCHEDULR	ACCTRNT	
WAIT		CTRLPROG	TYPE	
		IODEVICE	UNIT	

Figure 89 (Part 5 of 6). Cross-References Between Macro Instructions

Macro Instruction	Keyword	Macro Instruction Referred to	Keyword Referred to
SVCLIB	(All)	SVCTABLE	(All)
SVCTABLE	(All)	RESMODS SVCLIB	(All) (All)
UCS	(All)	IODEVICE	FEATURE UNIT
UNITNAME	UNIT	IODEVICE	ADDRESS

Figure 89 (Part 6 of 6). Cross-References Between Macro Instructions

# Appendixes

Each of the following is provided in an appendix:

- A list of device type names provided on every operating system.
- The additional preparation required if more than 96 I/O devices are to be specified during system generation
- The distribution methods.
- The messages produced during system generation.
- Descriptions of Functionally Equivalent I/O Devices



## 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 the devices to which they apply follow.

### Magnetic Tape Drives

<u>Device Type</u>	<u>Description</u>
2400	2400 9-track magnetic tape drive, which has either an 800 byte-per-inch (density) capability when the dual-density feature is not installed, or a 1600 byte-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, 2415, or 2420 9-track magnetic tape drive, which has only a 1600 byte-per-inch (density) capability
2400-4	2400 or 2415 9-track magnetic tape drive having an 800- and a 1600 byte-per-inch (density) capability
3400-2	3420 magnetic tape drive having 7-track capability and data conversion
3400-3	3410 or 3420 9-track magnetic tape drive having 1600 byte-per-inch (density) capability
3400-4	3410 or 3420 9-track magnetic tape drive having 800 byte-per-inch and 1600 byte-per-inch (density) capability

### Direct-Access Devices

<u>Device Type</u>	<u>Description</u>
2301	2301 drum storage
2302	2302 disk storage drive
2303	2303 drum storage drive
2305-1	2305-1 fixed-head disk storage (Model 1)
2305-2	2305-2 fixed-head disk storage (Model 2)
2311	2311 disk storage drive
2314	2314 direct-access storage facility
2321	2321 data cell drive
3330	3330 disk storage drive

## Unit Record Equipment

<u>Device Type</u>	<u>Description</u>
1052	1052 printer keyboard
1053	1053 Model 4 printer
1403	1403 printer or 1404 printer (continuous form only)
1442	1442 card read punch
1443	1443 printer
2495	tape cartridge reader
2501	2501 card reader
2520	2520 card read punch
2540	2540 card read punch (read feed)
2540-2	2540 card read punch (punch feed)
2671	2671 paper tape reader
3066	3066 console -- S/370 Model 165
3210	3210 console printer keyboard -- S/370 Model 155
3211	3211 printer
3215	3215 console printer keyboard -- S/370 Model 155
3505	3505 card reader
3525	3525 card punch with read and print feature

## Graphics Devices

<u>Device Type</u>	<u>Description</u>
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
2280	2280 film recorder
2282	2282 film recorder/scanner
3066	3066 integrated CRT operator's console -- S/370 Model 165
3277	3277 model 1 or model 2 display station
3284	3284 model 1 or model 2 printer
3286	3286 model 1 or model 2 printer
5450	Model 85 integrated CRT operator's console

Optical Character Readers

<u>Unit Name</u>	<u>Device Type</u>
1285	optical reader
1287	optical reader
1288	optical reader

Magnetic Character Readers

<u>Unit Name</u>	<u>Device Type</u>
1275	1275 optical reader sorter (available through World Trade branch offices only)
1419	1419 magnetic character reader

Audio Responses

<u>Unit Name</u>	<u>Device Type</u>
7770	7770 audio response unit





## Appendix B: Supporting Additional I/O Devices

If you want to increase the limits specified in the IOCTRL, IODEVICE, and UNITNAME macro instructions, you must follow a special procedure. There are two levels of additional I/O device support. The system generation with 248 devices can be performed only on a central processing unit that has at least 128K of main storage with a generating system that contains the Release 18 or later level of Assembler F, or the starter operating system must be used. The first limit allows a maximum of 248 I/O devices, 80 control units, and 100 group names. (Ten group names are assigned to each 2321 for which an IODEVICE macro instruction has been written.) The second limit allows up to 768 I/O devices, 255 control units, and 100 group names. The main storage requirement for the generating system must be more than 128K to generate a new system with 768 devices.

Before executing the system generation process, you must:

- Calculate the amount of storage required by the unit control blocks (UCB)
- Calculate the number of devices that can be specified in groups
- Redefine two members of SYS1.GENLIB before starting Stage I
- Insert two control statements in the system generation input deck

After system generation, you should reestablish the standard definitions.

Note: In this procedure, it is assumed that SYS1.GENLIB resides on a 2311 direct-access volume whose serial number is DLIB04. If SYS1.GENLIB resides on a different volume or device, the volume serial number and device type should be modified accordingly. All other parameters must be coded as shown.

### Calculating the Storage Required by Unit Control Blocks

The size of a unit control block (UCB) varies with the type of device. For example, a UCB for a magnetic tape is 24 bytes long and the UCB for a 2311 disk drive is 64 bytes long. In addition, 4 bytes are added to each UCB if TYPE=M65MP is specified in the CTRLPROG macro instruction. All UCBs in a system must reside in the first 32K bytes of fixed main storage along with all the request queue elements (RQEs), the control program nucleus, and the machine logout and PSW area. (The RQEs are specified by the parameter MAXIO in the CTRLPROG macro instruction.) The system generation process will be halted if the 32K limit is exceeded or if more than 768 devices are specified. Both of the following formulas must be satisfied in order not to exceed the limits.

Formula 1

$$TP + UR + DA + GR + TA \leq 768$$

TP is the number of teleprocessing devices.  
UR is the number of unit-record devices.  
DA is the number of direct-access devices.  
GR is the number of graphic devices.  
TA is the number of magnetic tape drives.

Formula 2

$$(24+A+B+BB) (TP+UR) + (64+A) (DA) + (12+C) (RQE) + (40+A) (GR) + (52+A) (TA) \leq 32,768-D-E$$

- A is 4 for M65MP.  
B is 4 for a printer with UCS buffer.  
BB is 4 for the devices 1285, 1287, 1288, and 2495.  
TP is the number of telecommunications lines.  
UR is the number of unit-record devices.  
DA is the number of direct-access devices. (A value of 3.4 is used for each 2321.)  
C is 4 for MVT or M65MP.  
RQE is the number of request queue elements.  
In addition, add 1 for the recovery management support (RMS) if it is in the system.  
Also, add 2 if the dynamic device reconfiguration (DDR) option is in the system.  
GR is the number of graphic devices.  
TA is the number of magnetic tape drives.  
D 384 for machine logout and PSW area for S/360 Models 40 to 75, MFT, and MVT.  
512 for machine logout and PSW area for S/370.  
4096 for machine logout and PSW area for M65MP.  
1978 for machine logout and PSW area for S/360 Models 85 and 91.  
2584 for machine logout and PSW area for S/360 Model 195.  
E 2000 for MFT nucleus resident below 4K.  
2400 for MFT nucleus resident below 4K with time-slicing option.  
168 for MVT nucleus resident below 4K.  
168 for M65MP nucleus resident between IOS and first 4K.

Calculating the Number of Devices for Groups

The total number of devices that can be included in groups is determined by either of the two formulas below.

For up to 248 devices use:

$$A=510-N$$

For up to 768 devices use:

$$A=1028-N$$

A is the total number of devices. A device may belong to more than one group, but it must be counted as a separate device each time it is used.

N is the number of uniquely named groups. The maximum number is 100 in both formulas.

## Redefining Two Members of SYS1.GENLIB

Two members of SYS1.GENLIB, SGGBLPAK and either SGPAK248 or SGPAK768, must be redefined for either level of limitation before Stage I of system generation can be executed. These members can be redefined by using the IEHPROGM utility program. The two utility control statements required to allow up to 248 devices are:

```
RENAME DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,MEMBER=SGGBLPAK,      X
        NEWNAME=SGPAK96
RENAME DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,MEMBER=SGPAK248,      X
        NEWNAME=SGGBLPAK
```

The two utility control statements required to allow up to 768 devices are:

```
RENAME DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,MEMBER=SGGBLPAK,      X
        NEWNAME=SGPAK96
RENAME DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,MEMBER=SGPAK768,      X
        NEWNAME=SGGBLPAK
```

The following DD statement is required in the step where the redefining is done:

```
//ddname DD DSNAME=SYS1.GENLIB,DISP=OLD
```

The two members are redefined during an execution of the IEHPROGM utility program using the utility control statements and the DD statement given above. (Detailed information about the IEHPROGM utility program can be found in OS Utilities.) Because IEHPROGM is also used to initialize the new system data sets during preparation for system generation, the utility control statements and the DD statement for redefining the members of SYS1.GENLIB can be included during the initialization run instead of executing IEHPROGM as a separate job.

## Inserting Two Control Statements

Before executing Stage I of the system generation process, you must insert two additional control statements in the input deck.

For the limit of up to 248 devices, the statements are:

```
    COPY SGGBLPAK
    &LIMIT(1) SETB 1
```

For the limit of up to 768 devices, the statements are:

```
    COPY SGGBLPAK
    &LIMIT(3) SETB 1
```

These statements must immediately follow the DD \* or DD DATA statement and precede any system generation macro instructions.

## Reestablishing the Standard Definitions

After the system generation process is finished, you should reestablish the standard definitions by using the IEHPROGM utility program.

The following two control statements are required if the limit selected was up to 248 devices:

```
RENAME  DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,          X
        MEMBER=SGGBLPAK,NEWNAME=SGPAK248
RENAME  DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,          X
        MEMBER=SGPAK96,NEWNAME=SGGBLPAK
```

The following two statements are required if the limit selected was up to 768 devices:

```
RENAME  DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,          X
        MEMBER=SGGBLPAK,NEWNAME=SGPAK768
RENAME  DSNAME=SYS1.GENLIB,VOL=2311=DLIB04,          X
        MEMBER=SGPAK96,NEWNAME=SGGBLPAK
```

The following DD statement is always required in the step where the redefining is done:

```
//ddname DD DSNAME=SYS1.GENLIB,DISP=OLD
```

### 128K Users

If SGPAK768 is required for the Stage I assembly, the IPL procedure for the distributed MFT starter system should be modified as follows:

Reply YES to the CHANGE PARTITIONS message.

Reply (R 00,'P0=(ABC,LAST),END,LIST') to the CONTINUE DEFINITION message.

After a SET DATE command issue:

```
MN JOBNAMES
START DSO.P0,00E
START RDR.S,00C
START INIT.A22
```

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

## Appendix C: Distribution Methods

The starter operating system package is provided by one of the following distribution methods:

- Complete starter operating system package for the 2311 disk storage drive: This package is distributed on five disk packs, or two tapes with 1600 BPI, or three tapes with 800 BPI, for restoring to disk packs. The volume serial numbers of the disk packs are DLIB01, DLIB02, DLIB03, DLIB04 and DLIB05.

The volume serial number of the disk pack is DLIB06.

- Complete starter operating system package for the 2314 direct access storage facility: This package is distributed on one tape with 1600 BPI, or two tapes with 800 BPI, for restoring to two 2316 disk packs. The volume serial numbers of the disk packs are DLIB01 and DLIB02.
- Complete starter operating system package for the 3330 direct access storage facility: This package is distributed on one tape with 1600 BPI, or two tapes with 800 BPI, for restoring to one 3336 disk pack. The volume serial number of the pack is DLIB01.

The starter operating system package distributed on disk packs is operable (though a backup copy of it should be made). It contains the starter operating system and the libraries as they are shown in Figures 90 and 91. (The data sets are not necessarily provided in the order shown.)

The starter operating system package distributed on tape includes all of the contents of the disk packs. Because the tapes must be restored onto disk packs, two additional independent utility programs -- IBCDASDI and IBCDMPRS -- are provided at the beginning of each tape for use in restoring the disk packs. In Figures 90 and 91, the tape layout is shown beside the corresponding disk pack. Only the libraries shown on the disk packs are restored from the tapes. The IBCDASDI and IBCDMPRS programs will always be at the beginning of the tapes, but the other data sets may not be in the order shown. The 2314 and 3330 distribution are available only on tape, and must be restored to the disk pack from the tape.

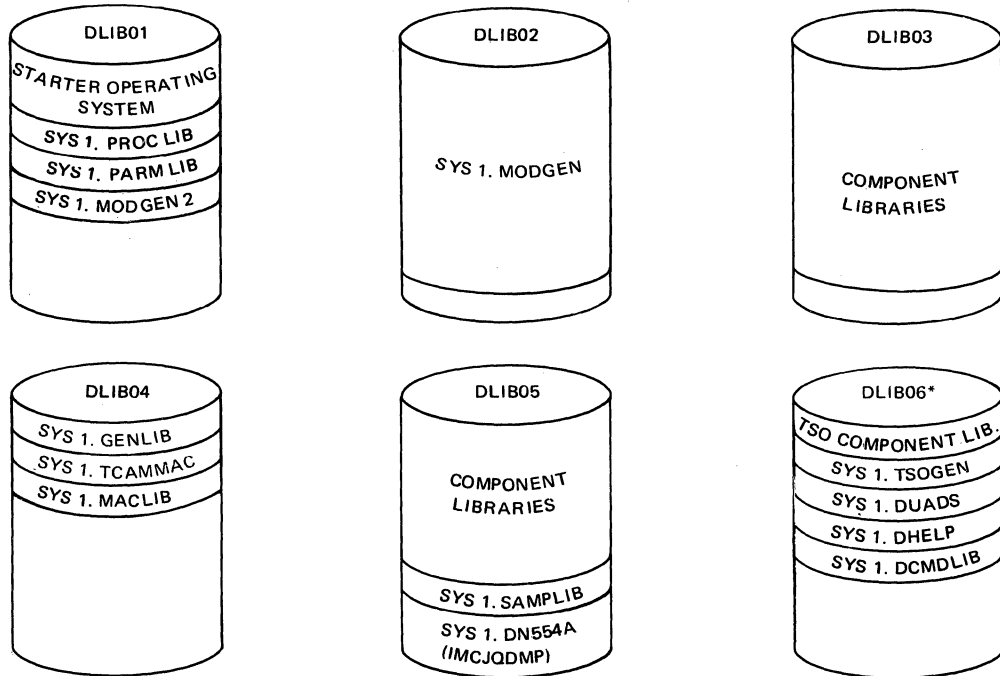
2311 DISTRIBUTION PACKS					
DLIB01	DLIB02	DLIB03	DLIB04	DLIB05	DLIB06
STARTER SYSTEM	SYS 1. MODGEN	SYS 1. DN554**	SYS 1. GENLIB	SYS 1. AL531	SYS 1. TSOGEN*
SYS 1. PARMLIB		SYS 1. CQ548*	SYS 1. TCAMMAC*	SYS 1. AS037	SYS 1. CI555*
SYS 1. PROCLIB		SYS 1. CI505	SYS 1. MACLIB	SYS 1. CB545	SYS 1. TSOMAC*
SYS 1. MODGEN2		SYS 1. CI535		SYS 1. COBLIB	SYS 1. DUADS*
		SYS 1. CQ513		SYS 1. CO503	SYS 1. DHELP*
		SYS 1. CQ519		SYS 1. ED521	SYS 1. DCMDLIB*
		SYS 1. DM508		SYS 1. FORTLIB	
		SYS 1. DM509		SYS 1. FO500	
		SYS 1. DN527		SYS 1. FO520	
		SYS 1. DN533		SYS 1. FO550	
		SYS 1. DN539		SYS 1. LD547	
		SYS 1. IO523		SYS 1. LM501	
		SYS 1. IO526		SYS 1. LM512	
		SYS 1. LM537		SYS 1. LM532	
		SYS 1. LM542		SYS 1. LM546	
		SYS 1. RC536		SYS 1. NL511	
		SYS 1. RC541		SYS 1. PL1LIB	
		SYS 1. RC543		SYS 1. PL552	
		SYS 1. RC551		SYS 1. RG038	
		SYS 1. UT506		SYS 1. SORTLIB	
				SYS 1. SAMPLIB**	
				SYS 1. DN554A	

2314 DISTRIBUTION PACKS		
DLIB01		DLIB02
STARTER SYSTEM	SYS 1. FO500	SYS 1. TSOGEN
SYS 1. TCAMMAC	SYS 1. FO520	SYS 1. GENLIB
SYS 1. TSOMAC	SYS 1. FO550	SYS 1. MODGEN
SYS 1. CQ548	SYS 1. IO523	SYS 1. MODGEN2
SYS 1. CI555	SYS 1. IO526	SYS 1. SAMPLIB
SYS 1. DUADS	SYS 1. LD547	SYS 1. DN554A
SYS 1. DHELP	SYS 1. LM501	
SYS 1. DCMDLIB	SYS 1. LM512	
SYS 1. DN554	SYS 1. LM532	
SYS 1. AL531	SYS 1. LM537	
SYS 1. AS037	SYS 1. LM542	
SYS 1. CB545	SYS 1. LM546	
SYS 1. CI505	SYS 1. NL511	
SYS 1. CI535	SYS 1. PL1LIB	
SYS 1. COBLIB	SYS 1. PL552	
SYS 1. CO503	SYS 1. RC536	
SYS 1. CQ513	SYS 1. RC541	
SYS 1. CQ519	SYS 1. RC543	
SYS 1. DM508	SYS 1. RC551	
SYS 1. DM509	SYS 1. SM023	
SYS 1. DN527	SYS 1. SORTLIB	
SYS 1. DN533	SYS 1. UT506	
SYS 1. DN539	SYS 1. PARMLIB	
SYS 1. ED521	SYS 1. PROCLIB	
SYS 1. FORTLIB		
SYS 1. MACLIB		

Figure 90 (Part 1 of 3). Arrangement of Data Sets

COMPLETE 2311 PACKAGE -- DISK DISTRIBUTION



\*DLIB06 -- Required if TSO is to be included in the new system.

COMPLETE 2314 PACKAGE -- DISK PACK CONTENTS AFTER RESTORING FROM TAPE

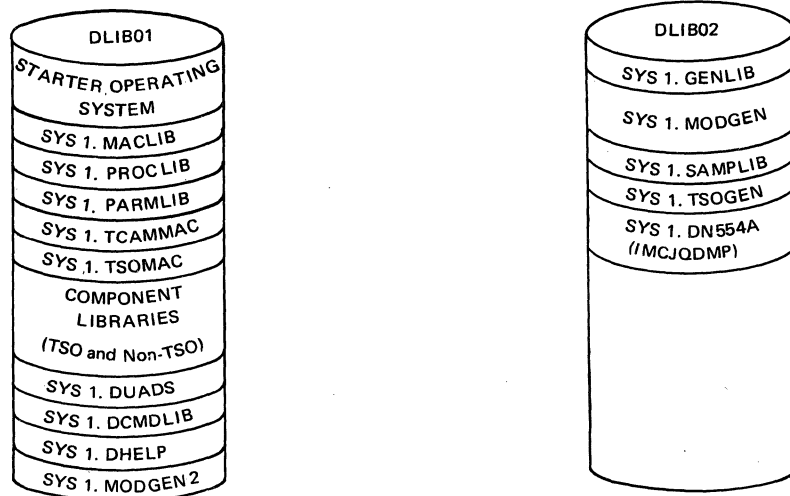


Figure 90 (Part 2 of 3). Arrangement of Data Sets

TAPE DISTRIBUTION FOR 2311 PACKAGE

Two tapes if 1600 BPI

Three tapes if 800 BPI

DASDI  
Dump/Restore  
T/M  
DLIB01  
Starter  
Operating  
System  
SYS 1. MODGEN 2  
SYS 1. PROCLIB  
SYS 1. PARMLIB  
T/M  
Trailer Record  
T/M  
DLIB02  
SYS 1. MODGEN  
T/M  
Trailer Record  
T/M  
DLIB03  
Component Libraries  
T/M  
Trailer Record  
T/M  
DLIB04  
SYS 1. GENLIB  
SYS 1. TCAMMAC  
SYS 1. MACLIB  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB05  
Component Libraries  
SYS 1. SAMPLIB  
SYS 1. DN554A  
(Service Aids)  
T/M  
Trailer Record  
T/M  
DLIB06  
All TSO  
Libraries  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB01  
Starter  
Operating  
System  
SYS 1. MODGEN 2  
SYS 1. PROCLIB  
SYS 1. PARMLIB  
T/M  
Trailer Record  
T/M  
DLIB02  
SYS 1. MODGEN  
T/M  
Trailer Record  
T/M  
DLIB03  
Component Libraries  
(First Half)  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB03  
Component Libraries  
(Second Half)  
T/M  
Trailer Record  
T/M  
DLIB04  
SYS 1. GENLIB  
SYS 1. TCAMMAC  
SYS 1. MACLIB  
T/M  
Trailer Record  
T/M  
Trailer Record  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB05  
Component Libraries  
SYS 1. SAMPLIB  
SYS 1. DN554A  
(Service Aids)  
T/M  
Trailer Record  
T/M  
DLIB06  
All TSO  
Libraries  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

TAPE DISTRIBUTION FOR 2314 PACKAGE

One tape if 1600 BPI.

Two tapes if 800 BPI

DASDI  
Dump/Restore  
T/M  
DLIB01  
Starter  
Operating  
System  
SYS 1. MACLIB  
SYS 1. PROCLIB  
SYS 1. PARMLIB  
SYS 1. TCAMMAC  
SYS 1. TSOMAC  
SYS 1. MODGEN 2  
Component Libraries  
T/M  
Trailer Record  
T/M  
DLIB02  
SYS 1. GENLIB  
SYS 1. MODGEN  
SYS 1. SAMPLIB  
SYS 1. TSOGEN  
SYS 1. DN554A  
(Service Aids)  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB01  
Starter  
Operating  
System  
SYS 1. MACLIB  
SYS 1. PROCLIB  
SYS 1. PARMLIB  
SYS 1. TCAMMAC  
SYS 1. TSOMAC  
SYS 1. MODGEN 2  
Component Libraries  
(First Half)  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

DASDI  
Dump/Restore  
T/M  
DLIB01  
Component Libraries  
(Second Half)  
T/M  
Trailer Record  
T/M  
DLIB02  
SYS 1. GENLIB  
SYS 1. MODGEN  
SYS 1. SAMPLIB  
SYS 1. TSOGEN  
SYS 1. DN554A  
(Service Aids)  
T/M  
Trailer Record  
T/M  
Trailer Label  
T/M

Figure 90 (Part 3 of 3). Arrangement of Data Sets



---

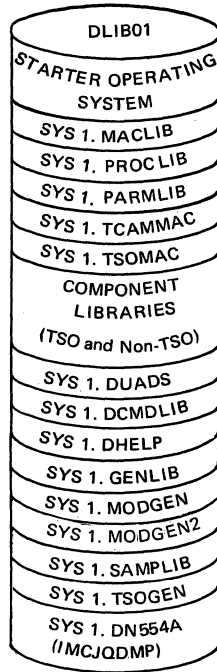
3330 DISTRIBUTION PACKS

DLIB01

STARTER SYSTEM	SYS 1. FO500	SYS 1. TSOGEN
SYS 1. TCAMMAC	SYS 1. FO520	SYS 1. GENLIB
SYS 1. TSOMAC	SYS 1. FO550	SYS 1. MODGEN
SYS 1. CQ548	SYS 1. IO523	SYS 1. MODGEN2
SYS 1. CI555	SYS 1. IO526	SYS 1. SAMPLIB
SYS 1. DUADS	SYS 1. LD547	SYS 1. DN554A
SYS 1. DHELP	SYS 1. LM501	
SYS 1. DCMDLIB	SYS 1. LM512	
SYS 1. DN554	SYS 1. LM525	
SYS 1. AL531	SYS 1. LM532	
SYS 1. AS037	SYS 1. LM537	
SYS 1. CB524	SYS 1. LM542	
SYS 1. CB545	SYS 1. LM546	
SYS 1. CI505	SYS 1. NL511	
SYS 1. CI535	SYS 1. PL1LIB	
SYS 1. COBLIB	SYS 1. PL552	
SYS 1. CO503	SYS 1. PT516	
SYS 1. CQ513	SYS 1. RC536	
SYS 1. CQ519	SYS 1. RC541	
SYS 1. DM508	SYS 1. RC543	
SYS 1. DM509	SYS 1. RC551	
SYS 1. DN527	SYS 1. RG038	
SYS 1. DN533	SYS 1. SM023	
SYS 1. DN539	SYS 1. SORTLIB	
SYS 1. ED521	SYS 1. UT506	
SYS 1. FORTLIB	SYS 1. PARMLIB	
SYS 1. MACLIB	SYS 1. PROCLIB	

---

Figure 91 (Part 1 of 2). Arrangement of Data Sets for the 3330



TAPE DISTRIBUTION FOR 3330 PACKAGE

DASDI  
 Dump/Restore  
 T/M  
 DLIB01  
 Starter  
 Operating  
 System  
 SYS 1. MACLIB  
 SYS 1. PROCLIB  
 SYS 1. PARMLIB  
 SYS 1. TCAMMAC  
 SYS 1. TSOMAC  
 Component Libraries  
 SYS 1. GENLIB  
 SYS 1. MODGEN  
 SYS 1. MODGEN2  
 SYS 1. SAMPLIB  
 SYS 1. TSOGEN  
 SYS 1. DN554A  
 (Service Aids)  
 T/M  
 Trailer Record  
 T/M  
 Trailer Label  
 T/M

One tape if 1600 BPI.

Two tapes if 800 BPI

DASDI  
 Dump/Restore  
 T/M  
 DLIB01  
 Starter  
 Operating  
 System  
 SYS 1. MACLIB  
 SYS 1. PROCLIB  
 SYS 1. PARMLIB  
 SYS 1. TCAMMAC  
 SYS 1. TSOMAC  
 Component Libraries  
 (First Half)  
 T/M  
 Trailer Record  
 T/M  
 Trailer Label  
 T/M

DASDI  
 Dump/Restore  
 T/M  
 DLIB01  
 Component Libraries  
 (Second Half)  
 SYS 1. GENLIB  
 SYS 1. MODGEN  
 SYS 1. MODGEN2  
 SYS 1. SAMPLIB  
 SYS 1. TSOGEN  
 SYS 1. DN554A  
 (Service Aids)  
 T/M  
 Trailer Record  
 T/M  
 Trailer Label  
 T/M

Figure 91 (Part 2 of 2). Arrangement of Data Sets for the 3330

## Appendix D: 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. Two types of messages are produced: error messages and informative messages.

### ERROR MESSAGES

Figure 92 shows the message code and format of system generation error messages. The messages follow.

IEIaaann text

Explanation: The error indicated by the message text is a coding error in the system generation macro instruction, aaa. The message serial number, nnn, identifies the message.

For the CHANNEL, IOCTRL, 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 these messages are:

5,\* \* \* IEICEN104 INSTSET VALUE NOT SPECIFIED

5,\* \* \* IEICHA102 CHANNEL2-ADDRESS VALUE NOT SPECIFIED

5,\* \* \* IEICHA102 CHAN#2-ADDRESS VALUE NOT SPECIFIED

The first example illustrates a message for the CENPROCS macro instruction.

The second example illustrates a message for a CHANNEL macro instruction. "CHANNEL2" is the name field of the macro instruction. The third example illustrates the same message, but in this case the name field of the macro instruction was omitted and "CHAN#2" was supplied by the macro instruction.

System Action: The assembler program did not produce a job stream in the SYSPUNCH data set. The program analyzed all remaining system generation macro instructions and printed any other required messages. Either messages IEIGEN113 or IEIGEN116 was printed, followed by the message: GENERATION TERMINATED. Then the system generation process was abnormally terminated.

Severity Code: 5

User Response: Correct the error or errors indicated and begin the system generation process from the start of Stage I.

IEIGEN113 QUIT SWITCH ON BEFORE GENERATE MACRO

Explanation: One or more errors, indicated by messages, were detected before the GENERATE macro instruction was expanded.

Severity Code: 7

IEIGEN116 QUIT SWITCH SET IN GENERATE MACRO

Explanation: One or more errors were detected during expansion of the GENERATE macro instruction.

Severity Code: 7

7,\* \* \* GENERATION TERMINATED \* \* \*

Explanation: The system generation process was abnormally terminated.

Severity Code: 7

WARNING MESSAGES

Figure 92 shows the message code and format of system generation warning messages. The messages follow.

IEIaaannn text

Explanation: The message text indicates a machine condition that may cause errors in the new system. For example:

0,\* \* \* IEIIOC114 PREVIOUS CONTROL UNIT HAS NO DEVICES

System Action: The job stream is produced.

Severity Code: 0

User Response: If necessary, correct the condition indicated and begin the system generation process from the start of Stage I.

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: This type of message will appear preceding any PUNCH statements that were produced by the macro expansion.

macro name

name of the macro instruction that produced any succeeding PUNCH statements.

component name

name of the component group responsible for maintaining the particular macro expansion.

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,\* \* \* IEIaaaannn text

s = Severity code:

- 0 Warning message; the condition indicated may cause errors in new system.
- 5 Error message; error in coding of a system generation macro instruction.
- 7 Error message; message is produced by GENERATE macro instruction.

aaa = Indication of system generation macro instruction at which error was detected;

<u>aaa</u>	<u>Macro Instruction</u>	<u>aaa</u>	<u>Macro Instruction</u>
AGL	ALGLIB	MAL	MACLIB
ALG	ALGOL	OUT	OUTPUT
ASM	ASSEMBLR	PLL	PL1LIB
CEN	CENPROCS	PL1	PL1
CHA	CHANNEL	POP	PTOP
CKP	CKPTREST	PRL	PROCLIB
CKR	CHECKER	PRM	PARMLIB
CMD	CMDLIB	PTM	PARTITNS
COB	COBOL	RES	RESMODS
COL	COBLIB	RPG	RPG
CTR	CTRLPROG	SCN	SECONSLE
DAT	DATAMGT	SEC	SECMODS
EDI	EDIT	SCH	SCHEDULR
EDT	EDITOR	SOL	SORTLIB
EMU	EMULATOR	SOR	SORTMERG
FOL	FORTLIB	SUP	SUPRSOR
FTC	FORTRAN	SVC	SVCTABLE
GEN	GENERATE or GENTSO	SVL	SVCLIB
GJC	GJOBCTL	SYS	YSUTILS
GPH	GRAPHICS	TEL	TELCLIB
HEL	HELP	TSO	TSOPTION
IMG	IMAGELIB	UAD	UADS
IOC	IOCONTRL	UCS	UCS
IOD	IODEVICE	UNI	UNITNAME
LDR	LOADER		
LNK	LINKLIB		

nn = Message serial number  
text = Message text

Figure 92. System Generation Error and Warning Messages



## Appendix E: Functionally Equivalent I/O Devices

---

I/O Device	Equivalent to	Usage
2319	2314	<p>The 2319 is a three drive disk storage unit that attaches to any S/360 or 370. The 2319 drives are functionally equivalent to the 2314 drives. As with the 2314, a maximum of eight drives can be used.</p> <p>When specifying a 2319, FEATURE= in the IODEVICE macro can be used for all models except the 145. To specify a 2319, see the IODEVICE macro.</p>
2596	1442	<p>The 2596 card read punch is functionally equivalent to the 1442 N1 card read punch. If the 2596 is specified as a 1442 N1 at system generation time, it becomes directly accessible by an assembler language program which enables it to read and punch 96-column cards. For information on the data management macros used to access a 2596, see <u>OS Data Management Services Guide</u>.</p> <p><u>Use:</u></p> <ul style="list-style-type: none"><li>• The device may only be used for direct input from or direct output to an assembler language program.</li><li>• The device may not be assigned as a job stream (SYSIN/SYSOUT) device.</li><li>• A data set on the device may not be opened as both an input and output data set at the same time.</li><li>• If both the 2596 and 1442 devices are being used in the same system, the device address cannot be included in a device class at system generation.</li><li>• If both the 2596 and the 1442 are being used in the same system, they must be assigned by their respective device addresses.</li><li>• If the system contains DDR, the 2596 cannot be replaced with the 1442 by the SWAP command.</li></ul>

---





Indexes to Systems Reference Library publications are consolidated in OS Master Index for System Reference Manuals, GC28-6644. For additional information about any subject listed below, refer to other publications listed for the same subject in the Master Index.

## A

- ABEND  
 by utility data sets 79  
 dump data set 217  
 standard system codes 261  
 (see also SYS1.DUMP)
- Access method  
 generation of 281,282  
 graphics programming services 325-326  
 optional 282  
 resident 439-440  
 (see also BDAM, ISAM, BTAM, QTAM)
- Accounting routine 24,98  
 example 33,116  
 to include 402
- Additional pairs of transient areas  
 description 157  
 specifying 276
- ALGLIB macro instruction 247  
 cross-reference table 467  
 specifying 245,246  
 type of (table) 244
- ALGOL compiler  
 component name 204  
 generation of 247,248  
 member names 247  
 sample program 98  
 to be included 204
- ALGOL library  
 contains 211  
 component name 204  
 to be included 211,245  
 (see also SYS1.ALGLIB)
- ALGOL macro instruction 247,248  
 cross-reference table 467  
 specifying 247-249  
 type of (table) 244
- ALGOL sample program 98
- Allocating space  
 (see space allocation)
- Alternative channel addressing  
 specifying 346-347
- Alternate console  
 description 163,164  
 specifying 397-401,403,425-430
- Alternate path retry (APR)  
 description 157,158  
 specifying 440
- American National Standard COBOL  
 component name 204,215  
 sample program 99  
 specifying 267-274  
 type of (table) 244  
 (see also COBOL macro instruction)
- American National Standard COBOL library  
 component name 204,215  
 to be included 265-266  
 (see also SYS1.COBLIB, COBLIB macro instruction, COBOL subroutines)
- APR  
 (see alternate path retry)
- ASMBLR  
 contained in 220  
 during Stage I 79,80  
 during Stage II 65,69,71,74-75  
 restart 84,86  
 user-written load module 24  
 (see also assembly steps, Stage II, restart procedures)
- ASCII 440
- Assembler language processor 166,167  
 alias 220  
 component names 204  
 contained in 221  
 decimal instructions 442  
 during Stage I 79,80  
 during Stage II 65,69,71,74-75  
 F-design level 32,204,252  
 generation of 251  
 sample program 100  
 sample steps, Stage II 69-70,74,76  
 system generation requirements 33,34  
 uses SYS1.MACLIB 224  
 uses utility data sets 28-31,32-33  
 (see also ASSEMBLR macro instruction)
- Assembler sample program 100
- ASSEMBLR macro instruction  
 specifying 251  
 type of (table) 244  
 (see also assembler language processor)
- Assembly steps  
 during Stage I 69,75,76  
 during Stage II 14,65,69,74  
 restart Stage I 79  
 restart Stage II 80,81  
 (see also ASMBLR)
- Asynchronous overlay 278
- ATTACH function 158-159  
 description 159  
 made resident description 158-159  
 specifying 440,444
- ATTACH macro instruction  
 (see SYS1.LINKLIB)
- Automatic restart 261
- Automatic volume recognition 413,414
- Auxiliary storage 10
- AVR 413,414

**B**

Background jobs 320,321,322  
 Backup copy 40,42  
   disk 52-55  
   examples 129  
   tape 42,48  
 Basic direct access method  
   (see BDAM)  
 Basic indexed sequential access method  
   (see BISAM, ISAM)  
 BDAM  
   component name 204  
   description of option 160,161  
   to be included 282  
 BISAM 282  
   (see also ISAM)  
 BLDL list 227  
 BLDL table made resident option  
   description 160,161  
   specifying 441  
 BSC1 telecommunications line  
   description 340  
   specifying 338,343,344,351,352  
 BSC2 telecommunications line  
   description 339,340  
   specifying 338,343,344,351,352  
 BSC3 telecommunications line  
   description 339,340  
   specifying 338,343,344,351,352  
 BTAM  
   component name 204  
   option description 186,187  
   required 322  
   specifying 281,361,362  
 Buffers  
   COBOL 269  
   FORTRAN 303-304  
   linkage editor 293-294  
   sections in 2840 346  
   write-to-log (WTL) 413-414  
   write-to-operator (WTO) 413-414  
 Burst devices  
   use of on channels 257,337,338  
   list of 337

**C**

Cards  
   restarting from 81,82  
   sample programs 98  
 Catalog 10  
   during restart 94  
   examples 10-15,126-127,130,132,140,145  
   listing 69,71,74,75  
   user-written routines 449,450  
   (see also system catalog, cataloged  
   procedures)  
 Cataloged procedures  
   for FORTRAN 295,301  
   for MVT or M65MP 251,267,291,301  
   group names for 39,463-465  
   IBM supplied 229

statements 11  
 user-written 229  
   (see also SYS1.PROCLIB)  
 Cataloged procedures library  
   (see SYS1.PROCLIB)  
 CATLG STATEMENT  
   during initialization 8,11-15,31  
   during reallocation 90,91,92,93-94  
 CENPROCS macro instruction  
   cross-reference table 468  
   specifying 253-256  
   type of (table) 244  
 Central processing unit 302,421-423,446  
   for starter system 34-35  
   specifying 253-256  
 Channel  
   (see selector channel, multiplexor  
   channel)  
 CHANNEL macro instruction  
   specifying 257-258  
   type of (table) 244  
 Channel programs 277  
 Channel-check handler 161,440  
 Character set  
   for ALGOL 248,249  
   for COBOL 267  
   for FORTRAN 304  
   for PL/I 371  
   for starter operating system 34  
   for 1403 343,344  
   for 1443 343,344  
   images 461,462  
   UCS macro instruction 461-462  
 CHECKER macro instruction  
   cross-reference table 467  
   specifying 260  
   type of (table) 244  
 Checkpoint/restart  
   cataloged procedure (IEFREINT) 163  
   description 162-163  
   specifying 261,443-445  
 CIRB macro instruction  
   requirements 160-161,175,186,281,361,  
   362  
 CKPTREST macro instruction 371  
   cross-reference table 467,469  
   specifying 261-262  
   type of (table) 244  
 CMDIB macro instruction  
   cross-reference table 468,469  
   specifying 263  
   type of (table) 244  
   (see also time sharing option)  
 COBLIB macro instruction 325,326  
   cross-reference table 467  
   specifying 265-266  
   type of (table) 244  
 COBOL compiler  
   component names 204  
   E-design level 204  
   generation of 267-274  
   sample display 110  
   sample program 109  
   in new system 204  
   sample programs 98,101  
 COBOL E sample program 101

COBOL library 203  
   (see also SYS1.COBLIB, COBLIB macro instruction, COBOL subroutines)  
 COBOL macro instruction 265-266  
   cross-reference table 467  
   specifying 267-268  
   type of (table) 244  
 COBOL subroutines  
   (see SYS1.COBLIB)  
 COBOL, American National Standard  
   (see American National Standard COBOL)  
 Coding conventions 157  
   for macro instructions 239-240  
 Command code groups 165  
 Complete operating system generation  
   description 1-3  
   definition 1-3  
   examples 3,123-130,136-148,149-153  
   macro instructions 239-472  
   preparations 7-59  
 Component libraries 204-206  
   catalog 26-30  
   description 203  
   example 130,132  
   mounting 15  
   in starter operating system 204-206  
 Components  
   list of 204-205  
   (see also component libraries)  
 Composite console  
   description 163-164  
   specifying 398-401,403-404,425  
 Configuration  
   (see machine configuration)  
 Console 418-419,429-430  
   alternate 163-164,397,425-426  
   composite 163-164,398-401,403,405,419-420  
   examples 419-420,429-430  
   master 165-166  
   multiple consoles option 1,164-167,398-401,402,425  
   primary 164  
   secondary 425-426  
   specifying 397-401,403-405,425-426  
 Contiguous data sets  
   with reallocating 92-94  
 Contiguous partitions 279  
 Control program  
   component name 204  
   in complete generation 2  
   in nucleus generation 2,275  
   resident portion 444-445  
   (see also SYS1.NUCLEUS)  
   starter operating system 30,34-35  
   to be included 275-280  
   type of 2  
   (see also MFT, MVT, PCP, Model 65 multiprocessing)  
 Control unit  
   specifying 331-336  
   generating additional 476-480  
   1051 334  
   2403 334,343,344,345,346,350  
   2404 334,337,350  
   2415 334,350  
   2701 335,341,345  
   2702 335,341,345,353  
   2703 335,341,345,353  
   2803 333-336,347  
   2804 333-336  
   2820 333-336  
   2821 333-336  
   2822 333-336  
   2840 333-336,346  
   2841 333-336,347  
   2848 333-336,343  
   3272 333,334,336  
   3803 333,336  
   3811 333  
 Conversational remote job entry (CRJE) 168-169  
 Core image dump 217  
 CPU  
   (see central processing unit)  
 CRJE 168-169  
 Cross-references in macro instructions 244,467-472  
 CTLG 2311  
   description 35  
   included in 35  
   use of 29-30  
 CTLG 2314  
   description 35  
   included in 35  
   use of 29-30  
 CTRLPROG macro instruction  
   cross-reference table 468  
   specifying 275-280  
   type of (table) 244  
 Current date 10  
 CVOL parameter  
   examples 14

**D**

Data conversion  
   feature 333-336  
   sample program 103  
 Data management routines  
   included in 232  
   macro instruction 281-282  
   primary, component name 204  
   (see also SYS1.SVCLIB, access methods)  
 DATAMGT macro instruction  
   cross-reference table 468  
   specifying 281-282  
   type of (table) 244  
 Data sets 197  
 Data set protection 10  
 Data set utilities  
   component name 204  
 Data transmission  
   processor to processor 391  
   sample program 103  
 DCB macro instruction 443

DCB parameter 208  
 DD statement, initialization 10  
 new system data sets 207

DCMLIB 283

DDR  
 (see dynamic device reallocation)

DD statement  
 DUMMY 316  
 example user-written module 24  
 format, initialization 10  
 for IEBPTPCH 81  
 for IEHLIST 27,28,86  
 for IEHPROGM 32  
 for punching job stream 81  
 for reallocating 88,94  
 for restarting Stage I 80  
 for SYS1.LOGREC 86  
 for utility data sets 20-22,30  
 input deck, initialization 8,9,10  
 input deck organization 20-22  
 sample programs 11-15  
 sample of Stage II 69-71,75-76

Decimal simulation for Model 91  
 description 170  
 specifying 253,254,442

Descriptor codes 165-167

Device type  
 for cataloged procedures 11  
 for initializing 9,10  
 generating additional 475-481  
 in macro instructions 245,266,298,  
 313-315,389,433,455,463-465  
 use of 337

Direct access  
 control units 333-336  
 I/O device 337-353  
 examples of allocation 15-20  
 examples of initializing 12-13  
 initializing generating system 7-8  
 initializing new system 8  
 mounting requirements 15-16,28  
 sample sort/merge program 117  
 unit names 463-464

Direct access volume serial number  
 description 170  
 specifying 442

Direct system output facility (DSO)  
 use of 25

Dump, core image  
 (see core image dump)

DISP parameter  
 DD statement, initialization 8

Display control module library  
 description of 216  
 specifying of 283

Distribution libraries 32

DRISAMP  
 (see IEHDASDR sample program)

DSNAME parameter  
 DD statement initialization 9,10  
 CATLG statement initialization 11

DSO  
 (see direct system output facility)

Dynamic device reallocation (DDR)  
 description 171,172  
 specifying 441

## E

EDIT macro instruction  
 cross-reference table 468  
 specifying 285-289  
 type of (table) 244

EDITOR macro instruction 435  
 cross-reference table 468  
 specifying 291-292  
 type of (table) 244

Emulator  
 for 709/7090/7094/7094II 294  
 specifying 293  
 where to find information 294

EMULATOR macro instruction 412  
 cross-reference table 468  
 specifying 293-294  
 type of (table) 244

Enqueue/dequeue function 444

Error messages  
 diagnostic, PL/I 374-376,377-378  
 during Stage I 1,22,65,79  
 for COBOL 271  
 for FORTRAN 298  
 for linkage editor 313,314  
 (see also messages)

Error recovery routines  
 included in 232  
 reduced procedure 443

RERP list 227  
 system 445-446  
 (see also SYS1.SVCLIB)

Error routines, I/O device 343

Examples  
 adding CSECT to nucleus 24  
 ASMBLR 20-21,127-128,131,132,141-144  
 backup of new system 43-44,53-55,  
 56-57  
 complete generation 3,123-131  
 creating restore tapes 129  
 deleting utility data sets 129  
 IBCDASDI 8,136,137,144  
 IBCDMPRS 136  
 IEBEDIT 81-84  
 IEBPTPCH 81-82,137  
 IEHDASDR 8,129,137  
 IEHLIST 137  
 IEHPROGM 11-15,69,76,79-80,88-94,  
 126-127,129,140-141  
 initializing new system data set 10-13,  
 139  
 initializing new system volumes 138,139  
 initializing starter system 136-137  
 initializing system residence volume  
 137,139,144  
 I/O device generation 144-147  
 machine configuration 124,134

## Examples (continued)

- MFT 133-148
- MVT 149-153
- nucleus generation 131-132
- obtaining generating system data sets 130
- of data set allocation 15-19
- of system generation 123-153
- processor/library generation 130-131
- punching job stream 81-82
- reallocation on same space 90-92
- reallocation with more space 91-94
- restart Stage I 79
- restarting with job stream on tape 82-83
- Stage I input deck 20,127,132,141-144
- steps in job stream 69-70,76,77
- user-written load module 24

EXEC statement

- during reallocation 88-94
- during Stage I 65,71
- during Stage II 65,71
- restart assembly 84-86
- sample program use 100,103,108,109, 112,118-126

Expiration date 10

Exponentiation 267,268

EXTRACT function 445

EXTRACT function made resident

- description 173
- specifying 445

## F

Fetch  
(see program fetch)

Floating point 267,268

Foreground jobs 320,321,322

FORTLIB macro instruction 325-326,391,392

- cross-reference table 468
- specifying 295-300
- type of (table) 244

FORTTRAN compiler

- component names 204
- error message facility 298
- G-design level 204,295,301-305
- generation of 301-305
- graphic subroutine package 300
- sample display
- sample program 108-110
- H-design level 204,295,301-305
- sample programs 104
- syntax checker
  - component name 204
  - specifying 259-260
  - use of 170,259,305,378-379

(see also SYS1.FORTLIB, FORTRAN macro instruction, FORTLIB macro instruction, FORTRAN library, FORTRAN subprograms, graphic subroutines package)

FORTTRAN macro instruction 259

- cross-reference table 468
- specifying 301-305
- type of (table) 244

Four-drive generation 17,19

## G

GENERATE macro instruction 10,65,68,69, 389,435

- exception 244
- specifying 307-316
- type of (table) 244
- use of for MVT 25
- utility data sets 21-22

Generating operating system

- data sets required 197
- space allocation for utility data sets 14-16

Generation library  
(see SYS1.GENLIB, SYS1.MODGEN)

GENTSO macro instruction

- cross-reference table 471
- specifying 317
- type of (table) 244

GENTYPE PARAMETER (GENERATE) 244

GETMAIN/FREEMAIN modules 444

GJOBCTL macro instruction 319,325,326

- cross-reference table 468
- specifying 319-323
- type of (table) 244

GJP  
(see graphic job processor)

Graphic job processor

- component name 204
- to be included 319,322

Graphic programming services

- component name 205
- description 173-175
- specifying 173
- to be excluded 326,362
- to be included 325

Graphic subroutine package

- component name 205
- to be included 325
- sample programs for FORTRAN IV, COBOL, PL/I 108-110

Graphic 2250 sample program 105-107

Graphic 2260 sample program 105-107

GRAPHICS macro instruction 265,266,300,319,351

- cross-reference table 469
- specifying 325,326
- type of (table) 244

Group name

- definition 242
- UNITNAME macro instruction 463-465

GSP  
(see graphic subroutine package)

GPSAMP  
(see graphic subroutine sample program)

## H

Hard copy log  
description 165  
devices can be used 419,420  
specifying 399-402,405-406,419,420  
HELP macro instruction  
specifying 327  
type of (table) 244  
Hierarchy  
(see storage hierarchies, main-storage  
hierarchy support)  
High speed multiplexor channel 257

## I

IBCDASDI utility program 136,137,139,144  
example of use  
included in 33  
used for initialization 7-8  
use of 43,53,56-57  
IBCDMPRS utility program  
included in 33,483  
obtaining  
use of 43,48,54,483  
IBCRVRP utility program  
included 33  
IDENTIFY function 442  
IDENTIFY function made resident  
description 174,175  
specifying 442  
IEAIPL00 utility program 7-8  
included in 7,33  
(see also IPL)  
IEBCOPY utility program  
during Stage II 68,71  
performance considerations 206  
required 220  
restarting 84-86  
storage requirements 453  
use of 14-15,65,84,85,88,92,361,362  
IEBEDIT utility program  
use of 84  
use restart Stage II 82  
IEBPTPCH utility program  
use during restart 82  
use of 45,48,52,81,82  
IEBUPDAT utility program 23,222  
IEBUPDTE utility program 23,220  
IEEVMPCR  
(see mount command -- cataloged  
procedure)  
IEFREINT  
(see checkpoint/restart facility)  
IEHDASDR sample program 111  
IEHDASDR utility program  
used for initialization 7-8  
use of 45  
IEHIOSUP step  
restarting 86  
sample step 65,71  
IEHIOSUP utility program 65,71  
included in 220  
requirements 86,88,220  
restart 86,88  
use of 68,71  
IEHLIST utility program  
example of listing data sets 137  
included in 220  
restarting 86  
use of 20,46,55,59,65,71  
IEHPRGM utility program  
during Stage II 65,71  
example of initializing 126-127,132,  
142-144  
example of scratching data sets 130  
included in 220  
restarting 86  
used for initialization 3  
used to catalog 9  
use of 9,69,75,79,80,86,480-481  
use to reallocate 119  
IEJESP  
(see FORTRAN E sample program)  
IEMSP2  
(see PL/I sample program)  
IEPSAMP  
(see COBOL E sample program)  
IERSP  
(see sort/merge sample program)  
IEUESP  
(see assembler sample program)  
IEWO 84  
steps 65,71  
use of 24,25,68  
(see also link edit steps)  
IEXSAMP  
(see ALGOL sample program)  
IEHYS  
(see FORTRAN G and H sample program)  
IFCDIP00 program  
included in 220  
restarting 86  
sample step Stage II 68,74  
IFCEREPO (EREP) program  
included in 220  
IHGSAMP  
(see update analysis program)  
IKDSAMPL sample program  
(see data transmission, data  
conversion)  
IMAGELIB 244,327  
Independent utility programs  
included in 33  
punching 40  
use of 7-8  
INDEX keyword parameter (GENERATE macro  
instruction)  
example 31  
use of 30-31  
Initialization  
description 7-14  
examples, input decks 8-14  
new system volumes 7-8  
of SYS1.LOGREC 9

Initialization (continued)  
 system data sets 8,9,10  
 (see also examples)  
 Initial program loader (see IPL)  
 Initiator 412  
 Input deck  
 CATLG statements, initialization 10,11  
 DD statements, initialization 10-11  
 errors 79  
 example of backup 129  
 example of cataloging 126-127  
 example of generation 126-129,141-144,147-149  
 example of initializing 126-127,132,140-141  
 example of scratching data sets 130  
 for generation 20-22  
 for including IEBUPDAT 23  
 for initializing system residence  
 volume 7-8  
 initializing system data sets 8,10-11  
 organization 20-21  
 sample programs 93-120  
 Instruction set  
 commercial 267  
 scientific 247,301  
 specified 254-255  
 universal 247,267,301,371  
 Integrated operator console  
 (see Model 85, 5450)  
 Interruption  
 machine 80  
 IOCONTROL macro instruction  
 additional devices 479  
 cross-reference table 469  
 specifying 331-336  
 type of (table) 244  
 IODEVICE macro instruction  
 additional devices 475-480  
 cross-reference table 469  
 specifying 337-353  
 type of (table) 244  
 IPL 10,11,279,405,407,408,410,441,444,445  
 how to obtain 7  
 included in 33  
 key 98-120  
 ISAM  
 component name 205  
 description of option 175  
 to be included 281  
 I/O device generation 418  
 description 3  
 examples 2,3,145-147  
 macro instructions used 244  
 restarting link edit steps 84  
 requirements 253,257,275,281,319,325,  
 331,337,347,421,425,439,451,453,461,  
 463  
 I/O recovery management  
 (see alternate path retry, dynamic  
 device reallocation, error recovery  
 routines)

## J

JCL  
 created for Stage II 20  
 required 20  
 purpose 20  
 sample 12-15,20-21  
 use of 8,9-11,100  
 JOB card  
 restart assembly 84,85  
 RESTART parameter 261  
 use of, restart 81  
 user created for Stage II 309-310  
 Job control language (see JCL)  
 Job scheduler  
 options 397-420  
 (see also SYS1.SYSJOBQE, SCHEDULR macro  
 instruction)  
 JOB statement  
 for Stage II 312  
 Job-step timing  
 description 176  
 specifying 446  
 Job stream  
 data about 21-22  
 description 65-76  
 during restart 79-94  
 generation of 1-2,20-22  
 problems 79-82  
 processing 1-2  
 punching 21-22  
 sample steps 69-71  
 tape to cards 82  
 Job termination 403,407  
 Job step CPU timing function 402,446

## L

LABEL parameter  
 DD statement, initialization 9,10  
 DD statement, input deck 20-22  
 Label processing 57  
 Labels  
 no labels 307  
 non-standard 24,449  
 standard 307  
 Language processors  
 example 2-3  
 included in 2,308  
 Libraries  
 (see operating system libraries)  
 Linkage editor  
 alias 291-292,301  
 contained in 220  
 component names 205  
 during Stage II 2,84,87

- F-design level 205,251,267,291,305,445
- input to 248,378,379
- performance considerations 206
- print options 312
- sample steps 65,71
- size required 313
- to be included 292-293
- use of 2,65,71,251,267,291,295,435
- uses utility data sets 28,32,307,316  
(see also link edit steps)
- Link edit steps
  - job stream -- Stage II 65-76
  - non-TSO generation 65-71
  - restarting 84,87
  - TSO generation 71-76
  - non-TSO 84
- Link library  
(see SYS1.LINKLIB)
- Link library list  
(see SYS1.PARMLIB)
- LINK macro instruction  
uses SYS1.LINKLIB 220
- Link pack area 179
- LINKEDIT
  - alias 291
- LINKLIB macro instruction
  - specifying 355
  - type of (table) 244
  - user-written modules 23-24
- LNKLST00  
(see SYS1.PARMLIB)
- LOADER macro instruction
  - cross-reference table 469
  - specifying 357-359
  - type of (table) 244
- Loader program
  - component name 205
  - macro instruction 357-359
  - notes about coding 359
  - planning for 359
  - specifying 357-359
  - test program 112
- LOAD macro instruction  
(see SYS1.LINKLIB)
- Log option
  - description 165-168,405
  - specifying 405-406,410

## M

- Machine-check handler (MCH)
  - part of SER 445
  - (see also SYS1.ASRLIB)
- Machine-check handler library  
(see SYS1.ASRLIB)
- Machine configuration
  - considered in 1,14
  - examples 123-125,133-135
  - macro instructions 244
  - M65MP 255
  - starter system requirements 34
- Machine interrupt 80
- MACLIB macro instruction  
specifying 361,362

- type of (table) 244
- Macro instructions 237-472
  - coding conventions 239
  - cross-references 467-472
  - description 240
  - table of 244
  - unit name in 242
  - (see also individual macro instructions)
- Macro library (see SYS1.MACLIB)
- Magnetic tape
  - control units 332-336
  - devices 337-353
  - nonstandard label routines 24
  - restarting from 81-83
  - standard labels 307
  - use of 21,82,270
  - user label sample program 120
  - (see also 9-track magnetic tape)
- Main storage hierarchy support
  - description 176-177
  - specifying 275-277
- Master scheduler  
uses SYS1.PARMLIB 227
- MCH  
(see machine-check handler)
- MCS  
(see multiple consoles support)
- Messages
  - CCH 162
  - requirements 221,232,275
  - specifying 253-255
  - sample programs 100
  - (see also error messages)
- MFT
  - considerations 25
  - log data sets 234,414-415
  - minimum partition 408
  - job scheduler for 397-398
  - options 397-398,440-444
  - partitions  
(see PARTTNS macro instruction)
  - requirements 244,367-370,408,448
  - task supervisor options 440-444
  - time-slicing 279-280
  - type of generation 2,3
  - with SMF 225
- Model 30
  - requirements 253
  - restrictions 253-254
  - specifying 253,255-256
- Model 40
  - requirements 253
  - restrictions 253-254
  - special storage 255
  - specifying 253,255-256
- Model 50
  - special storage 255
  - specifying 253,255
- Model 65 multiprocessing
  - component name 205
  - job scheduler for 401-402
  - log data sets 234
  - machine-check handler (MCH) 445-446
  - options 401-402
  - requirements 208,244,255,440-446
  - restrictions 255,277,336



Model 65 multiprocessing (continued)  
 task supervisor options 440-444  
 to be included 276,279-280  
 types of generation 2,3

Model 67  
 in 65 mode 37

Model 75  
 specifying 253

Model 85  
 channel-check handler (CCH) 441  
 machine-check handler (MCH) 445-446  
 specifying 253  
 system error recovery (SER) 445-446  
 with starter operating system 36  
 5450 326-339

Model 91  
 instruction set 253-254  
 restrictions 255,275,421  
 specifying 253  
 with COBOL 265,268  
 with FORTRAN 297  
 with starter operating system 34,35

Model 135  
 CCH 162  
 requirements 221,232,275  
 specifying 263-265

Model 145  
 CCH 162  
 requirements 232,275  
 specifying 253-255

Model 155  
 channel-check handler (CCH) 441  
 data sets 212,220,232  
 devices, additional 482  
 emulators 293  
 I/O device 3210 338,353,416-418,430  
 I/O device 3215 338,353,416-418,430  
 machine-check handler (MCH) 445-446  
 restrictions 275,277,421  
 specifying 253-255

Model 165  
 channel-check handler (CCH) 441  
 data sets 212,220,232  
 devices, additional 482  
 emulators 293  
 I/O device 3066 338,339,353,417,419,  
 430,480  
 machine-check handler (MCH) 445-446  
 specifying 253-255  
 start operating system 35-39  
 restrictions 275,277,421

Model 195  
 channel-check handler (CCH) 441  
 instruction set 255  
 requirements 253-255  
 restrictions 255,275  
 system error recovery (SER) 445  
 specifying 253-255  
 with COBOL 265  
 with starter operating system 34,35

MOUNT command  
 during restart 82

Multiple consoles support (MCS)  
 alternate console 165-168  
 buffers 414  
 command code groups 165

component name 205  
 composite console 398-402,403-405  
 description 165-168  
 devices can be used 416-419,430  
 placed in 205  
 SECONSLE macro instruction 165-168,  
 425-430  
 specifying 398-401,402-405,425-430  
 system log 165-166,405  
 system log data sets 234

Multiple extents 220

Multiplexor channel 257,334  
 restriction 257  
 specifying 257

Multiprogramming fixed number of tasks  
 (see MFT)

Multiprogramming variable number of tasks  
 (see MVT)

MVT  
 component name 205  
 considerations for 25  
 job scheduler for 399-401  
 log data sets 234,410  
 minimum region 408  
 options 399,440-444,461  
 requirements 244,291,301,448  
 rollout/rollin 277  
 simple generation 133-147  
 task supervisor options 440-444  
 time sharing 187  
 time slicing 279  
 type of generation 2,3

M65MP (see Model 65 multiprocessing)

## N

NAME keyword  
 UNITNAME macro instruction 242,463-465

New operating system  
 catalog data sets 28-32  
 data sets for 207-208  
 definition 1,3  
 examples of allocation 15-20  
 examples of initializing 10-15  
 initializing data sets 7-13  
 libraries constructed 2  
 types of 2-3

New system (see new operating system)

NIP  
 uses SYS1.PARMLIB 227

Nonstandard label routines 23-24,450

NOPATCH 442,448

Nucleus 275,443-445  
 member name 312  
 user/written programs 23-24  
 (see also SYS1.NUCLEUS)

Nucleus generation  
 definition 2  
 examples 4  
 macro instructions for 244

Nucleus initialization program (see NIP)

Nucleus library (see SYS1.NUCLEUS)

Nucleus modules, refreshable  
 (see SYS1.ASRLIB)

Null allocation  
for SYS1.PARMLIB 365  
for SYS1.PROCLIB 389

## O

Object modules  
from Stage II 65,71,314  
OBJPDS 65,74  
reallocation 88  
restart DD statement 65,84,87  
OBJPDS keyword (GENERATE macro  
instruction) 314  
OCR (see optical character reader, 1285,  
1287, 1288)  
OIU (optical image unit)  
specifying 343,350  
Online test executive program (OLTEP)  
buffers 415,442  
component name 205  
Operator  
action 10,15-21  
communications facility 441  
Optical character reader  
error routines made resident 232,443  
excluding 362  
specifying 338-339,348,444  
(see also 1285,1287,1288)  
Optical image unit (see OIU)  
Optimize  
COBOL 270  
FORTRAN 303  
PL/I 376-377  
OS/360 Loader (see Loader program)  
OUTPUT macro instruction  
cross-reference table 469  
specifying 363  
type of (table) 244  
Overlay supervisor 278

## P

Parameter library  
(see SYS1.PARMLIB)  
PARMLIB macro instruction  
specifying 365  
type of (table) 244  
PARTITNS macro instruction 276-277  
cross-reference table 468-469  
specifying 370  
type of (table) 244  
PATCH (NO PATCH) 442  
PCI (see program controlled interrupt)  
PL1LIB macro instruction 325,371,439  
cross-reference table 469,471  
specifying 381-387  
type of (table) 244  
PL1 macro instruction 381  
cross-reference table 467,469,472

specifying 371-379  
type of (table) 244  
PL1SAMP (see graphic subroutine package --  
sample programs)  
PL/I compiler 442  
automatic restart 261  
component name 205  
example 4  
generator of 371-379  
graphic subroutine package (GSP) 325  
optimizing 375-376  
graphic subroutine package (GSP) 325  
optimizing 375-376  
sample display 110  
sample program 108-110,113  
syntax checker  
component name 205  
specifying 259-260  
use of 168,169,259,305,378  
to be included 205,371  
(see also PL/I library, SYS1.PL1LIB, PL1  
macro instruction, PL1LIB macro  
instruction)  
PL/I F sample program 113  
PL/I library  
component name 205  
shared library  
description 177  
feature 383,387,469  
specifying 381-387  
(see also SYS1.PL1LIB, PL/I compiler,  
PL/I library, PL1 macro instruction,  
PL1LIB macro instruction)  
PL/I shared library feature (see PL/1  
library)  
POST module 443  
PRESRES list  
included in 227  
use of for MVT 25  
(see also SYS1.PARMLIB)  
Primary console  
description 163-164  
specifying 405  
Print output class 322  
Priority number 321  
Procedure library (cataloged) (see  
SYS1.PROCLIB)  
Processing programs  
macro instructions 244  
Processors 3-4  
Generation requirements 204  
(see also individual processors)  
Processor to processor (see PTOP macro  
instruction)  
Processor/library generation 3-4  
definition 3  
example 130-131  
macro instructions for 244  
PROCLIB macro instruction  
cross-reference table 469  
specifying 389,390  
type of (table) 244

Program controlled interrupt (PCI) option  
     description 178  
     specifying 276  
 Program options descriptions  
     additional pairs of transient areas 157  
     alternate path retry (APR) 157  
     ATTACH function 158  
     ATTACH function made resident 159  
     basic direct access method (BDAM) 160  
     BLDL table made resident 160  
     channel-check handler (CCH) 162  
     checkpoint/restart facility 162  
     consoles -- alternate and composite  
         consoles options 163-164  
     consoles -- multiple consoles support  
         (MCS) 164-168  
     conversational remote job entry (CRJE)  
         facility 168,169  
     decimal simulation option for  
         model 91 170  
     direct-access volume serial number  
         verification 170  
     dynamic device reconfiguration  
         (DDR) 171,172  
     EXTRACT function made resident 173  
     graphic programming services 174  
     IDENTIFY function made resident 174,175  
     indexed sequential access method  
         (ISAM) 175  
     job queue made resident 175  
     job step timing 176  
     main storage hierarchy support 176  
     multiple wait option 177  
     PL/I F shared library feature 177  
     program controlled interrupt (PCI) 178  
     reenterable load modules made  
         resident 179  
     remote job entry (RJE) facility 180  
     resident access method routines 180  
     rollout/rollin option 182  
     shared DASD 182  
     SPIE routines made resident 183  
     storage protection 183  
     system management facilities (SMF) 184,185  
     telecommunications option 186  
     timing options 186,187  
     time sharing option (TSO) 188  
     time-slicing facility 190  
     trace option 191  
     transient SVC table made resident 192  
     type 3 and 4 SVC routines made  
         resident 193  
     user-added SVC routines 193  
     validity check option 194  
     volume statistics facility 195,196  
 Protected data set 10  
 Protection function  
     central processing unit 253  
     programming 443  
 PTOP macro instruction 322  
     cross-reference table 468,469  
     specifying 391,392  
     type of (table) 244  
 PTOP sample program 103

Punch output  
     cards 65,71,81  
     classname 322  
     restarting 81

## Q

QISAM 282  
     (see also ISAM)  
 QTAM  
     component name 205  
     description 186  
     to be excluded 361,362  
     to be included 281  
 Queued indexed sequential access method  
     (see QISAM, ISAM)  
 Queued telecommunications access method  
     (see QTAM)  
 Queueing I/O requests 345

## R

RAM list  
     included in 227  
     (see also SYS1.PARMLIB)  
 RD parameter  
     use with checkpoint/restart 162,261,371  
 Reader  
     sample program use 98,120  
 Reallocation  
     OBJPDS 88  
     on same volume 89-91  
     on the same space 89-91  
     SYSUT1, SYSUT2, SYSUT3 89-90  
     SYS1.SYSJOBQE 94  
     with more space 91-93  
 Recovery management support (RMS) 205  
     (see also Model 85)  
 Reenterable load modules made resident  
     description 179  
     specifying 445  
 Region  
     minimum 408  
     size 321  
 Remote job entry  
     component name 205  
     description 180  
     requirements 281  
     specifying 410  
 Reply queue elements  
     number for write-to-operator (WTOR) 410  
 Report program generator (see RPG)  
 Requirements  
     starter system 34-35,37  
 Resident access methods  
     description 181  
     specifying 443-445  
 Resident supervisor functions 443-445  
 RESMODS macro instruction 451  
     cross-reference table 469,471  
     specifying 393,394  
     type of (table) 244

Restart  
 ASMBLR steps 84,86  
 automatic 261  
 from cards 81  
 from tape 81-83  
 IEBCOPY step 84-85,88  
 IEHIOSUP step 86,88  
 IEHLIST step 86,88  
 IEHPRGM 84,87  
 IEWL steps 84-85,87  
 IFCDIP00 step 86,88  
 procedures 79-84  
 reallocation of data sets 88-94  
 Stage I 79-80  
 Stage II 80-84  
 techniques 81-84  
 RESTART parameter (JOB card) 261  
 RMS (see recovery management support, model 85)  
 Rollout/rollin function  
 description 182  
 specifying 277  
 system data set 230  
 (see also SYS1.ROLLOUT)  
 Routing codes  
 description 165-168  
 specifying 398-402,405-406  
 RPG macro instruction  
 specifying 395  
 type of (table) 244  
 RPG processor  
 component name 205  
 sample program 114-116  
 (see also RPG macro instruction)  
 RPGSMPL (see RPG sample program)  
 RSVC list  
 included in 227  
 (see also SYS1.PARMLIB)

**S**

Sample accounting routine (SAMACTRT) 116  
 description 33  
 Sample library (see SYS1.SAMPLIB)  
 Sample programs 97-120  
 ALGOL (IEXSAMP) 98  
 American National Standard COBOL (IKFSAMP) 99  
 assembler (IEUESP) 100  
 COBOL E (IEPSAMP) 101  
 data generator sample program (IEBDATGN) 102  
 data transmission and conversion subroutines (IKDSAMPL) 103  
 FORTRAN G & H (IEYSP) 104  
 graphics (SAMP2250 and SAMP2260) 105-107  
 graphic subroutine package 108-110  
 IEHDASDR system utility (DRISAMP) 111  
 loader program test program 112  
 names of 97  
 PL/I F (IEMSP2) 113  
 RPG (RPGSMPL) 114-115

sample account routine (SAMACRT) 116  
 sort/merge (IERSP) 117  
 update analysis program (IHGSAMP) 118-119  
 user label 120  
 (see also SYS1.SAMPLIB)  
 SAMP2250 (see graphics 2250 sample program)  
 SAMP2260 (see graphics 2260 sample program)  
 Satellite graphic job processor  
 component name 206  
 to be included 322  
 SCHEDULR macro instruction  
 cross-reference table 467-471  
 specifying 397-420  
 type of (table) 244  
 Scratch volume  
 use during restart 85,87  
 SECMODS macro instruction 445  
 cross-reference table 470,471  
 specifying 421-423  
 type of (table) 244  
 Secondary allocation 208  
 Secondary console  
 (see SECONSLE macro instruction)  
 SECONSLE macro instruction 325,403,405  
 cross-reference table 469,471  
 specifying 425-430  
 type of (table) 244  
 Selector channel  
 specifying 257  
 Selector subchannel 346  
 SER 422,445-446,448  
 component name 206  
 Service aids  
 (see SYS1.DN554 and SYS1.DN554A)  
 SER0 (see SER)  
 SER1 (see SER)  
 SGJP (see satellite graphic job processor)  
 Shared DASD option  
 description 182  
 specifying 333,334,343,344,444  
 Shared library for PL/I (see PL/I library)  
 SMF  
 data sets 225  
 description 184-185  
 requirements 409,446  
 sample programs 33  
 specifying 398,402,408,446  
 SMF definition list (SMFDEFLT)  
 (see SMF)  
 SMFDEFLT (see SMF)  
 Sort library 227  
 component name 206  
 (see also SYS1.SORTLIB, SORTLIB macro instruction)  
 SORTLIB macro instruction  
 cross-reference table 471  
 required 435  
 specifying 433-434  
 type of (table) 244  
 SORTMERG macro instruction 433-434  
 cross-reference table 471  
 specifying 435-438  
 type of (table) 244  
 Sort/merge component name 206  
 Sort/merge sample program 117

## Space allocation

- done by IEHPROGM 9
- example 10-15
- for ALGOL library 245
- for COBOL 267
- for COBOL library 265
- for FORTRAN library 295
- for generation 308,315,316
- for GJP or SGJP 319-320
- for LINKLIB 247,251,267,301,355,371,395,435
- for MVT 25
- for minimum region 408
- for nucleus 393,412
- for parameter library 365
- for partitions 367-369
- for PL/I 372-377
- for PL/I library 378
- for procedure library 389
- for sort/merge library 433
- for SYSJOBQ 407-408
- for system macro instructions 361
- for telecommunications library 391
- for write-to-operator (WTO or WTOR) 415
- for write-to-log (WTL) 414
- new system data sets 207-236
- reallocation during restart 88-94
- restarting Stage I 79-80
- restarting Stage II 80-84
- utility data sets 26,28,32  
(see also system data sets)

SPACE parameter

- DD statement, initialization 10
- values, utility data sets 21

SPIE function 445

SPIE routines made resident

- description 183-184
- specifying 445

Stage I

- coding macro instructions 20
- description 1
- errors 79
- generation library (see SYS1.GENLIB)
- input deck 20-22
- product of 65,71
- restart procedures 80-84
- sample program 118-119
- sample restart deck 80
- space allocation 21
- system generation macro instructions 245-472

Stage II

- assemblies 2
- considerations for MVT 25
- description 1-2
- errors 80
- generation library (see SYS1.MODGEN)
- input 1
- link edit 2
- output data sets 315-316
- processing 1-2
- restart procedures 80-94
- sample program 118-119
- user-written functions 23-24  
(see also job stream)

## Status displays

- description of 165-166,168
- specifying of 397-401,406,409,426-429
- use 415

Starter operating system 33-62

- contents 33-34
- contents of SYS1.SAMPLIB 33
- data sets 33-35
- deleting libraries 59
- distribution 483
- example 136-138
- processing 40-59
- renaming data sets 30
- requirements 34
- sample initialization 136-137
- unit names 39

3330 32-35,42,123-132

START RDR command 81,82,412

START WTR command 413

Storage hierarchies 275-277,372-374

Storage protection

- description 184
- specifying 443,448

Storage requirements (see individual program options descriptions)

Storage size 253,255,422

Subchannel (see selector subchannel)

Subtasking (see ATTACH function)

SUPRVSOR macro instruction 439-447

- cross-reference table 471
- specifying 439-447
- type of (table) 244

SVC library (see SYS1.SVCLIB)

SVC routines

- resident 393
- transient 449
- type 1 and 2 451
- type 3 and 4 445,451
- user-written 393,449,451

SVC table 443

SVCLIB macro instruction 449,450

- cross-reference table 471
- specifying 449,450

SVCTABLE macro instruction 451,452

- cross-reference table 471
- specifying 451,452
- type of (table) 244

Synchronous overlay 278

Syntax checker

- component names 204
- for FORTRAN 259-260,305
- for PL/I 259-260,378
- specifying 259-260
- with CRJE 170

SYSCP 463-465

SYSCTLG

- description 209
- required 208,209
- serial number 10
- space allocation 209

SYSDA 463-465

SYSIN data set 9,10

- used during restart 79,80,81,82
- use of 90-92

SYSOUT data set 322

- used during sort/merge 438

SYSPRINT data set 7,27,315  
   use of 69-71,72-73,79-83  
 SYSPUNCH data set 88,89,90,91,93  
   purpose 22  
   use during restart 80  
   use of 69-71,75-76  
 SYSSQ 463-465  
 System catalog (see SYSCTLG)  
 System commands and responses with multiple  
   consoles 165  
 System data sets 199-236  
 System environment recording (see SER)  
 System error recovery (see SER)  
 System generating libraries  
   (see SYS1.GENLIB, SYS1.MODGEN, component  
   libraries)  
 System generation macro instructions  
   (see macro instructions)  
 System input unit  
   (see SYSIN)  
 System libraries  
   (see individual operating system  
   libraries)  
 System log data sets  
   (see SYS1.SYSVLOGX, SYS1.SYSVLOGY)  
 System macros library  
   (see SYS1.MACLIB)  
 System management facility  
   (see SMF)  
 System output unit  
   (see SYSOUT data set, SYSPRINT data set,  
   SYSPUNCH data set)  
 System queue area  
   specifying 278  
 System residence volume  
   data sets on 7,208  
   initializing 7-8  
   space allocation 7-10,12-14  
   (see also initialization)  
 System utility programs  
   example 111  
   IEBTPCH 30  
   IEBUPDAT 23  
   IEBUPDTE 23,319  
   IEHLIST 27  
   IEHPRGM 31-32  
   use of 2,21-22,29-32,301,381,433,435  
 SYSUTILS macro instruction  
   specifying 453  
   type of (table) 244  
 SYSUT1  
   DD statement 20-23  
   reallocation 89-90  
   space allocation 21  
   use during restart 75-76,85,87  
   use of 21-23,24  
 SYSUT2  
   DD statement 20-23  
   reallocation 89-90,93  
   space allocation 21  
   use during restart 75-76,85,87  
   use of 21-23,24  
 SYSUT3  
   DD statement 20-23  
   reallocation 89-90  
   space allocation 21  
   use during restart 75-76,85,87  
   use of 21-23,24  
 use during restart 75-76,85,87  
   use of 21-23,24  
 SYSUT4  
   DD statement 20-23  
   reallocation 89-90,93  
   space allocation 21  
   use of 20-23  
 SYS1.ACCT 208,210  
 SYS1.ALGLIB 208,211  
   generated from 204  
   INDEX parameter 29,31,33  
   reallocation 87,80  
   specifying 245-246  
   to be included 246  
   user assigned qualifier 312  
   (see also ALGOL macro instruction,  
   ALGLIB macro instruction, ALGOL  
   library)  
 SYS1.ASRLIB 208,213  
   initialized 65  
   requirement 445  
   restart DD statement 85  
   user assigned qualifier 312  
 SYS1.BROADCAST 208,213,457  
   (see also time sharing option)  
 SYS1.CMDLIB 208,214,264,317  
   (see also time sharing option)  
 SYS1.COBLIB 208,215,326  
   catalog 27,29  
   generated from 204  
   INDEX parameter 29,31,33  
   reallocation 87,80  
   rename 29  
   requirements 29,268  
   specifying 265-266  
   user assigned qualifier 312  
   (see also COBOL macro instruction,  
   COBOL library, COBLIB macro  
   instruction)  
 SYS1.DCMLIB 216  
 SYS1.DN554A  
   description 33  
   distribution of 33,484-486  
 SYS1.DN554 distribution 206  
 SYS1.DUMP 208,218  
   exception 8  
 SYS1.FORTLIB 208,219  
   catalog 27,295  
   generated from 204  
   generation of 313  
   INDEX parameter 29,30,31,33  
   reallocation 84-85,86-88  
   rename 30  
   space allocation 296  
   specifying 296  
   user assigned qualifier 312  
   user-written modules 30  
   (see also FORTRAN macro instruction,  
   FORTLIB macro instruction, FORTRAN  
   library, FORTRAN compiler, FORTRAN  
   subprograms)  
 SYS1.GENLIB 201  
   assembler requirements 251  
   catalog 27,30  
   demounting 15  
   distribution 38,484-486  
   (see also SYS1.MODGEN)

**SYS1.HELP** 208,220,326  
 (see also time sharing option)  
**SYS1.IMAGELIB** 149-150,208,220  
**SYS1.LINKLIB** 208,220,313-314  
   INDEX parameter 29,30,31  
   I/O device generation 3,308  
   mounting 15  
   nucleus generation 307  
   performance considerations 15  
   printing of 311  
   processor generation 308  
   reallocation 80,87  
   space allocation 10  
   specifying 355  
   use of 179,445  
   user assigned qualifier 312  
   user-written programs 24,355-358  
   64K 25  
**SYS1.LOGREC** 208,224  
   I/O device generation 3  
   printing of 314  
   restarting DD statement 84,85  
   user assigned qualifier 312  
**SYS1.MACLIB** 208,225  
   catalog 27,361  
   demonthing 14  
   distribution 34,483-486  
   generated from 206  
   generation of 68,80,361-362  
   INDEX parameter 20,30,31  
   in starter operating system 29-30  
   mounting 14  
   rename 29-30  
   requirements 361,362  
   restart 84-85  
   specifying 361-362  
   uncatalog 27-28  
   user assigned qualifier 312  
**SYS1.MANX** 208,225  
**SYS1.MANY** 208,225  
**SYS1.MODGEN, SYS1.MODGEN2** 201  
   catalog 26-27  
   demonthing 15  
   distribution 33-34,484-486  
   mounting 15  
   save 486  
   (see also SYS1.GENLIB)  
**SYS1.NUCLEUS** 10,208,226  
   catalog 393  
   generation of 315  
   INDEX parameter 29,30,31  
   mounting 15  
   performance considerations 15  
   reallocation 87  
   serial number 10  
   user assigned qualifier 312  
   user-written modules 393  
**SYS1.PARMLIB** 208,227  
   catalog 26-27  
   catalog if 365  
   generated from 205  
   generation of 365  
   INDEX parameter 29,30,31  
   in starter operating svstem 30  
   null allocation 10,365  
   reallocation 86  
   specifying 365  
   use of 359  
   user assigned qualifier 312  
   (see also parameter library)  
**SYS1.PL1LIB** 3,208,228  
   catalog 28,381  
   description 381-382  
   generated from 205  
   generation of 381-382  
   INDEX parameter 29,30,31  
   reallocation 86  
   rename 30  
   space allocation 381  
   specifying 381-387  
   user assigned qualifier 312  
   (see also PL/I library, PL1LIB macro  
   instruction, PL/I compiler)  
**SYS1.PROCLIB** 208,209  
   generated from 205  
   generation of 389-390  
   GJP cataloged procedures 319  
   INDEX parameter 29,30,31  
   in starter operating system 30  
   planning for 26-28  
   null allocation 10,389  
   rename 28-30  
   SGJP cataloged procedures 319  
   specifying 389-390  
   user assigned qualifier 312  
**SYS1.ROLLOUT** 208,230  
   example 11  
**SYS1.SAMPLIB** 33  
   contents 7,97,184-185  
   obtaining 98  
   sample programs 97-120  
**SYS1.SORTLIB** 208,231  
   catalog 27-28  
   generated from 206  
   INDEX parameter 29,30,31  
   rename 28-30  
   scratched 433,435  
   specifying 433  
   user assigned qualifier 312  
   (see also SORTLIB macro instruction,  
   SORTMERG macro instruction)  
**SYS1.SVCLIB** 208,232  
   generation of 450  
   INDEX parameter 29,30,31  
   I/O device generation 3  
   limit 449  
   mounting 15  
   performance considerations 15  
   printing of 311  
   serial number 10  
   user-written programs 24,449  
   user assigned qualifier 312  
   64K 25  
**SYS1.SYSJOBQE** 208,233  
   expiration date 10  
   reallocation 93-94  
   resident 413  
   SUPRVSOR keyword parameter 416  
**SYS1.SYSVLOGX** 208,234  
   buffer 414  
**SYS1.SYSVLOGY** 208,234  
   buffer 414

SYS1.TELCMLIB 208,235  
     BTAM 281  
     generated from 207  
     generation of 455-456  
     INDEX parameter 29,30,31  
     QTAM 281  
     reallocation 85,87  
     user assigned qualifier 312  
     (see also TELCMLIB macro instruction,  
     telecommunications library)  
 SYS1.TSOGEN 203  
     (see also time sharing option)  
 SYS1.UADS 208,236,459  
     (see also time sharing option)  
 S360  
     specifying 340,351  
 S370  
     specifying 254

## T

Task supervisor options 439-447  
 TCAM  
     cataloging 28  
     component 206  
     description 186  
     to exclude 361  
         blocking factor 483-486  
         job stream-Stage II 68,71  
 TELCMLIB macro instruction 391  
     specifying 455  
     type of (table) 244  
     use of 186  
 Telecommunication access method  
     (see TCAM)  
 Telecommunication devices 337  
 Telecommunications library 281  
     component name 206  
     to be included 455  
     (see also SYS1.TELCMLIB, TELCMLIB  
     macro instruction)  
 Telecommunications lines 337,343-344  
     transmission adapter 340  
 Telecommunications subroutines  
     contained in 455-457  
     (see also SYS1.TELCMLIB)  
 Terminal devices 343  
 Timer  
     feature 254  
     specifying 439,455  
 Time-slicing  
     description 190  
     specifying 279-280  
 Timing options  
     description 186-187  
     specifying 402,439,446  
 Time sharing option (TSO) 188-189  
     distribution library 483-486  
     examples 17,76,149-153  
     guidelines for restarting Stage II  
     86-88  
     job stream Stage II 71-76

Trace option  
     description 191  
     specifying 439,446  
 Trace table 446  
 Track stacking 407  
 Transient area 276  
 Transient SVC table option  
     description 192-193  
     specifying 367  
     (see also SVC routines, SVC table,  
     SVCTABLE macro instruction)  
 TSO  
     (see time sharing option)  
 TSOPTION macro instruction  
     specifying 457  
     type of (table) 244  
     (see also time sharing option)  
 TWX  
     specifying 338,341,351  
 Type 3 and 4 SVC routines made resident  
     description 193  
     specifying 445  
     (see also SVC routines, SVCLIB macro  
     instruction)  
 Types of generation 2-3  
     table 244

## U

UADS macro instruction  
     specifying 459  
     type of (table) 244  
     (see also time sharing option)  
 UCS macro instruction  
     cross-reference table 472  
     specifying 461-462  
     type of (table) 244  
 UNCT2311  
     description 32  
     included in 33  
 UNCT2314  
     description 32  
     included in 33  
 Unit address  
     definition 242  
 Unit name  
     (see device type)  
 UNIT PARAMETER  
     DD statement, initialization 9,10,22  
     JCL for Stage II 69-71  
     SYSPUNCH DD statement 21-22,81  
 UNITNAME macro instruction 320,337,389  
     additional devices 479  
     cross-reference table 472  
     specifying 463-465  
     type of (table) 244  
 Universal character set 244,267,301,343,  
     371,461  
     specifying 254  
     with starter operating system 35  
 Update analysis program 97  
     sample program 118-119  
 User label sample program 97,120



User totaling facility example  
 (see user label sample program)

User-written functions  
 error routines 343  
 for FORTRAN 206,295  
 include in 24,295,355,393,449,451  
 macro instructions 244  
 nonstandard label routines 449  
 space allocation 355,393  
 SVC 449,451,452  
 SYS1.LINKLIB 24,355  
 SYS1.NUCLEUS 24,393  
 SYS1.SVCLIB 24

User-written modules  
 (see user-written functions)

User-written programs  
 (see user-written functions)

User-written SVC routines  
 description 193  
 specifying 343,393,449  
 (see also user-written functions)

Utility data sets  
 catalog 22,28  
 deleting 79,307  
 demounted 14  
 during restart 79  
 example of allocation 15-20  
 example of scratching 130  
 for generation 20-22,29,32  
 mounting 14-15,59  
 names for 20  
 reallocation 88-94  
 required 14,26-28,29-30,32  
 reside 20  
 scratch 22,79  
 sizes 21  
 space allocation 11,21,28,32  
 specifying 20-22,307,315-316  
 (see also system data sets)

Utility programs 1,33,111  
 during Stage I 65  
 during Stage II 1,65,74  
 examples 126-127,129,136-137,140-141  
 storage available for 453  
 uses utility data sets 32  
 (see also system utility programs)

UT1SDS keyword (GENERATE macro  
 instruction)  
 requirement 307  
 specifying 309,316

UT2SDS keyword (GENERATE MACRO  
 instruction)  
 requirement 307  
 specifying 309,316

UT3SDS keyword (GENERATE macro  
 instruction)  
 requirement 307  
 specifying 309,316

UT4SDS keyword (GENERATE macro  
 instruction)  
 requirement 307  
 specifying 309,316

**V**

Validity check option  
 description 194  
 specifying 443

Verify DASD volume serial number 442

VOL parameter 9,11

Volume index  
 for SYSCTLG 7,8

Volume initialization  
 CATLG statements 10  
 DD statements 10  
 description 7  
 example 7  
 performed by 7

Volume label 71

Volume mounting  
 requirements 14-15

VOLUME parameter 9,10

Volume serial number 9,10,11,39

Volume statistics  
 description 195-196  
 SMF requirements 195,196,402

Volume table of contents  
 (see VTOC)

VTOC 7,8

**W**

WAIT macro instruction 446

WAIT options 446

WTP  
 (see write-to-programmer)

Write-to-log messages (WTL)  
 buffers 413-415  
 (see also SYS1.SYSVLOGX, SYS1.SYSVLOGY)

Write-to-operator  
 buffer default sizes 414  
 message buffers 414  
 messages contained in 234

Write-to-operator routines (WTO and/or  
 WTOR)  
 (see write-to-operator)

Write-to-programmer  
 message buffers 407-408

WTO  
 (see write-to-operator)

WTOR  
 (see write-to-operator)

WTR  
 (see Writer cataloged procedure)

WTTA  
 specifying 338,341

**X**

XCTL macro instruction  
 (see SYS1.LINKLIB)

7-track magnetic tape  
     for utility data sets 413  
     use of during generation 22  
 64K main storage 450  
     restrictions 355  
 83B3  
     specifying 338,341,350  
 115A  
     specifying 338,341,350  
 1030  
     specifying 338,348  
 1050  
     specifying 338,341,348  
 1050X  
     specifying 338,339,343,344,348  
 1051  
     specifying 334  
 1052  
     device type 476  
     specifying 334,338,348,350  
 1053  
     device type 476  
     specifying 338,348  
 1060  
     specifying 338,341,343,344,348  
 1130  
     specifying 338,341,348  
 1275  
     device type 476  
     restriction 338  
     specifying 338,342,348  
 1285  
     error routines made resident 444-445  
     specifying 338,348,444-445  
 1287  
     error routines made resident 444,445  
     specifying 338,348,444,445  
 1288  
     error routines made resident 444,445  
     specifying 338,348,444,445  
 1403  
     device type 476  
     specifying 338,343,344,348,350  
 1404  
     specifying 350  
 1419  
     specifying 338,342,348  
 1442  
     device type 476  
     specifying 338,348  
 1443  
     device type 476  
     specifying 338,343,344,348  
 2020  
     specifying 338,339,341,348  
 2150  
     specifying 350  
 2250  
     specifying 338,339,343,344,346,349,  
         353,376  
 2260  
     specifying 338,339,343,344,349,353  
 2265  
     specifying 338,341,349,353  
 2280  
     specifying 176,338,343,344,349

2282  
     specifying 338,343,344,349  
 2301  
     device type 475  
     specifying 338,347,349  
     system residence 8,315  
 2302  
     device type 475  
     specifying 337,338,347,349  
 2303  
     device type 475  
     specifying 338,347,349  
     system residence 8,315  
 2305  
     device type 475  
     specifying 337,338,347,349  
     system residence 8,315  
 2311  
     device type 475  
     specifying 337,338,347,349  
     system residence 8,315  
 2314  
     device type 475  
     specifying 337,338,343,344,347,349  
     system residence 8,315  
 2319  
     device type 493  
     specifying 340  
 2321  
     device type 475  
     specifying 338,347,349  
     use of 465  
 2400  
     device type 475  
     specifying 344  
 2401  
     specifying 337,338,343,344,346,349  
 2402  
     specifying 337,338,343,344,346,349  
 2403  
     specifying 334,343,344,346,349  
 2404  
     specifying 334,337,338,343,346,350  
 2415  
     specifying 333,334,338,343,344,350  
 2420  
     specifying 337,338,346,350  
 2495  
     device type 476  
     restriction 340  
     specifying 338,343  
 2501  
     device type 476  
     specifying 338,350  
 2520  
     device type 476  
     specifying 338,350  
 2540P  
     device type 476  
     specifying 338,340,350  
 2540R  
     device type 476  
     specifying 338,340,350  
 2596  
     description 443  
     specifying 339

2671	device type	476			
	specifying	338,350			
2701	specifying	333,334,341,345,353			
2702	specifying	333,334,341,345,353			
2703	specifying	333,334,341,345,353			
2715	BSCA	341			
2740	specifying	338,341,344,350			
2740C	specifying	338,340,350			
2740X	specifying	338,340,350			
2741	specifying	341			
2741C	specifying	338,340,350			
2741P	specifying	338,340,350			
2760	(see OIU)				
2780	specifying	176,186,388,341,343,344,351			
2803	specifying	334			
2816	specifying	334			
2820	specifying	334			
2821	specifying	334			
2822	specifying	334			
2840	specifying	334,340			
2841	specifying	334,346			
2844	specifying	333,343,344			
2845	specifying	353			
2848	specifying	333,335,343,353			
2880	specifying	258			
2972	specifying	341			
2955	specifying	338,340			
3066	specifying	338,340			
3210	specifying	338,351			
3211	device type	477			
	specifying	338			
3215	specifying	338,477			
3330	device type	475			
	distribution	487,488			
	specifying	337,338,347,351			
	starter operating system	32-39,42,123-132			
	system residence	8,315			
3410	device type	39,475			
	specifying	338,343,351			
	use	37			
3420	device type	39,475			
	specifying	338,343,344,351			
	use	37			
3505	device type	39,476			
	specifying	338			
	use	37			
3525	description	343,344			
	device type	39,476			
	specifying	338			
	use	37			
3803	specifying	333,334,336			
5450	device type	476			
	specifying	338,340,351			
7770	specifying	338,351,477			



**International Business Machines Corporation**  
**Data Processing Division**  
**1133 Westchester Avenue, White Plains, New York 10604**  
**(U.S.A. only)**

**IBM World Trade Corporation**  
**821 United Nations Plaza, New York, New York 10017**  
**(International)**