

MRX/OS System Generation

Reference Manual

2200.008-01

MEMOREX

Computer System
Products

May 1973 Edition

This edition is a major revision and obsoletes all previous editions. It documents the MRX/OS system generation programs at their level in MRX/OS Release 2.

Technical changes are marked with a vertical bar in the outer margin. Changes due to subsequent releases will be documented in future publications bulletins or revisions.

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

Comments may be addressed to the Memorex Corporation, Publications Dept., 8941 Tenth Avenue North (Golden Valley) Minneapolis, Minnesota 55427.

© 1972, 1973 MEMOREX CORPORATION

PREFACE

This publication describes the system generation programs, including a discussion of the structure of the programs, libraries and files used by the programs, and system generation operating procedures.

Input to the system generation programs is a series of statements produced by filling out a question-answer checklist presented in the publication **MRX/OS System Generation Checklist**. Section 5 in this reference manual discusses the possible answers to the Checklist. The programmer should review the reference manual before filling out the Checklist.

TABLE OF CONTENTS

Section		Page
1	SYSTEM GENERATION OVERVIEW	1-1
	Operating Systems	1-1
	Minimum System	1-2
	Resident Extension System	1-2
	Operating Environment	1-2
2	SYSGEN STRUCTURE	2-1
	SYSGEN Components	2-1
	Build Parameter Table (\$OSSGBPT)	2-1
	Checklist Processor (\$OSSGCLP)	2-1
	Reassembly Control	2-1
	Card Image Generator	2-2
	Linkage Editor Control (\$OSRSDNT)	2-2
	Build System Resident (\$BLDRES)	2-2
	Job Flow	2-2
	Flow of JOBONE	2-2
	Flow of JOBTWO	2-4
	Copy Procedures	2-4
	File Clean-Up Procedures	2-4
3	LIBRARIES AND FILES USED BY THE SYSGEN PROGRAMS	3-1
	SYSGEN Object Library	3-1
	System Procedure Library	3-1
	SYSGEN Source Library	3-2
	Nucleus Library	3-2
	System Message Library	3-3
	Operating System Resident Library	3-3
	System Message Library Input	3-3
	System Macro Library	3-3
	SYSGEN Macro Backup Library	3-4
	System Load Library	3-4
	SYSGEN Load Library	3-5
	Checklist Data File	3-5
	SYSGEN Data File	3-5
	BLDRES Directive File	3-5
4	OPERATING PROCEDURES FOR SYSTEM GENERATION	4-1
	Type 1 SYSGEN (Steps for Procedures A and B)	4-2
	Multi-Drive User	4-2

TABLE OF CONTENTS (Continued)

Section	Page
4 (Cont)	4-2
Step 1	4-2
Step 2	4-2
Step 3	4-2
Step 4	4-3
Step 5	4-3
Step 6	4-3
Step 7	4-4
Step 8	4-4
Step 9	4-4
Step 10	4-4
Step 11	4-4
Step 12	4-5
Step 13	4-5
One-Drive User	4-5
Step 1	4-5
Step 2	4-5
Step 3	4-6
Step 4	4-6
Step 5	4-6
Step 6	4-7
Step 7	4-7
Step 8	4-7
Type 2 and Type 3 SYSGENS (Steps for Procedures A and B)	4-7
5	5-1
DESCRIPTION OF CHECKLIST QUESTIONS	5-1
SYSGEN Control	5-1
System Control	5-2
Output Spooler	5-3
I/O Devices	5-4
Telecommunications	5-7
Control Language Services	5-8
Assembler Language	5-10
COBOL Language	5-11
APPENDIX A – SYSTEM GENERATION PROCEDURES AND ERROR RECOVERY	A-1
APPENDIX B – CONTROL LANGUAGE STATEMENTS FOR SYSGEN	B-1
APPENDIX C – SPECIAL SYSGEN PROCEDURES	C-1
APPENDIX D – ERROR MESSAGES	D-1

1. SYSTEM GENERATION OVERVIEW

The system generation (SYSGEN) programs have been designed so that you can define a complete operating system that meets your individual needs, and place that system on a system resident pack. The SYSGEN programs can be executed under the operating system that is on the MRX distribution pack, or under an operating system that has been previously generated from such a distribution pack.

There are three distinct types of system generation:

Type 1 Complete system generation

Type 2 Modification of the resident operating system

Type 3 Modification of Programming Services

There are two system generation categories:

Procedure A destroys your input pack.

Procedure B preserves the integrity of your input pack.

See Section 4 for more information on the different types and categories of system generation.

The resident operating system executes in the reserved system area of main memory. The Programming Services programs execute in a user partition. Programming Services includes the following programs: Control Language Services, Data Management, Linkage Editor, Librarian, Assembler Language, COBOL, FORTRAN, and RPG II. The three types of system generation are discussed in Section 4 of this manual.

The particular type of system generation you choose is defined by filling out the System Generation Checklist. The statements that result from the Checklist are combined with the appropriate Control Language statements (Appendix B) and any necessary procedure //CALL statements (Appendix C) to provide the input for the SYSGEN programs.

OPERATING SYSTEMS

Memorex offers two basic resident operating systems. They are called the Minimum system and the Resident Extension system. The first is an 8KB system and the latter is a 10KB system. The size of the user partition(s) depends on the size of your resident operating system. For example, if the main memory size of your computer is 32KB and you have a

10KB resident operating system, you will have a user partition that is 22KB. You can elect to have two partitions, if you choose; the size of the partitions is defined by the system Generation Checklist. However, no partition can be smaller than 4KB (4096 bytes). Referring to the previous example, you could create two partitions from the 22KB available, one of them 4KB and the other 18KB. The size of the resident operating system varies from system to system according to the options you select (via the Checklist) and to the hardware configuration. The features of each type of resident operating system are outlined in the following paragraphs.

MINIMUM SYSTEM

The Minimum resident operating system is an 8KB system. It supports one buffered card reader, one buffered line printer, two 200 cylinder disc drives (3664)*, and one operator console (1240). Table D-1 in Appendix D of the Checklist manual shows the byte requirements of additional features that can be added to the Minimum system.

RESIDENT EXTENSION SYSTEM

The Resident Extension resident operating system is a 10KB system. It not only supports the same basic hardware that the Minimum system supports, but it also supports up to a maximum of eight disc drives and a maximum of 16 external controllers to which additional printers, and tape drives can be attached, depending upon the options selected from Table D-1 in Appendix D of the **System Generation Checklist** manual. The telecommunications driver can be added for an additional 2046 bytes. Each telecommunications line requires 100 bytes for internal tables. Each equipment type configured, in addition to the operator console equipment type which is 80, requires from 300-800 bytes. Logical communications capability can be added to the Resident Extension system for an additional 1650 bytes. When logical communications is being used, it requires additional space in the user partition. Tables D-1 and D-2 in Appendix D of the Checklist manual show the byte requirements of additional features that can be added to the Resident Extension system.

OPERATING ENVIRONMENT

In addition to one card reader, one line printer, one operator console, and at least one disc drive, the user must have approximately 800 tracks of storage on-line to store the files used by the SYSGEN programs. These programs can be executed under the operating system that resides on all MRX software distribution packs, or they can be executed under any operating system previously created from such a distribution pack.

The SYSGEN programs use macro language and Control Language procedures, which are explained in the **MRX/OS Assembler Reference** manual and the **MRX/OS Control Language Services, Basic Reference** manual, respectively. Prior to performing a SYSGEN, the user must refer to the **MRX/OS Utility Programs Reference** manual for a description of the procedures he must follow to initialize a disc pack as a system resident pack. Any pack that is used as a system resident pack must be initialized as a system resident pack before the SYSGEN programs can be executed.

*It is possible to have a system containing only one disc drive.

2. SYSGEN STRUCTURE

The system generation statements defined by the programmer are executed by a series of related programs that perform the system generation process. The input of these programs is submitted in a series of different jobs. The input for JOBONE is the set of statements produced from the Checklist questions. The output of JOBONE is the input for JOBTWO. This section discusses the various SYSGEN programs and shows the job flow of the system generation process. (Detailed information on Control Language for SYSGEN is presented in Appendix B.)

SYSGEN COMPONENTS

The components of the SYSGEN jobs include: Build Parameter Table Program, Checklist Processor, Reassembly Control, Card Image Generator, Linkage Editor Control, and Build System Resident. Figures 2-1 and 2-2 show the flow of the SYSGEN jobs. When used in conjunction with the information presented in Appendix A, this information will help you debug your programs. The major job steps of the SYSGEN programs are described in the following paragraphs. The name of the job step appears in parentheses after the paragraph heading.

BUILD PARAMETER TABLE (\$OSSGBPT)

This program performs validity checks on input parameters and formats the data for the Checklist Processor.

CHECKLIST PROCESSOR (\$OSSGCLP)

The Checklist Processor analyzes the input data for SYSGEN JOBONE (answers to Checklist questions prepared by you) and modifies the relevant SYSGEN macros to reflect the SYSGEN options specified. This job step performs two additional functions that are described in the next two paragraphs. They are Reassembly Control and Card Image Generator.

REASSEMBLY CONTROL

Based on the macros that have just been modified, the Reassembly Control component determines which parts of the system must be assembled to produce the system tailored to your unique needs. If parts of the operating system that execute in the user partition are modified, this component causes the Card Image Generator to produce Control Language statements to linkage edit those parts of the system (Control Language Services, Assembler Language, and COBOL Language).

CARD IMAGE GENERATOR

The Card Image Generator builds the macro named \$CALLGEN that is called by the source modules \$CRDGEN1 and \$CRDGEN2. When these modules are assembled, they produce the Control Language procedure called GENPROC2. When these procedures are executed, they cause the Assembler and the Linkage Editor to produce the following sets of modules:

- Updated system object modules
- Relocatable load modules (for system components that execute in the user partition)
- Absolute load modules (for system components that are part of the resident operating system or associated overlays)

LINKAGE EDITOR CONTROL (\$OSRSDNT)

When this component is assembled, it calls the SYSGEN macros named SGCNTRL and SGTCDSEFS. These macros may have been modified by the Checklist Processor. They contain information describing the resident operating system structure desired to control the selection of the appropriate main resident operating system Linkage Editor directives.

BUILD SYSTEM RESIDENT (\$BLDRES)

This component verifies that the volume designated as the system resident pack has been properly initialized. (The Disc Initialize utility program must be executed before you attempt to execute the SYSGEN programs.) Then it reads absolute load modules from the library file built by the Linkage Editor and enters them in a program file called \$NUCLIB (Nucleus Library). This library has a special structure that makes possible high-speed loading of resident overlays. After the library has been created, a similar program file called \$MSGLIB, containing system messages, is built. The relocatable load modules are then merged with the system load library, \$SYSLODLIB.

JOB FLOW

As shown by Figures 2-1 and 2-2, the SYSGEN job flow has been designed to minimize your need for previous experience in performing a system generation. All you need to do is complete the Checklist and code the appropriate Control Language statements. The Checklist questions are fully described in Section 5 of this manual. The Control Language statements are shown in Appendix B. The two following paragraphs explain the flow of each job

FLOW OF JOBONE

While looking at Figure 2-1, refer to the steps for SYSGEN JOBONE outlined below. The dotted line in the figure isolates the Checklist Processor component which consists of steps 2 through 4.

1. Validate input parameters and build them into virtual table format.
2. Edit SYSGEN macros.

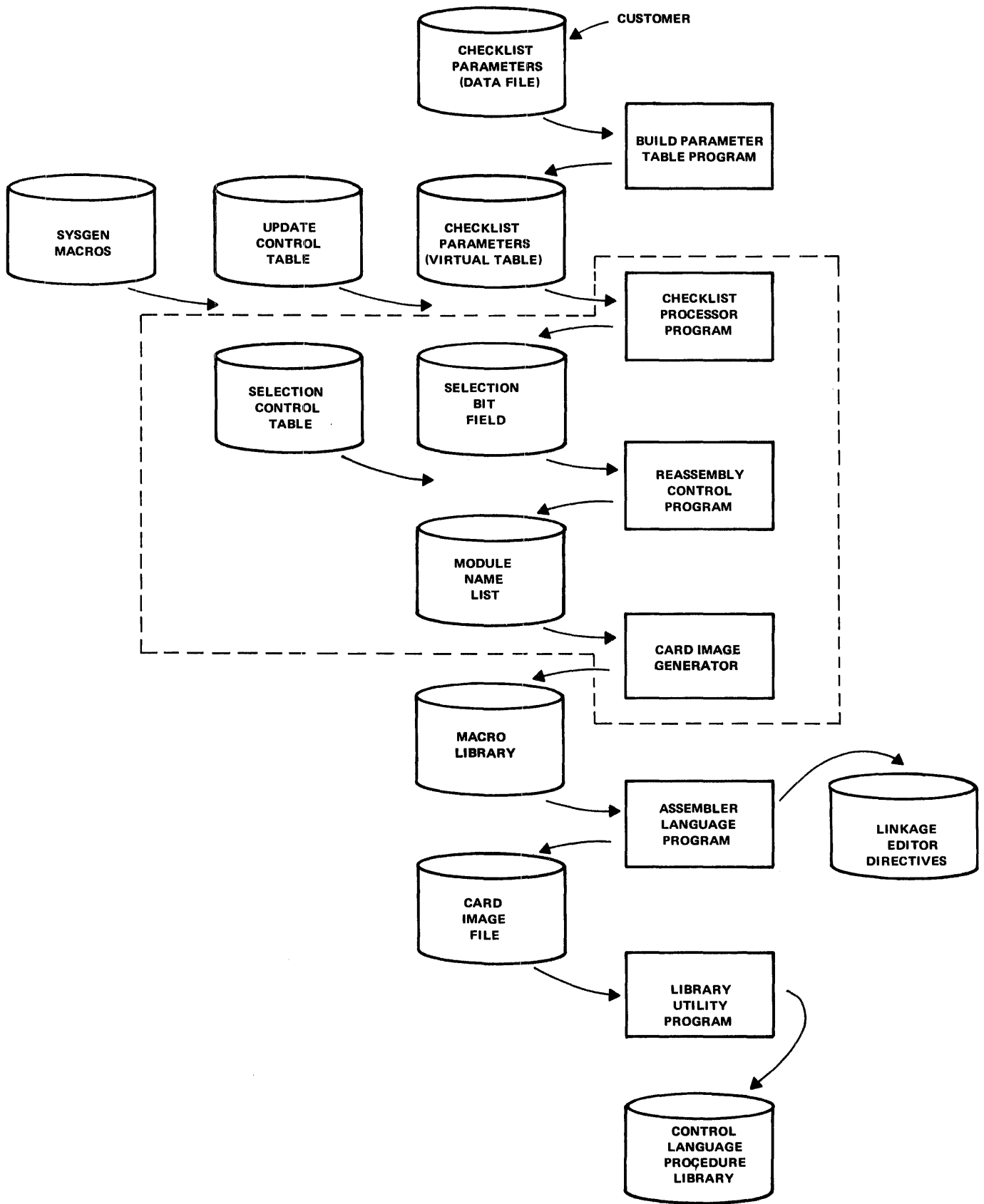


Figure 2-1. SYSGEN JOBONE

3. Select system source modules that require assembly.
4. Build card images of the Control Language Services procedure required to assemble selected system source modules and linkage edit the system.
5. Add the preceding Control Language Services procedure to SYSGEN JOBTWO.

FLOW OF JOBTWO

While looking at Figure 2-2, refer to the steps for SYSGEN JOBTWO outlined below.

1. Execute the Assembler to produce updated system object modules.
2. Execute the Linkage Editor to produce the resident operating system absolute load modules.
3. Unblock the absolute modules and write them in the Nucleus Library (\$NUCLIB).
4. Execute the Linkage Editor to produce relocatable load modules for system programs that run in user partitions.

GENPROC3 (steps 2 and 3 above) can be executed independently as a separate job, if you so desire.

COPY PROCEDURES

Until you have executed the copy procedures of JOB2A and JOB2B you have not completed your formal system generation procedures. The Control Language statements required to execute these jobs are listed in Appendix B.

FILE CLEAN-UP PROCEDURES

JOBTHREE, JOBFOUR, and JOBFIVE are copy procedures like JOB2A and JOB2B, but they are not part of the formal system generation procedures. Use JOBTHREE to copy the libraries from the distribution pack onto your new resident pack. Use JOBFOUR to copy the internal files from the distribution pack to your new resident pack. Execute JOBFIVE to produce a clean listing of the contents of the \$SYSLODLIB (System Load Library) that is on your new resident pack. This listing, however, will not show the files that you copied in JOBFOUR.

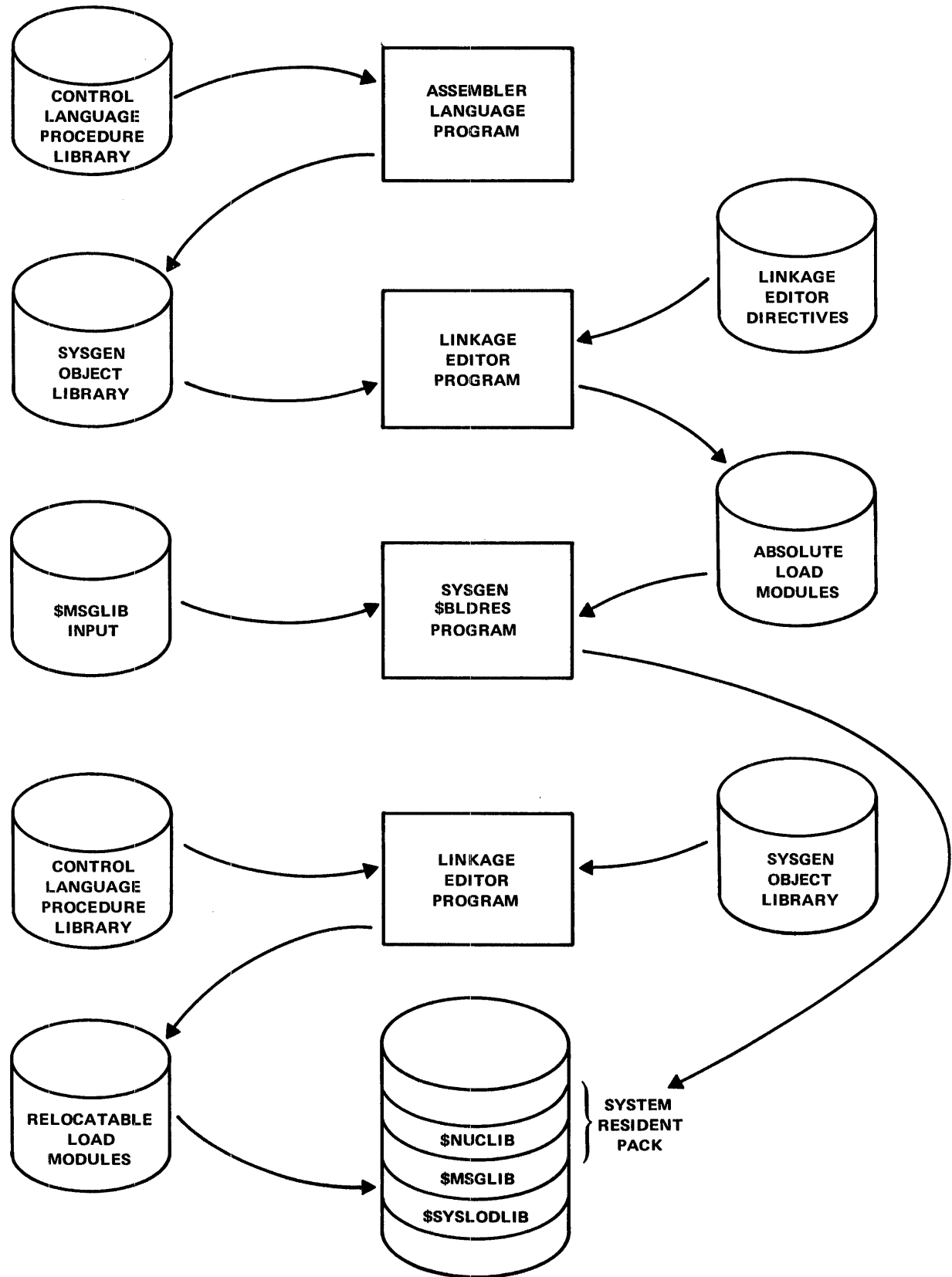


Figure 2-2. SYSGEN JOBTWO

3. LIBRARIES AND FILES USED BY THE SYSGEN PROGRAMS

This section contains descriptions of all the libraries and files used by the SYSGEN programs. Included in these descriptions are recommendations for renaming and labeling libraries and files that you will need in order to perform Type 2 and Type 3 SYSGEN's, as well as the libraries and files that you must save in order to give yourself the capability of applying either absolute or relocatable patches (or both) to your system.

SYSGEN OBJECT LIBRARY

This library is called \$SGOBJ. It contains all of the object modules required to support the SYSGEN programs. This library must be cataloged in the CCAT and must be on-line in order to execute the SYSGEN programs. Because the SYSGEN programs modify the contents of this library during execution, we recommend that you always create a backup copy of your present version of your system resident pack before you execute the SYSGEN programs.

Execute JOB2B to copy \$SGOBJ onto your new resident pack so that you have the capability of modifying your system using a Type 2 or Type 3 SYSGEN or by applying relocatable patches. You can also rename this library to reflect the current version of your system. However, the name you choose must be limited to eight alphanumeric characters because this name is a parameter used internally by the SYSGEN programs to create GENPROC2 and GENPROC4. On the MRX Distribution Pack, this library is called \$SGOBJ. The name of this library in your current system is the answer to question 3 in the Control Section of the Checklist.

Members of the SYSGEN object library that may be modified by the SYSGEN programs include:

- \$OSRSDNT
- \$OSYSTAB
- \$TPCDRV
- \$OSCCRES
- \$IOBIO
- \$CPRES1
- \$IODRV
- \$JMIR
- ASM
- COBOL
- \$OSSVCTR

SYSTEM PROCEDURE LIBRARY

This library is called \$SYSPROCLIB. It contains all of the procedures provided by MRX/OS. This library must be on-line and cataloged in the system CCAT in order to execute the SYSGEN programs. Two of the four SYSGEN procedures reside in this library. They are GENPROC1 and GENPROC3. GENPROC2 and GENPROC4 are created during

JOBONE and added to this library. After JOBTWO has terminated, GENPROC2 and GENPROC4 can be deleted. You can delete any MRX/OS procedures from this library that you do not intend to use. If you do this, we recommend that you make backup copies of these procedures so that you have them available if you decide to use them at a later date. You can also add procedures of your own to this library, or you can create your own procedure library (or libraries). If you want to save space, you can copy GENPROC1 and GENPROC3 and store them off-line. However, before executing the SYSGEN programs, you must then merge GENPROC1 and GENPROC3 into the system procedure library and make sure that that library is on-line and cataloged in the CCAT. The members of this library that are related to the SYSGEN programs include:

- SGALLOC
- SGCOPY
- SGMERGE
- SGPURGE
- SGCATDSP
- SGCAT
- SGUNCAT
- SGCOPYA
- SGCPYCAT
- SGRTNCAT

SYSGEN SOURCE LIBRARY

This library is called \$SYSGENSCRLIB. It contains all of the source modules required to support the SYSGEN programs. It must always be present in the system CCAT and on-line in order to execute the SYSGEN programs. This library must be saved. It is needed for all types of SYSGEN that you will perform. However, it need not be on-line except when you are executing the SYSGEN programs.

All of the members of the SYSGEN source library are listed below.

- \$OSRSDNT
- \$OSYSTAB
- \$TPCDRV
- \$OSCCRES
- \$IOBIO
- \$IODRV
- \$CPRES1
- \$JMIR
- ASM
- COBOL
- \$OSSVCTR
- \$CRDGEN1
- \$CRDGEN2

NUCLEUS LIBRARY

This library is called \$NUCLIB. It contains your resident operating system and its overlays. This library resides on the MRX Distribution Pack. The \$NUCLIB is built by Type 1

SYSGEN's, and it is modified by Type 2 SYSGEN's. The final contents of \$NUCLIB will reflect the answers you have coded to the Checklist questions. The contents of \$NUCLIB are logically equivalent to the contents of \$OSRSDNTLIB (Operating System Resident Library). However, the contents of \$NUCLIB are put in the format of a variable record length program file to facilitate high-speed loading of system overlays.

SYSTEM MESSAGE LIBRARY

This library is called \$MSGLIB. It also has the format of a variable record length program file. It contains all of the system messages that are printed on the SYSOUT file. The messages in this library are written in EBCDIC. The space the library occupies is always allocated adjacent to \$NUCLIB. Both \$NUCLIB and \$MSGLIB are built and accessed at the physical I/O level. The cataloged entries simply serve to reserve the space that they occupy.

OPERATING SYSTEM RESIDENT LIBRARY

This library is called \$OSRSDNTLIB. It contains all of the absolute load modules that are the resultant output of the Linkage Editor program in SYSGEN JOBTWO. The \$BLDRES program builds a program file using these modules in SYSGEN JOBTWO. This program file is called \$NUCLIB. You must save the \$OSRSDNTLIB if you intend to make absolute patches to your system. We suggest that you copy the library, rename it using your SYSGEN version number (see description of \$SGOBJ), and then purge the original file from your system. Store your copy off-line.

This library is never cataloged in the system CCAT. Because it is allocated on the system resident pack that you are creating, it must never be present in the PCAT of that pack when you begin executing the SYSGEN programs.

Because the final content of this library reflects the answers you have coded to the Checklist questions, its content will vary.

SYSTEM MESSAGE LIBRARY INPUT

This file is called \$MSGLIBINPUT. It contains all of the system messages organized in a conventional library manner. The file is used as input to the \$BLDRES program when the System Message Library (\$MSGLIB) is built. It must be on-line and cataloged in the central catalog while the SYSGEN programs are executing. However, it is not used at any other time.

SYSTEM MACRO LIBRARY

This library is called \$SYSMACLIB. It contains all of the macros provided by MRX/OS. You may add macros of your own to the \$SYSMACLIB. The contents of this library are modified by the SYSGEN programs. These modifications are determined by the answers you code in answer to the Checklist questions. When performing a Type 2 or Type 3 SYSGEN, you must have your current version of this library cataloged in the CCAT (central catalog). If you plan to use macros other than those defined within a particular source

module, you must have this library on-line and cataloged in the system CCAT so that this library can be accessed by the Assembler.

The Assembler allows specification of a macro library other than \$SYSMACLIB. However, the library specified must contain all of the macros called during a given assembly, because the Assembler cannot search more than one library for a macro definition.

SYSGEN MACRO BACKUP LIBRARY

This library is called SYSGEN-MAC-BACKUP. It contains all of the SYSGEN macros in their original state. The content of this library is never modified. This is your back-up library of the SYSGEN macros that are modified in \$SYSMACLIB as a part of the execution of the SYSGEN programs.

The members of the SYSGEN-MAC-BACKUP Library are:

- SGCNTRL
- RSDNTTYP
- SGCPDEFS
- SCNDCHN
- CARDMAC
- RDPCHMAC
- PRINTMAC
- TAPEMAC
- DISCMAC
- SGIODEFS
- TCOMMACH
- SGTCDEFS
- SGCLSDEF
- SYSINPUT
- SGASMDF
- SGOBDEF
- SDISCMAC
- SDISCDEF
- PTUCSMAC
- PTUCSDEF
- SGSPLDEF
- SGRSDDEF

SYSTEM LOAD LIBRARY

This library is called \$SYSLODLIB. It contains all of the relocatable load modules that you purchased. This library is cataloged in the CCAT of the MRX Distribution Pack. You must always have a load library on-line in order to execute MRX/OS system programs.

The members of the System Load Library are variable and for SYSGEN execution, must include:

- \$OSSGBPT
- \$OSSGCLP
- \$BLDRES
- LIBUTIL
- ASM
- LNKEDT

SYSGEN LOAD LIBRARY

This library is called \$SYSLODLIBA. It contains all of the relocatable load modules produced by the SYSGEN programs. This library is created each time the SYSGEN programs are executed if system programs that execute in a user partition are affected by Checklist specifications.

The library is cataloged in the CCAT while the SYSGEN programs are executing, but it must be merged with \$SYSLODLIB and purged from the system before your SYSGEN is complete. These procedures are part of JOBTWO. Refer to Appendix C for a description of the merge (SGMERGE) and purge (SGPURGE) procedures.

CHECKLIST DATA FILE

This file is called CHECKLISTDATA. It is a temporary file that contains your answers to the Checklist questions which you have punched into data cards.

SYSGEN DATA FILE

This file is called \$SYSGENDATA. It contains the same information contained in the CHECKLISTDATA file, but this file is structured as a virtual table. During JOBONE this file is allocated as a permanent file to facilitate error recovery procedures. It is purged from the system at the conclusion of JOBONE.

BLDRES DIRECTIVE FILE

This file is called \$BLDRESDIRECTIVES. It contains the directives that are used to control the building of the Nucleus Library (\$NUCLIB) and the System Message Library (\$MSGLIB). It must be on-line and cataloged in the central catalog while the SYSGEN programs are executing. However, it is not used at any other time.

4. OPERATING PROCEDURES FOR SYSTEM GENERATION

This section contains step-by-step procedures for generating a system. The first part of this section lists the steps that you must perform to generate a complete system. This type of SYSGEN is called a Type 1 or complete SYSGEN. The first group of instructions pertains to the multi-drive user (two or more disc drives in a system). The second group of instructions pertains to the user who has a one-drive system. The second half of this section lists the steps for modifying an existing system; modification SYSGEN's are of two types, called Type 2 and Type 3. To modify the system programs that form your resident operating system, perform a Type 2 SYSGEN. To modify the system programs that execute in a user partition, perform a Type 3 SYSGEN. The first group of instructions in this area pertains to the multi-drive user. The second group of instructions pertains to the one-drive user.

The libraries and files referred to in this section are fully described in Section 3 of this manual. Appendix C contains descriptions of the special procedures referenced in this section. Appendix B lists and describes the Control Language statements necessary to execute the SYSGEN programs. Those statements that are shaded are a required part of the job stream if you want to maintain the integrity of your input pack. When those statements are used, omit JOB2B and omit the four statements (beginning with //EXEC PGM=LIBUTIL) preceding the call to GENPROC1 in JOBONE. *The type of SYSGEN where you invalidate your input pack is called Procedure A, and the type where you maintain the integrity of your input pack is called Procedure B in this manual.*

Before you begin your SYSGEN, you must complete the **MRX/OS System Generation Checklist** and keypunch the information on cards to form your data deck for SYSGEN JOBONE. Next code and keypunch a //CALL statement for the SGALLOC procedure. This procedure will allocate space for and put the location of the Nucleus Library (\$NUCLIB) and the System Message Library (\$MSGLIB) into the pack catalog (PCAT) of your new system resident pack. Then code and keypunch a //CALL statement for the procedure SGCOPYA to copy the \$SYSLODLIB onto your new pack. Insert these //CALL statements immediately after the //JOB statement for JOBONE. Then code and keypunch //CALL statements for the procedures SGMERGE and SGPURGE. These procedures will first merge \$SYSLODLIBA with \$SYSLODLIB of your new system pack and then purge \$SYSLODLIBA from your new pack. Insert these //CALL statements after the //CALL PROC=GENPROC4 statement in SYSGEN JOBTWO. Now keypunch the Control Language statements shown in Appendix B for SYSGEN JOBONE and JOBTWO. Also keypunch the statements for JOB2A and JOB2B. JOB2A copies the \$SYSPROCLIB and the special SYSGEN procedures defined in Appendix C to the new system pack. It also allocates space for the System Error Log (\$SYSELOG) and the Job Accounting feature (\$OSSPLJA). JOB2B copies \$SGOBJ (SYSGEN Object Library) and \$SYSMACLIB (System Macro Library). Although JOBTHREE, JOBFIVE and JOBFIVE are not part of the formal SYSGEN process, if you plan to execute those jobs immediately after you complete your system generation, keypunch the Control Language statements for those jobs. Refer to Appendix B for the statements. Appendix C defines the procedures named SGALLOC, SGMERGE and SGPURGE.

After you have coded and keypunched all of this information, you will be ready to execute your SYSGEN jobs.

TYPE 1 SYSGEN (STEPS FOR PROCEDURES A AND B)

Perform the steps listed below in the order in which they appear to generate a full system. The first group pertains to the multi-drive user. The second group pertains to the one-drive user.

MULTI-DRIVE USER

STEP 1

Execute the Disc Initialize utility program to prepare your new system resident pack. Refer to the **MRX/OS Utility Programs Reference** manual for a description of the Disc Initialize program.

The following directions assume that you are generating your system onto an empty pack that has an empty central catalog.

STEP 2

Perform the IPL (Initial Program Load) procedure outlined below:

1. Mount on drive 0 the distribution pack or any existing system resident pack that has the following libraries in its central catalog: System Macro Library (\$SYSMACLIB), Procedure Library (\$SYSPROCLIB), Load Library (\$SYSLODLIB), SYSGEN Source Library (\$SYSGENSRCLIB), System Message Library Input File (\$MSGLIBINPUT), \$BLDRES Directives File (\$BLDRESDIRECTIVES) and SYSGEN Object Library (\$SGOBJ). Mount the pack that will contain your new system on drive 1.
2. Perform the AUTOLOAD (IPL) procedures described in the **MRX/OS Operating Procedures** manual.

STEP 3

Put the SYSGEN input deck for JOBONE in the system input reader. Push the READY pushbutton on the system input reader.

STEP 4

At the beginning of the various steps of SYSGEN JOBONE, the following messages should appear in the system print file (SYSOUT).

- BEGIN VERIFYING PARAMETERS AND BUILDING \$SYSGEN DATA
- BEGIN UPDATING SYSGEN MACROS
- BEGIN BUILDING GENPROC2
- FILE GENPROC2 IN THE PROCEDURE LIBRARY
- BEGIN BUILDING GENPROC4
- FILE GENPROC4 IN THE PROCEDURE LIBRARY

If all of the messages do not appear, refer to the Control Language statements for GENPROC1 shown in Appendix A for a description of the error recovery procedures that you must perform.

Because of the size of the assemblies in JOBTWO, it may be necessary to issue a SHARE command via the console to share the resident pack with the new system pack before initiating JOBTWO.

STEP 5

Initiate SYSGEN JOBTWO. (Remember to place JOBTWO in the reader before attempting to initiate it.)

STEP 6

Consult the listings of the assembly of the modules, \$CRDGEN1 and \$CRDGEN2, that created GENPROC2 in the print file for SYSGEN JOBONE to monitor the progress of SYSGEN JOBTWO. Each completed assembly and linkage editing step will be announced on the console. These steps are called processing milestones. If SYSGEN JOBTWO terminates in error, refer to Appendix A for appropriate error recovery procedures.

At the completion of SYSGEN JOBTWO, check the listings of the Assembler, Linkage Editor, \$BLDRES, and the SYSOUT file for error messages. Error recovery procedures are explained in Appendix A.

STEP 7

Remove the plug of the distribution pack (or your previously generated system resident pack that meets the requirements outlined in Step 2) from drive 0. Plug in your new system resident pack on drive 0, keeping the original pack on-line.

STEP 8

Perform the AUTOLOAD (IPL) procedures described in the **MRX/OS Operating Procedures** manual. During this time the \$JMDATA file for the new system is built. If the IPL fails, you can retry the IPL by entering 0003 in the M register before replying YES to the question, "RETAIN JOB QUEUE?" A YES reply must be given because this is not the initial autoload of the system. If you are building a two-partition system, you must perform a second IPL.

STEP 9

Initiate JOB2A. (Remember to place JOB2A in the reader before attempting to initiate it.) This job is a substep of JOBTWO which has been broken up into three jobs (JOBTWO, JOB2A, and JOB2B) for more efficient operation. JOB2A copies \$SYSPROCLIB which includes GENPROC1, GENPROC2, GENPROC3, GENPROC4, and all special SYSGEN procedures defined in Appendix C. It also allocates space for the System Error Log (\$SYSELOG). If you answer question 8 in the Output Spooler section of the Checklist with YES, you must insert the optional //DEF card shown in Appendix B into this job to allocate space for your Job Accounting feature.

STEP 10

Initiate JOB2B. (Remember to place JOB2B in the reader before attempting to initiate it.) JOB2B copies \$SGOBJ and \$SYSMACLIB onto your new system resident pack. (Skip JOB2B when using Procedure B.)

At this time you have completed the format system generation. You may now execute JOBTHREE, JOBFOUR, and JOBFIVE or you may execute these three jobs at another time.

STEP 11

Initiate JOBTHREE. (Remember to place JOBTHREE in the reader before attempting to initiate it.) JOBTHREE copies \$SYSOJLIB and all the libraries from your distribution pack that you specify as wanting on your new system resident pack. See Appendix B for a description of the Control Language statements needed to execute this job.

STEP 12

Initiate JOBFOUR. (Remember to place JOBFOUR in the reader before attempting to initiate it.) JOBFOUR uses the File-to-File Utility program (see **MRX/OS Utility Programs Reference** manual) to copy the message text files for the Assembler, COBOL, and RPG II programs as well as the \$BLDRESDIRECTIVES file and the \$COPYRIGHT file and all other files listed in Appendix B.

STEP 13

Initiate JOBFIVE. (Remember to place JOBFIVE in the reader before attempting to initiate it.) JOBFIVE produces a clean record of the contents of the \$SYSLODLIB on your new system resident pack. The listing does not show the names of the files that you copied in JOBFOUR using the File-to-File Utility program.

When you have completed your formal system generation, your input pack will be a scratch pack. It will no longer contain a valid operating system. If you want to maintain the use of the input pack as a system resident pack, follow Procedure B. See Appendix B for details.

ONE-DRIVE USER

Perform the steps listed below in the order in which they appear to generate a full system. You cannot build a completely new system in the \$NUCLIB under which you are executing. So if you perform your IPL from the standard \$NUCLIB location, you must build your new system in the alternate \$NUCLIB location. Similarly, if you build your new system in the standard location, you must perform your IPL from the alternate location.

STEP 1

Perform the IPL (initial program load) procedure outlined below:

1. Mount on drive 0 the distribution pack or any existing system resident pack that has the following libraries in its central catalog: System Macro Library (\$SYSMACLIB), Procedure Library (\$SYSPROCLIB), Load Library (\$SYSLODLIB), SYSGEN Source Library (\$SYSGENSRCLIB), System Message Library Input File (\$MSGLIBINPUT), \$BLDRES Directives File (\$BLDRESDIRECTIVES) and SYSGEN Object Library (\$SGOBJ).
2. Perform the AUTOLOAD (IPL) procedures described in the **MRX/OS Operating Procedures** manual.

STEP 2

Put the SYSGEN input deck for JOBONE in the system input reader. Push the READY pushbutton on the system input reader.

STEP 3

At the beginning of the various steps of SYSGEN JOBONE, the following messages should appear in the system print file (SYSOUT).

- BEGIN VERIFYING PARAMETERS AND BUILDING \$SYSGEN DATA
- BEGIN UPDATING SYSGEN MACROS
- BEGIN BUILDING GENPROC2
- FILE GENPROC2 IN THE PROCEDURE LIBRARY
- BEGIN BUILDING GENPROC4
- FILE GENPROC4 IN THE PROCEDURE LIBRARY

If all of the messages do not appear, refer to the Control Language statements for GENPROC1 shown in Appendix A for a description of the error recovery procedures that you must perform.

STEP 4

Initiate SYSGEN JOBTWO. (Remember to place JOBTWO in the reader before attempting to initiate it.)

STEP 5

Consult the listings of the assembly of the modules, \$CRDGEN1 and \$CRDGEN2, that created GENPROC2 in the print file for SYSGEN JOBONE to monitor the progress of SYSGEN JOBTWO. Each completed assembly and linkage editing step will be announced on the console. These steps are called processing milestones. If SYSGEN JOBTWO terminates in error, refer to Appendix A for appropriate error recovery procedures.

At the completion of SYSGEN JOBTWO, check the listings of the Assembler, Linkage Editor, \$BLDRES, and the SYSOUT file for error messages. Error recovery procedures are explained in Appendix A.

STEP 6

Perform the AUTOLOAD (IPL) procedures described in the **MRX/OS Operating Procedures** manual. During this time the \$JMDATA file for the new system is built. If the IPL fails, you can retry the IPL by entering 0003 in the M register before replying YES to the question, "RETAIN JOB QUEUE?" A YES reply must be given because this is not the initial autoloading of the system.

STEP 7

Initiate JOB2A. (Remember to place JOB2A in the reader before attempting to initiate it.) This job is a substep of JOBTWO which has been broken up into three jobs (JOBTWO, JOB2A, and JOB2B) for more efficient operation. JOB2A copies \$SYSPROCLIB which includes GENPROC1, GENPROC2, GENPROC3, GENPROC4, and all special SYSGEN procedures defined in Appendix C. It also allocates space for the System Error Log (\$SYSELOG). If you answer question 8 in the Output Spooler section of the Checklist with YES, you must insert the optional //DEF card shown in Appendix B into this job to allocate space for your Job Accounting feature.

STEP 8

Initiate JOB2B. (Remember to place JOB2B in the reader before attempting to initiate it.) JOB2B copies \$SGOBJ and \$SYSMACLIB onto your new system resident pack. (Skip JOB2B when using Procedure B.)

TYPE 2 AND TYPE 3 SYSGENS (STEPS FOR PROCEDURES A AND B)

If you are performing a Type 2 SYSGEN, complete the following Checklist sections: SYSGEN Control, System Control, Output Spooler, I/O Devices*, and Telecommunications. Key punch this data. Also key punch all the Control Language statements listed in Appendix B that are needed to execute the SYSGEN programs.

If you are performing a Type 3 SYSGEN, complete the following Checklist sections: SYSGEN Control, Control Language Services, Assembly Language, and COBOL. Key punch this data. Also key punch all of the Control Language statements listed in Appendix B except the //CALL statement to GENPROC3 in SYSGEN JOB2. Insert your data deck in the job stream as shown in Appendix B.

Remember that the questions you do not answer when coding your data for a Type 2 or Type 3 SYSGEN default to the values you specified when you generated the system you are about to modify. These defaults will not be the standard defaults printed on the Checklist except in cases where you previously selected the standard default. It is not possible to modify back to the standard defaults without performing a Type 1 SYSGEN.

* If any changes are made that affect the device address of the system input reader, question 2 in the Control Language Services section of the Checklist must also be changed.

Type 2 and Type 3 SYSGEN's require that the following files be cataloged in CCAT and on-line: \$SYSMACLIB containing the SYSGEN macros that reflect the system you are modifying, \$SGOBJ containing the SYSGEN object modules that reflect the system you are modifying*, \$SYSLODLIB, \$MSGLIBINPUT, \$BLDRESDIRECTIVES, \$SYSPROCLIB, and \$SYSGENSCRLIB. For a Type 2 SYSGEN, the following files must not be in the PCAT of the pack being modified: \$SYSLODLIBA and \$OSRSDNTLIB. For a Type 3 SYSGEN, the \$SYSLODLIBA must not be in the PCAT of the pack being modified.

Use the Catalog Display procedure described in the **MRX/OS Utility Programs Reference** manual for this purpose. If the files are on the PCAT, use the SGPURGE procedure to get rid of them. The SGPURGE procedure is described in Appendix C.

If you are performing your modifications on your existing pack, follow the step-by-step procedures outlined for the one-drive user for Type 1 SYSGEN's. If you are modifying your system and building the modified system on a second pack, follow the step-by-step procedures outlined for the multi-drive user.

Type 2 and Type 3 SYSGEN's require considerably less time than Type 1 SYSGEN's because the number of assemblies is greatly reduced. However, they can only be used to upgrade your system; for example, adding additional features via the Checklist.

* This file may have been renamed by you at the end of your previous SYSGEN. If so, be sure to supply the correct name in answer to question 3 in the SYSGEN Control section of the Checklist.

5. DESCRIPTION OF CHECKLIST QUESTIONS

This section is your guideline for completing the System Generation Checklist. The headings and question numbers used in this section are exactly the same as those used in the Checklist.

SYSGEN CONTROL

1. What type of SYSGEN are you performing? Specify one: 1, 2, 3. Default is 1.

If you answer 1, it means that you are performing a complete system generation so you must read all of the questions in the Checklist and answer those that describe your system. If you answer 2, it means that you want to change the resident portion of your existing operating system, and you need only answer the questions in the first five sections of the Checklist. If you answer 3, it means that you want to add to or change features in the Programming Services portion of your existing operating system*. To do this, you need answer only the relevant questions in the SYSGEN Control section and the questions in the last three sections of the Checklist.

2. Which resident operating system do you have? Specify one: MINIMUM, RESIDENT EXTENSION. Default is RESIDENT EXTENSION.

If you answer MINIMUM, it means that you have the basic 8KB operating system. If you answer RESIDENT EXTENSION, it means that you have the basic 10KB operating system. Table D-1 in Appendix D of the Checklist shows the byte requirements of the various features that can be added to both types of operating systems. Table D-2 shows the byte requirements for the Output Spooler printer that can be added to the Resident Extension system. Table D-3 shows the byte requirements for telecommunications lines.

3. What is the name of your library that contains the file of object modules that you want updated by the SYSGEN programs? The name must be a 1 to 8 character alphanumeric name that matches the name you supply on the //CALL statement in GENPROC3. Default is \$SGOBJ.

If you accept the default, your library containing the object modules will have the same name as the library on the distribution pack that contains these modules. If you decide to rename this library for your operating system, you must make sure that the name you specify matches the name you supply on the //CALL statement in GENPROC3. Appendix B shows the //CALL statement for GENPROC3.

In Section 3 of this manual is a description of the contents and purpose of the \$SGOBJ library.

*A change to question 2 in the Control Language Services section is an exception to this rule.

SYSTEM CONTROL

1. What is the main memory size of your computer? Specify one: 16KB, 24KB, 32KB, 48KB, 64KB.

When answering this question, specify the amount of central storage you have in your configuration. Because this question has no default, you must code an answer to it when you perform a Type 1 SYSGEN. The link edit map, produced when the resident operating system is linked, gives the exact number of bytes used by the resident operating system. The external label in the map of the highest byte in the resident operating system is \$OSEND.

2. Does your system have the hardware Error Correction (ECC) feature? Specify one: YES, NO. Default is NO.

This feature is not available on the MRX/40. It provides automatic correction of memory errors. If you want a log of the errors that have been corrected, the software resident byte requirement is 100 bytes. The log is allocated via a procedure described in Appendix C. This log is useful as a preventative maintenance tool for field service representatives.

3. If you have a two-partition system, specify in bytes the size of partition 1. Otherwise omit this question. No partition can be smaller than 4096 bytes.

A two partition system gives you the ability to have two jobs running concurrently, which increases the productivity of your system. The method you should use to calculate the size of your partitions is explained in the Checklist.

4. If you have the Resident Extension operating system, specify whether or not you have selected Performance Option 1. Specify one: YES, NO. Default is NO.

This option expands the resident area of your system to increase job performance. This expansion makes it possible to eliminate one look-up disc access for every \$NUCLIB load operation. You can select both this option and Performance Option 2 (see question 5 in this section). However, the combination of the two options does not unconditionally eliminate one disc access per \$NUCLIB load operation. Table D-1 in Appendix D of the Checklist shows the byte requirement of this option.

5. If you have the Resident Extension operating system, specify whether or not you have selected Performance Option 2. Specify one: YES, NO. Default is NO.

This option also expands the resident area of your system to increase job performance. This expansion makes it possible to eliminate all look-up disc accesses for \$NUCLIB elements which make up an associative list. The contents of this associative list are managed dynamically and contain load control information for the most recently referenced \$NUCLIB elements. The size of the associative list is

automatically increased from the nominal size if you select the TCOM option or the Output Spooler option. If both TCOM and Output Spooler options are selected, the largest possible associative list is generated. Table D-1 in Appendix D of the Checklist shows the byte requirement of this option.

OUTPUT SPOOLER

1. Does your system have the Output Spooler feature? Specify one: YES, NO. Default is NO.

This feature allows you to spool the output of several jobs and print or punch them at a later time. It is an increased performance feature. The byte requirement of this feature is 4096 bytes of resident memory plus 452 bytes for required system labels and buffers. Additional buffer and FDT space is required depending on the total number of spooled devices and the blocking factor. Refer to Tables D-1 and D-2 in Appendix D of the **MRX/OS System Generation Checklist** for details.

2. If your system has the Output Spooler feature, what is the maximum number of printers that you want dedicated to the Spooler at any one time? Specify a number in the following range: 1-14. Default is 1.

The number of device addresses coded in answer to question 4 in this section must not exceed the number of devices specified in answer to this question. The buffer space required for printers is determined from the answer to this question and the answer to question 5 in this section. Refer to Table D-2 in Appendix D of the **MRX/OS System Generation Checklist** for specific details. An answer greater than 1 to this question means that you must answer YES to question 1 in the I/O Devices section.

3. If you have a reader/punch, do you want it dedicated to output spooling? Specify one: YES, NO. Default is NO.

In order to answer YES to this question, you must also answer YES to question 3 in the I/O Devices section. The device will be released to the system if the Spooler is not initiated or by the use of the UNSPOOL command. If this reader/punch is your standard input device, it can be assigned to the Spooler after initialization by using the SPOOL command.

4. If your system has the Output Spooler, specify a list of the device addresses for the printers that you want assigned to the Spooler. Separate the device addresses one from another with a comma. Default is 21E.

All device addresses specified in answer to this question must also be specified in answer to question 4 in the I/O Devices section of the Checklist. Note also that the device type as well as the device address is a part of the answer to question 4 in the I/O Devices section. The number of printers that can be assigned is conditional upon the blocking factor that you select. See Table D-2 in Appendix D of the **MRX/OS System Generation Checklist** for details. The unit table entries for all specified

devices will be marked as assigned to the Spooler when the system is initialized. The devices will be released to the system if the Spooler is not initiated or by the use of the UNSPOOL command. The number of printers assigned is conditional upon the blocking factor selected from Table D-2 in Appendix D of the **MRX/OS System Generation Checklist**.

5. If your system has the Output Spooler, specify the printer blocking factor. Refer to Table D-2 in Appendix D of the **MRX/OS System Generation Checklist** for the ranges allowed. Default is 1.

The value that you specify in answer to this question is the number of blocks that the Output Spooler will read (using multiblock reads) when processing unblocked, spooled print files. The answer to this question also determines the maximum block size that the Spooler will accept. (The maximum block size is 142 multiplied by the printer blocking factor.)

6. If your system has the Output Spooler specify the punch blocking factor. Specify a number in the following range: 1-2. Default is 1.

The value that you specify in answer to this question is the number of blocks that the Output Spooler will read (using multiblock reads) when processing unblocked, spooled punch files.

7. If your system has the Output Spooler, what is the maximum number of files that you want to allow queued on the Spooler queue? Specify a number in the following range: 45-255. Default is 45.

The answer to this question is the maximum number of files that can be queued to the Spooler at any one time.

8. If your system has the Output Spooler, do you want the Job Accounting feature? Specify one: YES, NO. Default is NO.

If you answer this question YES, an accounting file is allocated on an account record for each print and/or punch file written. Each account record contains a job name, file name, number of lines printed or number of cards punched, date, and time of completion. This feature requires 250 additional bytes.

I/O DEVICES

1. Does your system have the Second Selector Channel feature? Specify one: YES, NO. Default is NO.

The software byte requirement for this feature is 150 bytes. The Second Selector Channel has eight additional control unit positions. I/O channel units, other than those supported by integrated attachment features, can be connected to either the First Selector Channel or the Second Selector Channel if you answer YES to this question.

The second digit of the device address should be increased by a factor of 2 to denote additional external controllers. The exception to this rule applies only to the first printer adapter (controller) which should be assigned to device address 21x.

Therefore, addresses available to denote additional external controllers on Selector Channel 1 are:

22x	2Ax
24x	2Cx
26x	2Ex
28x	

This rule also applies to controllers attached to Selector Channel 2. Device addresses available are:

10x	16x	1Cx
12x	18x	1Ex
14x	1Ax	

2. Do you have a card reader attached to your integrated card reader attachment feature? Specify one: YES, NO.

If you specify YES, the SYSGEN programs will automatically assign your card reader to device address 204. The device type will be 23. Only one reader can be attached to either the MRX/40 or the MRX/50. *Because this question has no default, it must be answered.*

3. Do you have a card reader/punch attached to your integrated card reader/punch attachment feature? Specify one: YES, NO.

Because this question has no default, it must be answered. If you specify YES, the SYSGEN programs will automatically assign your buffered reader/punch to device address 20C. The device type will be 11. Only one reader/punch can be attached to either a MRX/40 or a MRX/50.

4. Specify, on separate data cards, the device type and device address of each line printer in your configuration. Default is 51,21E.

Your system output device must always be assigned to device address 21E. Use a comma to separate the device type from the device address. If you have more than one printer, code data cards for each one, including the default (51,21E), even if your first printer is a buffered printer located at device address 21E. Defaults are no longer valid once you code an answer to the question. If you want the default plus another printer described to the system, code two cards, one containing 51,21E and another containing the device type and device address of the second printer. Appendix B in the System Generation Checklist lists supported device types. If you begin your device address with a 1, it means that you must have answered YES to question 1 in this section.

5. Specify, on separate data cards, the device type and device address of each magnetic tape drive in your configuration.

The second digit of your three-digit hexadecimal device address specifies the external controller that your device is attached to. We recommend that you use increments of two when specifying your controllers. A maximum of four tape drives can be attached to one tape controller. If you had seven tape drives, your device addresses could be 221, 222, 223, 224 for the first four drives and 245, 246, 247, or 241, 242, 243 for the next three drives (seven total). If the first digit of your device address is a 1, it means that you must have answered YES to question 1 in this section. Do not attempt to attach 1600 bpi drives to a MRX 3237 Model 12 Controller (800 bpi controller).

6. Specify, on separate data cards, the device type and device address of each disc drive in your configuration. Defaults are 70,300 and 70,301.

The device addresses of your discs must begin with 30. Addresses available are 300 – 307. You must have one disc located at device address 300, because 300 is the address of the system disc. A maximum of eight discs can be configured.

7. Do you have the disc Seek-on-Position software feature in your system? Specify one: YES, NO. Default is NO.

This feature saves time when seeking records in a multi-disc system. The feature requires 30 bytes of resident memory. It is available under both resident operating systems.

8. If you want automatically shared discs at IPL (initial program load) time, enter the addresses of the discs you want shared. Separate the addresses one from another using a comma. If you do not want this feature, omit this question.

If later you do not want the devices that you specify in answer to this question shared, you can use the UNSHAR operator console directive to change the status of any device that you specified as shared in answer to this question. The format of this directive is:

```
*UNSHAR u[,vvvvvv]
```

where:

u is the disc drive unit number (1-7)

[,vvvvvv] is an optional 6-character volume identifier of the volume specified as mounted on the drive. This variable should be entered only if the operator chooses to verify the currently mounted volume.

All devices specified as shared in answer to this question must be available during IPL. Do not attempt to use the UNSHAR directive until after IPL. The device address of the system resident (300) must not be specified in answer to this question.

9. What UCS (universal character set) images do you want associated with each of your printers? Enter the printer address and the associated image for each UCS printer. Separate the printer address from the character image with a comma. Separate each pair one from another with a comma. Example: 21E, PN, 21A, XX.

Because this question has no default, it must be answered if you have any UCS printers configured in your system.

TELECOMMUNICATIONS

1. How many telecommunications lines do you have in your configuration? For MRX/40 Resident Extension users, specify a decimal number from 1 through 7. For MRX/50 Resident Extension users, specify a decimal number from 1 through 15.

This question is for documentation purposes. It also serves as a reminder that the number you specify here and the number of data cards you code to answer question 2 should be the same.

2. Specify, on separate data cards, the device type and device address of each telecommunications line in your configuration. Use a comma to separate the device type from the device address.

As with I/O devices, this is the way you make your telecommunications network known to the system. The device addresses available for use are 001 through 00F. The device address must correspond to the line number. For example, line 2 must be assigned to device address 002. Appendix B in the System Generation Checklist lists supported device types. Appendix D in the same document lists the byte requirements for each type.

3. If you have the Resident Extension operating system, do you want the logical communications capability in addition to the physical communications system that you have just configured? Specify one: YES, NO. Default is NO.

This feature requires 1650 bytes of resident memory. It also uses space in the user partition. The Minimum system does not support the logical communications capability. For a complete description of the logical communications feature, refer to the **MRX/OS Telecommunications Reference** manual.

CONTROL LANGUAGE SERVICES

1. What is the maximum number of entries that you will allow in your job queue? Specify a number from 1 – 62. Default is 62.

The SYSGEN programs allocate one track on the disc for the job queue. The minimum block size on the disc is 18, so 62 job entries will fit on one track. That is why the upper limit has been specified as 62. For each job that is in the queue, a SYSIN file is built.

2. What device are you using as your standard input device for job control purposes? Specify one: READER, READER/PUNCH. Default is READER.

The system must know which of your input devices you are using as your system standard input device. Device address 21E must always be assigned to the standard output device (see question 4 in the I/O Control section of the Checklist) so there is no separate question about the standard output device.

3. What value do you want Control Language Services to supply for the keyword PRIORITY when the keyword is omitted from //JOB statements? Specify a number from 1 – 9. Default is 1.

One is the lowest priority and 9 is the highest priority. If your site has many jobs that have different levels of priority, we suggest that you accept the default number 1, and encourage your user programmers to always specify the PRIORITY keyword on all //JOB statements.

4. What value do you want Control Language Services to supply for the keyword TIME when the keyword is omitted from //EXECUTE statements? Specify a number from 1 through 1439. Default is 1439 (23 hours, 59 minutes).

When answering this question, specify the maximum amount of time in minutes that you will allow a job to execute before the system aborts it. This parameter acts as a safety factor for programs that get hung in a loop and tie up the system, so choose the value you specify here carefully. We recommend that the programmer use this parameter on programs that have not been completely debugged.

5. Do you want the keyword USER to be a required keyword on all //JOB statements? Specify one: YES, NO. Default is NO.

This parameter gives you an accounting tool if you answer the question YES because it allows you to keep track of the time used by all programs.

6. Which entry do you want Control Language Services to supply for the keyword SPL= (SPOOL=) when the keyword is omitted from //ROUTE statements? Specify one: YES, NO. Default is NO.

If SPL=NO is selected, the physical unit record device (printer or punch) directly receives the output from the program. If SPL=YES is selected, the output from the program goes to a disc file that serves as a temporary buffer. Spooled files are then later printed and/or punched by the Output Spooler.

7. What value do you want Control Language Services to supply for the first positions/parameter for the keyword NUM= (primary allocation) when the keyword is omitted from the //ROUTE statement? Specify a number in the following range: 1-32767. Default is 1000. Do not include a comma in your response.

The first positions/parameter indicates the size of the temporary file when it is initially allocated on disc.

8. What value do you want Control Language Services to supply for the second positions/parameter for the keyword NUM= (expand increment) when the keyword or this parameter is omitted from the //ROUTE statement? Specify a number in the following range: 1-32767. Default is 1000. Do not include a comma in your response.

This parameter indicates the size of the incremental increase that is used for dynamic expansion of the file specified by the first positions/parameter.

9. What value do you want Control Language Services to supply for the keyword NUM= when the keyword is omitted from //DATA statements? Specify a number in the following range: 1-32767. Default is 1000. Do not include a comma in your response.

This keyword specifies the maximum number of records that will be accepted for a given //DATA file. If the number of records in the file exceeds the answer to this question, the file will not be expanded and the job will be aborted. An answer that is excessively large will cause considerable waste of disc space.

10. What character set do you want to be the default for the UCS= parameter on //ROUTE statements? Answer this question only if you have one or more printers in your configuration that have the universal character set. Default is PN.

If you do not have any printers with the UCS feature, the PN default will not in any way affect printer assignment. If you do have UCS printers, printer assignment follows the algorithm shown in Table 5-1.

Table 5-1. UCS Printer Assignment Algorithm

Number	Decision/Action	Yes	No	Uncond.
1	Was DEV= coded?	2	8	
2	Is that device available?	3	11	
3	Was UCS= coded?	4	7	
4	Is this device a UCS printer?	5	12	
5	Is correct UCS mounted?	7	6	
6	Issue 'MOUNT UCS' message.			5
7	Assign printer.			Exit
8	Was UCS= coded?	9	13	
9	Is printer available with this UCS already mounted?	7		
10	Is any UCS printer available?	5	11	
11	Shortage of available devices.			Exit
12	Unable to honor UCS request.			Exit
13	Is any standard printer available?	7	11	
14	Is any printer available?	7	11	

ASSEMBLER LANGUAGE

1. How many cards are there in your average Assembler Language program? Specify a number from 1-65535. Do not include a comma in your answer. Default is 1000.

Depending upon the needs of your installation, specify a number here that reflects the average length of most Assembler Language programs. The maximum number (65,535) reflects the word length limit of the MRX computer. It is the upper limit for the number of cards in your program. This parameter is used for internal file allocation. By using the //PAR statement, a programmer can override the SYSGEN parameter specified in answer to this question at execution time.

2. In which column do you want the Assembler to quit processing Assembler Language statements? Specify a number from 41-120. Default is 72.

The minimum number (41) is the natural limit for the length of the language statements. The maximum number (120) allows the telecommunications user to structure his input as efficiently as possible. The variance is due to the type of input device used. Card image input has restrictions that are different from telecommunications input.

3. How many lines per page do you want printed on your Assembler Language listing? Default is 56.

Specify a number that meets the needs of your installation. This number will vary depending upon the size of your printer paper, the number of lines that your printer prints, and whether or not you want logical pages to exceed physical pages.

4. Does your computer have the floating-point hardware feature? Specify one: YES, NO. Default is NO.

COBOL LANGUAGE

1. How many cards are there in your average COBOL program? Specify a number from 1-65535. Do not include a comma in your answer. Default is 1000.

Specify a number that reflects the average length of most COBOL programs. The average program length is used by the compiler for internal file allocation. By using the //PAR statement, a programmer can override the SYSGEN parameter, specified in answer to this question, at execution time. The maximum number (65,535) reflects the word length limit of the MRX computer.

2. In which column do you want the COBOL compiler to quite processing COBOL source statements? Specify a number from 41-120. Default is 72.

The minimum number (41) is the natural limit for the length of compiler statements. The maximum number (120) allows the telecommunications user to structure his input as efficiently as possible. The variance is due to the type of input device used. Card image input has restrictions that are different from telecommunications input.

3. How do you want your COBOL USASI errors handled? Specify one: WARNING, FATAL. Default is FATAL.

Answer this question WARNING or accept the default FATAL, depending upon the needs of the COBOL users. This is the only time (SYSGEN-time) that you can override the fatal nature of USASI errors.

A. SYSTEM GENERATION PROCEDURES AND ERROR RECOVERY

This appendix contains the four basic procedures that form SYSGEN JOBONE and JOBTWO. They are called GENPROC1, GENPROC2, GENPROC3, and GENPROC4. Within the coding shown for each procedure, each //EXECUTE statement represents a restart point in the SYSGEN job. The //TELL statements refer to the program segment that appears after the first //EXECUTE statement that precedes each //TELL statement. Although the //TELL statement is shown at the end of the program segment, the actual message will appear on the operator console before the segment is executed.

For example, the first program segment of GENPROC1 begins with an //EXECUTE statement, is followed by three //DEF statements, and ends with a //TELL statement. That //TELL statement refers back to the //EXECUTE statement. The message contained in the //TELL statement will be printed out on the operator console before this program segment is executed.

GENPROC1

```
//EXEC   PGM=$OSSGBPT,NAME=GENSTEP1

//DEF    ID=INPUT,FIL=CHECKLISTDATA,STA=T

//DEF    ID=OUTPUT,FIL=$SYSGENDATA,STA=(P,O),CSD=NO,SIZE=256,BLK=1,
//       CAT=YES,VER=YES,NUM=20,CON=YES

//DEF    ID=LIST,DEV=PRINTER

//TELL   OP=BEGIN VERIFYING PARAMETERS AND BUILDING $SYSGENDATA

//EXEC   PGM=$OSSGCLP,NAME=GENSTEP2

//DEF    ID=DATA,FIL=$SYSGENDATA,STA=(P,I)

//DEF    ID=MACRODEF,FIL=$SYSMACLIB,STA=(P,O)

//DEF    ID=LIST,DEV=PRINTER

//TELL   OP=BEGIN UPDATING SYSGEN MACROS

//EXEC   PGM=ASM,NAME=GENSTEP3,TIME=30

//PAR    IMEM=$CRDGEN1,OBJECT=NO,OMEM2=GENPROC2

//DEF    ID=LIST,DEV=PRINTER

//DEF    ID=INPUT,FIL=$SYSGENSRCLIB,STA=(P,I)

//DEF    ID=OUTPUT2,FIL=$SYSGENPROC2,STA=(P,O),CSD=NO,VER=YES,BLK=1,
//       SIZ=80,CAT=YES,NUM=80,CON=YES
```

```

//TELL    OP=BEGIN BUILDING GENPROC2
//EXEC    PGM=LIBUTIL
//PAR     COMMAND=UPDATE,MEM=(,,GENPROC2,PRO)
//DEF     ID=LIST,DEV=PRINTER
//DEF     ID=SEQIN,FIL=$SYSGENPROC2,STA=(P,I)
//DEF     ID=OUTPUT,FIL=$SYSPROCLIB,STA=(P,O)
//TELL    OP=FILE GENPROC2 IN THE PROCEDURE LIBRARY
//EXEC    PGM=ASM,NAME=GENSTEP4,TIME=30
//PAR     IMEM=$CRDGEN2,OBJECT=NO,OMEM2=GENPROC4
//DEF     ID=LIST,DEV=PRINTER
//DEF     ID=INPUT,FIL=$SYSGENSRCLIB,STA=(P,I)
//DEF     ID=OUTPUT2,FIL=$SYSGENPROC4,STA=(P,O),CSD=NO,VER=YES,BLK=1,
//        SIZ=80,CAT=YES,NUM=140,CON=YES
//TELL    OP=BEGIN BUILDING GENPROC4
//EXEC    PGM=LIBUTIL
//PAR     COMMAND=UPDATE,MEM=(,,GENPROC4,PRO)
//DEF     ID=LIST,DEV=PRINTER
//DEF     ID=SEQIN,FIL=$SYSGENPROC4,STA=(P,I)
//DEF     ID=OUTPUT,FIL=$SYSPROCLIB,STA=(P,O)
//TELL    OP=FILE GENPROC4 IN THE PROCEDURE LIBRARY
//EXEC    PGM=UTSSPF
//PAR     PURGE=$SYSGENDATA
//PAR     PURGE=$SYSGENPROC2
//PAR     PURGE=$SYSGENPROC4

```

GENPROC2

The following Control Language statements make up the second SYSGEN procedure called GENPROC2. This is the first procedure in SYSGEN Job 2. Your output listing will show the statements.

The set of statements enclosed in braces is executed for every system source module that is reassembled at SYSGEN time. Source modules that must be reassembled are determined by the answers you code when you complete the System Generation Checklist.

```
//EXEC PGM=ASM,TIME=1440
//PAR IMEM=&MEMNAME,OMEM1=&MEMNAME*
//DEF ID=INPUT,FIL=$SYSGENSRCLIB,STA=(P,I)
//DEF ID=OUTPUT1,FIL=&OBJNAME**,STA=(P,O)
//DEF ID=LIST,DEV=PRINTER
//TELL OP=BEGIN ASSEMBLY OF &MEMNAME
```

GENPROC3

The following Control Language statements make up the third SYSGEN procedure. This is the second procedure in SYSGEN JOBTWO. This procedure is the optional SYSGEN procedure. Your output listing will show the statements.

```
//DEC OBJNAME=$SGOBJ,FULLGEN=YES,STDNUC=YES,RSLOC=YES,
//      SYSVOL,DEVADDR
//EXEC PGM=LNKEDT,TIME=1440
//PAR PGM=$OSRSDNT,LST=XREF,LSD=NO,SRH=NO,ORG='0000'
//DEF ID=INPUT,FIL=&OBJNAME,STA=P,VOL=&SYSVOL**
//DEF ID=OUTPUT,FIL=$OSRSDNTLIB,STA=(P,O),VER=YES,CAT=NO,
//      CON=YES,VOL=&SYSVOL,LOC=$RSLOC,SIZ=256,BLK=1,CSD=NO,
//      NUM=4000
//DEF ID=LIST,DEV=PRINTER
//TELL OP=BEGIN LINK-EDIT OF RESIDENT OPERATING SYSTEM -- PART 1.
//EXEC PGM=LIBUTIL,TIME=1440
//PAR COMMAND=COPY,MEM=($OSRSDNT,ABS,$OSCEPL,ABS)
//DEF ID=INPUT,FIL=$OSRSDNTLIB,STA=(P,I),VOL=&SYSVOL
//DEF ID=OUTPUT,FIL=&OBJNAME,STA=(P,O),VOL=&SYSVOL
//DEF ID=LIST,DEV=PRINTER
```

*&MEMNAME is the name of the particular source module that is being assembled at this time.

**&OBJNAME is the name of the library containing the SYSGEN object modules specified in answer to question 3 of the SYSGEN Control section of the Checklist or \$SGOBJ if the default has been selected.

```

//TELL      OP=BEGIN COPY OF COMPOSITE ENTRY POINT LIST, PART 1.
//EXEC      PGM=LNKEDT,TIME=1440
//PAR       PGM=$OSSVCTR,LST=XREF,LSD=NO,SRH=ABS,ORG=$CPINIT
//DEF       ID=INPUT,FIL=&OBJNAME,STA=P,VOL=&SYSVOL
//DEF       ID=OUTPUT,FIL=$OSRSDNTLIB,STA=P,VER=YES,VOL=&SYSVOL
//DEF       ID=LIST,DEV=PRINTER
//TELL      OP=BEGIN LINK-EDIT OF OPERATING SYSTEM -- SVC TRANSIENTS
//EXEC      PGM=LIBUTIL,TIME=1440
//PAR       COMMAND=COPY,MEM=($OSSVCTR,ABS,$OSCEPL2,ABS)
//DEF       ID=INPUT,FIL=$OSRSDNTLIB,STA=(P,I),VOL=&SYSVOL
//DEF       ID=OUTPUT,FIL=&OBJNAME,STA=P,VOL=&SYSVOL
//DEF       ID=LIST,DEV=PRINTER
//TELL      OP=BEGIN COPY OF COMPOSITE ENTRY POINT LIST, PART 2.
//EXEC      PGM=LNKEDT,TIME=1440
//PAR       PGM=SPTABLE,LST=XREF,SRH=ABS,LSD=NO,ORG='0000'
//DEF       ID=INPUT,STA=P,FIL=&OBJNAME,VOL=&SYSVOL
//DEF       ID=OUTPUT,FIL=$OSRSDNTLIB,STA=(P,O),VER=YES,CAT=NO,
//          CON=YES,VOL=&SYSVOL,LOC=NO,SIZ=256,BLK=1,CSD=NO
//DEF       ID=LIST,DEV=PRINTER
//TELL      OP=BEGIN LINKAGE EDITING OF SPTABLE
//EXEC      PGM=LNKEDT,TIME=1440
//PAR       PGM=SPTABLE2,LST=XREF,SRH=ABS,LSD=NO,ORG='0000'
//DEF       ID=INPUT,STA=P,FIL=&OBJNAME,VOL=&SYSVOL
//DEF       ID=OUTPUT,FIL=$OSRSDNTLIB,STA=(P,O),VER=YES,CAT=NO,
//          CON=YES,VOL=&SYSVOL,LOC=NO,SIZ=256,BLK=1,CSD=NO
//DEF       ID=LIST,DEV=PRINTER
//EXEC      PGM=LNKEDT,TIME=1440
//PAR       PGM=SPTABLE3,LST=XREF,SRH=ABS,LSD=NO,ORG='0000'
//DEF       ID=INPUT,STA=P,FIL=&OBJNAME,VOL=&SYSVOL
//DEF       ID=OUTPUT,FIL=$OSRSDNTLIB,STA=(P,O),VER=YES,CAT=NO,
//          CON=YES,VOL=&SYSVOL,LOC=NO,SIZ=256,BLK=1,CSD=NO
//DEF       ID=LIST,DEV=PRINTER
//EXEC      PGM=$BLDRES,TIME=1440
//PAR       FULLGEN=&FULLGEN,STNDLOC=&STDNUC,DEVADDR=&DEVADDR
//DEF       ID=INPUT1,FIL=$OSRSDNTLIB,STA=(P,I),VOL=&SYSVOL
//DEF       ID=INPUT2,FIL=$MSGLIBINPUT,STA=(P,I)
//DEF       ID=LIST,DEV=PRINTER
//DEF       ID=DRECTVES,FIL=$BLDRESDIRECTIVES,STA=(P,I)
//DEF       ID=OBJLIB,FIL=&OBJNAME,STA=(P,O),VOL=&SYSVOL

```

```

//TELL    OP=BEGIN BUILDING NUCLIB
//EXEC    PGM=LNKEDT,TIME=1440
//PAR     PGM=SYSPATCH,LST=XREF,PRIV=YES,SRH=ABS,LSD=(&OBJNAME)
//DEF     ID=INPUT,FIL=$SYSOBJLIB,STA=P
//DEF     ID=LIB1,FIL=&OBJNAME,STA=P,VOL=&SYSVOL
//DEF     ID=OUTPUT,FIL=$SYSLODLIB,STA=P,VOL=&SYSVOL,CAT=NO
//DEF     ID=LIST,DEV=PRINTER
//TELL    OP=BEGIN LINKAGE EDITING OF SYSPATCH

```

GENPROC4

```

{ //EXEC PGM=LNKEDT,TIME=1440
  [//PAR PGM=$JMIR,PRIV=YES,SRH=ABS,LST=XREF,LSD=NO
  //DEF ID=INPUT,FIL=&OBJNAME,STA=(P,I)*
  [//DEF ID=OUTPUT,FIL=$SYSLODLIBA,STA=(P,O),CSD=NO,VER=YES,CON=YES,
  // BLK=1,SIZ=256,CAT=YES,NUM=400]
  //DEF ID=LIST,DEV=PRINTER
  [//TELL OP=BEGIN LINKAGE EDITING OF $JMIR]
}

{ //EXEC PGM=LNKEDT,TIME=1440
  [//PAR PGM=ASM,POOLSIZ='101',PRIV=YES,LST=XREF,LSD=NO]
  //DEF ID=INPUT,FIL=&OBJNAME,STA=(P,I)
  [//DEF ID=OUTPUT,FIL=$SYSLODLIBA,STA=(P,O)]
  //DEF ID=LIST,DEV=PRINTER
  [//TELL OP=BEGIN LINKAGE EDITING OF ASM]
}

{ //EXEC PGM=LNKEDT,TIME=1440
  [//PAR PGM=COBOL,POOLSIZ='1A4',LST=XREF,LSD=NO]
  //DEF ID=INPUT,FIL=&OBJNAME,STA=(P,I)
  [//DEF ID=OUTPUT,FIL=$SYSLODLIBA,STA=(P,O)]
  //DEF ID=LIST,DEV=PRINTER
  [//TELL OP=BEGIN LINKAGE EDITING OF COBOL]
}

```

*&OBJNAME is the name of the library containing the SYSGEN object modules specified in answer to question 3 of the SYSGEN Control section of the Checklist or \$SGOBJ if the default has been selected.

The three sets of statements enclosed in braces are executed when the Control Language Services (\$JMDF)*, Assembler Language (ASM), and COBOL relocatable load modules are being produced at SYSGEN time. When the system module containing the unit tables (\$OSYSTAB) is reassembled during SYSGEN, sets of statements to linkage edit the following system programs are included in GENPROC4: UTCVIM, UTCVMI, UTSSCL, UTSSCD, UTSSCU, RESTART, UTSSLU and \$BLDRES. Notice that the //PAR statement and the //TELL statement vary according to the name of the module being produced. Notice also that the first time one of the modules is executed it has the following //DEF statement.

```
//DEF      ID=OUTPUT,FIL=$SYSLODLIBA,STA=(P,O),CSD=NO,VER=YES,  
  
//        BLK=1,SIZ=256,CAT=YES,NUM=400
```

For example, if the modules named \$JMDF and ASM were not executed, the //DEF statement shown above would replace the //DEF statement defining output for \$SYSLODLIBA that is shown in the COBOL group of statements. This statement causes the allocation of \$SYSLODLIBA. If all three modules are being link edited, the first module will contain the //DEF statement that allocates \$SYSLODLIBA and the other modules will contain the following version of the //DEF statement.

```
//DEF      ID=OUTPUT,FIL$SYSLODLIBA,STA=(P,O)
```

The //EXEC statement in each group of statements is your restart point. The processing milestone corresponds to the //TELL statement in each group of statements. The milestone occurs after each relocatable load module has been produced. Additional sets of Control Language statements needed to execute the Linkage Editor are included for each system program that refers to operating system definitions that have been modified by the SYSGEN programs.

*When \$JMDF is link edited, sets of statements to linkage edit the following related modules will also be included in GENPROC4: \$JMIR, \$JMSI, \$JMST1, \$JMSTM, \$JMOIS, \$JMSOS, \$JMJT, \$JMIR, \$JMOIR and \$JMJI.

B. CONTROL LANGUAGE STATEMENTS FOR SYSGEN

CLS FOR JOBONE

The Control Language statements necessary to call SYSGEN JOBONE are listed below. Put the data cards that you have coded in answer to the Checklist questions after the //DATA card. Shaded statements are valid only for SYSGEN, Type 1, Procedure B jobs.

Because all SYSGEN jobs must be run sequentially, do not put the next job in the card reader until the preceding job has finished executing.

```
//JOB  NAME=JOBONE,TYPE=1[,USER=] [,PRIORITY=]
//CALL PROC=SGALLOC,NUCLOC=____,MSGLOC=7,SYSVOL=____
//CALL PROC=SGCOPYA,LIST=YES,INFILE=$SYSLODLIB,
//      OUTFILE=$SYSLODLIB,
//      OCAT=NO,OLOC=____,ONUM=8000,INVOL=____,OUTVOL=____
//CALL PROC=SGCPYCAT,INVOL=____,OUTVOL=____
//CALL PROC=SGCOPY,INFILE=SYSGEN-MAC-BACKUP,
//      OUTFILE=$SYSMACLIB,INVOL=____,OUTVOL=____
//EXEC  PGM=LIBUTIL
//PAR   COMMAND=DELETE,MEM=$OSCEPL,MEM=$OSCEPL2,
//      MTYPE=ABS
//DEF   ID=LIST,DEV=PRINTER
//DEF   ID=OUTPUT,STA=(P,O),FIL=$SGOBJ,VOL=____
//CALL  PROC=GENPROC1
//DATA  FIL=CHECKLISTDATA
●
● Insert input data (answer to Checklist questions) here.
●
//EOJ
```

Omit for
Types 2
and 3

The keyword parameters in JOBONE that are variable have the following meanings:

NUCLOC= the location (cylinder number) of the Nucleus Library which is 4 if STDNUC=YES in the GENPROC3 //CALL statement and 10 if STDNUC=NO.

SYSVOL= the volume identifier of your new system resident pack.

- OLOC= 10 if STDNUC=YES in the GENPROC3 //CALL statement and 13 if STDNUC=NO. OLOC can also be specified as NO and \$SYSLODLIB will then be placed in the first available space on the disc.
- INVOL= the volume identifier of your distribution or other resident input pack.
- OUTVOL= the volume identifier of the system resident pack you are building.
- VOL= the volume identifier of your distribution pack (or other input pack).

Both the Message Library and the Nucleus Library occupy three cylinders. Omit the //CALL statement to copy SYSGEN-MAC-BACKUP and the //EXEC PGM=LIBUTIL statement as well as its associated //PAR and //DEF statements if you are performing a Type 2 or Type 3 SYSGEN (modification type SYSGENS). If you are following Procedure B, omit the //EXEC PGM=LIBUTIL and associated statements.

CLS FOR JOBTWO

Immediately after the //EOJ card that terminates SYSGEN JOBONE, insert the following Control Language statements to call SYSGEN JOBTWO. Remember that these jobs must be run consecutively. No other job can intervene.

```

//JOB  NAME=JOBTWO,TYPE=1[,USER=] [,PRIORITY=]
//CALL PROC=GENPROC2
//CALL PROC=GENPROC3,SYSVOL=____,DEVADDR=____
//      [,FULLGEN= {YES}
//      [NO}]
//      [,STDNUC= {YES}
//      [NO}] [,OBJNAME={your name}
//      [$SGOBJ}]
//      [RSLOC={nnn}
//      [YES]
//      [NO}] *
//CALL PROC=GENPROC4
//CALL PROC=SGMERGE,SYSVOL=____
//CALL PROC=SGPURGE,FIL=$SYSLODLIBA
//CALL PROC=SGRTNCAT,INVOL=____,OUTVOL=____
//EOJ

```

Omit for Type 3

Make these statements into a separate job if you have selected several SYSGEN options.

The keyword parameters in JOBTWO that are variable have the following meanings:

- SYSVOL= the volume identifier of your new system resident pack.
- DEVADDR= the device address where the new SYSRES pack is mounted. This is a two-digit hexadecimal number.

*Defaults are underlined for the parameters in GENPROC3.

- FULLGEN= YES for a Type 1 or Type 2 SYSGEN; NO only if you are doing a field update or putting in a patch.
- STDNUC= specifies the location of the Nucleus Library. If you omit this keyword and accept the default YES, it means that your Nucleus Library and System Message Library that begin one cylinder after the end of your CCAT, are in the standard location. If you specify NO, it means that your Nucleus Library is in the system alternate location which immediately follows the standard location. Because these areas are referenced using physical I/O, it is mandatory that you make sure that this area is available before you select this option.
- OBJNAME= the name you specified in answer to Checklist question 3 or the default, \$SGOBJ, which is the standard SYSGEN object library (*SYSGEN Control* section).
- RSLOC= the value used in the LOC= parameter on the //DEF statement when \$OSRSDNTLIB is allocated.
- INVOL= the volume identifier of your distribution or other resident input pack.
- OUTVOL= the volume identifier of the system resident pack you are building.

Omit the //CALL statement to GENPROC3 if you are performing a Type 3 SYSGEN. If you are performing a Type 2 SYSGEN, you must have an existing \$NUCLIB ready to be used. It must either be in the standard location, or it must be in the alternate location as directed by the keyword STDNUC= on the //CALL statement to GENPROC3. The pack specified by the keyword SYSVOL= on the //CALL statement to GENPROC3 must contain the volume number of your new system resident pack.

In summary, if you are performing a Type 1 or Type 2 SYSGEN, you must call GENPROC1, GENPROC2, GENPROC3, and GENPROC4. If you are performing a Type 3 SYSGEN, call only GENPROC1, GENPROC2, and GENPROC4.

NOTE

If you answered YES to question 5 in the *Control Language Services* section of the Checklist, USER= is a required keyword on your SYSGEN //JOB statements.

CLS FOR JOB2A

This job copies the Procedure Library which includes GENPROC1, GENPROC2, GENPROC3, GENPROC4, and all special SYSGEN procedures defined in Appendix C. It also allocates space for the System Error Log (\$SYSELOG). If you answered YES to question 8 in the *Output Spooler* section of the Checklist, the second to the last //DEF statement must also be included in this job. This statement allocates space for the Job Accounting feature (\$OSSPLJA).

```

//JOB   NAME=JOB2A,TYPE=1[,USER=] [,PRIORITY=]
//EXEC  PGM=LIBUTIL
//PAR   COMMAND=COPY
//DEF   ID=INPUT,FIL=$SYSPROCLIB,STA=P,VOL= ____
//DEF   ID=OUTPUT,FIL=$SYSPROCLIB,STA=P,SIZ=80,
//      NUM=1560,ORG=S,CON=YES,VOL= ____
//DEF   ID=CAT1,FIL=$SYSELOG,STA=P,ORG=S,CON=YES,NUM=520
//      SIZ=40,CSD=NO,BLK=1,CAT=YES,LOC=____
Optional { //DEF   ID=CAT2,FIL=$OSSPLJA,STA=P,ORG=R,CON=YES,NUM=200†
           //      SIZ=54,CSD=NO,BLK=1,CAT=YES,[,LOC=____ ] [,VOL=____ ]
           //DEF   ID=LIST,DEV=PRINTER
           //EOJ

```

†Note: The NUM= entry should match the answer to question 2507.

The keyword parameters that are variable in JOB2A have the following meanings:

In the first //DEF statement:

VOL= the volume identifier of your distribution or other resident input pack.

In the second //DEF statement:

VOL= the volume identifier of the system resident pack you are building.

In the third //DEF statement:

LOC= the cylinder location of your System Error Log (\$SYSELOG). The usual location is 199.

In the fourth (optional) //DEF statement:

NUM= is a parameter that can be changed by the user to meet his needs. Two hundred allows room for accounting 200 jobs.

[LOC=] the cylinder location of your Job Accounting feature (\$OSSPLJA). If specified, the number must be on a cylinder boundary.

[VOL=] can be specified if you want the Job Accounting feature located on a shared volume.

CLS FOR JOB2B

This job copies SYSGEN Object Library (\$SGOBJ) and the System Macro Library (\$SYSMACLIB) onto your new system resident pack. If you are using Procedure B, omit this job.

```

//JOB   NAME=JOB2B,TYPE=1[,USER=] [,PRIORITY=]
//CALL  PROC=SGCOPYA,INFILE=$$GGOBJ,OUTFILE=$$GGOBJ,
//      ONUM=8000,INVOL=____,OUTVOL=____,LIST=YES
//CALL  PROC=SGCOPYA,OSIZ=80,OCSD=YES,OLOC=NO,
//      INFILE=$$SYSMACLIB,OUTFILE=$$SYSMACLIB,
//      ONUM=17550,INVOL=____,OUTVOL=____,LIST=YES
//EOJ

```

The keyword parameters that are variable in JOB2B have the following meanings:

INVOL= the volume identifier of your distribution or other resident input pack.

OUTVOL= the volume identifier of the system resident pack you are building.

CLS FOR JOBTHREE

JOBTHREE should be used to copy \$SYSOBJLIB and all the libraries from your distribution pack that you want on your new system resident pack. This job is not part of the formal SYSGEN procedures.

```

//JOB   NAME=JOBTHREE
//CALL  PROC=SGCOPYA,LIST=YES,INFILE=$$SYSOBJLIB,
//      OUTFILE=$$SYSOBJLIB,
//      ONUM=1000,OLOC=NO,INVOL=____,OUTVOL=____

```

-
- Insert a series of //CALL PROC=SGCOPYA statements to allocate and copy the files you need in your new system.
- These files are listed below. The required parameters for the //CALL
- statements are explained in Appendix C.

```

//EOJ

```

Filenames of the modules that can be copied in this job include:

CBLLODLIB	(800)	\$STANDALONE (1950) OSIZ=80,CSD=YES
CBLRTLIB	(200)	SRTOBJLIB (1500)
RPGLODLIB	(800)	UTCVEMLIB (400)
RPGRTLIB	(400)	\$MSGLIBINPUT (1000)
EMULODLIB	(1000)	SYSGEN-MAC-BACKUP (390) OSIZ=80,CSD=YES
EMUMACLIB	(3900)	IQROBLIB (360) OSIZ=252,CSD=YES
EMUOBJLIB	(1000)	IQRSRCLIB (7020) OSIZ=80,CSD=YES
FTNLODLIB	(800)	\$SYSGENSRCLIB (23400) OSIZ=80,CSD=YES
FTNRTLIB	(1200)	

ONUM= parameter specifications shown after the module name in parenthesis.
For modules having specific, required OSIZ and CSD parameter entries,
these are shown after the ONUM entry.

The OLOC, INVOL, and OUTVOL parameters are the same for all the //CALL PROC=
statements you code as those shown in the first //CALL statement of JOBTHREE.

The keyword parameters that are variable in JOBTHREE have the following specifications:

INVOL= the volume identifier of your distribution or other resident input pack.

OUTVOL= the volume identifier of the system resident pack you are building.

CLS FOR JOBFOUR

JOBFOUR should be used to copy files that you will need in your new system that cannot
be copied using the SGCOPYA procedure. The program used to copy these files is called the
File to File Utility program (UTFF). Use this program to copy message text files for the
Assembler, COBOL, and RPG II programs and also to copy the \$BLDRESDIRECTIVES file
used by the \$BLDRES program, the \$COPYRIGHT file and the Inquiry Retrieval
installation file.

```
//JOB NAME=JOBFOUR,TYPE=1 [,USER=] [,PRIORITY=]
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=ASMMTFIL,CAT=NO,STA=(P,O),VOL=___
//DEF ID=OUTPUT,FIL=ASMMTFIL,STA=(P,O),CAT=YES,
// SIZ=256,ORG=R,NUM=60,CSD=NO,VOL=___
//DEF ID=LIST,DEV=PRINTER
//TELL OP=BEGIN FILE-TO-FILE COPIES. NO PRINTER OUTPUT
// (EXCEPT CLS).
//TELL OP=JOBFOUR FILE-TO-FILE TAKES ABOUT 10 MINUTES.
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=RPGERMSG,CAT=NO,STA=P,VOL=___
//DEF ID=OUTPUT,FIL=RPGERMSG,CAT=YES,STA=(P,O),
// SIZ=80,ORG=R,NUM=500,CSD=YES,VOL=___
//DEF ID=LIST,DEV=PRINTER
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=$COPYRIGHT,CAT=NO,STA=(P,O),VOL=___
//DEF ID=OUTPUT,FIL=$COPYRIGHT,CAT=YES,STA=(P,O),
// SIZ=80,BLK=1,CSD=YES,ORG=S,NUM=500,VOL=___
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=$BLDRESDIRECTIVES,CAT=NO,STA=(P,O),
// VOL=___
```

```

//DEF ID=OUTPUT,FIL=$BLDRESDIRECTIVES,CAT=YES,STA=(P,O),
// SIZ=80,BLK=1,CSD=YES,ORG=S,NUM=780,VOL=____
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=COBMTFIL,CAT=NO,STA=(P,O),VOL=____
//DEF ID=OUTPUT,FIL=COBMTFIL,CAT=YES,STA=(P,O),
// SIZ=256,ORG=R,NUM=100,CSD=NO,VOL=
//DEF ID=LIST,DEV=PRINTER
//EXEC PGM=UTFF
//DEF ID=INPUT,FIL=IRINST,CAT=NO,STA=P,VOL=____
//DEF ID=OUTPUT,FIL=IRINST,CAT=YES,STA=P,
// SIZ=80,CSD=YES,ORG=S,NUM=1000,VOL=____
//DEF ID=LIST,DEV=PRINTER
//EOJ

```

The VOL= keyword parameter is the volume identifier of the system resident pack you are building.

CLS FOR JOBFIVE

Execute this job to produce a clean listing of the contents of the \$SYSLODLIB (System Load Library) that you have just built. This listing will not show the files that you copied in JOBFOUR.

```

//JOB NAME=JOBFIVE,TYPE=1[,USER=] [,PRIORITY=]
//EXEC PGM=UTSSCD
//DEF ID=PCAT1,FIL=$PCAT,DEV=DISC,STA=(P,I),VOL=____
//DEF ID=CCAT,FIL=$CCAT,DEV=DISC,STA=(P,I),VOL=____
//DEF FIL=OUTPUT,DEV=PRINTER,ID=PRINTER,CSD=NO,
// BLK=1,SIZ=133
//EXEC PGM=LIBUTIL
//DEF ID=LIST,DEV=PRINTER
//PAR COMMAND=PTOC,ILIB=IN30
//DEF ID=IN30,STA=P,FIL=$MSGLIBINPUT,VOL=____
//PAR COMMAND=PTOC,ILIB=IN40
//DEF ID=IN40,STA=P,FIL=$OSRSDNTLIB,VOL=____
//PAR COMMAND=PTOC,ILIB=IN50
//DEF ID=IN50,STA=P,FIL=$SGOBJ,VOL=____
//PAR COMMAND=PTOC,ILIB=IN60

```

```

//DEF ID=IN60,STA=P,FIL=$STANDALONE,VOL=___
//PAR COMMAND=PTOC,ILIB=IN70
//DEF ID=IN70,STA=P,FIL=$SYSGENSRCLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN80
//DEF ID=IN80,STA=P,FIL=$SYSLODLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN90
//DEF ID=IN90,STA=P,FIL=$SYSMACLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN100
//DEF ID=IN100,STA=P,FIL=$SYSOBJLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN120
//DEF ID=IN120,STA=P,FIL=$SYSPROCLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN130
//DEF ID=IN130,STA=P,FIL=CBLLODLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN140
//DEF ID=IN140,STA=P,FIL=CBLRTOLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN150
//DEF ID=IN150,STA=P,FIL=EMUOBJLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN155
//DEF ID=IN155,STA=P,FIL=FTNLODLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN156
//DEF ID=IN156,STA=P,FIL=FTNRTOLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN170
//DEF ID=IN170,STA=P,FIL=IQROBJLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN160
//DEF ID=IN160,STA=P,FIL=IQRSRCLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN175
//DEF ID=IN175,STA=P,FIL=RPGLODLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN176
//DEF ID=IN176,STA=P,FIL=RPGRTOLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=PT178
//DEF ID=PT178,STA=P,FIL=SRTOBJLIB,VOL=___
//PAR COMMAND=PTOC,ILIB=IN180
//DEF ID=IN180,STA=P,FIL=SYSGEN-MAC-BACKUP,VOL=___
//PAR COMMAND=PTOC,ILIB=IN199
//DEF ID=IN199,STA=P,FIL=UTCVEMLIB,VOL=___
//EOJ

```

The VOL= keyword parameter is the volume identifier of the system resident pack you are building.

C. SPECIAL SYSGEN PROCEDURES

Included in this appendix are the special procedures that you will use to tailor your libraries to the needs of your installation. The Control Language statements are listed for each procedure. Each procedure is called via a //CALL statement that you insert in your job stream. Appendix B shows the SYSGEN job stream and Section 4 tells you where you may want to use these procedures.

SGCOPYA

Use this procedure to create a copy of an existing library for which you need to allocate space for the output file.

```
//EXEC      PGM=LIBUTIL
//PAR      COMMAND=COPY,LIST= &LIST
//DEF      ID=INPUT, FIL= &INFILE, STA=P, VOL= &INVOL
//DEF      ID=OUTPUT, FIL= &OUTFILE, STA=P, VOL= &OUTVOL,
//          SIZ= &OSIZ, BLK= &OBLK, CON=YES, CSD= &OCSD,
//          CAT= &OCAT, NUM= &ONUM, LOC= &OLOC
//DEF      ID=LIST, DEV=PRINTER
```

The //CALL PROC=SGCOPYA statement that you use to call this procedure requires the following specifications on it:

LIST=	Enter YES if you want to list the members of the library that have been copied.
INFILE=	Name of the library you are copying.
OUTFILE=	Name of new library you are creating.
INVOL=	Label of the pack containing the input library.
OUTVOL=	Label of the pack containing the output library.
OSIZ=	Enter record length in bytes if different from 256.
OCSD=	Enter YES if library members are in common stored data format.
OCAT=	Enter NO if output library need not be cataloged.
OBLK=	Enter number of records per block in the output library if number is not 1.

ONUM= Number of records for which space should be allocated.

OLOC= Cylinder number of the start of the output library.

SGCOPY

Use this procedure to make a copy of an existing library for which space has already been allocated for the output file.

```
//EXEC PGM=LIBUTIL  
  
//PAR COMMAND=COPY, LIST= &LIST  
  
//DEF ID=LIST, DEV=PRINTER  
  
//DEF ID=INPUT, FIL= &INFILE, VOL= &INVOL, STA=P  
  
//DEF ID=OUTPUT, FIL= &OUTFILE, VOL= &OUTVOL, STA=P
```

The //CALL PROC=SGCOPY statement that you use to call this procedure requires the following specifications on it:

LIST= Enter YES if you want to list the members of the library that have been copied.

INFILE= Name of the library you are copying.

OUTFILE= Name of the new library you are creating.

INVOL= Label of the pack containing the input library.

OUTVOL= Label of the pack containing the output library.

SGMERGE

Use this procedure to merge \$SYSLODLIBA with \$SYSLODLIB.

```
//EXEC PGM=LIBUTIL  
  
//PAR COMMAND=COPY  
  
//DEF ID=INPUT, FIL=$SYSLODLIBA, STA=P, VOL=&SYSVOL  
  
//DEF ID=OUTPUT, FIL=$SYSLODLIB, STA=P  
  
//DEF ID=LIST, DEV=PRINTER
```


The //CALL PROC=SGMERGE statement that you use to call this procedure requires the following specification:

SYSVOL= Label of the pack containing the new operating system.

SGPURGE

Use this procedure to purge a cataloged file.

```
//EXEC      PGM=UTSSPF
//PAR       PURGE= &FILE
```

The //CALL PROC=SGPURGE statement that you use to call this procedure requires one additional specification: FILE= name of the file to be purged which must be centrally cataloged in the current system.

SGALLOC

Use this procedure to allocate space for \$NUCLIB and \$MSGLIB on a new system resident pack that has been previously initialized using the Disc Initialize utility program but has never been used as a system resident pack.

```
//EXEC      PGM=$OSSGDN
//DEF       ID=DUMMYA, FIL=$NUCLIB, STA=P,
//          SIZ=256, NUM=1200, CSD=NO, BLK=1,
//          CAT=NO, CON=YES, VOL=&SYSVOL, LOC=&NUCLOC
//DEF       ID=DUMMYB, FIL=$MSGLIB, STA=P,
//          SIZ=256, NUM=1200, CSD=NO, BLK=1,
//          CAT=NO, CON=YES, VOL= &SYSVOL, LOC=&MSGLOC
```

The //CALL PROC=SGALLOC statement that you use to call this procedure must be inserted in the job stream immediately after the //JOB card for SYSGEN JOBONE. This //CALL statement requires these additional specifications on it:

SYSVOL= Label of the system resident pack upon which the new system is being built.

NUCLOC= Cylinder number where \$NUCLIB must be allocated. \$NUCLIB must be allocated 1 cylinder after the end of the central catalog (see example below).

MSGLOC= Enter the cylinder number that reflects NUCLOC + 3 for the standard location or NUCLOC - 3 for the alternate NUCLOC specification.

EXAMPLE

If the CCAT (central catalog) begins at cylinder 1 and is 2 cylinders long, NUCLOC=4 and MSGLOC=7.

SGCAT

Use this procedure to catalog an existing file.

```
//EXEC      PGM=UTSSCU
//DEF       ID=IN,FIL=$CCAT,VOL=&SYSVOL
//DEF       ID=CTLG01,FIL=&FILE,STA=(P,O),VOL=&DATAVOL
```

The //CALL PROC=SGCAT statement that you use to call this procedure requires the following specifications on it:

SYSVOL= Label of the pack that contains the central catalog (CCAT).
FILE= Name of the file that is being cataloged.
DATAVOL= Label of the pack that will contain the file being cataloged.

SGUNCAT

Use this procedure to uncatalog a file.

```
//EXEC      PGM=UTSSCU
//DEF       ID=IN,FIL=$CCAT,VOL=&SYSVOL,STA=P
//DEF       ID=UNCTLG01,FIL=&FILE,STA=(P,O),VOL=&DATAVOL
```

The //CALL PROC=SGUNCAT statement that you use to call this procedure requires the following specifications:

SYSVOL= Label of the pack that contains the central catalog (CCAT).
FILE= Name of the file that is being cataloged.
DATAVOL= Label of the pack that will contain the file being cataloged.

SGCATDSP

Use this procedure to display the pack catalog (PCAT) and the central catalog (CCAT) of the system resident pack.

```
//EXEC      PGM=UTSSCD

//DEF       ID=PRINTER,DEV=PRINTER

//DEF       ID=PCAT1,FIL=$PCAT,VOL=&SYSVOL,STA=P

//DEF       ID=CCAT,FIL=$CCAT,VOL=&SYSVOL,STA=P
```

The //CALL PROC=SGCATDSP card that you use to call this procedure requires the following specification:

SYSVOL= Label of the pack that contains the catalogs being displayed.

SGCPYCAT (USED WITH PROCEDURE B ONLY)

Use this procedure in JOBONE if you want to maintain the integrity of your input pack. This procedure copies \$SYSOBJ and \$SYSMACLIB from your existing resident pack to your new pack. It also calls the SGCAT and SGUNCAT procedures (described earlier in this appendix) so that assembled modules and updated modules will appear on the new pack.

```
//DEC  INVOL,OUTVOL,
//     OBJFILI=$SGOBJ,OBJFILO=$SGOBJ,OBJNUM=8000,OBJLOC=NO,
//     MACFILI=$SYSMACLIB,MACFILO=$SYSMACLIB,MACNUM=23400,
//     MACLOC=NO,LIST=YES
//CALL PROC=SGCOPYA,OSIZ=256,OCSD=NO,OCAT=NO,OBLK=1,
//     LIST=&LIST#,SELECT#=E#,MEM#=$OSCEPL#,MEM#=$OSCEPL2#,MTYPE#=ABS,
//     ONUM=&OBJNUM,OLOC=&OBJLOC,INFILE=&OBJFILI,OUTFILE=&OBJFILO,
//     INVOL=&INVOL,OUTVOL=&OUTVOL
//CALL PROC=SGCOPYA,LIST=&LIST,OSIZ=80,OCSD=YES,OCAT=NO,OBLK=1,
//     ONUM=&MACNUM,OLOC=&MACLOC,INFILE=&MACFILI,OUTFILE=&MACFILO,
//     INVOL=&INVOL,OUTVOL=&OUTVOL
//CALL PROC=SGUNCAT,FILE=&OBJFILI,SYSVOL=&INVOL,DATAVOL=&INVOL
```

```
//CALL PROC=SGCAT,FILE=&OBJFILO,SYSVOL=&INVOL,DATAVOL=&OUTVOL
//CALL PROC=SGUNCAT,FILE=&MACFILI,SYSVOL=&INVOL,DATAVOL=&INVOL
//CALL PROC=SGCAT,FILE=&MACFILO,SYSVOL=&INVOL,DATAVOL=&OUTVOL
```

The //CALL PROC=SGCPYCAT statement that you use to call this procedure requires the following specifications:

INVOL= the volume identifier of the resident pack you are using to generate your new system.

OUTVOL= the volume identifier of the new resident pack you are building.

SGRTNCAT (USED WITH PROCEDURE B ONLY)

Use this procedure in JOBTWO if you want to maintain the integrity of your input pack.

```
//DEC      INVOL,OUTVOL,
//          OBJFILI=$SGOBJ,OBJFILO=$SGOBJ,
//          MACFILI=$SYSMACLIB,MACFILO=$SYSMACLIB
//CALL     PROC=SGUNCAT,FILE=&OBJFILO,SYSVOL=&INVOL,
//          DATAVOL=&OUTVOL
//CALL     PROC=SGCAT,FILE=&OBJFILI,SYSVOL=&INVOL,
//          DATAVOL=&INVOL
//CALL     PROC=SGCAT,FILE=&OBJFILO,SYSVOL=&OUTVOL,
//          DATAVOL=&OUTVOL
//CALL     PROC=SGUNCAT,FILE=&MACFILI,SYSVOL=&INVOL,
//          DATAVOL=&OUTVOL
//CALL     PROC=SGCAT,FILE=&MACFILI,SYSVOL=&INVOL,
//          DATAVOL=&INVOL
//CALL     PROC=SGCAT,FILE=&MACFILO,SYSVOL=&OUTVOL,
//          DATAVOL=&OUTVOL
```

The //CALL PROC=SGRTNCAT statement that you use to call this procedure requires the following specifications:

INVOL= the volume identifier of the resident pack you are using to generate your new system.

OUTVOL= the volume identifier of the new resident pack you are building.

D. ERROR MESSAGES

Table D-1. Console Messages

Error Number	Error Type	Message
1	OSSG0018	<p>SYSRES NOT INITIALIZED. MOUNT NEW PACK AND TYPE C TO CONTINUE OR A TO ABORT SYSGEN.</p> <p>When the operator types an A in response to this message, the following message will appear in the SYSOUT file: SYSRES NOT INITIALIZED. OPERATOR ABORTED JOB.</p>

Table D-2. SYSOUT Messages

Error Number	Error Type	Message
2	OSSG0028	<p>SYSRES NOT INITIALIZED. OPERATOR ABORTED JOB.</p> <p>When the operator types in an A in response to error number 1, this message will appear in the SYSOUT file.</p>
3	OSSG0038	<p>nnnn BAD DIRECTIVES ENCOUNTERED.</p> <p>The total number of bad directives printed on the listing will be supplied for the value nnnn in this message.</p>
4	OSSG0048	<p>IMPOSSIBLE TO FILE, DIRECTORY FULL.</p> <p>Inadequate space remains in the \$NUCLIB directory so an entry cannot be created there for the current program.</p>
5	OSSG0058	<p>IMPOSSIBLE TO FILE, FILE FULL.</p> <p>Insufficient space remains in \$NUCLIB program file for the current program. The last card printed on the listing is the last card successfully filed.</p>

Table D-2. SYSOUT Messages (Continued)

Error Number	Error Type	Message
6	OSSG0068	<p>nnnn BAD DATA CARDS ENCOUNTERED, SYSGEN JOB ONE TERMINATED. SYSGEN DATA PURGED.</p> <p>The total number of erroneous data cards printed on the listing will be supplied for the value nnnn in this message.</p>
7	OSSG0078	<p>NO DATA CARDS FOUND, SYSGEN JOB ONE TERMINATED. SYSGEN DATA PURGED.</p>
8	OSSG0088	<p>SYSGEN MACRO xxxxxxxx MISSING FROM MACRO LIBRARY.</p> <p>The one to eight character alphanumeric name of the missing macro will be supplied for the value xxxxxxxx in this message.</p>
9	OSSG0098	<p>INVALID SYSGEN PARAMETER TABLE SPECIFIED. PROGRAM TERMINATED.</p> <p>This means that the data is bad. Could be caused by hardware error.</p>
10	OSSG0108	<p>DEVICE ADDRESS OF SYSRES, SPECIFIED BY THE PARAMETER DEVADDR, CANNOT BE FOUND.</p> <p>This device address is where the SYSGEN programs perform the physical I/O necessary to put the \$NUCLIB on the disc.</p>
11	OSSG0118	<p>INVALID PARAMETERIZATION ON //PAR CARD FOR BLDRES.</p> <p>This message appears when an erroneous parameter or a misspelled parameter appears on the //PAR card. Refer to Appendix A in this document.</p>

Table D-3. Listing Messages

Error Number	Error Type	Message
12	Serious*	<p>DATA CARD IN ERROR, IGNORED:</p> <p>The name of the erroneous data card will appear on the listing after the colon. For the total number of bad cards that appear on the listing, refer to error message 6 in the SYSOUT file.</p>
13	Serious**	<p>THE FOLLOWING BLDRES DIRECTIVE IS IN ERROR:</p> <p>The name of the erroneous directive will appear on the listing after the colon. For the total number of bad directives that appear on the listing, refer to error message 3 in the SYSOUT message file.</p>
14	Serious**	<p>MODULE CANNOT BE FOUND IN INPUT LIBRARY:</p> <p>The name of the module will appear after the colon. The absolute load module that you specified on the BLDRES directive cannot be found in the input library.</p>
15	Serious**	<p>INPUT MODULE SIZE EXCEEDS 2048 BYTES:</p> <p>The name of the module will appear after the colon. The BLDRES program uses an internal buffer of 2KB so no absolute load module can exceed this size.</p>

*The Build Parameter Table will continue execution even when it encounters invalid data cards. However, at the completion of the program, the SYSGEN programs will terminate processing. You must then correct all bad cards and re-execute the entire job using the corrected data cards.

**The module named after the colon is not processed and will not appear in your NUCLIB at the completion of your SYSGEN. However, the NUCLIB that you have created does contain the other valid modules and so can be updated.

INDEX

Assembler Language	5-10	Libraries used by SYSGEN programs	3-1
		Linkage Editor Control	2-2
BLDRES directive file	3-5		
Build Parameter Table	2-1	Minimum system	
Build System Resident	2-2	definition	1-1
		description	1-2
Card Image Generator	2-2		
Checklist Data File	3-5	Nucleus Library	3-2
Checklist Processor	2-1		
Checklist sections	5-1	Operating environment	1-2
SYSGEN Control	5-1	Operating procedures	
System Control	5-2	introduction	4-1
Output Spooler	5-3	multi-drive user	4-2
I/O Devices	5-4	one-drive user	4-5
Telecommunications	5-7	Operating System Resident Library	3-3
Control Language Services	5-8	OSRSDNT	2-2
Assembler Language	5-10	Output Spooler	5-3
COBOL Language	5-11		
CHECKLISTDATA	3-5	Procedure A	1-1,4-1,C-6
COBOL Language	5-11	Procedure B	1-1,4-1,C-5
Control Language Services	5-8		
Control Language statements for SYSGEN	Appendix B	Reassembly Control	2-1
Copy procedures	2-4	Resident extension system	
		definition	1-1
		description	1-2
Error messages	Appendix D		
Error recovery	Appendix A	SGALLOC	C-3
		SGCAT	C-4
File clean-up procedures	2-4	SGCATDSP	C-5
Files used by SYSGEN programs	3-1	SGCOPY	C-2
		SGCOPYA	C-1
GENPROC 1	A-1	SGCPYCAT	C-5
GENPROC 2	A-3	SGMERGE	C-2
GENPROC 3	A-3	SGPURGE	C-3
GENPROC 4	A-4	SGRTNCAT	C-6
		SGUNCAT	C-4
Initial Program Load (IPL)	4-2,4-5	SYSGEN components	2-1
I/O Devices	5-4	SYSGEN control	5-1
		SYSGEN Data File	3-5
		SYSGEN Load Library	3-5
Job flow		SYSGEN Object Library	3-1
JOBONE	2-2	SYSGEN-MAC-BACKUP	3-4
JOBTWO	2-4	SYSGEN Macro Backup Library	3-4
JOBONE	4-1	SYSGEN Source Library	3-2
JOBTWO	4-1	System control	5-2
JOB2A	4-1	System Generation Procedures and	
JOBTHREE	2-4,4-1,B-5	Error Recovery	A-1
JOBFOUR	2-4,4-1,B-6	System Load Library	3-4
JOBFIVE	2-4,4-1,B-7		

System Macro Library	3-3	\$BLDRES	2-2
System Message Library	3-3	\$BLDRESDIRECTIVES	3-5
System Procedure Library	3-1	\$MSGLIB	3-3
		\$MSGLIBINPUT	3-3
Telecommunications	5-7	\$NUCLIB	3-2
Type 1		\$OSRSDNTLIB	3-3
overview	1-1	\$OSSGBPT	2-1
procedures	4-2	\$OSSGCLP	2-1
Type 2		\$SGOBJ	3-1
overview	1-1	\$SYSGENDATA	3-5
procedures	4-8	\$SYSGENSCRLIB	3-2
restriction	3-3	\$SYSLODLIB	3-4
Type 3		\$SYSLODLIBA	3-5
overview	1-1	\$SYSMACLIB	3-3
procedures	4-8	\$SYSPROCLIB	3-1
restriction	3-3		

MEMOREX

First Class
Permit No. 14831
Minneapolis,
Minnesota 55427

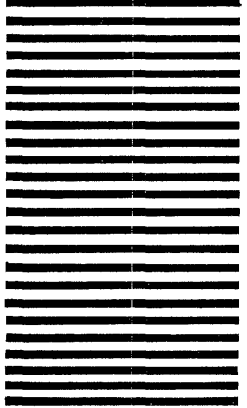
Business Reply Mail

No Postage Necessary if Mailed in the United States

Postage Will Be Paid By

Memorex Corporation

Midwest Operations – Publications
8941 Tenth Avenue North
Minneapolis, Minnesota 55427



Thank you for your information.

Our goal is to provide better, more useful manuals, and your
comments will help us to do so.

.Memorex Publications