PERKIN-ELMER

SYSTEM GENERATION/32 (SYSGEN/32)

Reference Manual

48-037 F00 R02

The information in this document is subject to change without notice and should not be construed as a commitment by The Perkin-Elmer Corporation. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place, Oceanport, New Jersey 07757

Printed in the United States of America

TABLE OF CONTENTS

CHAPTERS (Continued)

APPENDIXES

FIGURES

TABLES

TABLES (Continued)

PREFACE

This manual describes the System Generation/32 (Sysgen/32) program and the procedures to produce a 32-bit operating system. This manual is intended for system programmers and operators.

Chapter 1 introduces the sysgen command substitution system (CSS) procedure and describes its phases. Chapter 1 also explains how to load and start the Sysgen/32 program. The Sysgen/32 commands that cause a configuration input file to be processed are described in detail. Chapter 2 details the sysgen configuration input statements that make up the input file, and includes data communication configuration statements. Chapter 3 lists and describes the libraries required by the Sysgen/32 program and explains how to tailor library modules to user needs and include the modified modules in the appropriate library. Chapter 3 also explains how to include user-written drivers and modules in the system. Chapter 4 details the OS/32-supported devices. Chapter 5 is written for the less experienced user and contains sample Sysgen/32 sessions.

Appendix A compares the Sysgen/32 and the OS/32 Configuration Utility Program (CUP) configuration statement defaults. Appendix B lists the sysgen messages, and Appendix C is a summary of the Sysgen/32 commands and configuration statements. Appendix D compares OS/32 CUP and Sysgen/32 configuration statements.

Revision 02 introduces the new Model 3205 System. New configuration statements have been added to be used in the sysgen interactive mode. Enhancements have been made to the file manager. New system source modules and supported features have been included. With the R07.2 release, new macros have been introduced to the Sysgen/32 Macro Library. Support for virtual disk and mirror disk is documented. Configuration statements to enable the generation of an operating system for a Model 3200MPS System are clearly marked as pertaining to the Model 3200MPS System only. This manual applies to the OS/32 R07.2 software release and higher.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1
INTRODUCTION TO SYSTEM GENERATION/32 (SYSGEN/32)

1.1 INTRODUCTION

Sysgen/32 is a program designed to enable a user to create and tailor an operating system to accommodate particular system requirements. In Sysgen/32, hardware and software features for the operating system are selected and defined through the use of sysgen configuration statements. These statements are defined in a sysgen configuration input file. Driver and system modules provided in the OS/32 package are selected by Sysgen/32, based on the requirements indicated in these sysgen statements.

The user can create a new configuration input file or modify an existing configuration input file using Sysgen/32 commands. The EDIT command enables the user to use all of the OS/32 Edit command repertoire. The HELP and the CONVERSATIONAL commands are available to help the user create a configuration input file.

Once a configuration input file is created, it is processed by the Sysgen/32 program to produce macro calls. These macros are subsequently expanded, assembled and linked to yield an operating system. Figure 1-1 details the step-by-step process that results in the generation of an operating system using Sysgen/32.

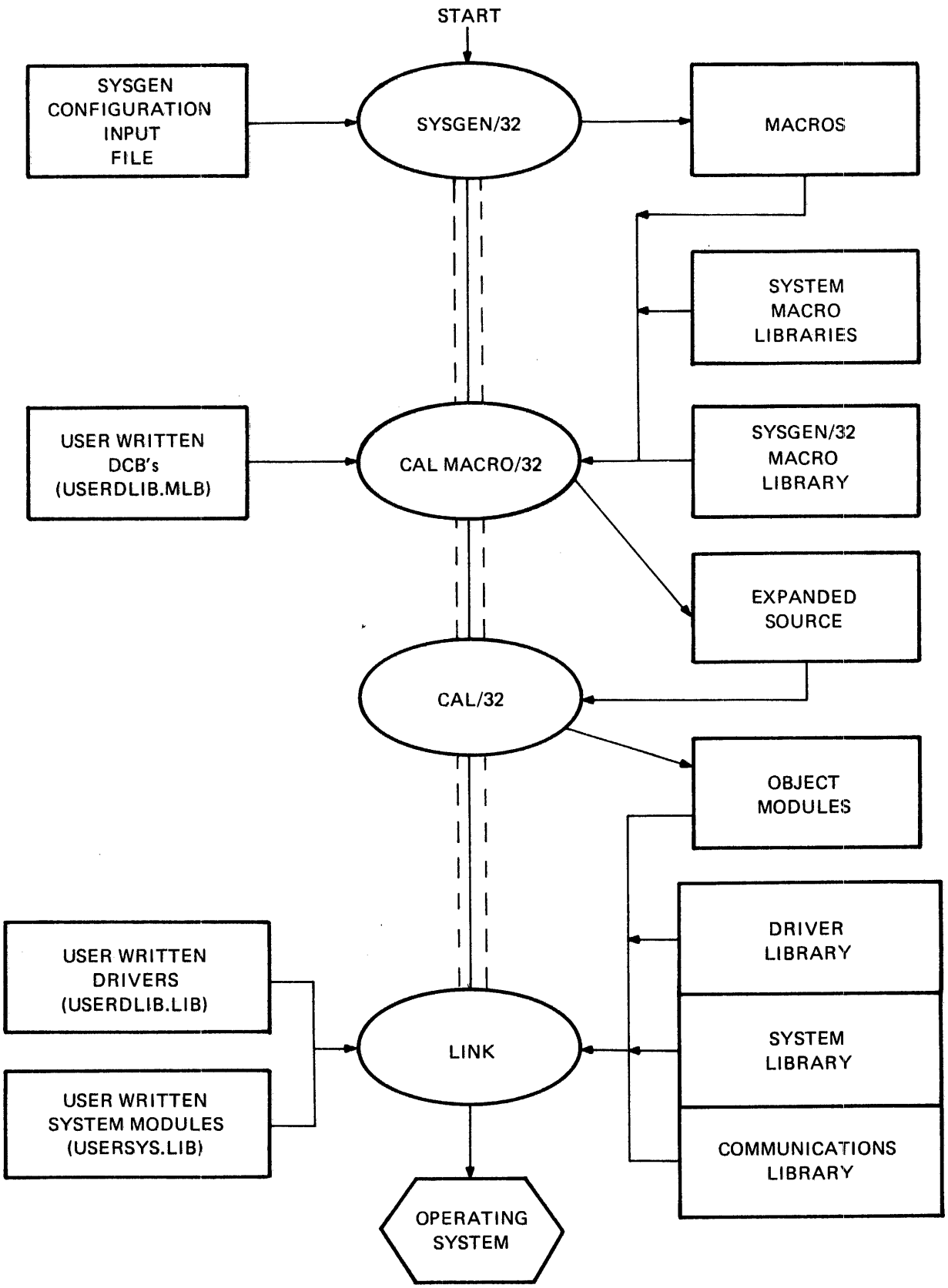


Figure 1-1 Generation of an Operating System Using Sysgen/32

As shown in Figure 1-1, the sysgen process involves four general steps.

1. A configuration input file is created and processed via the Sysgen/32 program, resulting in a set of macro calls.

2. The macro calls are expanded using Common Assembly Language (CAL) Macro/32.

3. The expanded macros are assembled using CAL/32, resulting in object modules.

4. The object modules are linked using OS/32 Link to yield an operating system.

The user has the option to perform these steps in a single process by using a command substitution system (CSS) procedure called sysgen CSS, and executing each program in the overall procedure.

1.2 CREATING AN OPERATING SYSTEM USING THE SYSTEM GENERATION (SYSGEN) COMMAND SUBSTITUTION SYSTEM (CSS)

The sysgen CSS can be used to create an operating system with minimal user interaction. The user creates the sysgen configuration input file and passes the filename and other optional parameters to the CSS.

The macro generation, expansion, assembly and linkage steps are automatically performed according to the sysgen CSS instructions. Figure 1-2 details the components of the sysgen CSS.

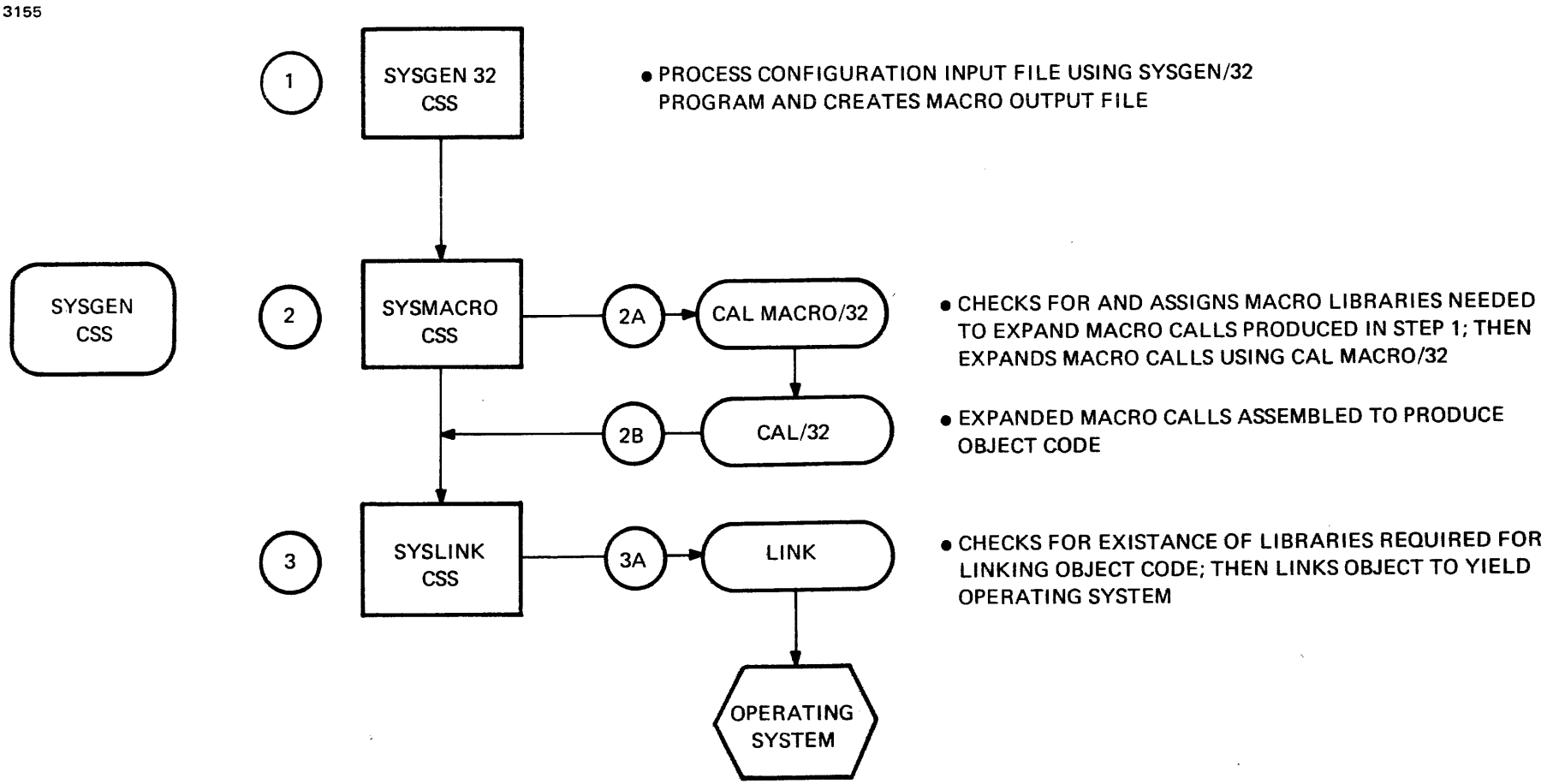


Figure 1-2 Generation of an Operating System Using the Sysgen CSS

1.2.1 Starting the System Generation (Sysgen) Command Substitution System (CSS)

The following format is used when starting the SYSGEN CSS.

Format:

SYSGEN input filename [,segsize increment] [,{library volume / user volume}]

[,Macro and CAL listing flag] [,file save flag]

[,list filename]

Parameters:

input filename	is the 1- to 8-character name of the sysgen configuration input file. The CSS assumes a default extension of .SYS for the configuration input file. If the file specified does not exist, an error message is generated and the CSS will abort.
segsize increment	is a decimal number specifying the number of kb of workspace required by the sysgen task. The default is 40kb.
library volume	is a 1- to 4-character name (with colon) of the disk volume containing the libraries used by CAL Macro/32 and Link during CSS execution. The libraries can be on the user's private account (/P) or on the system account (/S). If this parameter is omitted, the user volume is the default.
Macro and CAL listing flag	is the parameter that determines what files will be printed in the list file. If this parameter is entered, all intermediate files are sent to the list file after the sysgen process is completed. If this parameter is omitted, only error messages are sent to the list file after the sysgen process is completed. Enter a YES to indicate that intermediate files are to be printed.
file save flag	is the parameter that causes the CSS to save all intermediate files created during the sysgen process. If this parameter is omitted, all of the intermediate files created during sysgen are deleted. Enter a YES if intermediate files are to be saved.

list filename	is the file descriptor (fd) of the list file to which all messages are written. If this parameter is omitted, a default file with the input filename.PRT is allocated.

Example:

```
SYSGEN OS32SYS,30,M300:,PRINT1,YES,LIST1
```

This example initiates the sysgen CSS and passes the following information to the CSS:

- OS32SYS is the configuration input file. An extension of .SYS is assumed by the CSS.
- The segment size increment is 30.
- M300: is the library volume containing the libraries used by CAL Macro/32 and Link.
- PRINT1 is the file that will receive the CAL Macro/32 or CAL/32 list output for debugging purposes.
- The YES flag indicates all intermediate files are not to be deleted.
- LIST1 is the list file for messages.

Functional Details:

If an error occurs during the execution of the sysgen CSS, an error message is written to the console and the CSS is aborted. If the CSS reaches a successful completion, the following information is displayed to the console:

- Operating system map filename = (input filename).LST or user-specified fd
- Operating system object filename = (input filename).OBJ
- Operating system task filename = (output volume name + input filename).OS

To use newOSFYL.OS
Rename to;
OS32XXXX.001
must be → — Must Match Loc 7E Number.

1.2.2 Components of the System Generation (Sysgen) Command Substitution System (CSS)

The SYSGEN CSS is comprised of three CSS modules:

- SYSGEN32.CSS
- SYSMACRO.CSS
- SYSLINK.CSS

These modules can be run independently, outside of the sysgen CSS procedure. However, when they are executed independently, the user must pass the appropriate parameters to each CSS. It is particularly useful to be able to execute the SYSLINK CSS routine independently, since this enables a previously linked operating system to be relinked to incorporate changes in system and driver library modules. The following sections detail the procedures for executing the SYSGEN32, SYSMACRO and SYSLINK CSS routines independently.

1.2.2.1 Executing the SYSGEN32 Command Substitution System (CSS)

The SYSGEN32 CSS executes the Sysgen/32 program, which processes the sysgen configuration input file and generates macro calls.

Format:

```
SYSGEN32 input filename [,segsize increment]
```

Parameters:

input filename	is the 1- to 8-character name of the sysgen configuration input file. The CSS assumes a .SYS extension for the configuration input file. If the file specified does not exist, an error message is generated and the CSS aborts.
segsize increment	is a decimal number specifying the number of kb of workspace required by the sysgen task. The default is 40kb.

Functional Details:

If the CSS encounters an error while executing (end of task code other than 0), an error message is generated and the CSS aborts. The list file should be checked for errors. If the CSS reaches a successful completion (end of task code = 0), the file containing the macro calls generated by the Sysgen/32 program will be contained in a file named:

(input filename).MAC

This file is used by the next CSS procedure, SYSMACRO.

1.2.2.2 Executing the SYSMACRO Command Substitution System (CSS)

The SYSMACRO CSS takes the macro calls generated by the Sysgen/32 program, expands them using CAL Macro/32, and assembles the expanded source modules using CAL/32.

Format:

SYSMACRO macro output filename [,{library volume / user volume}]

[,listing flag] [,save files flag] [,list filename]

Parameters:

macro output filename	is the 1- to 8-character name of the file containing the macro calls generated by the Sysgen/32 program. The CSS assumes an extension of .MAC. If the specified file does not exist, an error message is generated and the CSS aborts.
library volume	is the name (colon included) of the disk volume containing the libraries used by CAL Macro/32 during CSS execution. The macro libraries (.MLB) can be on the user's private account (/P) or on the system account (/S). If this parameter is omitted, the user volume is the default.
listing flag	means a complete listing of macro expansion and assembly is desired. If this parameter is omitted, only error messages are listed.

save files flag	means that intermediate files created during the sysgen process are to be saved. If this parameter is not entered, all intermediate files are deleted.
list filename	is the fd of the last file to which all messages are written. If this parameter is omitted, a default file with the name input filename.LST is allocated.

Functional Details:

If the CSS encounters an error (end of task code other than 0), an error message is generated and the CSS aborts.

If the CSS reaches a successful completion (end of task code = 0), the object module assembled by CAL/32 will be contained in a file named:

(macro output filename).OBJ

This file will be used by the next CSS routine, SYSLINK CSS.

1.2.2.3 Executing the SYSLINK Command Substitution System (CSS)

The SYSLINK CSS takes the object modules assembled in the SYSMACRO CSS and links the object with the appropriate libraries to produce an operating system.

Format:

SYSLINK object filename [,{library volume / user volume}] [,files save flag]

Parameters:

object filename	is the 1- to 8-character name of the file containing the CAL object code generated during the SYSMACRO CSS. If the specified file does not exist, an error message is generated and the CSS is aborted.

library volume	is the name (colon included) of the volume where the system and driver libraries reside. The system libraries (.LIB) can be on the user's private account (/P) or on the system account (/S). If this parameter is omitted, the user volume is the default.
file save flag	is the parameter that causes the CSS to save all intermediate files created during the sysgen process. If this parameter is omitted all of the intermediate files created during the sysgen procedure are deleted.

Functional Details:

If an error occurs during execution of the SYSLINK CSS (end of task code other than 0), an error message is generated and the CSS is aborted.

If the CSS reaches a successful completion, the following information is displayed:

- Operating system map filename = (object filename).LST or a user-specified fd
- Operating system object file = (object filename).OBJ
- Operating system task file = (output volume name + object filename).OS

1.3 USING THE SYSTEM GENERATION/32 (SYSGEN/32) PROGRAM

The Sysgen/32 program enables a user to create and/or process a sysgen configuration input file to generate macro calls. The option to process a previously generated configuration input file or interactively create and process a new configuration input file is available.

The Sysgen/32 program runs in batch and interactive environments. In a batch environment, the program processes the configuration input file without user interaction. In an interactive environment, the user can create the configuration input file directly from the command device and correct run-time errors as they occur.

1.3.1 Loading and Starting the System Generation/32 (Sysgen/32) Program

The following commands are used to load and start the Sysgen/32 program.

1.3.1.1 LOAD Command

The system LOAD command loads Sysgen/32 into memory.

Format:

```
LOAD SYSGEN32 [,segsize increment]
```

Parameter:

segsize increment — is an optional parameter that specifies the workspace required by the sysgen task. This increment must be large enough to accommodate processing of all the configured devices. The default is 40kb.

If the segsize is not large enough, the message:

```
LINE____ADDR____STACK OVERFLOW
TASK PAUSED
```

will be displayed. The task must then be cancelled and reloaded with a larger increment.

| START |

1.3.1.2 START Command

The system START command begins execution of the Sysgen/32 program. All parameters are optional.

Format:

START [,INPUT=fd_1] [,OUTPUT=fd_2] [,COMMAND=fd_3] [,LIST=fd_4]

Parameters:

INPUT= fd_1 specifies the configuration input file to be processed by the Sysgen/32 program. The input file must be an existing file.

OUTPUT= fd_2 specifies the output file to which macro calls generated by Sysgen/32 will be written. The output file cannot be an existing file.

COMMAND= fd_3 specifies the command input device. This parameter establishes whether the environment is batch or interactive. If an interactive device is specified, the environment is interactive. CON: is the default for a command device.

LIST= fd_4 specifies the device or file to which all list output and messages generated during sysgen execution are sent. If the list parameter specifies a file, the file must already exist. PR: is the default list device.

Functional Details:

If the START command is entered with both INPUT and OUTPUT parameters specified and the command device is omitted, the Sysgen/32 program will immediately process the configuration input file and assume a batch environment unless the input file is empty.

If the specified input file is empty, the program will display the following message:

READY FOR CONFIGURATION INPUT

The program then waits for the user to interactively input configuration input statements that will be used to create the configuration input file.

If the START command is entered without the INPUT and OUTPUT parameters, the Sysgen/32 commands described in the following sections can be used to:

- Create a sysgen input file in a conversational mode (CONVERSATIONAL).
- Specify the configuration input file (INPUT).
- Specify the macro output file (OUTPUT).
- Modify the sysgen configuration input file (EDIT).
- Initiate processing of the sysgen configuration input file (PROCESS).

The HELP command is also available as a user aid.

1.4 SYSTEM GENERATION/32 (SYSGEN/32) COMMANDS

The following Sysgen/32 commands are available:

- INPUT
- OUTPUT
- CONVERSATIONAL
- EDIT
- HELP
- PAUSE
- PROCESS
- END

1.4.1 INPUT Command

The INPUT command specifies a configuration input file to be used as input to the Sysgen/32 program. This command is used if the INPUT parameter was not specified with the START command.

Format:

```
INPUT fd
```

Parameter:

fd	is the file descriptor of a configuration input file to be processed by the Sysgen/32 program.

Functional Details:

If the specified input file is empty and the command device is an interactive device, the input file can be created either by entering configuration statements directly from the command device after the PROCESS command is entered or conversationally (see Section 1.4.3).

1.4.2 OUTPUT Command

The OUTPUT command specifies an output file to receive the macro calls generated by Sysgen/32. This command is used if the OUTPUT parameter was not specified with the START command.

Format:

```
OUTPUT fd
```

Parameter:

fd	is the file descriptor of the file that receives the Sysgen/32-generated macro calls.

Functional Details:

The specified output file must be an empty, nonexistent file. If the OUTPUT command is entered with the name of an existing file, an error is generated.

1.4.3 CONVERSATIONAL Command

The CONVERSATIONAL command initiates a sysgen prompt and user response session in an interactive environment. This prompt and response session is a user aid in creating a configuration input file.

Format:

CONVERSATIONAL

Functional Details:

The Sysgen/32 program issues interactive prompts relating to the hardware configuration and software options. The possible responses, in parentheses, and the defaults, in brackets, are displayed after each prompt, where applicable. Defaults are taken if a carriage return (CR) is depressed for all but device prompts. A response must be entered for all device prompts.

The program will create sysgen configuration statements based on the responses to the prompts. Acceptable statements will be written to the specified input file. Nonacceptable statements or responses will cause the program to generate an error message and to reissue the appropriate prompt, or prompts, until an acceptable response is entered.

If the CONVERSATIONAL command is entered before an input file was defined via the INPUT command or INPUT parameter of the START command, the following message is displayed:

INPUT MUST BE ENTERED

When in CONVERSATIONAL mode, the user can access the HELP file by entering a question mark (?) in response to a prompt. The question mark causes the HELP file to display pertinent information about a configuration statement or parameter and briefly describes its use. The program then reissues the prompt sequence for the configuration statement. After all prompts have been issued, the following message is displayed:

CONVERSATIONAL PROCESSING COMPLETE

The PROCESS command can then be entered to start the processing of the newly created input file. Certain errors will not be detected during the conversational session, but will be detected in the processing phase. These errors can be corrected using the EDIT command (see Section 1.4.4).

Example:

```
 PROCESSOR MODEL (7/32,8/32,3200MPS,3205,3210,3220,3230,3240,
                  3250) [3220]
>?
 CPU:  SPECIFIES THE 32-BIT PROCESSOR MODEL.

       COMMAND FORMAT:  CPU [N][,R]


                  N IS THE MODEL NUMBER OF THE PROCESSOR.

                    THE VALUE FOR N MAY BE:
                          7/32, 8/32, 3200MPS,3205,3210,3220,
                          3230,3240,3250.

                  R SPECIFIES THE NUMBER OF REGISTER SETS,
                    2 OR 8.


                  THE DEFAULT VALUES FOR THE CPU COMMAND ARE
                    3220 WITH 8 REGISTER SETS.
 PROCESSOR MODEL (7/32,8/32,3200MPS,3205,3210,3220,3230,3240,
                  3250) [3220]
>CR
 NUMBER OF REGISTER SETS (2 OR 8) [8]
>CR
 O/S VERSION (8 CHAR. ALPHANUMERIC STRING) [BLANKS]
```

1.4.4 EDIT Command

The EDIT command makes the entire OS/32 Edit command repertoire available to the user. This command is available during program execution in interactive mode only.

Format:

```
EDIT
```

Functional Details:

All of the OS/32 Edit commands can be used to correct or modify the configuration input file. When the changes are made and the file is saved, the END or DONE command terminates the edit session and returns control to the Sysgen/32 program.

Example:

```
>INPUT  SYS1.SYS          }     Specifies input and output
>OUTPUT  SYS1.MAC         }     files.
>PROCESS                  }

ILLEGAL VALUE     3280    }     Error is encountered.
READY FOR SYSGEN COMMANDS }     Sysgen program enters command
                                input mode.
>EDIT                     )
READY FOR EDIT COMMANDS   |
>GET SYS1.SYS             |
>OPT LIST=CON:            |     Enters edit mode and corrects
>T/3280/                  }     error.
2 CPU 3280                |
>SU/3280/3230/            |
>DONE                     |
WORKFILE = FIXD:SYS1.000  )
RENUMBERED INPUT FILE AVAILABLE FIXD:SYS1.SYS
READY FOR SYSGEN COMMANDS }     resume Sysgen/32 processing
>PROCESS                  }
```

| HELP |

1.4.5 HELP Command

The HELP command accesses the HELP file and displays sysgen commands and configuration statements. A brief description of each command or configuration statement and information on how to use each is also displayed.

Format:

HELP [{name} | {*}]

Parameters:

name	specifies the name of a specific sysgen statement or command to be displayed.
*	specifies that all sysgen statements and commands be displayed.

Functional Details:

If the HELP command is entered without a parameter, the following message is displayed:

```
For a list of commands type HELP *
For help on any command mnemonic, type HELP <mnemonic>
```

Example 1:

```
>HELP *
 I(NPUT)           O(UTPUT)          CONV(ERSATIONAL)   ED(IT)
 PA(USE)           PR(OCESS)         END                ACC(OUNTING)
 B(ACKGROUND)      CL(OCK)           CM(DLEN)           CP(U)
 CS(S)             DA(TE)            DEVA(DS)           DEVI(CES)
 DIR(ECTORY)       DI(SCBLOCK)       DS(YS)             ENDC
 ERRORR(EC)        F(LOAT)           IOC(LASS)          IL(EVEL)
 INT(ERCEPT)       IR(EADER)         IT(AM)             J(OURNAL)
 L(OGLEN)          LPU               MAXA(PU)           MAX(TASK)
 MCON(FIG)         MEMCHECK          ME(MORY)           MIR(ROR)
 NOSE(G)           TG(D)             QU(EUE)            MO(DULE)
 SP(OOL)           SPL(32)           ST(ARTUP)          R(OLL)
 TC(OM)            TE(MP)            VER(SION)          SST(ABLE)
 COOR(DINATION)    COPY              POW(ERFAIL)        V(OLUME)
                                                        SL(ICE)
 For HELP on any of the above command mnemonics, type HELP
 <mnemonic>
```

Example 2:

```
>HELP ACC
 ACCOUNTING: INCLUDES ACCOUNTING SUPPORT.

       COMMAND FORMAT:  ACCOUNTING [{N}] [,NOFILEACCOUNTING]

           N = DECIMAL VALUE FROM 2 THROUGH 32, SPECIFYING
           THE MAXIMUM NUMBER OF ACCOUNTING CLASSES.
           DEFAULT FOR N = 4.

           NO FILE ACCOUNTING SPECIFIES THAT FILE ACCOUNTING
           SUPPORT IS EXCLUDED.
```

| PAUSE |

1.4.6 PAUSE Command

The PAUSE command interrupts execution of the Sysgen/32 program and returns control to the operating system.

Format:

```
PAUSE
```

1.4.7 PROCESS Command

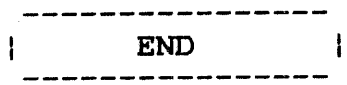
The PROCESS command initiates processing of the sysgen configuration input file.

Format:

PROCESS

Functional Details:

Do not use the PROCESS command if both a nonempty input file and an output file were specified as parameters of the START command, but no command device was entered.



1.4.8 END Command

The END command ends the Sysgen/32 program.

Format:

```
END
```

Functional Details:

An end of task code other than zero indicates that an error occurred during Sysgen/32 execution in a batch environment.

CHAPTER 2
SYSTEM GENERATION/32 (SYSGEN/32) CONFIGURATION STATEMENTS

2.1 INTRODUCTION

Sysgen configuration statements make up the configuration input file that defines the hardware and software features of the target operating system. The sysgen configuration statements are:

ACCOUNTING	FLOAT	NOSEG
BACKGROUND	ILEVEL	POWERFAIL
CLOCK	INTERCEPT	QUEUE
CMDLEN	IOCLASS	ROLL
COORDINATION	IREADER	SLICE
COPY...ENDCOPY	ITAM	SPOOL
CPU	JOURNAL	SPL32
CSS	LOGLEN	SSTABLE
DATE	LPU	STARTUP...ENDS
DEVADS	MAXAPU	TCOM
DEVICES...ENDD	MAXTASK	TEMP
DIRECTORY	MCONFIG	TGD
DISCBLOCK	MEMCHECK	VERSION
DSYS	MEMORY	VOLUME
END	MIRROR	
ERRORREC	MODULE...ENDM	

Certain statements may span more than one line. A comma, input as the last nonblank character, indicates that the statement is continued on the next line.

Any characters following an asterisk (*) are treated as comments. If an asterisk is in column 1, the entire line is treated as a comment line. Comment lines are copied to the list device.

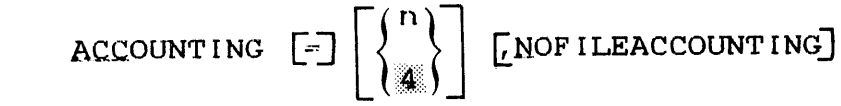
2.2 SYSTEM GENERATION (SYSGEN) CONFIGURATION STATEMENTS

The following sections alphabetically present, and describe in detail, each sysgen configuration statement.

2.2.1 ACCOUNTING Statement

The ACCOUNTING statement specifies that accounting support is included in the system.

Format:



Parameters:

n	is a decimal number from 2 to 32 that specifies the maximum number of device or file classes to be supported by the Accounting Facility. The minimum number of classes must be two, because two classes are needed to handle the four file types. If this parameter is omitted, 4 is the default.
NOFILEACCOUNTING	prevents logging of accounting data when files are deleted or renamed.

Functional Details:

Each device or file class supported by the Accounting Facility must be defined by the IOCLASS statement. Each input/output (I/O) class supported by the Accounting Facility occupies 12 bytes in the user task control block (TCB) and four bytes in the multi-terminal monitor (MTM) for each MTM user of the Accounting Facility. If the ACCOUNTING statement is omitted, accounting support is not included in the system.

See the IOCLASS statement or IOCLASS parameter of the device statement for file or device classes.

The NOFILEACCOUNTING option allows the user to reduce the size of the account transaction file (ATF).

Accounting support is required by the load-leveling executive (LLE) if used on the Model 3200MPS System.

NOTE

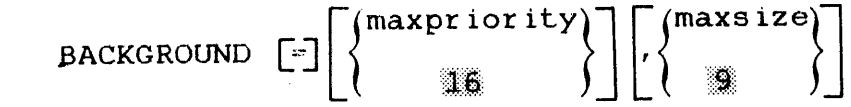
The maximum number of I/O classes that are currently processed by the Accounting Reporting Utility is 10. See the OS/32 System Support Utilities Reference Manual.

| BACKGROUND |

2.2.2 BACKGROUND Statement

The BACKGROUND statement establishes the maximum priority and maximum amount of system space for a background task in the system.

Format:



Parameters:

maxpriority	is a decimal value from 11 to 248 specifying the highest priority at which a background task can run. If this parameter is omitted, 16 is the default.
maxsize	is a decimal value in increments of 0.25kb specifying the largest system space area in which system data structures (i.e., file control block (FCB), task queue entry (TQE), TCB) of a background task can be stored. If this parameter is omitted, 9 is the default.

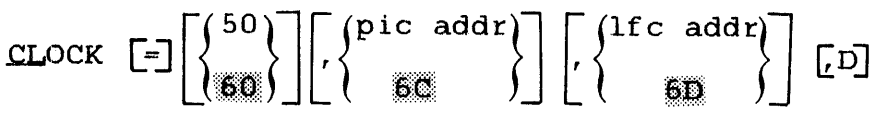
Functional Details:

If this statement is omitted, the default parameters are assumed.

2.2.3 CLOCK Statement

The CLOCK statement sets the line frequency of the clock and device addresses of both the precision interval clock (PIC) and line frequency clock (LFC) for the system. Together, these clocks are called the universal clock (UCLOCK) module.

Format:



Parameters:

50	is a hexadecimal number indicating the line frequency value. If this parameter is omitted, 60 is the default.
pic addr	is a hexadecimal number specifying the physical device address of the PIC. The user-specified address must not be greater than the maximum device address specified by the DEVADS statement. If this parameter is omitted, 6C is the default.
lfc addr	is a hexadecimal number specifying the physical device address of the LFC. The user-specified address must not be greater than the maximum device address specified by the DEVADS statement. If this parameter is omitted, 6D is the default.
D	is the alphabetic character D specifying that the date and time are to be displayed on the display panel. This parameter should be specified only if the central processing unit (CPU) statement indicates that the target system is a Perkin-Elmer Model 7/32 or 8/32 processor.

Functional Details:

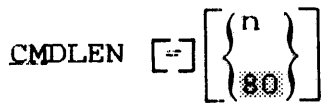
If this statement is omitted, the default parameters are assumed.

| CMDLEN |

2.2.4 CMDLEN Statement

The CMDLEN statement specifies the maximum length of the system command buffer or buffers if the command substitution system (CSS) is supported in the system.

Format:



Parameter:

n	is a decimal number from 32 to 1,024 specifying the number of bytes in the system's command buffers. If this parameter is omitted, 80 is the default.

Functional Details:

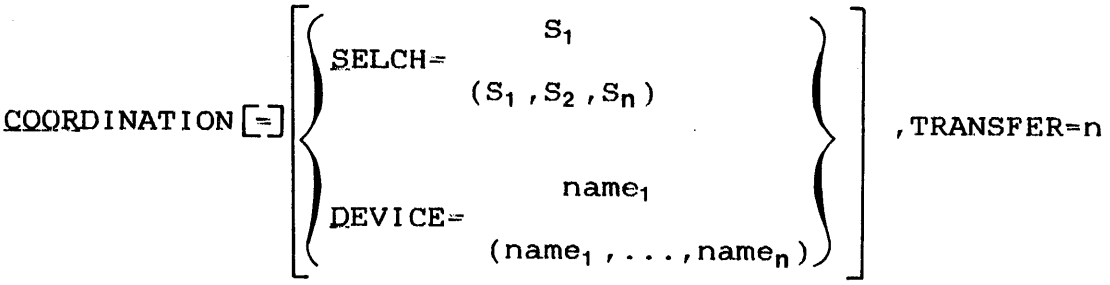
If CSS is supported, or commands are read from devices or files with record lengths greater than 80, a larger command buffer length must be specified. If CSS is supported, parameter substitution causes a small input line length to be expanded to a greater length that must be less than or equal to the system command buffer length.

If this statement is omitted, 80 is the default.

2.2.5 COORDINATION Statement

The COORDINATION statement establishes the maximum number of simultaneous data transfers allowed in the target system. This statement must be entered if the number of selector channels (SELCHs) is greater than the number of simultaneous data transfers allowed in the target system.

Format:



Parameters:

SELCH= S_1 - S_n specifies the SELCH to be coordinated.

DEVICE= $name_1$ - $name_n$ specifies the devices requiring coordination.

TRANSFER= n specifies the maximum number of simultaneous transfers. If 1 is specified, the resulting extended direct memory access (EDMA) node functions like a supernode (the main coordination node that coordinates the other nodes and their attached devices), providing coordination only, without impact on the transfer rate. Table 2-1 provides recommended numbers of simultaneous transfers for the various Perkin-Elmer 32-bit processors.

TABLE 2-1 RECOMMENDED NUMBER OF SIMULTANEOUS DMA TRANSERS

NUMBER OF TRANSFERS	SYSTEM
1	Model 7/32, 3205
1	Model 8/32 with one or more nonbuffered SELCHs; extended SELCH (ESELCH)
4	Model 8/32 with buffered SELCH (BSELCH)
4	Model 3210, 3220 and 3230 with BSELCHs
*	Model 3240, 3250 and 3200MPS

* This value is dependent on the hardware configuration.

Functional Details:

The COORDINATION statement should be used if the number of SELCHs in a system with MSM80 or MSM300 disks or 6250 bits per inch (bpi) magnetic tape drives is greater than the number of simultaneous data transfers allowed in the system.

One COORDINATION statement must be entered for every group of SELCHs or devices requiring coordination.

2.2.6 COPY Statement

The COPY statement copies data into the macro output file. Statements are copied to the file in the format in which they are read until the ENDCOPY statement is encountered in column 1.

Format:

```
COPY
[line1]
   .
   .
   .
[linen]
ENDCOPY
```

Parameter:

line	specifies data to be copied into the macro output file.

Functional Details:

The COPY statement is useful when a user wants to include user-specified macros in the generated macro file. It is recommended that the macros that are more commonly called in a normal sysgen should be included.

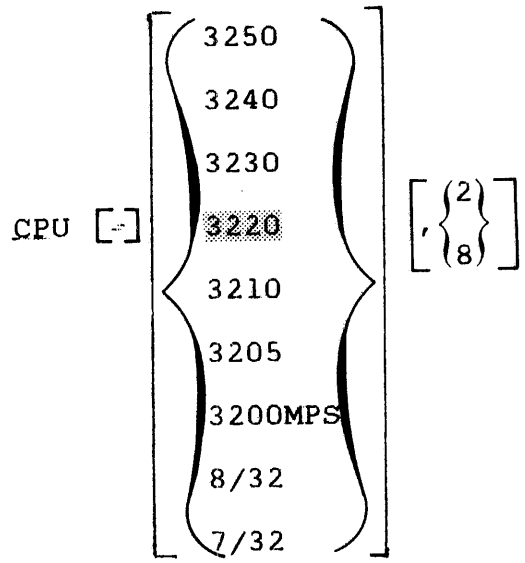
Example:

```
COPY
 MCALL  DCBI,CCBI,CONVNUM,EVNGEN
 MCALL  MMDGEN,DCB53,DCB54
 MCALL  MTRI,DCB65
 MCALL  BIOCGEN,DCB39
ENDCOPY
```

2.2.7 CPU Statement

The CPU statement specifies the Perkin-Elmer 32-bit processor for which the system is being configured.

Format:



Parameters:

3250 3240 3230 3220 3210 3205 3200MPS 8/32 7/32	specify the target system processor model number. If this parameter is omitted, 3220 is the default. The Model 3200MPS System is valid for the multiprocessor system only.
2 8	are decimal numbers that specify the number of register sets supported by the hardware. The Perkin-Elmer Model 3200MPS, 3205, 3210, 3220, 3230, 3240, 3250 and 8/32 Systems support eight register sets. If this parameter is omitted for the Model 7/32 processor, 2 is the default. If this parameter is omitted for all other processors, 8 is the default.

Functional Details:

If this statement is omitted, the default parameters are assumed.

2.2.8 CSS Statement

The CSS statement specifies the maximum number of nested CSS calls allowed in one routine for the target system.

Format:

$$\underline{\text{CSS}}\ [=]\ \left[\begin{Bmatrix} n \\ 5 \end{Bmatrix}\right]$$

Parameter:

n	is a decimal number from 1 through 249 specifying the maximum number of nested CSS calls in one routine; i.e., the number of routines that can be active at one time. If CSS is not supported, 1 must be specified. If this parameter is omitted, 5 is the default.

Functional Details:

The operating system allocates the amount of memory required for a CSS by the following equation:

```
MEMORY = CMDLEN n * CSS n
```

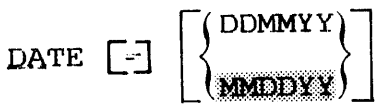
If this statement is omitted, the default is 5.

```
----------------
|      DATE      |
----------------
```

2.2.9 DATE Statement

The DATE statement specifies the format in which the current date is expressed for the system.

Format:



Parameters:

DDMMYY	is day-month-year format. If this parameter is omitted, MMDDYY is the default.
MMDDYY	is month-day-year format.

2.2.10 DEVADS Statement

The DEVADS statement specifies the maximum number of devices, maximum device address, maximum number of bytes occupied by the interrupt service pointer (ISP) table and the starting address of the memory access controller (MAC) or SSTABLE in the system.

Format:

DEVADS [=] [{ 3 / 1 / 0 }]

Parameters:

3
1

are decimal numbers specifying a table entry containing established maximum values. If not specified, 0 is the default. The established values specified in the DEVADS statement must correspond to the hardware configuration. Table 2-2 lists the maximum values associated with each DEVADS statement.

TABLE 2-2 DEVADS STATEMENT VALUES

ENTRY NUMBER	NUMBER OF DEVICES	MAXIMUM DEVICE ADDRESS	NUMBER OF BYTES IN ISP	MAC, MAT OR SST STARTING ADDRESS
3	*1,023	X'3FF'	2,048	X'900'
1	511	X'1FF'	1,024	X'500'
0	255	X'0FF'	512	X'300'

* This number refers to the number of device addresses supported by the operating system.

| DEVICES...ENDD |

2.2.11 DEVICES...ENDD Statements

The DEVICES...ENDD statements are used to delimit the device descriptor statements. Every device to be configured in the target system must be defined by a device descriptor statement.

Each device descriptor statement requires three parameters, the device name specification, the device address and the device code. These three parameters must be entered in the order described. Optional parameters describing other device details can be entered in any order. Not all configuration statements are supported by the DEVICES...ENDD statement.

Format:

```
DEVICES

   dev name:,dev address,dev dcode [,CLOCK={XA|XB|XC|XD}] [,CM=n]

      [,CONSOLE] [,CONTROLLER=n] [,DISC] [,DUAL] [,EOV] [,ILEVEL=n]

      [,INTIMER={n|10}] [,IOCLASS={n|global I/O class}] [,IOLIMIT={n|10}]

      [,LEADCOUNT=n] [,LINESTATUS=n] [,MAXFRAMES=n]

      [,MAXWRITEBUFF={n|1024}] [,MINREADBUFF={n|64}] [,MTO=n] [,N2={n|10}]

      [,NCS={n|E8E8}] [,NODISC] [,NONSHARED] [,OUTIMER={n|10}]

      [,PADCOUNT=n] [,POLLDELAY=n] [,POLLIMIT=n] [,POLLTIME={n|3}]

      [,QUEUE=name] [,READCONTROL=n] [,RECLEN=n]

      [,RESPONSETIME={n|7FFF,3}] [,SCREENTIME=n] [,SELCH=n] [,SIZE=n]

      [,SPINDLE=n] [,SSA=name] [,TO2={n|30}] [,TRANSLATE=name]

      [,USCI=n] [,USCO=n] [,USER=(uparm[=parameter])]

      [,VIRT=(startcyl,ncyl)] [,WAKEUP={n|30}] [,WRITECONTROL=n]

      [,XDCOD=n]

.
.
.
ENDD
```

Parameters:

dev name — is a 1- to 4-character device mnemonic specifying a unique device name. The first character must be alphabetic, the remaining alphanumeric.

dev address — is a hexadecimal number specifying the physical address of a device. This number must not be greater than the maximum device address specified in the DEVADS statement. For pseudo devices, specify 0.

dev dcode — is a decimal number ranging from 16 to 254 specifying the device code. If the system is to be configured with devices defined as pseudo output devices, spooling must be supported. A pseudo device is a virtual device that establishes a correlation between a user and a physical output device. Each pseudo device is logically linked to a physical output device so many users or tasks can output to the same logical device simultaneously. If pseudo device PR1: is logically linked to PRT1: and PRT2:, an assignment to PR1: will eventually generate output on either PRT1: or PRT2:.

If the pseudo spooling output device is chosen, the extended device code (XDCOD=), the record length (RECLEN=) and the size (SIZE=) parameters can be optionally chosen.

Example:

```
PR1:, 0, 1, XDCOD=114, RECLEN=80, SIZE=56
```

*CLOCK= — XA specifies the A clock.
XB specifies the B clock.
XC specifies the C clock.
XD specifies the D clock.

CM= — n is a hexadecimal number that specifies the channel manager device address. This parameter specifies a channel manager supported device.

CONSOLE — identifies the device as the console device.

CONTROLLER= — n is a 2-digit hexadecimal number specifying the controller address.

DISC — device specifies OS/32 directory support. This is the default parameter for disk devices only.

DUAL — specifies that dual port option is in effect for mass storage module (MSM) disks.

EOV — indicates that end of volume (EOV) labels on magnetic tapes are to be processed. If this parameter is not specified, EOV labels are not supported.

ILEVEL= — n is a decimal number from 0 to 3 specifying the hardware interrupt level of each individual device. If this parameter is omitted, the global interrupt level (see the ILEVEL statement) is assumed for this device.

INTIMER= — n specifies in seconds the input error timer used at the supervisor call 15 (SVC15) level. The default is 10 seconds.

IOCLASS= — n is a decimal number from 0 to 31 specifying the I/O class of each individual device. This number must not be greater than the maximum number of device or file classes specified in the ACCOUNTING statement. If this parameter is omitted, the global IOCLASS is assumed.

IOLIMIT= — n specifies the number of retries that will be attempted for an I/O error. The default is 10 retries.

*LEADCOUNT= — n is a decimal number specifying the number of leading synchronous characters within a unit of data.

LINESTATUS= — n defines the static line status (SLS) as used in zero-bit insertion/deletion (ZBID) data link control (ZDLC). This parameter also defines the subprotocol that ZDLC will use on the communications line. Options include normal response mode, asynchronous response mode, symmetrical response mode, LAP, LAPB, etc. All protocols may not be supported.

MAXFRAMES= — n specifies the maximum number of unacknowledged frames that can be outstanding in ZDLC. The maximum possible values are 7 for normal control mode and 127 for extended control mode.

MAXWRITEBUFF= — n specifies the maximum number of bytes in size for a write buffer that can be received or transmitted. The default is 1,024 bytes.

MINREADBUFF= n specifies the minimum number of bytes required for a read buffer. The default is 64 bytes.

MTO= n specifies the number of tenths of a second that must expire before devices are repolled if there were no positive responses to the previous polls. If this parameter is not specified, devices are not repolled.

N2= n specifies the N2 count as defined in the X.25 specifications and indicates the number of times an information frame can be retransmitted following a T1 time-out or similar error. The default is 10 times.

NCS= n is a hexadecimal number that defines the numbered and information control commands that can be used on a ZDLC line. The default is E8E8.

NODISC specifies no OS/32 directory support for a device.

NONSHARED specifies that an existing shared-busy condition for the device should be ignored.

OUTIMER= n specifies the output error timer used at the SVC15 level. The default is 10 seconds.

*PADCOUNT= n is a decimal number specifying the length of a pad sequence within a unit of data. For asynchronous devices, this parameter specifies the length of the carriage return (CR)/line feed (LF) sequence that is appended at the end of the user buffer. For synchronous devices, this parameter is for the number of bits (XFF) appended to the end of the write transmission.

POLLDELAY= n specifies the amount of time certain data communication protocols take to go into a delay following a nonresponsive polling cycle. If this parameter is not specified, there is no delay.

*POLLIMIT= n is a decimal number specifying the limit of polling retries allowed on a line.

POLLTIME= n specifies, in seconds, the time allowed for a response following the transmission of a poll sequence. The default is 30 units of 100ms (3 seconds).

QUEUE= name is a 1- to 8-character alphanumeric string indicating the name of an alternate disk I/O scheduling routine.

*READCONTROL= n is a hexadecimal or decimal mask specifying read control characters. If specified in hexadecimal, the value must be preceded by an X. Setting bits in this field can enable the special character handling required by the terminal. Changing this value enables specification of special characters that can be used for termination of a line, for a line delete or for a backspace. See Table 2-3 for a description of options.

TABLE 2-3 SPECIAL ASYNCHRONOUS CHARACTERS

TYPE	CHARACTER	ASCII	MEANING	READ MASK	WRITE MASK
Termination characters	CR	X'0D'	Carriage return	X'8000'	X'8000'
	ETX	X'03'	End of text (CTRL C)	X'4000'	X'4000'
	EOT	X'04'	End of transmission (CTRL D)	X'2000'	X'2000'
	User-defined		Terminate read/write	X'0200'	X'0200'
	Any enabled line delete character		Terminate read on line delete	X'0100'	
Backspace	BS	X'08'	Backspace (CTRL H)	X'0080	
	<--	X'5F'	Back arrow or underscore (shift 0)	X'0040'	
	User-defined	---		X'0010'	
Line delete	#	23	Number sign	X'0108'	
	User-defined			X'0102'	
	NAK or CAN	15 18	NAK (CTRL-U) CANCEL (CTRL-X)	X'0101' X'0101'	
Control	DC1, DC2	11,12	Carousel/printer (START)		X'0400'
	DC3, DC4	13,14	Buffer controls (STOP)		X'0400'

TABLE 2-3 SPECIAL ASYNCHRONOUS CHARACTERS (Continued)

TYPE	CHARACTER	ASCII	MEANING	READ MASK	WRITE MASK
Break	BREAK		Break fulfills prepare		X'0001'
			Do not begin a write if break status is present; return break error.		X'0010'
	ESC	X'1B'	Allow Escape to break write		X'0008'

RECLEN= n specifies the physical record length of a device.

*RESPONSETIME= n specifies, in seconds, the amount of response time to a nonscreen read. For multidrop terminals involving a microcomputer response, this should be a short time. For point-to-point terminals involving human response time, the amount of time should be longer.

The defaults are:

For device control blocks (DCBs) 156 and 157	7FFF seconds
For DCB 158	3 seconds

SCREENTIME= n is a decimal or hexadecimal number specifying the amount of time required to read a full screen of data. This parameter is used for error recovery purposes. If the time allotted is not large enough, an I/O could fail in the middle of a screen read. If too much time is allotted, there could be excessive error recovery time expended. The recommended times are:

For DCBs 156 and 157	25 seconds
For DCB 158	20 seconds

SELCH= n is a 2-digit hexadecimal number specifying the SELCH address.

SIZE= n specifies the page size for pseudo devices.

SPINDLE= n is a decimal number from 0 to 3 specifying the floppy disk spindle number.

SSA= name specifies the secondary station address polling address generated for a ZDLC station and applies only to stations defined at sysgen. Additional stations can be dynamically generated later. The recommended polling addresses are:

For DCBs 181 and 183	01
For DCB 186	01 for the first station and 03 for the second

TO2= n is a hexadecimal number that specifies T1 time-out specified in the X.25 protocol. This parameter indicates the time that can be allotted for a response after transmitting an information frame. The default is 30 seconds.

TRANSLATE= name specifies a 1- to 8-character alphanumeric name of the translation table used for the device.

USCI= n specifies a hexadecimal 32-bit mask that defines the unnumbered protocol commands used with the ZDLC protocol. This parameter defines the input commands accepted on the line. See the USCO parameter for the default fields of this parameter.

USCO= n specifies a hexadecimal 32-bit mask that defines the unnumbered protocol commands used within the ZDLC protocol. This parameter defines the output commands that can be sent on the line.

The default for these fields differs depending upon the subprotocol defined by the LINESTATUS parameter.

Examples:

Normal response mode, primary, extended control:

```
UCSO = 00800010
UCSI = 10884000
```

Normal response mode, secondary, extended control:

```
UCSO = 10884000
UCSI = 00800010
```

Normal response mode, primary, normal control:

```
UCSO = 00808000
UCSI = 10884000
```

Normal response mode, secondary, normal control:

```
UCSO = 10884000
UCSI = 00808000
```

Asynchronous response mode, primary, extended control:

```
UCSO = 00904000
UCSI = 10884000
```

Asynchronous response mode, secondary, extended control:

```
UCSO = 10884000
UCSI = 00904000
```

Asynchronous response mode, primary, normal control:

```
UCSO = 10804000
UCSI = 10884000
```

Asynchronous response mode, secondary, normal control:

```
UCSO = 10884000
UCSI = 10804000
```

Symmetrical response mode, extended control:

```
UCSO = 10984000
UCSI = 10984000
```

Symmetrical response mode, normal control:

```
UCSO = 10884000
UCSI = 10884000
```

LAP response mode, extended control:

```
UCSO = 10984000
UCSI = 10984000
```

LAP response mode, normal control:

```
UCSO = 10884000
UCSI = 10884000
```

LAPB response mode, extended control:

```
UCSO = 10894000
UCSI = 10894000
```

LAPB response mode, normal control:

```
UCSO = 11884000
UCSI = 11884000
```

USER= uparm is a user-defined parameter of up to seven characters that is defined in the DCB macro definition.

Parameter is a maximum of 30 characters and must be preceded by the equal to sign (=). Multiple user statements are allowed.

This option allows a user to have Sysgen/32 pass a user-defined macro parameter to the .MAC file. The user parameter is written to the macro call first, allowing positional parameters to be used.

Examples:

```
TTY1:,2,16,USER=(MYPARM)

CAR1:,3,22,USER=(PARM1=(12,15))

CAR2:,4,22,USER=(PARM1),USER=(PARM2)
```

VIRT= startcyl defines the starting cylinder, counting from zero, of the virtual disk specified by this device statement.

ncyl defines the number of cylinders in this virtual disk. Values default to decimal unless preceded by an X to indicate a hexadecimal value. See Section 4.11.2.2 for a discussion of virtual disk.

Example:

VIRT= (0,323)	323	cylinders starting at cylinder zero
VIRT= (323,250)	250	cylinders starting at cylinder 323
VIRT= (573,250)	250	cylinders starting at cylinder 573

	823	= Total cylinders

NOTE

This is a reasonable layout for a MSM300 removable disk (823 cylinders).

WAKEUP= n defines the amount of time in seconds that a certain protocol will go into a time delay if it has nothing to do. The default is 30 units of 100ms (3 seconds).

*WRITECONTROL= n is a hexadecimal mask specifying the write control character. The value must be preceded by an X. By setting bits in this field, the user can enable the special character handling required by the terminal.

XDCOD= xdcod is a decimal or hexadecimal halfword specifying additional configuration information within a device. If a hexadecimal value is specified, it must be preceded by an X.

This parameter can be used to request full bidirectional I/O control (BIOC) support by specifying X08 and a clock. A clock is selected from the following:

0A = A clock
0B = B clock
0C = C clock
0D = D clock

Example:

XDCOD = X080D

In this example, X08 specifies full BIOC support, and 0D specifies the D clock. See the OS/32 Operator Reference Manual and the Multi-Terminal Monitor (MTM) Reference Manual for a description of the features of the BIOC driver.

NOTE

The asterisk (*) preceding some of the parameters denotes parameters applicable to data communication devices only (see Section 2.2.11.2).

Table 2-4 shows the extended device codes for data communication devices. See Table 4-3 for BIOC XDCOD bit descriptions.

TABLE 2-4 EXTENDED DEVICE CODES FOR DATA COMMUNICATION DEVICES

BIT	HEX MASK (DECIMAL VALUE)	MEANING
0	8000	Is master/slave bit (processor-to-processor link only).
	8000 (32768)	Indicates that this end of processor-to-processor link is master.
	0000 (0)	Indicates that this end of processor-to-processor link is slave.
1-3	7000	Is reserved; must be zero.
4-5	0C00	Are line configuration bits.
	0800 (2048)	Indicates automatic dial-in or manual dial-out.
	0400 (1024)	Indicates a leased line.
	0000 (0)	Indicates a directly connected null modem cable.
6-7	0300	Are line protocol bits.
	0300 (768)	Indicates half-duplex 2-wire.
	0200 (512)	Indicates simplex write.*
	0100 (256)	Indicates simplex read.*
	0000 (0)	Indicates half-duplex 4-wire.*
8	0080	Is explicit connect request bit.
	0000	Indicates that the system will do an automatic connect if an SVC1 read/write request is issued to a line that is not connected. Status returned is 8225.

TABLE 2-4 EXTENDED DEVICE CODES FOR DATA COMMUNICATION DEVICES (Continued)

BIT	(DECIMAL VALUE)	MEANING
8 (con't)	0000 (con't)	If the line is disconnected during read/write request, A0XX status is returned. Next read/write issued will cause system to automatically connect the line.
	0080	Indicates system will return error A018 if SVC1 read/write request is issued to a line that is not connected.
9	0040	Is reserved; must be zero.
10-11	0030	Are clock bits (PALS/PASLA only).
	0030 (48)	Indicates clock D.
	0020 (32)	Indicates clock C.
	0010 (16)	Indicates clock B.
	0000 (0)	Indicates clock A.
12-15	000F	Makes up the default option index for Models 1200 and 1250/1251. Must be zero for all other devices.

* Requires adapter, strapped full-duplex

The common physical record lengths for data communication terminals are shown in Table 2-5.

TABLE 2-5 PHYSICAL RECORD LENGTHS FOR DATA COMMUNICATION TERMINALS

TERMINAL	RECORD LENGTH
Model 550/550B	80
Model 1100 VDU	80
Model 1200 VDU	80
VDU Models 1250/1251	80
Model 6100	80
Carousel	128
M33 TTY	72
M35 TTY	80
SIGMA 10 terminal	73
Remote line printer	132

Functional Details:

Device codes and device addresses determine shared-busy conflicts between devices, such as fixed and removable disks in the same drive, cassettes and TTY/KP with TTY/RP on the same device. The shared-busy conflicts can be overridden by the NONSHARED parameter. Only one channel control block (CCB) is created for disks with a shared-busy conflict.

Shared device leafs will be created by specifying the same device address for each device sharing a leaf. In the following example, each device will share the same device leaf.

Example:

```
CRT1:,10,39

CRT2:,10,39

CRT3:,10,39
```

If ILEVEL and IOCLASS statements precede a group of devices, they remain in effect until another ILEVEL or IOCLASS statement is read. The ILEVEL and IOCLASS parameters in the device statement can be used to override the global setting specified in the global ILEVEL or IOCLASS statement for a specific device.

Default values for all standard devices supported by Sysgen/32 are maintained in the Sysgen/32 Macro Library. These defaults represent driver initialization and termination routines, disk sizes, device attributes, defaults for communication devices, etc. Certain defaults can be overridden by entering the appropriate parameter in the device statements.

2.2.11.1 Coding Examples of Device Statements

The following examples show the coding of the OS/32 Configuration Utility Program (CUP) device statements on the left and the Sysgen/32 device statements on the right.

Examples:

```
OS/32 CUP
STATEMENTS

DEVICES
1:F0,0
2:B6,0
3 DSC1:C6,51,D

* DSC2:C7,50,D
ILEVEL 1
1:F1,0
2:0,0
3 MAG1:85,65
ILEVEL 2
1 CON:10,39,C
ILEVEL 3

1 PRT:62,114
1 VDU1:12,39
1 VDU2:14,39
ILEVEL 1
1:F2,0
2:0,0
3 MAG2,C5,65
ENDD
```

```
SYSGEN/32 STATEMENTS

DEVICES
DSC1:, C6, 51, SELCH=F0, CONTR=B6
DSC2:, C7, 50, SELCH=F0, CONTR=B6
MAG1:, 85, 65, SELCH=F1, CONTR=0
IL=1
CON:, 10, 39, CONSOLE, ILEVEL=2
ILEVEL 3
PRT:, 62, 114
VDU1:, 12, 39
VDU2:, 14, 39
ILEVEL 1
MAG2:, C5, 65, SELCH=F2, CONTR=1
*NOTE THAT THE CONTROLLERS FOR MAG1:
*and MAG2: ARE UNIQUE

ENDD
```

2.2.11.2 Configuring Data Communication Devices

The following sysgen configuration statements can be used to configure data communications devices in the system. The parameters are explained under the DEVICES...ENDD statements in Section 2.2.11.

- Asynchronous communication lines - device code 144

```
LINE:, 40, 144, XD=X0830, REA=XE1C9, WRI=XE809, PAD=2

LIN2:, 42, 144, XD=X0020
```

- Remote line printer and letter quality printer - device code 145

```
RPRT:, 90, 145, XD=X0830
```

- SIGMA 10 terminal - device code 146

```
SGMA:, 100, 146, XD=X0300, REC=80, REA=X8181, PAD=2
```

- Nonediting video display unit (VDU) - device code 147

```
CRT3:, 50, 147, XD=XFFFF, REC=80, REA=X8181, WRI=X0400,
     PAD=3

CRT4:, 52, 147
```

- Model 1200 and 6100 VDUs - device code 156

```
OWL1:, 60, 156, XD=X0830, REC=80, PAD=2

OWL2:, 62, 156
```

- Model 1250 point-to-point VDU - device code 157

```
SWL1:, 70, 157
```

- Model 1250 multidrop VDU and Series 7000 8-bit data transfer - device code 157

```
MULT:, 80, 158, XD=X0000, REC=80, PAD=2

MUL2:, 82, 158, XD=X0820
```

- Binary synchronous communications line on 201 DSA - device code 160

```
BQL1:, 120, 160, XD=X8000, REC=80, PAD=1, LEA=2,
     TR=BECB.TOP

BQL2:, 122, 160
```

- IBM 3780 RJE emulation on 201 DSA - device code 161

```
BQZ1:, 160, 161, REC=80, PAD=1, LEA=2

BQZ2:, 162, 161
```

- IBM 2780 RJE emulation on 201 DSA - device code 162

```
Q278:, 170, 162
```

- Binary synchronous processor-to-processor link on 201 DSA - device code 163

```
P2P1:, 200, 163, REC=80, PAD=1, LEA=2

P2P2:, 202, 163
```

- Binary synchronous communications line on quad-synchronous adapter (QSA) - device code 168

```
BQSL:, 130, 168, XD=X8000, REC=80, PAD=1, LEA=2,
     TR=BEBC.TOP

BQS2:, 132, 168
```

- IBM 3780 RJE emulation on QSA - device code 169

  SSA:, 170, 169, REC=80, PAD=1, LEA=2

  SSA2:, 172, 169

- IBM 2780 RJE emulation on QSA - device code 170

  Q378:, 190, 170

- Binary synchronous processor-to-processor link on QSA - device code 171

  PQP:, 210, 171

- 3270 binary synchronous support on QSA - device code 185

  EMT1:,BA,185,XDCOD=X1B00,PADCOUNT=3,LEADCOUNT=3

- DMA I/O subsystem (DIOS) - device code 192

  DIOS:, 100, 192

  DIOS is always configured with a device code of 192. The addresses of the devices under the DIOS must have an even address in increments of 2.

- Ethernet Data Link Controller (EDLC) - device code 212 and 213

  The general formats for the Ethernet device statements are:

  rdev name:,address,212,SELCH=m,NONSHARED,CONTROLLER=n

  wdev name:,address,213,SELCH=m,NONSHARED,CONTROLLER=n

  Where:

rdev name	refers to the read device name.
wdev name	refers to the write device name.
address	is identical for both device names.

m	refers to the SELCH address location.
n	refers to a logical controller number. The chosen number must not conflict with any physical controller device number or any other logical controller number, such as for magnetic tape.

A sample configuration is as follows:

```
ETHR:,68,212,SELCH=F0,NONSHARED,CONTROLLER=0

ETHW:,68,213,SELCH=F0,NONSHARED,CONTROLLER=0
```

Table 2-6 shows the sysgen default parameter values for each communication device.

TABLE 2-6 SYSGEN DEFAULT PARAMETER VALUES FOR DATA COMMUNICATION DEVICES

	DEVICE CODE	XDCOD	READ CONTROL	WRITE CONTROL	PAD COUNT	RECLEN	LEAD COUNT	POL LIMIT	SCREEN TIME	RESPONSE TIME	TRANSLATE
ASYNCHRONOUS DEVICES	144	X400	E1C9	E809	-	-	-	-	-	-	ASYN.XLT
	145	X400	-	X410	2	132	-	-	30	300	ASYN.XLT
	146	X300	X8181	-	3	80	-	-	30	300	ASYN.XLT
	147	X400	X8181	X400	3	80	-	-	30	300	ASYN.XLT
	156	-	-	-	2	80	-	-	25	X7FFF	ASYNCTOP
	157	-	-	-	2	80	-	-	25	X7FFF	ASYNCTOP
	158	-	-	-	2	80	-	-	20	3	ASYNCTOP
BINARY SYNCHRONOUS DEVICES	160	-	-	-	1	80	2	-	-	-	BEBC.TOP
	161	-	-	-	1	80	2	-	-	-	BEBC.TOP
	162	-	-	-	1	80	2	-	-	-	BEBC.TOP
	163	-	-	-	1	80	2	-	-	-	BEBC.TOP
	168	-	-	-	1	80	2	-	-	-	BEBC.TOP
	169	-	-	-	1	80	2	-	-	-	BEBC.TOP
	170	-	-	-	1	80	2	-	-	-	BEBC.TOP
	171	-	-	-	1	80	2	-	-	-	BEBC.TOP

2.2.11.3 Configuring Zero-Bit Insertion/Deletion (ZBID) Data Link Control (ZDLC) Devices

The following coding examples include ZDLC data communication devices in the system.

- 3270 binary synchronous support on QSA - device code 172

```
SPT1:,BA,172,XDCOD=X0400,IOL=10,POLLTIME=10,WAKEUP=10,
     PADCOUNT=3,LEADCOUNT=3
```

- ZBID on QSA; SVC15 access - device code 176

```
ZBD1:,B8,176,XDCOD=X0300
```

- ZBID on QSA; SVC15 access; simplex I/O - device code 177

```
ZBD2:,B8,177,XDCOD=X0100 (simplex read)

ZBD3:,B9,177,XDCOD=X0200 (simplex write)
```

- ZBID channel terminal manager (CTM) on QSA; SVC1 access; simplex write - device code 178

```
ZB4R:,BA,178,XDCOD=X0100,POLLIMIT=5,IOL=5,MINREAD=264,
     LINESTATUS=X00F9,USCI=XD0884024
```

- ZBID CTM on QSA; SVC1 access; simplex read; daughter to device code 178 - device code 179

```
ZB4W:,BB,179,XDCOD=X0200
```

- ZBID CTM on QSA; SVC1 access; - device code 180

```
ZBID:,B8,180,XDCOD=X0300,POLLIMIT=5,IOL=5,MINREAD=264,
     LINESTATUS=X0099,UCSI=XD0884020
```

- ZBID CTM on QSA; SVC1 access; simplex write - device code 181

```
ZB:,B8,181,XDCOD=X0100,POLLIMIT=5,IOL=5,MINREAD=264,
   LINESTATUS=X00F9,UCSI=XD0884020
```

- ZBID CTM on QSA; SVC1 access; simplex read; daughter to device code 181 - device code 182

```
ZB:,B9,182,XDCOD=X0200
```

- ZBID CTM on QSA; SVC1 access; half-duplex with assembled DCT - device code 183

```
ZB:,BA,183,,XDCOD=X0300,POLLIMIT=5,MINREAD=264,
   LINESTATUS=X0099,UCSI=XD0884020,UCSO=X18900030,
   NCS=XF8F8
```

2.2.12 DIRECTORY Statement

The DIRECTORY statement specifies that secondary directory support is included in the system.

Format:

```
DIRECTORY
```

Functional Details:

If this statement is omitted, no secondary directory support is included. If a disk is marked on with the CDIRECTORY parameter specified in the operator MARK command, and the system is built with directory support, file search time is reduced. Marked on disks require the following additional working storage areas:

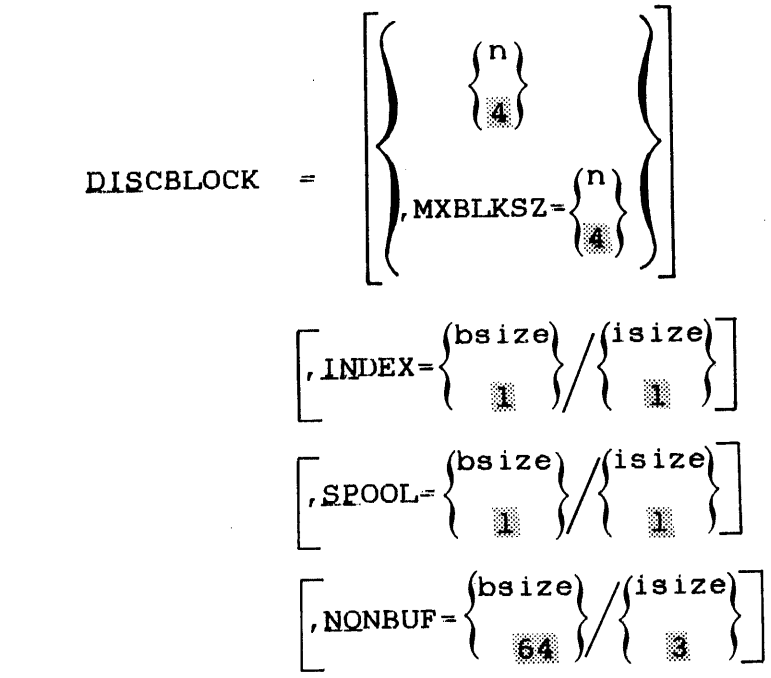
- system space of 64 bytes for access control blocks (ACBs), and
- a secondary directory buffer with a default buffer size equal to 1,024 bytes. The default value can be overridden when the disk is marked on.

See the OS/32 Application Level Programmer Reference Manual for detailed information.

2.2.13 DISCBLOCK Statement

The DISCBLOC statement specifies the maximum physical block size for data and index blocks for indexed files and nonbuffered files. The parameters are interactively independent. If any are omitted, the default values are used.

Format:



Parameters:

n	is a decimal number from 1 to 255 indicating the maximum number of 256-byte segments that can be specified for data or index blocks in an ALLOCATE command or an SVC7. If this parameter is omitted, 4 is the default. This parameter is invalid if MXBLKSZ is specified because it is another way of specifying MXBLKSZ.
MXBLKSZ=	is a decimal number from 1 to 255 indicating the maximum number of 256-byte segments that can be specified for data or index blocks in an ALLOCATE command or an SVC7. If this parameter is omitted, 4 is the default. This parameter is invalid if the first positional parameter, n, is specified.

INDEX= specifies the default data and indexed blocks for INDEX FILES, where bsize is the data block size, and isize is the index block size. These values are used as defaults in an ALLOCATE command and an SVC7. If bsize is omitted, the default is 1. If isize is omitted, the default is 1. Both bsize and isize are decimal numbers that range from 1 to MXBLKSZ.

SPOOL= specifies the data and indexed blocks for SPOOL FILES, where bsize is the data block size, and isize is the index block size. These values are used as defaults in an ALLOCATE command and an SVC7. If bsize is omitted, the default is 1. If isize is omitted, the default is 1. Both bsize and isize are decimal numbers that range from 1 to MXBLKSZ.

NONBUF= specifies the data and indexed blocks for NONBUFFERED FILES, where bsize is the data block size, and isize is the index block size. These values are used as defaults in an ALLOCATE command and an SVC7. If bsize is omitted, the default is 64. If isize is omitted, the default is 3. bsize ranges from 1 to 65,535 and isize ranges from 1 to MZBLKSZ.

Functional Details:

The values specified at sysgen will serve as system defaults for all other commands that allow the user to specify index or data block size. If index or data block size is specified via the BLOCK command in SPL/32 or the operator command SET BLOCKS, these block sizes override the default block sizes established by DISCBLOCK at sysgen.

2.2.14 DSYS Statement

The DSYS statement specifies the default number of kilobytes of available dynamic system space. The size of system space can be adjusted by the SET SYS operator command after the system is built. The following dynamic control blocks are allocated in system space:

- Private file control block (PFCB)
- File control block (FCB)
- Task control block (TCB)
- Task queue element (TQE)
- Access control block (ACB)
- Segment description element (SDE)
- Private segment table (PST)

Format:

DSYS [=] [{ n / 25 }]

Parameter:

n	is a decimal number from 1 to the total number of kilobytes of memory. If n is omitted, 25kb is the default.

Functional Details:

If this statement is omitted, 25kb is the default.

2.2.15 ENDC Statement

The ENDC statement must be the last sysgen statement specified; it indicates the end of all sysgen configuration statements.

Format:

```
ENDC
```

| ERRORREC |

2.2.16 ERRORREC Statement

The ERRORREC statement specifies that error recording support is included in the system.

Format:

ERRORREC [=] fd,size,period

Parameters:

fd	is the file descriptor of the default error recording file. The specified file is assigned by the system as the default error recording file.
size	is a decimal number from 1 to 32,767 specifying the maximum number of 256-byte records in the default error recording file.
period	is a decimal number from 1 to 1,440 specifying the minutes that elapse between memory error recording readouts on the Model 3205, 3200MPS, 3210, 3220, 3230, 3240 and 3250 processors. The recommended period is two minutes.

Functional Details:

Error recording supports recording of:

- I/O errors for all processors
- System errors for all processors
- Memory errors for Perkin-Elmer Series 3200 processors

No significant increase in overhead cost is incurred by recording I/O and system errors.

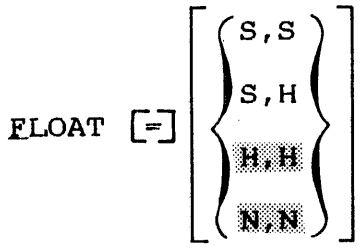
The Perkin-Elmer Series 3200 processors contain error recording memory and an optional error logger. If this statement is specified, the hardware error logger is periodically read and the data is written to the error recording file for subsequent reporting. If this statement is omitted, error recording support is not included in the system.

If this statement is specified when configuring a Perkin-Elmer Series 3200 processor in the system, the MCONFIG statement must also be included.

2.2.17 FLOAT Statement

The FLOAT statement specifies that floating point support (software or hardware) is included in the system.

Format:



Parameters:

S	indicates software floating point is supported for single precision floating point (SPFP) in the first parameter and for double precision floating point (DPFP) in the second parameter.
H	indicates hardware floating point is supported for SPFP in the first parameter and for DPFP in the second parameter.
N	indicates no floating point is supported.

Functional Details:

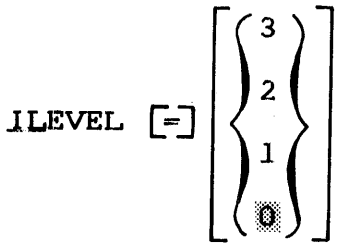
Software floating point support should be included only in systems that do not support hardware floating point. If the hardware floating point parameter is not specified for a system with hardware floating point, memory is wasted and unpredictable results occur after a power fail/restore sequence. SPFP support occupies 1.9kb of memory, and DPFP support occupies 2.9kb of memory.

If the FLOAT statement is omitted, H,H is the default for the Perkin-Elmer Series 3200 processors, and N,N is the default for the Model 7/32 or 8/32 processors.

2.2.18 ILEVEL Statement

The ILEVEL statement specifies the hardware interrupt levels for all devices that are specified between this ILEVEL statement and another ILEVEL statement or the ENDD statement.

Format:



Parameters:

3	is a decimal number indicating the fourth and lowest interrupt level at which a device can interrupt. If this parameter is omitted, 0 is the default.
2	is a decimal number indicating the third interrupt level at which a device can interrupt. If this parameter is omitted, 0 is the default.
1	is a decimal number indicating the second interrupt level at which a device can interrupt. If this parameter is omitted, 0 is the default.
0	is a decimal number indicating the first and highest interrupt level at which a device can interrupt. If this parameter is omitted, 0 is the default.

Functional Details:

If multiple I/O interrupt levels are not to be included in the system, omission of this statement causes all devices to be configured at the highest interrupt level, 0. A group of devices configured with the same SELCH or disk controller address must be configured at the same interrupt level. Therefore, the ILEVEL statement should directly precede the group of devices to be configured at a specific interrupt level.

The ILEVEL parameter in the device statement is used to redefine the interrupt level for a particular device without affecting the global interrupt level setting.

2.2.19 INTERCEPT Statement

The INTERCEPT statement indicates that SVC interception support is to be included in the system.

Format:

```
INTERCEPT
```

Functional Details:

If this statement is omitted, SVC interception is excluded from the operating system.

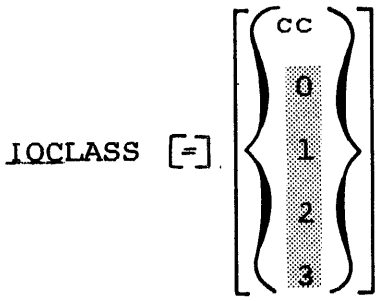
This is a required parameter for SPL/32 Spooling support and Reliance R05 or higher.

| IOCLASS |

2.2.20 IOCLASS Statement

The IOCLASS statement specifies the global class setting to be associated with a particular device or group of devices used for accounting.

Format:



Parameter:

cc	is a decimal number from 0 to 31 specifying the class associated with devices or files. If this statement or the parameters are omitted, 0, 1, 2 and 3 are the default classes.

Functional Details:

A global IOCLASS can be specified or changed by entering the IOCLASS statement immediately preceding the device or group of devices to be associated with that class. All devices are associated with that IOCLASS statement until the next IOCLASS statement is entered.

The IOCLASS parameter in the device statement redefines the I/O class of a particular device and does not affect the global I/O class setting.

User-specified I/O classes must be within the range specified by the ACCOUNTING statement (see Section 2.2.1). Table 2-7 lists the default device and file classes.

TABLE 2-7 DEFAULT DEVICE AND FILE CLASSES

CLASS	DEFAULT
0	Logical I/O (indexed files)*
1	Physical I/O with SELCH* (contiguous files)
2	Physical I/O with multiplexor channel (VDUs)
3	Logical spooled I/O (spooled output)

* The Accounting Facility requires classes 0 and 1.

IREADER

2.2.21 IREADER Statement

The IREADER statement specifies that internal reader support is to be configured in the system. This statement allows a task loaded at the system console to send input messages to the command processor.

Format:

```
IREADER
```

Functional Details:

If this statement is omitted, internal reader support cannot be configured in the system.

This statement must be used in order to utilize the operator IRBUFF command and SVC2 code 14.

2.2.22 ITAM Statement

The ITAM statement indicates that communications support is to be included in the system.

Format:

ITAM

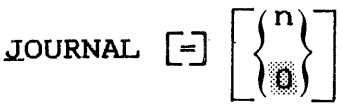
Functional Details:

Communications support consists of system modules, drivers and DCBs, CCBs, etc. The drivers are stored in either the communications driver library or extended communications driver library. The system modules are stored in the system communications library (see Chapter 3).

2.2.23 JOURNAL Statement

The JOURNAL statement specifies the maximum number of journal entries for the system. The system journal is a list of data entries that records operating system events and is used for tracing the cause of a system failure.

Format:



Parameter:

n	is a decimal number from 0 to 12,999 specifying the maximum number of journal entries. The number 0 is the default.

Functional Details:

The amount of memory required for the user-specified number of journal entries is calculated as:

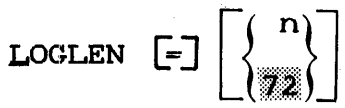
number of bytes for journal = 20(n + 8)

A source assembly of the operating system must be done with SGN.JRNL equal to one to include journal code within the operating system.

2.2.24 LOGLEN Statement

The LOGLEN statement specifies the maximum number of bytes of message buffer size in the system.

Format:



Parameter:

n	is a decimal number from 32 to 132 specifying the maximum number of bytes of message buffer size. If this parameter is omitted, 72 is the default.

Functional Details:

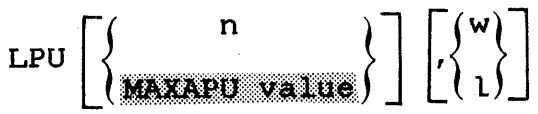
This statement sets the message buffer size for user tasks (u-tasks) executing SVC2 code 7 log message calls. If the length of the user-specified message is greater than the message buffer size, the right-most bytes of the message are truncated.

When a u-task running under MTM issues an SVC2 code 7 to the system console, the user buffer is truncated to LOGLEN. If the SVC2 code 7 is directed to the user terminal, LOGLEN has no effect. If this statement is omitted, 72 is the default.

2.2.25 LPU Statement

The LPU statement is used to specify that one or more logical processing units (LPUs) are to be configured in the system. An LPU is used to assign tasks to processors. This statement is valid for a Model 3200MPS system only.

Format:



Parameters:

n	is a decimal number that specifies the maximum number of LPUs that can be configured in the system. If this parameter is omitted, the default is the value specified in the MAXAPU statement (see Section 2.2.26). This ensures a one-to-one LPU-to-auxiliary processing unit (APU) mapping capability.
w	specifies the size of the LPU/APU mapping table. This parameter must be 1.

Functional Details:

An LPU is always mapped to an APU task execution queue. Each LPU can be mapped to only one APU queue. However, several LPUs can be mapped to the same APU queue.

Each task is assigned an LPU at link time. The LPU assignment can be changed by the task itself, by another task or by the system operator. The LPU mapping can be disabled or enabled by the task itself or by another task. The default is: mapping disabled for LPU = 0 and enabled for the rest of the LPUs.

2.2.26 MAXAPU Statement

The MAXAPU statement is used to specify the maximum number of APUs to be configured in the system and map these APUs to their APU controllers. This configuration statement is valid only for a Model 3200MPS System.

Format:

```
MAXAPU [{n}]
       [{9}]

[APU1 ,contr1 ]
[APU2 ,contr2 ]
[     .       ]
[     .       ]
[     .       ]
[APUn ,contrn ]
ENDAPU
```

Parameters:

n	is a decimal number specifying the maximum number of APUs to be configured in the system. If this parameter is omitted, the default is 9.
$APU_1 \ldots APU_n$	is the identity of the APU to be configured in the system. The only allowable values for this parameter are 1 through 9.
$contr_n$	is a hexadecimal address specifying the number of the APU controller (real-time support module (RTSM) digital I/O (DIO)) for APU n. This parameter is called an APU controller statement.

Functional Details:

One APU controller statement must be entered for each APU to be configured in the system.

Example:

```
MAXAPU

 APU1,0A
 APU2,50
 APU3,52
   .
   .
   .
 APU9,5E
ENDAPU
```

The hexadecimal numbers in this example of the standard controller assignment are device addresses.

2.2.27 MAXTASK Statement

The MAXTASK statement specifies the maximum number of tasks (including rolled-out tasks) that can be in the system at one time.

Format:

Parameter:

n	is a decimal number from 1 to 252 specifying the maximum number of tasks in the system at one time. If this parameter is omitted, 32 is the default.

Functional Details:

If this statement is omitted, 32 is the default.

2.2.28 MCONFIG Statement

The MCONFIG statement must be used if memory error recording is desired for a Perkin-Elmer Series 3200 processor.

Format:

```
MCONFIG  BLOCK=nn,START=xx,RANGE=yy  [,INTERL={2|4|0}]
         [,SHARED={NORECORD|NVRECORD|RECORD}]
```

Parameters:

BLOCK= nn is a decimal number from 0 to 15 specifying the area of memory in megabytes for which a particular memory controller is strapped. Blocks have to be specified in numerically ascending order starting with 0 for the lowest megabyte address. For Perkin-Elmer Series 3200 processors, 1Mb is the minimum specified amount of memory.

START= xx is a decimal number from 0 to 15 specifying the starting megabyte address of the block. Fractional megabytes should be rounded up to the next full megabyte.

RANGE= yy is a decimal number from 0 to 16 specifying the number of megabytes in the block. For processors containing less than 1Mb, specify 1.

INTERL= 2 specifies 2-way interleaving.
4 specifies 4-way interleaving.
0 specifies noninterleaving of memory.

This parameter is optional and if omitted, 0 is the default.

SHARED= defines the memory block as part of shared memory. This parameter is optional.

NORECORD — inhibits the processor from reading the error logger for the designated block. The block configuration is not verified.

NVRECORD — specifies that the processor should read the error logger for the designated block. The block configuration is not verified. This facilitates bringing an error logger readout on-line in a multiprocessor configuration without destroying valid data in shared memory. If this parameter is omitted, the designated block is assumed to be in local memory.

RECORD — specifies that the processor should read the error logger for the designated block. The block configuration is verified at system start-up time.

Functional Details:

As part of the system configuration process, the physical memory configuration must be defined if memory error recording is included in the system.

In multiprocessor systems with shared memory, only one processor should be designated to read the shared memory error logger. This prevents the scattering of error logger recordings.

One MCONFIG statement is required for each group of memory controllers that is strapped for a different address range. Typically, there is only one group of controllers unless the configuration contains shared memory.

Examples:

For a Model 3250 processor with 4Mb 2-way interleaved local memory:

```
MCONFIG BLOCK=0,START=0,RANGE=4,INTERL=2
```

For a Model 3220 processor with 3/4Mb local memory:

```
MCONFIG BLOCK=0,START=0,RANGE=1
```

For a Model 3240 processor with 3Mb 4-way interleaved local memory, 1.5Mb 2-way interleaved shared memory starting at Y'500000', and 2Mb noninterleaved shared memory starting at Y'800000':

```
MCONFIG BLOCK=0,START=0,RANGE=3,INTERL=4
MCONFIG BLOCK=1,START=5,RANGE=2,INTERL=2,SHARED=RECORD
MCONFIG BLOCK=2,START=8,RANGE=2,SHARED=NORECORD
```

For a Model 3230 processor with 3.5Mb local memory and 1.5Mb shared memory starting at Y'400000':

```
MCONFIG BLOCK=0,START=0,RANGE=4
MCONFIG BLOCK=1,START=4,RANGE=2,SHARED=NVRECORD
```

For a Model 3205 processor with 2Mb local memory:

```
MCONFIG BLOCK=0,START=0,RANGE=3
```

2.2.29 MEMCHECK Statement

The MEMCHECK statement indicates that memory diagnostics support is included in the system.

Format:

```
MEMCHECK
```

Functional Details:

The memory diagnostics program is executed at initial program load (IPL) time. If any bad or unavailable pages exist in memory (256-byte pages for Model 7/32, 8/32 and 3220 processors; 2,048-byte pages for Model 3210, 3230, 3240, 3250 and 3200MPS processors; 4,096-byte pages for the Model 3205 processor), the operating system marks them as unavailable. Memory can also be tested, marked off and marked on by the operator MEMORY command if memory diagnostics support is included.

2.2.30 MEMORY Statement

The MEMORY statement specifies the maximum number of kilobytes of available local memory for the system. Local memory is a contiguous memory area starting at absolute address 0 and consists of the:

- operating system,
- dynamic system space,
- reentrant library segments (optional),
- task common segments (optional), and
- pure and impure segments.

Format:

MEMORY [=] [{ n / 256 }]

Parameter:

n	is a decimal number specifying the maximum number of kilobytes of available local memory. The number is in increments of 16 and ranges from 256kb to 1,024kb for the Model 7/32, 8/32 and 3220 processors. The maximum number of kilobytes of available memory for the Model 3205 and 3210 processors is 4,096. The maximum number of kilobytes available for the Model 3230, 3240, 3250 and 3200MPS processors is 16,384kb. If this parameter is omitted, 256 is the default.

Functional Details:

If the operating system memory size exceeds the memory size specified by n, the error is not detected until a load module is created by OS/32 Link. The size of local memory can be changed by the operator MEMORY command. If this statement is omitted, the default is 256.

2.2.30A MIRROR Statement

The MIRROR statement specifies that mirror disk support is included in the system.

Format:

MIRROR

Functional Details:

If this statement is omitted, mirror disk support is not included in the system. If mirror disk support is not included and an attempt is made to mark on disks with the MIRROR= option of the operator MARK ON command, an error will result.

Mirror disk support occupies approximately 12kb of memory.

2.2.31 MODULE...ENDM Statements

The MODULE...ENDM statements substitute a user-written or user-modified system module for a Perkin-Elmer standard system module.

Format:

```
MODULE
[new module name1]
[new module name2]
        .
        .
        .
[new module namen]
ENDM
```

Parameter:

new module name	is a 4-character name, a period and a 3-character variation (ffff.xxx) indicating the user-written or user-modified module to be selected. If the name (ffff) is the same as the standard module name, the variation (xxx) overrides the variation normally selected at sysgen. However, if the name is not the same as a standard module name or USER.xxx, an error is generated.

| NOSEG |

2.2.32 NOSEG Statement

The NOSEG statement specifies that memory segmentation support is excluded from the system.

Format:

NOSEG

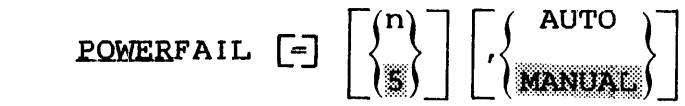
Functional Details:

When segmentation support is excluded, tasks that were previously established in pure and impure segments cannot be loaded into the system. Exclusion of segmentation support causes inefficient use of memory through the loss of shared pure segments. If this statement is omitted, segmentation support is included.

2.2.33 POWERFAIL Statement

The POWERFAIL statement is used to specify the action to be taken after a power failure occurs.

Format:



Parameters:

n	is a decimal value from 0 to 65,535 specifying the number of seconds to wait after a power failure has occurred. If n is omitted, the default is five seconds. A value of zero indicates no wait is to occur.
AUTO	is specified so that the system will automatically restart without operator intervention. If this parameter is omitted, MANUAL is the default.
MANUAL	is specified so that the system will issue the following message from the system console:

POWER RESTORE-RESET PERIPHERALS AND ENTER GO

2.2.34 QUEUE Statement

The QUEUE statement defines the maximum number of entries in the system queue used to schedule driver operation.

Format:

Parameter:

n	is a decimal number from 1 to 64,999 specifying the maximum number of entries in the system queue. If this parameter is omitted, the total number of devices in the system is the default.

Functional Details:

The minimum number of entries should be equal to the total number of devices (including nodes, channels and controllers), because driver termination routines do not check for sufficient room on the system queue when adding entries.

If this statement is omitted, the total number of devices in the system is the default.

2.2.35 ROLL Statement

The ROLL statement specifies that roll support is included in the system.

Format:

```
ROLL [=] [rvoln]
```

Parameter:

rvoln	is a 1- to 4-character volume name specifying the default roll volume. The first character of the volume name must be alphabetic, the remaining alphanumeric. If this parameter is omitted, blanks are generated as the volume name.

Functional Details:

When roll support is specified, at least one direct access device must be included. If this statement is omitted, roll support is excluded from the system.

2.2.36 SLICE Statement

The SLICE command is used to invoke the time-slice scheduling option.

Format:

SLICE [=] [n]

Parameter:

n	is zero or a decimal number from 20 to 65,535 specifying the time-slice, in milliseconds, allocated to each task. If n is not specified, time-slice scheduling is disabled.

Functional Details:

If n is zero, time-slice scheduling is disabled: otherwise, n represents the maximum time in milliseconds that any one task can remain active if another task of equal priority is ready. The time-slice option is initially disabled.

The SLICE command provides a SET SLICE capability at system initialization. This allows time-slice scheduling to be activated automatically by the system. The SET SLICE command can be used to override the SLICE statement. See the OS/32 Operator Reference Manual for the SET SLICE command.

2.2.37 SPOOL Statement

The SPOOL statement specifies that OS/32 Spooler support is included in the system.

Format:

SPOOL [=] [spvoln]

Parameter:

spvoln	is a 1- to 4-character volume name specifying the spool volume. The first character of the volume name must be alphabetic, the remaining alphanumeric. If this parameter is omitted, blanks are generated as the volume name.

Functional Detail:

When spool support is specified, at least one direct access device must be included. If this statement is omitted, spool support is excluded from the system. The statement cannot be used if the SPL32 statement is used.

```
----------------
|     SPL32     |
----------------
```

2.2.38 SPL32 Statement

The SPL32 statement specifies that SPL/32 support is to be included in the target system.

Format:

SPL32

Functional Details:

This statement and the SPOOL statement explained in Section 2.3.37 are mutually exclusive statements. If this statement is entered, the user cannot specify any spooler pseudo devices in the DEVICES...ENDD statements.

When this statement is entered, INTERCEPT is automatically included in the system.

2.2.39 SSTABLE Statement

The SSTABLE statement specifies the maximum number of shared segment table (SST) entries to be reserved in the system.

Format:

Parameter:

n	is a decimal number from 1 to 8,192 specifying the maximum number of SST entries allowed in the system. If this parameter is omitted, the default is 32.

Functional Details:

The Model 3205, 3210, 3230, 3250, 3200MPS and 3240 processors use an SST. Each SST entry requires eight bytes. Space for the table is allocated in 256-byte blocks. If this statement is omitted, 32 is the default.

| STARTUP...ENDS |

2.2.40 STARTUP...ENDS Statements

The STARTUP...ENDS statements define a start-up CSS procedure, executed at system start-up. This feature allows FASTCHEK to run automatically and loads the necessary tasks to create the system environment.

Format:

```
[STARTUP
.
.
.
ENDS   ]
```

Functional Details:

The CSS procedure is executed before the SET TIME request is issued by the system. If a SET TIME request is issued from the start-up CSS, the operating system ENTER DATE AND TIME request is not made.

The CSS commands are stored in memory in packed format. This memory is not reused during system operation. A large number of start-up CSS routines could affect the amount of memory available for task execution, so start-up requests should be as brief as possible, making use of such commands as $TRANSFER.

The BUILD...ENDB and $BUILD...$ENDB commands are not allowed in a start-up CSS procedure, but are allowed in a CSS called from a start-up CSS procedure.

Example:

```
STARTUP
SE T  1/02/84,07:59:59
$J
MA DSC1:,ON,,CD=200
$T
$IFNE 0
   $JOB
   MA DSC1:,ON,P
   L .BG,FASTCHEK,80
   T .BG
   MA DSC1:,OFF
   ST,CH=DSC1:,LI:CON
   MA DSC1:,ON,,CD=ALL
   $TERMJOB
   $IFNE 0
      $WR *** MARK OR CHECK ERROR ON DSC1: ***
      $EX
   $EN
$EN
$TR STARTUP
ENDS
```

| TCOM |

2.2.41 TCOM Statement

The TCOM statement defines and reserves system storage for TCOM segments in global memory. Global memory (shared memory) is located outside of local memory. Local memory is defined by the operator MEMORY command.

Format:

TCOM [=] $name_1$,$address_1$,$size_1$ [/.../$name_n$,$address_n$,$size_n$]

Parameters:

name	is a TCOM segment name from 1- to 8-characters that corresponds to a labeled common segment name in a user program. The first character of the segment name must be alphabetic, the remaining alphanumeric.
address	is a hexadecimal number within the ranges specified in Table 2-8 specifying the absolute address of a TCOM segment located outside local memory. The user-specified number is rounded down to the nearest 256-byte page address.
size	is a decimal number in increments of .25kb from .25kb to the maximum amount of global memory.

Functional Details:

Global memory is located above MTOP, the area specified in the operator MEMORY command, and is limited by the physical memory of the system. If the address specified is greater than the physical memory of the machine, the error is not displayed by Sysgen/32. All TCOM segments must be specified in order of ascending physical addresses; a maximum of 14 TCOM segments can be defined. Overlapping TCOM segments are not allowed.

TABLE 2-8 TCOM ADDRESS RANGE

CPU MODEL	ADDRESS RANGE	SEGMENT SIZE (KB) INCREMENT
7/32	1-FFF00	.25
8/32	1-FFF00	.25
3205	1-3FF000	4.0
3210	1-3FF800	2.0
3220	1-FFF00	.25
3230	1-FFF800	2.0
3240	1-FFF800	2.0
3250	1-FFF800	2.0
3200MPS	1-FFF800	2.0

2.2.42 TEMP Statement

The TEMP statement specifies the default volume name to be used for temporary (scratch) file support.

Format:

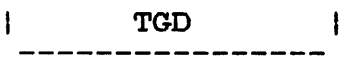
```
TEMP [=] [tvoln]
```

Parameter:

tvoln	is a 1- to 4-character volume name specifying the default temporary volume. The first character of the volume name must be alphabetic, the remaining alphanumeric. If this parameter is omitted, blanks are generated as the volume name.

Functional Details:

Temporary file support is always included in the system. If this statement is omitted, the default temporary volume name is set to blanks. Temporary files are allocated at assign time and deleted at close time. To exclude temporary file support, a source sysgen must be performed.



2.2.43 TGD Statement

The TGD statement specifies that TGD device support is included in the system.

Format:

```
TGD
```

Functional Details:

If this statement is omitted, TGD support is excluded. See the OS/32 Application Level Programmer Reference Manual.

2.2.44 VERSION Statement

The VERSION statement specifies a user version that is associated with a particular operating system sysgen.

Format:

```
VERSION [=] [vvvvvvvv]
```

Parameter:

vvvvvvvv	is an 8-character alphanumeric string specifying the version of a particular operating system sysgen. If this parameter is omitted, blanks are generated as the version number.

Functional Details:

The 8-character version, in addition to the operating system revision and update number, is displayed on the system console at system initialization time in this format:

```
OS32MTrr-uu.vvvvvvvv
```

If this statement is omitted, blanks are generated as the version.

2.2.45 VOLUME Statement

The VOLUME statement specifies the name of the default system volume.

Format:

VOLUME [=] [voln]

Parameter:

voln	is a 1- to 4-character volume name specifying the default system volume. The first character of the volume name must be alphabetic, the remaining alphanumeric. If this parameter is omitted, blanks are generated as the volume name.

Functional Details:

After sysgen, the operator can change the volume name through the operator VOLUME command.

If this statement is omitted, blanks are generated as the volume name.

CHAPTER 3
LIBRARIES REQUIRED FOR SYSTEM GENERATION (SYSGEN)

3.1 INTRODUCTION

The Sysgen/32 process requires standard system libraries, general-purpose driver libraries to support the devices listed in Chapter 4, and system and Sysgen/32 macro libraries. These libraries are required for the macro expansion and link phases of the sysgen procedure. Other system libraries that may be required by the Sysgen/32 process are system communications libraries if communication devices are configured in the operating system, and user libraries if user-written devices are configured in the operating system.

OS/32 supports two types of sysgen:

- Object-level
- Source-level

The object-level sysgen enables a user to configure an operating system tailored to specific needs by selecting driver and system modules provided in the OS/32 package. Assemblies of system modules are not required in performing an object-level sysgen.

The source-level sysgen enables a user to modify the OS/32 system modules and drivers. This procedure requires reassembling one or more systems and/or driver source modules, replacing existing versions of these modules in the system or driver libraries with the user-modified modules, and executing the object-level sysgen procedure. Section 3.8 presents sample source sysgen procedures.

3.2 STANDARD LIBRARIES (OPERATING SYSTEM)

The standard libraries required for the Sysgen/32 process are:

- The standard system library (SYS.LIB)
- The standard driver library (DRIVER.LIB)
- The end of system module (UBOT.OBJ)

These libraries are used in the link phase of the Sysgen/32 process and are available in both object and source versions.

3.2.1 Standard System Library (SYS.LIB)

Table 3-1 presents a description of the source sysgen options that can be included in a target system.

The source sysgen modules of SYS.LIB are listed in Table 3-2. These modules can be altered by the sysgen parameters in the sysgen parameter file. Table 3-3 is a list of the source sysgen parameters and the amount of memory the desired sysgen option occupies.

Each module has an extension number that corresponds to functional variations that support various sysgen options. Table 3-4 is a list of module variations in SYS.LIB. The standard variations provided in SYS.LIB are a subset of possible variations. A source-level sysgen can be optionally performed to include or eliminate options in a specific module. A source-level sysgen can also be optionally performed to eliminate specific source-level sysgen options included in all module variations.

TABLE 3-1 SOURCE SYSGEN OPTION DEFINITIONS

OPTION	MNEMONIC	DEFINITION
Journal support	JRNL	Supports recording of normal internal system events on a circular list in memory. Used as a debugging tool for operating system development and is recommended for installations where user-written or user-modified modules are tested.
Safety check support	SAFE	Supports consistency checking within the operating system. Detected inconsistencies result in a system failure.
System debug software support	DBUG	Supports debugging of the operating system. Includes consistency checks not intended for operational systems.
Bulk command module support	BCMD	Supports bulk storage operator commands such as WRITE FILEMARK, FORWARD FILEMARK, FORWARD RECORD, BACKSPACE FILEMARK, BACKSPACE RECORD and REWIND.
Contiguous file support	CO	Supports contiguous files on direct access devices.

TABLE 3-1 SOURCE SYSGEN OPTION DEFINITIONS (Continued)

OPTION	MNEMONIC	DEFINITION
Indexed and nonbuffered indexed file support	INX	Supports indexed and nonbuffered indexed files on direct access devices.
Direct access device support	DA	Supports direct access bulk storage devices (disks) beyond the basic input/output (I/O) device level, including primary directory support, sector allocation, bit map support and volume mechanism support. Includes the DISPLAY FILES, DELETE, XDELETE, VOLUME and MARK operator commands.
Secondary directory support	DIR	Supports the secondary directory feature that provides an in-memory, paged index to the disk directory to reduce access time for directory operations. Requires direct access support.
Mirror disk support	MIRR	Supports the mirrored disk feature, which provides a duplicate write to a secondary disk drive. Requires direct access support.
Spooling support	SPOL	Supports OS/32 Spooler. Requires direct access support.
SPL/32 support	SP32	Supports SPL/32 Spooler. Requires direct access support.
Roll support	ROLL	Supports rolling of tasks to disks to execute more tasks than can fit into available task memory. Requires direct access support.
Temporary file support	TEMP	Supports temporary files. Requires direct access support.
Shared segmentation support	SEG	Supports use of sharable segments such as pure (reentrant) segments, run-time libraries (RTLs) and task commons.

TABLE 3-1 SOURCE SYSGEN OPTION DEFINITIONS (Continued)

OPTION	MNEMONIC	DEFINITION
Link overlay support	AOVL	Supports loading and executing overlaid tasks produced by OS/32 Link. Requires direct access support.
Communications support	ITAM	Supports communication devices.
Trap-generating device support	TGD	Supports the 8-line interrupt module, providing task traps in response to external interrupts.
Double precision floating point (DPFP) software support	DF	Supports DPFP in a system without the hardware to support it. Includes software emulation routines for double precision instructions.
Single precision floating point (SPFP) software support	SF	Supports SPFP in a system without the hardware to support it. Includes software emulation routines for single precision instructions.
Seek support	SEEK	Supports C-scan seek scheduling on disk devices. Selection of a scheduling algorithm is specified in each disk device control block (DCB). Standard driver library has seek scheduling enabled for all disks except floppy. This option is required to support those disks. Requires direct access support.
Extended direct memory access (EDMA) support	EDMA	Supports coordination of multiple EDMA devices. Required if the EDMA devices exceed the bandwidth capability of the EDMA bus. This coordination mechanism is also used for floppy disk systems where the adapter bandwidth is limited. Requires direct access support.
Memory diagnostics support	MCHK	Supports memory testing features for initial system load and at the request of particular operator commands. Finds and marks off bad memory.

TABLE 3-1 SOURCE SYSGEN OPTION DEFINITIONS (Continued)

OPTION	MNEMONIC	DEFINITION
Memory error recording support	MERC	Supports recording of memory errors generated by the system. Requires that general error recording is selected. Available for the Perkin-Elmer Series 3200 processors.
General error recording	GERC	Supports error recording of memory, I/O, system errors and system milestones.
Model 3210, 3220, 3230, 3240 and 3250 support	3200	Supports the Perkin-Elmer Series 3200 processors.
Memory address translator (MAT) Model 3200MPS, 3205, 3210, 3230, 3240 and 3250 support	ATM	Supports MAT on Model 3210, 3230, 3240, 3200MPS, 3205 and 3250 processors. Must be removed for Model 7/32, 8/32 and 3220 processors.
Accounting facility support	ACCT	Supports job accounting of system resources. Requires direct access support.
File accounting	ACCF	Generates accounting records for file renames and deletes. Requires accounting support. Requires direct access support.
Internal reader	IRDR	Supports sending input messages to the command processor by console-loaded tasks.
Auxiliary processors	APU	Supports the Perkin-Elmer Model 3200MPS System.

TABLE 3-2 SYSTEM SOURCE MODULES AND SUPPORTED FEATURES

SOURCE MODULES		SUPPORTED FEATURES																																		
FUNCTION	NAME	JRNL	SAFE	DBUG	BCMD	CO	INX	DA	DIR	MIRR	SP32	SPOL	ROLL	TEMP	SEG	AVOL	ITAM	TGD	DF	SF	SEEK	EDMA	CMHK	MERC	GERC	320	ATM	ACCT	ACCF	IRDR	APU
Auxiliary Processing Unit (APU) support	APSV.MAC	x	x	x																																x
Console Driver	CDVR.MAC																																			
Display and Bulk Command	CMDB.MAC			x	x			x	x	x	x	x	x	x			x																			
Command Executors	CMEX.MAC									x	x	x					x						x													
Internal Reader	CMIR.MAC																																		x	
Console Monitor	CMON.MAC																																			
Console and CSS Super-visor	CMSP.MAC	x								x													x													
Error Recording	ERRC.MAC																					x		x	x											
Account-ing Facility	EXAC.MAC																												x	x						
Interrupt Handler	EXIN.MAC	x		x												x	x						x	x		x	x								
I/O	EXIO.MAC	x	x							x							x				x	x						x								
Loader and Segment Control	EXLD.MAC							x					x		x	x						x	x				x									
Memory Manager	EXMY.MAC		x					x					x				x									x	x								
Super-visor Services	EXSP.MAC	x										x															x									
SVC3,4, 5,6,7,14	EXSV.MAC							x					x		x	x		x																		

TABLE 3-2 SYSTEM SOURCE MODULES AND SUPPORTED FEATURES (Continued)

SOURCE MODULES		SUPPORTED FEATURES																																
FUNCTION	NAME	JRNL	SAFE	DBUG	BCMD	CO	INX	DA	DIR	MIRR	SP32	SPOL	ROLL	TEMP	SEG	AOVL	ITAM	TGD	DF	SF	SEEK	EDMA	MCHK	MERC	GERC	320	ATM	ACCT	ACCF	IDIR	APU
Timer Manager	EXTI.MAC		x														x									x		x			
Task Manager	EXTM.MAC	x	x										x		x											x		x			
Floating Point Traps	FLTP.MAC																		x	x												
Conti-guous File SVC1 Interrupt	FMCO.MAC					x				x																						
Indexed File SVC1 Interrupt	FMIN.MAC						x																									
SVC7 Interrupt	FMA7.MAC					x	x	x				x		x			x								x							
	FMB7.MAC					x	x	x	x			x		x			x								x			x	x			
	FMS7.MAC							x				x					x															
File Manager Utility Routine	FMUT.MAC			x				x	x		x		x			x																
Intercept Manager	INTC.MAC																															
	INTD.MAC																															
Memory Diagnos-tics	MCHK.MAC			x																			x									
Panic Dump	DUMP.MAC																															
End of System Modules	UBOT.MAC																															
Communi-cations Support	ITFM.MAC																															

x = supported source module features

Certain features supported by the system source modules might not be desired in the user operating system. Therefore, the user can change the features that a source module supports. By choosing a module whose value equals 0, support for that feature is excluded. To include support for a feature, choose a module whose value for the specified sysgen option equals 1.

Example:

```
SGN.SPOL EQU 1 Spooling support included

SGN.SPOL EQU 0 Spooling support excluded
```

Table 3-3 lists the system source modules, source sysgen parameters and the approximate amount of memory the desired sysgen option occupies.

TABLE 3-3 DEFINITIONS OF SOURCE SYSGEN PARAMETERS

SYSGEN PARAMETERS	DEFINITION	APPROXIMATE MEMORY REQUIRED
SGN.JRNL = 0	System journal support excluded	N/A
SGN.JRNL = 1	System journal support included	2.5kb
SGN.SAFE = 0	Safety check support excluded	N/A
SGN.SAFE = 1	Safety check support included	1.0kb
SGN.DBUG = 0	System debug support excluded	N/A
SGN.DBUG = 1	System debug support included	2.0kb
SGN.BCMD = 0	Bulk file command support excluded	N/A
SGN.BCMD = 1	Bulk file command support included	0.5kb
SGN.CO = 0	Contiguous file support excluded	N/A
SGN.CO = 1	Contiguous file support included	1.0kb

TABLE 3-3 DEFINITIONS OF SOURCE SYSGEN PARAMETERS (Continued)

SYSGEN PARAMETERS	DEFINITION	APPROXIMATE MEMORY REQUIRED
SGN.INX = 0	Indexed and nonbuffered indexed file support excluded	N/A
SGN.INX = 1	Indexed and nonbuffered indexed file support included	3.8kb
SGN.DIR = 0	Secondary directory support excluded	N/A
SGN.DIR = 1	Secondary directory support included	2.5kb
SGN.MIRR = 0	Mirror disk support excluded	N/A
SGN.MIRR = 1	Mirror disk support included	12.75kb
SGN.SPOL = 0	Spooling support excluded	N/A
SGN.SPOL = 1	Spooling support included	1.25kb
SGN.SP32 = 0	SPL/32 Spooling support excluded	N/A
SGN.SP32 = 1	SPL/32 Spooling support included	1.25kb
SGN.ROLL = 0	Roll-in support excluded	N/A
SGN.ROLL = 1	Roll-in support included	3.1kb
SGN.TEMP = 0	Temporary file support excluded	N/A
SGN.TEMP = 1	Temporary file support included	0.5kb
SGN.SEG = 0	Sharable segmentation support excluded	N/A
SGN.SEG = 1	Sharable segmentation support included	0.5kb
SGN.AOVL = 0	Link overlay support excluded	N/A
SGN.AOVL = 1	Link overlay support included	1.75kb

TABLE 3-3 DEFINITIONS OF SOURCE SYSGEN PARAMETERS (Continued)

SYSGEN PARAMETERS	DEFINITION	APPROXIMATE MEMORY REQUIRED
SGN.ITAM = 0	Communications support excluded	N/A
SGN.ITAM = 1	Communications support included	0.5kb
SGN.TGD = 0	Trap-generating device support excluded	N/A
SGN.TGD = 1	Trap-generating device support included	0.25kb
SGN.DF = 0	DPFP support excluded	N/A
SGN.DF = 1	DPFP support included	2.9kb
SGN.SF = 0	SPFP support excluded	N/A
SGN.SF = 1	SPFP support included	1.9kb
SGN.SEEK = 0	Seek scheduling support excluded	N/A
SGN.SEEK = 1	Seek scheduling support included	400 bytes
SGN.EDMA = 0	Extended direct memory access support excluded	N/A
SGN.EDMA = 1	Extended direct memory access support included	250 bytes
SGN.MCHK = 0	Memory diagnostics support excluded	N/A
SGN.MCHK = 1	Memory diagnostics support included	3.5kb
SGN.MERC = 0	Memory error recording support excluded	N/A
SGN.MERC = 1	Memory error recording support included	3.7kb

TABLE 3-3 DEFINITIONS OF SOURCE SYSGEN PARAMETERS (Continued)

SYSGEN PARAMETERS	DEFINITION	APPROXIMATE MEMORY REQUIRED
SGN.GERC = 0	General error recording support excluded	N/A
SGN.GERC = 1	General error recording support included	2.5kb
SGN.3200 = 0	Perkin-Elmer Series 3200 processors support excluded	N/A
SGN.3200 = 1	Perkin-Elmer Series 3200 processors support included	1.5kb
SGN.MAT = 0	Model 7/32, 8/32 and 3220 memory management excluded	N/A
SGN.MAT = 1	Model 3205, 3210, 3230, 3240, 3250 and 3200MPS System memory management included	N/A
SGN.ACCT = 0	Accounting support excluded	N/A
SGN.ACCT = 1	Accounting support included	2.0kb
SGN.ACCF = 0	Delete/rename file accounting reporting support excluded	N/A
SGN.ACCF = 1	Delete/rename file accounting reporting support included	0.25kb
SGN.IRDR = 0	Internal reader support excluded	N/A
SGN.IRDR = 1	Internal reader support included	1.50kb
SGN.APU = 0	APU support excluded	N/A
SGN.APU = 1	APU support included	20.0kb

TABLE 3-4 SYSTEM OBJECT MODULES AND SUPPORTED FEATURES

SYSTEM OBJECT MODULE		SUPPORTED FEATURES																																				
FUNCTION	NAME	JRNL	SAFE	DBUG	BCMD	CO	INX	DA	DIR	MIRR	SP32	SPOL	ROLL	TEMP	SEG	AOVL	ITAM	TGD	DF	SF	SEEK	EDMA	MCHK	MERC	GERC	3200	ATM	ACCT	ACCF	IRDR	APU
APU Support	APSV.F01																																				0	
	APSV.F02	0	0	0																																	1	
Console Driver	CDVR.F01																																					
Display and Bulk Commands	CMDB.F31		0	0	1			0	0	0	0	0	0	0			1																					
	CMDB.F32		0	0	1			1	0	0	0	1	1	1			1																					
	CMDB.F33		0	0	1			1	1	1	1	1	1	1			1																					
Command Executors	CMEX.F33									1	1	1					1						1															
Internal Reader	CMIR.F01																																				0	
	CMIR.F02																																				1	
Console Monitor	CMON.F01																																					
Console and CSS Supervisor	CMSP.F33	0								1													1															
Error Recording	ERRC.F01																							0	0													
	ERRC.F02																							0	1													
	ERRC.F03																							1	1													
Accounting	EXAC.F01																											0	0									
	EXAC.F02																											1	0									
	EXAC.F03																											1	1									
Interrupt Handler	EXIN.F51	0														1	1						1			0	0	1									
	EXIN.F53	0	0	0												1	1						1	1		1	1	1									
I/O Supervisor	EXIO.F11	0	0							1											1	1						0									
	EXIO.F12	0	0							1											1	1						1									
Loader and Segment Control	EXLD.F50							1					0		1	1							1				0										
	EXLD.F51							1					1		1	1							1				0										
	EXLD.F52							1					0		1	1							1				1										
	EXLD.F53							1					1		1	1							1				1										
Memory Manager	EXMY.F52		0					1					0				1									1	1										
	EXMY.F53		0					1					1				1									1	1										

TABLE 3-4 SYSTEM OBJECT MODULES AND SUPPORTED FEATURES (Continued)

SYSTEM OBJECT MODULE		SUPPORTED FEATURES																														
FUNCTION	NAME	JRNL	SAFE	DBUG	BCMD	COX	INX	DAT	DIR	MIRR	SP2	SPOL	ROLL	TMP	SEG	AVL	AIM	TGD	DF	SF	SEK	EDMA	MCHK	MERC	CERC	320	ATM	ACCT	ACCF	IRDR	APU
Supervisor Services	EXSP.F51	0										1															0				
	EXSP.F53	0										1															1				
SVC3,4, 5,6,9,14	EXSV.F01							0					0		1	0		1														
	EXSV.F52							1					0		1	1		1														
	EXSV.F53							1					1		1	1		1														
Timer Manager	EXTI.F01		0														0									0					
	EXTI.F02		0														1									0					
Task Manager	EXTM.F50	0	0										0													0		1			
	EXTM.F51	0	0										1		1											0		1			
	EXTM.F52	0	0										0													1		1			
	EXTM.F53	0	0										1		1											1		1			
Floating Point Traps	FLTP.F02																		0	1											
	FLTP.F03																		1	0											
	FLTP.F04																		1	1											
Contiguous File Manager	FMCO.F33					1				1																					
Indexed File Manager	FMIN.F33						1																			1					
SVC7 Assign/ Close	FMA7.F31					0	0	0				0		0			1									0					
	FMA7.F33					1	1	1				1		1			1									1		1	1		
	FMA7.F34					1	1	1				0		1			1									1		1	1		
SVC7 Allo/Del	FMB7.F31					0	0	0	0			0		0			1									0		0	0		
	FMB7.F33					1	1	1	1			1		1			1									1		1	1		
SVC7 Interrupt	FMS7.F31							0				0					1														
	FMS7.F33							1				1					1														
File Manager Utility	FMUT.F31							0	0			0		0			1														
	FMUT.F32			0				1	0			1		1			1														
	FMUT.F33			0				1	1			1		1			1														
SVC Interception Support	INTC.F01																														
	INTC.F02																														
Memory Diagnostics	MCHK.F01																						0								
	MCHK.F02			0																			1								
Mirror Disk	MIRR.F01									0																					
	MIRR.F02									1																					
Panic Dump	DUMP.F01																														

0 = feature is not supported
1 = feature is supported
shading = variation

3.2.2 Standard Driver Library (DRIVER.LIB)

The standard driver library, DRIVER.LIB, is required in the link phase of the Sysgen/32 process. DRIVER.LIB is available in object and source versions. The source modules can be modified and included in the driver or user library.

3.3 STANDARD COMMUNICATION LIBRARIES

There are two standard communication libraries required in the link phase of the Sysgen/32 process if communication devices are to be configured in the operating system:

- The standard communications system libraries
- The standard communications driver libraries

3.3.1 Standard Communications System Libraries

Table 3-5 presents the standard communications system libraries.

TABLE 3-5 STANDARD COMMUNICATIONS SYSTEM LIBRARIES

LIBRARY DESCRIPTION	LIBRARY NAME
Basic communications package library	ITBSYS.LIB
ZDLC enhancement package library	ITESZDLC.LIB
3270 emulator package souce library	ITES327E.LIB

These libraries are available in object and source versions.

3.3.2 Standard Communications Driver Libraries

Table 3-6 presents the standard communications driver libraries.

TABLE 3-6 STANDARD COMMUNICATIONS DRIVER LIBRARIES

LIBRARY DESCRIPTION	LIBRARY NAME
Basic communications package driver library	ITEDLIB.LIB
ZDLC driver library	ITEDZDLC.LIB
3270 emulator package driver library	ITED327E.LIB
Ethernet data link controller (EDLC) driver library	ITEDETHR.LIB

3.4 USER-DEFINED LIBRARIES

If a user-written communication protocol and/or driver is to be included in the operating system, it must be included in one of the following user-defined libraries:

- User-defined system library
- User-defined driver library

3.4.1 User-Defined System Library

Table 3-7 presents the user-defined system library provided to contain user-written/enhanced system modules.

TABLE 3-7 USER-DEFINED SYSTEM LIBRARY

LIBRARY DESCRIPTION	LIBRARY NAME
User-defined system library	USERSYS.LIB

3.4.2 User-Defined Driver Library

Table 3-8 presents the user-defined driver library provided to contain user-written device drivers.

TABLE 3-8 USER-DEFINED DRIVER LIBRARY

LIBRARY DESCRIPTION	LIBRARY NAME
User-written drivers	USERDLIB.LIB

3.4.3 User-Defined Driver Device Control Block (DCB) Macro Library

The DCB macro library provided for user-written drivers is shown in Table 3-9.

TABLE 3-9 USER-DEFINED DRIVER DCB MACRO LIBRARY

LIBRARY DESCRIPTION	LIBRARY NAME
User-written DCB macros	USERDLIB.MLB

3.5 MACRO LIBRARIES

The macro libraries presented in Table 3-10 are required in the macro expansion phase of the Sysgen/32 process. The macros that comprise the Sysgen/32 macro library are listed and explained in Table 3-11.

TABLE 3-10 MACRO LIBRARIES

LIBRARY DESCRIPTION	LIBRARY NAME
Sysgen/32 macro library	SYSGEN32.MLB
Macro library containing system structures	SYSSTRUC.MLB
Basic communication system macro library (required only if communication support is included)	ITMS.MLB
General driver macro library	DVRM.MLB
System macro library	SYSMACRO.MLB
Model 3270 emulator macro library	ITED327E.MLB

TABLE 3-11 MACROS IN THE SYSGEN/32 MACRO LIBRARY

MACRO CALL NAME	FUNCTION
AND	Used to AND two hex character strings, of up to eight characters, and returns result in %RESLT.
ANDD	Used to AND two 1-byte characters and returns answer to %Z.
APBGEN	Generates and initializes the APU processor block.
BIOCGEN	Initializes the device-dependent part of the DCB for bidirectional input/output control (BIOC) driven CRTs.
CONVNUM	Converts a decimal number to 2-digit hexadecimal character string.
CCBI	Generates and intializes the channel control block (CCB) for non-ITAM devices.
DCBFORM	Includes DCB format with instructions to aid the user in creating DCB macro for user-written drivers.

TABLE 3-11 MACROS IN THE SYSGEN/32 MACRO LIBRARY (Continued)

MACRO CALL NAME	FUNCTION
DCBI	Creates the DCBs and intializes the DCB fields for non-ITAM devices.
DCBINIT	Sets all structure copy flags to zero and starts the DCB module.
DCBTERM	Terminates the DCB module.
DCBxxx	User-defined DCB macro.
DCHX	Converts a decimal arithmetic value to hex string and returns result in %HXVAL.
DEF	Defines storage or constants; optionally repeats the definition the specified number of times.
DFLIST	Defines a list and generates a label and entry based on the parameters passed.
DMT	Generates the device mnemonic table (DMT). Entries are generated using the information in the sysgen device statements.
DMTEND	Generates the end of the DMT.
EVNGEN	Generates device, directory and bit map leaves; generates required nodes (coordination, selector channel (SELCH), controller).
FLTPINIT	Generates external entries (EXTRNS) for the appropriate floating point modules depending on support specified in the FLOAT configuration statement. (The EXTRN for a user-specified module is generated by the Sysgen/32 program.) Generates labels and reserves memory for the program status word (PSW) and register save areas.
GENAPBD	Generates and initializes the directory to the APU processor blocks.
HXDC	Used by HXDCP macro to generate decimal strings.
HXDCP	Used to convert hex strings to decimal strings. The result is returned in global variable %DCVAL.
ITAMCCB	Generates and initializes the CCBs for ITAM devices.

TABLE 3-11 MACROS IN THE SYSGEN/32 MACRO LIBRARY (Continued)

MACRO CALL NAME	FUNCTION
ITDCE	Creates and initializes the Ethernet specific portion of the DCB.
ITDC1 ITDC2 ITDC3	Creates and initializes the device-dependent portion of the DCB for ITAM devices.
IVTGEN	Generates the initial value table (IVT) from information entered in the configuration input file.
LPMTGEN	Generates and initializes the logical processor mapping tables.
MMDGEN	Generates the mass storage media (MSM) or multimedia diagnostic (MMD) device-dependent portion of the DCB for MSM or MMD disks.
MTPI	Initializes the additional DCB fields needed for magnetic tapes.
SVTGEN	Generates and initializes the system value table.
TTBGEN	Generates and intializes trap blocks for APUs.
USEREND	Used to end user-determined DCBs.
USERINIT	Resets structure copy flags. Allows a copy of the desired DCB structure to be included in the user's DCBxxx macro expansion.
$CRDP	Contains the extended DCB structures for card reader/punches.
SLABEL	Used to create a label and an entry for the passed parameter.
SMCONFIG	Defines the memory configuration for Perkin-Elmer Series 3200 processors based on the information in the MCONFIG statement.
SPDMT	Flags virtual devices in the DMT.

TABLE 3-11 MACROS IN THE SYSGEN/32 MACRO LIBRARY (Continued)

MACRO CALL NAME	FUNCTION
SPTINIT	Defines and initializes the system pointer table (SPT); sets the panic address and system queue pointer; generates the interrupt service pointer table (ISPT), task control blocks (TCBs) and the TCB table; generates the journal and pointer stack. Defines and initializes the system pointers and tables, and reserves memory for system structure.
STARTUP	Builds the command stream to run at system start-up using STARTUP...ENDS statement input.
TCOMINIT	Generates and reserves memory for the global task common segments using TCOM statement input.
VMTGEN	Generates a label and reserves space for the volume mnemonic table (VMT).
$XEVIN	Allows EVN macro to be called in a follow on program module.
$UP $REV	Used by the SPTINIT macro (when generating the SPT) to update the operating system revision ID.

3.6 INCLUDING USER-WRITTEN DRIVERS

If a nonstandard device is to be included in the operating system, it must be defined in the sysgen device statements. The user-defined DCB macro library (USERDLIB.MLB) must be specified during the macro expand phase. The library containing the user-written driver (USERDLIB.LIB) for the device must be specified during the link phase of the Sysgen/32 process.

Use the reserved device codes 240-254 to configure a user-written driver in the system.

Each device configured in the system gets an appropriate DCBxxx macro call written to the .MAC output file (xxx is the device code; e.g., DCB39, DCB147, DCB245, etc.) The DCBxxx macro creates the device DCB and external references to the device driver (in DRIVER.LIB or USERDLIB.LIB). The user must create the DCBxxx macro definition and put it in the user USERDLIB.MLB file.

3.6.1 Creating the DCBxxx Macro

Creating the DCBxxx macro entails a few intermediate steps:

1. Use MLU32 to get the DCBFORM macro from the SYSGEN32.MLB file to use as the pattern.

2. Make the appropriate changes noted in DCBFORM to create the DCB macro.

3. Save the file as DCBxxx.MAC.

4. Use the MLU32 (macro library) utility to add the DCBxxx macro definition to your USERDLIB.MLB file. This library will be searched by Macro/32 before the SYSGEN32.MLB in the normal sysgen process. Use care when creating definitions of macros with names identical to macro names in other libraries.

5. Use COPY/32 or the LIBLDR utility to add the driver code to your USERDLIB.LIB file. The USERDLIB.LIB file will be edited by OS/32 Link before the standard DRIVER.LIB file. Therefore, modified Perkin-Elmer drivers that use standard Perkin-Elmer device codes can also be placed in USERDLIB.LIB, preempting the standard Perkin-Elmer driver.

6. Perform a sysgen using the standard SYSGEN.CSS. The USERDLIB.MLB file will be assigned and the DCBxxx definition will be used.

See the DCBFORM macro in the SYSGEN32.MLB file.

The following is a sample CSS used to build USERDLIB.MLB.

```
XDEL      USERDLIB.MLB
$BUILD    USERDLIB.CMD
ESTAB     USERDLIB.MLB
INCL      DCB240.MAC
INCL      DCB241.MAC
INCL      DCB242.MAC
SAVE      *
END
$ENDB
LO   .BG,MLU32;TA .BG
AS   5,USERDLIB.CMD,SRO
AS   6,CON:
ST
XDEL USERDLIB.CMD
$EXIT
```

The following is a sample CSS used to build USERDLIB.LIB.

```
XAL     USERDLIB.LIB,IN,126
$BUILD  LIBLDR.CMD
COPY    6,1
COPY    7,1
COPY    8,1
END
ENDB
LO   .BG,LIBLDR;  TA .BG
AS   1,USERDLIB.LIB
AS   5,LIBLDR.CMD,SRO
AS   6,USERD240.OBJ,SRO
AS   7,USERD241.OBJ,SRO
AS   8,USERD242.OBJ,SRO
ST
XDEL LIBLDR.CMD
$EXIT
```

3.7 INCLUDING USER-WRITTEN MODULES

The Sysgen/32 interface allows a user to include user-written or user-modified modules in the system. To include user-written or user-modified source modules, follow these steps:

1. Copy the source sysgen parameters from the appropriate file, SYSGEN.MAC, to a back-up file.

2. Copy the system source module library to a back-up file.

3. Change the source parameters in SYSGEN.MAC to include or exclude the desired options.

4. If the name of the modified module differs from the original module name, the PROG, ENTRY and EQUATE statement labels must be changed to the modified module name. The $OSPROG macro can be used to change the original module name to the modified module name.

5. The following assignments must be made to expand the internal operating system macros using the Common Assembly Language (CAL) Macro/32 processor:

```
AL usermodule.EXP,IN,80
L  .BG,MACRO32
T  .BG
AS 1,usermodule.CAL
AS 2,usermodule.EXP
AS 3,NULL:
AS 8,SYSSTRUC.MLB,SRO
AS 9,SYSMACRO.MLB,SRO
AS 10,ITMS.MLB,SRO
ST ,ML=(8,9,10),MLIST=(ND,NG)
```

To assemble module EXSV, logical unit 11 (lu11) must be assigned to SYSMAC32.MLB. To assemble module FMS7, lu11 must be assigned to SYSMAC32.MLB, and lu12 must be assigned to DVRM.MLB.

NOTE

If user-written macros are to be included in the expansion, assign lu11 to the appropriate macro library and start Macro/32 by entering:

```
ST ,ML=(8,9,10,11),MLIST=(ND,NG)
```

When assembling a driver from the general purpose driver library, assign lu11 to DVRM.MLB and start Macro/32 by entering:

```
ST ,ML=(8,9,10,11),MLIST=(ND,NG)
```

6. Assemble the modified source module using CAL/32 with the CROSS and SQUEZ options specified. (The SQUEZ option is required to assemble the floating point module.)

```
AL usermodule.OBJ,IN,126
L  .BG,CAL32,55
T  .BG
AS 1,usermodule.EXP
AS 2,usermodule.OBJ
AS 3,PR:
TE 5,EC
AS 7,SYSGEN.MAC,SRO
ST ,CROSS,ERS,SQUEZ=99
```

7. Use the following LIBLDR commands to create the USERSYS.LIB.

```
LO .BG,LIBLDR
TA .BG
AL newsys.LIB,IN,126
AS 0,NULL:
AS 1,USERSYS.LIB
AS 2,NEWSYS.LIB
AS 3,CON:
AS 4,usermod.OBJ
AS 5,CON:
ST
.BG>TAB 1,3
.BG>RW 1
.BG>TAB 4,3
.BG>RW 4
.BG>DUPE 1,2 sysmod.nnn
.BG>COPY 4,2
.BG>COPY 1,0
.BG>DUPE 1,2
.BG>RW 2
.BG>TAB 2,3
.BG>END
```

8. Perform a sysgen using the modified user system library. Include the MODULE...ENDM statements to include the modified module if the module name changed.

3.8 SOURCE-LEVEL SYSTEM GENERATION (SYSGEN) EXAMPLE

To include the code to allow JOURNAL support in the operating system, the system source module EXSP.MAC must be reassembled by following these steps:

1. See Table 3-4 for the source module EXSP.MAC affected by the sysgen variable SGN.JRNL.

2. Make a back-up copy of the EXSP.MAC file.

3. Change the option variables in the SYSGEN.MAC file to:

```
SGN.JRNL EQU 1
```

   This will include JOURNAL support. If the system is a Model 7/32, 8/32 or 3220, SGN.ATM must also be set to zero.

4. Use the EDIT commands to change the source sysgen options in the SYSGEN.MAC file.

```
LO .BG,EDIT32
TA .BG
ST
.BG>GET SYSGEN.MAC
.BG>T /SGN.JRNL/
      8    SGN.ITAM     EQU   0
.BG>CH / 0/,/ 1/
      8/   SGN.JRNL     EQU   1
.BG>SAVE NEWSGN.MAC
.BG>END
```

5. Make the following assignments to expand the internal operating system macros using the CAL Macro/32 processor.

```
LO .BG,MACRO32,20
TA .BG
AL EXSP.EXP,IN,80/4
AS 1,NEWEXSP.MAC
AS 2,EXSP.EXP
AS 3,NULL:
AS 8,SYSSTRUC.MLB,SRO
AS 9,SYSMACRO.MLB,SRO
AS 10,ITMS.MLB,SRO
ST ,ML=(8,9,10)
```

6. Assemble the user-modified EXIN.EXP module using CAL/32.

```
AL EXSP.OBJ,IN,126
LO .BG,CAL32,55
TA .BG
AS 1,EXSP.EXP
AS 2,EXSP.OBJ
AS 3,PR:
TEMP 5,EC
AS ,7NEWSGN.MAC,SRO
ST ,CROSS,SQUEZ=99,ERS
```

7. Include the new object module, the user system library.

8. Perform a sysgen using the modified system library.

```
REM     USERSYS.LIB,USERSYS.OLD
AL      USERSYS.LIB,IN,126
LO      .BG,LIBLDR; TA .BG
AS      0,NULL
AS      1,USERSYS.OLD,SRO
AS      2,USERSYS.LIB
AS      5,CON:
AS      6,EXSP.OBJ,SRO
ST
.BG> DUPE   1,2  EXSP.F53
.BG> COPY   6,2
.BG> COPY   1,0
.BG> DUPE   1,2
.BG> RW     2
.BG> TAB    2,5
.BG> END
.
```

CHAPTER 4
STANDARD OS/32-SUPPORTED DEVICES

4.1 INTRODUCTION

OS/32-supported devices and device characteristics are described in this chapter. These characteristics must be defined in the System Generation/32 (Sysgen/32) device statements when a device is included in the operating system. Nonstandard devices supported by user-written drivers must use reserved device codes and must also be defined in the device statements to be included in the system. See Chapter 3 for information on configuring user-written modules and drivers in the system.

4.2 OS/32-SUPPORTED LOCAL AND REMOTE DEVICES

Table 4-1 lists, by local and remote device codes, the OS/32-supported devices. Table 4-2 lists the categories (card reader/punch, local video display unit (VDU), etc.) of OS/32-supported devices and their local and remote device codes.

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
0-15		Reserved	
16		Model 33 Teletype (TTY) keyboard/printer	CLI
17		Model 35 TTY keyboard/printer	CLI
18		Nonediting VDU	CLI, CLCM RS-232C
19-20		Reserved	
21		Carousel 15, 30, 35, 80-character line	CLI
22		Carousel 15, 30, 35, 132-character line	CLI
23		Model 1100 VDU	CLI, CLCM
24-33		Reserved	
34	147	Nonediting VDU, 2- or 8-line	RS-232C, CLCM, CLI

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES (Continued)

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
35		Reserved	
36	147	Graphic display terminal (GDT), 2- or 8-line	RS-232C, CLCM
37	147	Carousel 300, 2- or 8-line	RS-232C, CLCM, CLI
38	147	Carousel 300 with electronic forms control, 2- or 8-line	RS-232C, CLCM,CLI
39		Models 550, 1100 and 6100 VDU, 2- or 8-line	RS-232C, CLCM,CLI
40		Line printer bidirectional input/output control (BIOC)	
41		Reserved	
42		50Mb CDD50 fixed portion (25Mb)	
43		50Mb CDD50, removable portion (25Mb)	
44		256Mb disk, fixed	
45		1.5Mb head per track (HPT)	
46		160Mb, fixed	
47		1.5HPT	
48		2.5Mb disk, fixed	
49		2.5Mb disk, removable	
50		5Mb disk, fixed	
51		5Mb disk, removable	
52		40Mb disk, removable	
53		67Mb disk, removable	
54		256Mb disk, removable	
55		Floppy disk	
56		68.5Mb disk, fixed	
57		1.5Mb HPT	
58		67Mb disk, fixed	
59		16Mb disk, removable	
60		16Mb disk, fixed	
61		48Mb disk, fixed	
62		80Mb disk, fixed	
63		675Mb disk, fixed	
64		800 mag tape	
65		800/1600 mag tape	
66		Intertape cassette	
67		Reserved	
68		6250/1600/800 STC mag tape	
69		6250/1600/800 STC mag tape (halfword mode)	
70		6250/1600/800 TELEX mag tape (halfword mode)	

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES (Continued)

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
71-79		Reserved	
80		High-speed paper tape reader/ punch	
81		Model 33 TTY reader/punch	CLI
82		Model 35 TTY reader/punch	CLI
83		Carousel 35 with paper tape reader, 80-character line	
84		Carousel 35 card reader, 132-character line	
85-95		Reserved	
96		Card reader with software translate, 029 card encoding	
97		Card reader with hardware translate, 029 card encoding	
98		Card reader with software translate, 026 card encoding	
99-103		Reserved	
104		Card reader, high-speed, 029 card encoding	
105		Card reader/punch, 029 card encoding	
106		Card reader/punch with print option separate, 029 card encoding	
107-111		Reserved	
112		Low-speed line printer	
113		Medium-speed line printer	
114		High-speed line printer	
115-127		Reserved	
128		8-line interrupt module	
129		Digital (MUX) controller	
130-135		Reserved	
136		Real-time analog system with internal clock	
137		Real-time analog system with user-supplied external clock	
138		Mini input/output (I/O) analog input	
139		Mini I/O analog output	
140		Mini I/O digital	
141-142		Reserved	
143		I/O bus switch	

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES (Continued)

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
	144	Asynchronous communications line	RS-232C, MPC,CLCM
	145	Remote line printer (ITAM) and letter quality printer	
	146	SIGMA 10 terminal (communications)	
18	147	Nonediting VDU (communications)	RS-232C, MPC,CLCM
	148-155	Reserved	
	156	Model 1200 VDU (communications)	RS-232C, MPC,CLCM
	157	Model 1250 point-to-point VDU; Series 7000 8-bit data transfer	
	158	Model 1250 multidrop VDU	RS-232C, MPC,CLCM
	159	Reserved	
	160	Binary synchronous communications line on 201 DSA	RS-232C, MPC,CLCM
	161	IBM 3780 remote job entry (RJE) emulation on 201 DSA	
	162	IBM 2780 RJE emulation on 201 DSA	
	163	Binary synchronous processor-to-processor link on 201 DSA	
	164-167	Reserved	
	168	Binary synchronous communications line on quad-synchronous adapter (QSA)	QSA,SSA, MPC
	169	IBM 3780 RJE emulation on QSA	
	170	IBM 2780 RJE emulation on QSA	
	171	Binary synchronous processor-to-processor link on QSA	
	172	Binary synchronous IBM 3270 support on QSA	
	173-175	Reserved	
	176	ZBID line driver on QSA; supervisor call 15 (SVC15) access only; half-duplex	
	177	ZBID line driver on QSA; SVC15 access only; simplex I/O	
	178	ZBID CTM on QSA; two-way simultaneous (TWS) operational mode; SVC1 access only; simplex input; "mother" to device 179	QSA,SSA, MPC

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES (Continued)

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
	179	Zero-bit insertion/deletion (ZBID) channel terminal manager (CTM) on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 178	QSA,SSA, MPC
	180	ZBID CTM on QSA; TWA operational mode; SVC1 access only; half-duplex	
	181	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex input; "mother" (with assembled DCT) to device 182	QSA,SSA, MPC
	182	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 181	QSA,SSA, MPC
	183	ZBID CTM on QSA; TWA operational mode; SVC1 access only; half-duplex; with assembled DCT	
	184	Reserved	
	185	Binary synchronous IBM 3270 emulation on QSA	
	186	ZBID CTM for X.25 level 3 on QSA; TWS operational mode; SVC1 access only; simplex input; "mother" to device 187	QSA,SSA, MPC
	187	ZBID CTM for X.25 level 3 on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 186	QSA,SSA, MPC
	188-191	Reserved	
	192	Direct memory access I/O subsystem (DIOS)	
	193-211	Reserved	
	212	Ethernet line driver on Ethernet controller; SVC15 access only; simplex input; read side (must be used together with write side)	EDLC
	213	Ethernet line driver on Ethernet controller; SVC15 access only; simplex output; write side (must be used together with read side)	EDLC

TABLE 4-1 DEVICE CODES FOR OS/32-SUPPORTED DEVICES (Continued)

LOCAL DEVICE CODE	REMOTE DEVICE CODE	DEVICE TYPE	INTERFACE
	214-223	Reserved	
	224-239	Reserved for CAMAC access method	
	240-254	Reserved for user-written drivers	
	255	Null device	

CLI = Current loop interface
CLCM = Current loop communications multiplexor
RS-232C = programmable asynchronous line system (PALS), programmable asynchronous single line adapter (PASLA), 2- or 8-line COMM MUX
QSA = Quad-synchronous adapter
SSA = Single synchronous adapter
MPC = Multiperipheral controller
EDLC = Ethernet data link controller

NOTES

1. CLI is equivalent to TTY interface.

2. Device code 145 now supports the remote serial line printer (ITAM) and the letter quality printer.

3. Device codes 34 and 36 through 40 can be utilized to invoke the BIOC driver and can be used for local or dial-in applications. The BIOC driver supports vertical forms control (VFC).

4. Full VFC read and write operations are supported by the BIOC driver, the line printer driver and the point-to-point terminal manager. However, the point-to-point terminal manager supports VFC in conversational mode only.

5. The TTY driver and the asynchronous terminal manager will not support VFC read or write operations. However, the affected software (TTY and asynchronous terminal manager) is modified to treat a VFC write operation as if a formatted write operation had been received. The VFC character will be ignored rather than intepreted or printed.

6. The Model 3270 communication devices are not part of the standard Perkin-Elmer operating system. To have Model 3270 support in the target system, the modules must be ordered from a Perkin-Elmer sales office. If Model 3270 devices are configured for the standard operating system, their specification will result in undefined symbols at link time.

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Card reader	With software translate, 029 card encoding	96	
	With hardware translate, 029 card encoding	97	
	With software translate, 026 card encoding	98	
	Carousel 35 reader	84	
Card reader/ punch	High-speed, 029 card encoding	104	
	029 card encoding	105	
	With print option separate, 029 card encoding	106	
TTY reader/ punch	Model 33 (CLI)	81	
	Model 35 (CLI)	82	
	Carousel 35 with paper tape reader, 80-character line	83	
	Carousel 35 card reader, 132-character line	84	
TTY keyboard printer	Model 33 (CLI)	16	
	Model 35 (CLI)	17	
	Nonediting VDU (CLI)	18	
	Carousel 15, 30, 35, 80-character line (CLI)	21	
	Carousel 15, 30, 35, 132-character line (CLI)	22	
	Model 1100 VDU (CLI)	23	
High-speed paper tape reader/punch	High-speed	80	

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES (Continued)

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Line printer (ITAM) and letter quality printer	Low-speed	112	
	Medium-speed	113	
	High-speed	114	
	Remote		145
Line printer (BIOC)	Local	40	
Tape cassette	Intertape	66	
Magnetic tape	800 bpi (bits per inch)	64	
	1600/800 bpi	65	
	6250/1600/800 bpi STC	68	
	6250/1600/800 bpi STC halfword mode	69	
	6250/1600/800 bpi TELEX tape with standard controller	70	
Disks	50Mb CDD50, fixed portion (25Mb)	42	
	50Mb CDD50, removable portion (25Mb)	43	
	256Mb disk, fixed	44	
	1.5Mb, HPT	45	
	160Mb, fixed	46	
	1.5Mb HPT	47	
	2.5Mb disk, fixed	48	
	2.5Mb disk, removable	49	
	5Mb disk, fixed	50	
	5Mb disk, removable	51	

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES (Continued)

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Disks (Continued)	40Mb disk, removable	52	
	67Mb disk, removable	53	
	256Mb disk, removable	54	
	68.5Mb disk, fixed	56	
	1.5Mb disk, HPT	57	
	67Mb disk, fixed	58	
	16Mb disk, removable	59	
	16Mb disk, fixed	60	
	48Mb disk, fixed	61	
	80Mb disk, fixed	62	
	675Mb disk, fixed	63	
Floppy disk	Floppy disk	55	
Local VDU	Nonediting VDU (CLCM) (RS-232C)	34	147
	GDT, (CLCM) (RS-232C)		
	Carousel 300 (CLCM) (RS-232C)	37	147
	Carousel 300 with electronic forms control (CLCM) (RS-232C)	38	147
	Models 550, 1100 and 6100 (CLCM) (RS-232C)	39	147
8-line interrupt module	8-line interrupt module	128	
Digital MUX	Digital MUX controller	129	

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES (Continued)

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Conversion equipment	Real-time analog system with internal clock	136	
	Real-time analog system with user-supplied external clock	137	
Analog I/O Controller	Mini I/O analog input	138	
	Mini I/O analog output	139	
Digital I/O Controller	Mini I/O digital	140	
I/O Bus Switch	I/O Bus Switch	143	
Communication Devices	Asynchronous communications line, line driver only		144
	Remote line printer and letter quality printer		145
	SIGMA 10 terminal		146
	Nonediting VDU (CLI)	18	147
	Models 550 and 1100 VDU (CLI)	39	147
	TEC 455 editing VDU		155
	Model 1200 VDU		156
	Model 1250 point-to-point VDU; Series 7000 8-bit data transfer		157
	Model 1250 multidrop VDU		158
	Binary synchronous on 201 DSA, line driver only		160
	IBM 3780 RJE emulation on 201 DSA		161

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES (Continued)

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Communication Device (Continued)	IBM 2780 RJE emulation on 201 DSA		162
	Binary synchronous processor-to-processor on 201 DSA		163
	Binary synchronous on QSA, line driver only		168
	IBM 3780 RJE emulation on QSA		169
	IBM 2780 RJE emulation on QSA		170
	Binary synchronous processor-to-processor link on QSA		171
	Binary synchronous IBM 3270 link on QSA		172
	ZBID line driver on QSA; SVC15 access only; half-duplex		176
	ZBID line driver on QSA; SVC15 access only; simplex I/O		177
	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex input; "mother" to device 179		178
	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 178		179
	ZBID CTM on QSA; TWA operational mode; SVC1 access only; half-duplex		180

TABLE 4-2 CATEGORIES OF OS/32-SUPPORTED DEVICES (Continued)

TYPE	DEVICE	LOCAL DEVICE CODE	REMOTE DEVICE CODE
Communication Devices (Continued)	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex input; "mother" (with assembled DCT) to device 182		181
	ZBID CTM on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 181		182
	ZBID CTM on QSA; TWA operational mode; SVC1 access only; half-duplex; with assembled DCT		183
	Binary synchronous IBM 3270 emulation on QSA		185
	ZBID CTM for X.25 level 3 on QSA; TWS operational mode; SVC1 access only; simplex input; "mother" to device 187		186
	ZBID CTM for X.25 level 3 on QSA; TWS operational mode; SVC1 access only; simplex output; "daughter" to device 186		187
	DIOS		192
	Ethernet line driver on Ethernet controller; SVC15 access only; simplex input; read side (must be used together with write side)		212
	Ethernet line driver on Ethernet controller; SVC15 access only; simplex output; write side (must be used together with read side)		213

4.3 DEVICES USING CURRENT LOOP INTERFACE (CLI)

Devices using CLI can be used as the console device. These devices connected to the CLI can be configured in the system:

- Carousel 15, 30, 35
- Carousel 300
- Model 33 TTY
- Model 35 TTY
- Nonediting VDU
- Model 550
- Model 1100

When the paper tape reader/punch and keyboard/printer features of the TTY and Carousel devices are both supported, a shared-busy condition exists, since these devices are configured at the same address.

NOTE

If the reader/punch feature is supported by an (ASR) TTY console device, the reader/punch feature must not be assigned to a user task (u-task).

If a Carousel supports paper tape, it can only support the paper tape reader feature.

Examples:

```
TTY:,2,016,CONS          *Model 33, console device
TRP1:2,081               *Reader/punch, same device
```

4.4 DEVICES USING STANDARD RS-232C INTERFACE (PALS, PASLA, 2- OR 8-LINE COMMUNICATIONS MULTIPLEXOR) OR CURRENT LOOP COMMUNICATIONS MULTIPLEXOR (CLCM)

Devices using the standard RS-232C interface or CLCM can be used as the console device. The following devices connected to the standard RS-232C interface or CLCM can be configured in the system:

- Nonediting VDU
- Graphic display terminal
- Carousel 300
- Carousel 300 with electronic format control (EFC)
- Model 550 (nonediting) VDU
- Model 1100 (nonediting) VDU
- Model 1200 (editing) VDU
- Model 1250 (editing) VDU
- Model 6100 VDU

Because the editing features of Model 1200 and 1250 are not currently supported by the VDU driver, both models must be configured as a nonediting VDU. Communication supports the editing features of Model 1200 and 1250. See the OS/32 Basic Data Communications Reference Manual for details of this support.

CAUTION

DO NOT USE MODEL 1200 AND 1250 SYSTEMS WITH THE EDITING FEATURE AS THE CONSOLE DEVICE, BECAUSE DATA WRITTEN TO THE DEVICE CAN LOCK THE KEYBOARD.

The default strapping for PALS, PASLA, 2- and 8-line communications MUX, and CLCM is as follows:

- The clocks are strapped to a baud rate between 110 and 9600. A clock is selected when the device begins an I/O operation.
- The function switches for Model 1100, 1200 and 1250 Systems should be set for even parity. All other devices should be set for no parity.
- The interface must be strapped full-duplex, disable CARR status, disable DSRDY status and disable CL2S for nonediting VDUs, graphic display terminals and Carousels. For the Model 1100, 1200 and 1250 VDUs, the interface must be strapped full-duplex, enable CARR status, enable DSRDY status and enable CL2S status.

4.4.1 The Bidirectional Input/Output Control (BIOC)

With the release of OS R06.2, the standard local CRT driver is the BIOC driver, which will be configured in the system by default at sysgen time. Certain features of the BIOC driver can be enabled and disabled. To do this, the following parameters can be specified in the DEVICES...ENDD statements:

RECLEN=n	where n is the number of characters per line.
SIZE=n	where n is the number of lines per page.
XCOD=n	where n specifies a halfword with bit definitions explained in Table 4-3.

NOTE

The XDCOD= and CLOCK= parameters are mutually exclusive when establishing a BIOC DCB. The user can mask in clock support through the XDCOD parameters shown in Section 2.3.11.

TABLE 4-3 BIOC XDCOD BIT DESCRIPTIONS

BIT	HEX	MEANING
0	8000	Support ITAM extended option connect/disconnect.
1	4000	Suppress BREAK/ESC function. Selection of this option prevents actuation of a BREAK from causing an I/O error on remote printers.
2	2000	Enable CTRL-A to cause entry into baud adjust routine. Selection of this option allows local terminals to select different baud rates.
3	1000	Suppress write time-out if carrier is off. Selection of this option allows a dial-in user to be welcomed by multi-terminal monitor (MTM)'s ID message at the completion of baud adjust. This bit takes precedence over bit 0 on write requests and should not be set if bit 0 is set.

TABLE 4-3 BIOC XDCOD BIT DESCRIPTIONS (Continued)

BIT	HEX	MEANING
4	0800	Support full BIOC functions. Selection of this option enables the full complement of BIOC features. If this option is not selected, the ASCII read functions will be limited to CTRL-H, as a nondestructive backspace, and CTRL-X, as a cancel line request. In addition, CTRL-Q through CTRL-T will be stripped from the input to allow full implementation of flow control as used by some data-concentrators.
5	0400	Suppress mark off of devices at sys-init time for false sync. At system boot or hot-start, BIOC addresses every device that has been sysgened with its DCBs. If BIOC receives a false sync in response to an output command, it marks that device off. If a system has devices on bus-switches, the power fail delay may result in these devices being marked off. If the above conditions cause problems, it is advisable to suppress this feature at sysgen time.
6	0200	Support carriage return (CR) as termination for image read. Normally, image reads are terminated only with buffer-full or break. Selection of this option enables CTRL-M to terminate the read. If CTRL-M is used to terminate the read, it will not be echoed to the terminal but it will be placed into the buffer as data.
7-15	01FF	Select command 2 for choice of stop bits, data bits, parity and clock rate. (Consult the appropriate interface, i.e., 2- and 8-line COMM MUX, MPC, etc., programming manual for specific information.) If bit 7 is a 1, bits 8-15 will be used as the entire command 2. If bit 7 is a 0, bits 8-11 must be 0 and bits 12-15 can be A, B, C, D or 0 to select only the clock rate. When bit 7 is 0, the default defined by the DCB is used for the other options.
	LN	Specifies terminal/printer logical record length. Normally this length is the number of characters that can be printed on one line.
	PS	Specifies terminal/printer page size. Normally this size is the number of lines that can be printed on one page.

The BIOC default interface strappings are as follows:

- The clocks are strapped to baud rates between 50 and 19,200. A clock is selected when the device begins an I/O operation. The initial clock can be selected by the SYSGEN statement, but can be changed through the use of the auto-baud adjustable feature.

- The function switches for Perkin-Elmer Model 1100 and 1200 terminals (device code 39) should be set for even parity. All other devices (device codes 34, 38 and 40) should be set for no parity.

- The interface must be strapped full-duplex. In most local applications, CARR, DSRDY and CL2S status lines should be disabled. For Perkin-Elmer Model 1100 and 1200 terminals and dial-in applications, CARR, DSRDY and CL2S are usually enabled.

- For dial-in applications, a modem cable should be used. BIOC uses the CARR status to detect the disconnection of dial-in lines. When CARR drops, BIOC returns I/O errors to ensure that the user is terminated from an active MTM environment. If this response is not desired, CARR status should be disabled.

- Enabling CL2S status will prevent writes from proceeding if a port is not connected. The time-out status that would result from this condition will usually result in loss of the data being written, the MTM ID message, for example. For this reason, BIOC allows the user to selectively suppress write time-outs if CARR status is also absent. This is typically used on dial-in lines when the user wants to display the MTM ID message upon completion of the auto-baud adjust.

4.5 INTERTAPE CASSETTES

The two cassettes located in a transport are interlocked in the hardware, causing a shared-busy condition.

Example:

```
CAS1:,45,066                    *Two cassettes on
CAS2:,55,066                    *same transport
```

4.6 CARD READERS

These card readers can be configured in the system:

- Card reader 029 with software code translation

- Card reader 029 with hardware code translation
- Card reader 026 with software code translation

There is an additional feature of the 026 and 029 card reader drivers that can be included in the system at sysgen time. The driver can translate the Hollerith code into EBCDIC instead of ASCII code. This feature can be included by specifying the following translate tables in the device statement:

- INITRE26, which converts all 026 card reader codes directly to EBCDIC representation in memory
- INITRE29, which converts all 029 card reader codes directly to EBCDIC representation in memory

Example:

```
CR29:,04,96          *Standard card reader, software
                      translate, 029 card encoding

CRHW:,04,97          *Card reader, hardware
                      translate, 029 card encoding

CR26:,04,98          *Standard card reader, software
                      translate, 026 card encoding
```

4.7 CARD PUNCHES

These card punches can be configured in the system:

- High-speed 029 card encoding
- Card reader/punch (interpreting)
- Card reader/punch 029 card encoding with print option

As supplied, these devices translate 029 Hollerith code to/from ASCII. There is an additional feature of the card punch and card reader/punch that can be included in the system. Optionally, the devices can translate 026 Hollerith code to/from ASCII, 026 Hollerith code to/from EBCDIC, or 029 Hollerith code to/from EBCDIC. These features can be included by referencing the following translate tables in the device statement:

- INITPA26 or INITRA26, which converts all 026 card codes directly to ASCII representation in memory

- INITPE29 or INITRE29, which converts all 029 card codes directly to EBCDIC representation in memory
- INITPE26 or INITRE26, which converts all 026 card codes directly to EBCDIC representation in memory

Example:

```
CP:,7,104                  *High-speed card punch
CRP:,8,105                 *Card reader/punch
CRPS:,9,106                *Card reader/punch with print separate
```

4.8 LINE PRINTERS

These line printers can be configured in the system:

- 60-200 lpm (lines per minute) printer
- 300 lpm printer
- 600 lpm, 1000 lpm printer
- Remote serial line printer and letter quality printer

Example:

```
LPR1:,62,112               *Low-speed printer (60-200 lpm)
LPR2:,63,113               *Medium-speed printer (300 lpm)
LPR3:,72,114               *High-speed printer (600 lpm,
                            100 lpm)
RMOT:,48,145,XDCD=X'800     Serial line printer (110 baud
                            through 9600)
```

4.9 HIGH-SPEED PAPER TAPE READER/PUNCH

High-speed paper tape reader/punches can be configured in the system.

Example:

```
PRTP:,13,80                *High-speed paper tape reader/punch
```

4.10 MAGNETIC TAPE CONTROLLERS AND TAPES

Magnetic tape controllers are specified in the controller option of the device statements. A magnetic tape controller supports from one-to-four transports.

The device statement for the tapes must identify the tapes for each controller. Each tape requires a device statement that allows more than one tape to be active at one time. The following magnetic tapes can be configured in the system:

- 800 bpi magnetic tapes
- 1600 bpi magnetic tapes
- 1600/800 bpi dual density magnetic tape drives
- 6250 bpi magnetic tapes - special systems controller (STC)
- 6250 bpi - halfword mode controller (STC)
- 6250 bpi - TELEX tape device

The 6250 tape drives may require extended direct memory access (EDMA) bandwidth coordination. The 6250 tape drives can run with gaps or in gapless mode.

Example:

```
MAG1:,85,64,CONTR=0,SELCH=F0   *Tape1 (800 bpi)
MAG2:,95,64,CONTR=0,SELCH=F0   *Tape2 (800 bpi)
MAG3:,C5,65,CONTR=1,SELCH=F0   *Tape3 (1600 bpi)
MAG4:,D5,65,CONTR=1,SELCH=F0   *Tape4 (1600 bpi)
MAG5:,80,68,SELCH=F0,CONTR=2   *Tape5 (6250 bpi)
MAG6:,90,68,SELCH=F0,CONTR=2   *Tape6 (6250 bpi)
```

4.11 DISKS

All Perkin-Elmer-supported disks have one physical disk volume per disk drive, except for the 10Mb disk (M46-416), which is composed of two 5Mb disk volumes, and the 68.5Mb disk, which is composed of a 1.5Mb disk and a 67Mb disk.

The driver allows many of the mass storage media (MSM) disks to be divided into two or more virtual disks. This capability is described in Section 4.11.2.2. Three algorithms are available for scheduling I/O queued to a disk: priority, first-in/first-out (FIFO) and seek optimization (C-SCAN). The C-SCAN algorithm is supplied as a default and is used to produce the highest throughput on the disk by reducing the number and range of seek operations. It includes an adjustment to prevent one task from queuing many consecutive I/Os. If a task request is queued ahead of all other requests, and the current I/O is being executed for this same task, then this request is scheduled on the next scan of the arm.

The priority and FIFO algorithms are selected by specifying the QUEUE option in the device statement. For priority scheduling, DISKQ is changed to COMQ. For FIFO, it is changed to COMFIFO. Priority schedules by calling task priority and then FIFO within each priority level. FIFO schedules without regard to priority or disk position.

A second option is the elimination of the EDMA coordination overhead in the systems that do not require memory bandwidth monitoring. This option is not selectable by an option on the device statement. This is accomplished in the DCB macro for the desired disk type (see Section 4.11.1) in the macro library SYSGEN32.MLB.

To remove EDMA coordination from an MSM300 removable disk (device code 54), use the LIST command of the Macro Library Utility to save macro DCB54 of SYSGEN32.MLB to a new file. Then, use the OS/32 EDIT GET command to retrieve this new file and locate the string EDMA=SEEKCHK, an argument to the DCBI macro call. Delete the string and save the edited file. Using the Macro Library Utility, include this edited file in the USERDLIB.MLB, the DCB macro library provided for user-written drivers.

NOTE

This option is not recommended unless a careful analysis of the hardware configuration has verified that memory bandwidth monitoring is not required.

4.11.1 Moving-Head Disks

A moving-head disk controller supports from one-to-four disk drives. The device statements for the disks must identify the SELCH and controller address.

The following disks can be configured in the system:

- 2.5Mb, fixed

- 2.5Mb, removable
- 5Mb, fixed
- 5Mb, removable

} Compose a 10Mb disk

- 40Mb, removable

The 10Mb disk is composed of two 5Mb disks. These two disks share the same drive, causing a shared-busy condition. Two device statements must be specified for a 10Mb disk. The shared-busy conflict is resolved by Sysgen/32, based on the device code/device address.

Example:

```
DSC1:,C6,51,SELCH=F0,CONTR=B6    *Removable
DSC2:,C7,50,SELCH=F0,CONTR=B6    *Fixed
DSC3:,D6,51,SELCH=F0,CONTR=B6    *Removable
DSC4:,D7,50,SELCH=F0,CONTR=B6    *Fixed
```

DSC1 and DSC2 have a shared-busy condition, as do DSC3 and DSC4.

4.11.2 Mass Storage Media (MSM) Disks

These disks can be configured in the system:

- MSM300 256Mb, fixed
- MSM300 256Mb, removable
- MSM80 67Mb, removable
- MSM80 67Mb, fixed
- MSM80 HPT 1.5Mb, fixed

} Compose a 68.5Mb disk

- MSM80 HPT 68.5Mb, fixed and removable
- MSM1.5 HPT of 675, fixed
- MSM160, fixed

The 68.5Mb MSM HPT disk is composed of a 1.5Mb fixed-head disk and an MSM 67Mb fixed disk.

4.11.2.1 Mass Storage Media (MSM) Dual Port Option

If the disk drive is configured with a dual port option, meaning it is shared with another system, then the dual port option (DUAL) must be specified in the device statement. If this parameter is omitted, the disk driver will treat the alternate channel busy condition as an error.

An example of the MSM disk with dual port option is as follows:

```
D300:,EC,54,SELCH=F1,CONTR=EB,DUAL 256Mb removable - dual port
```

4.11.2.2 Mass Storage Media (MSM) Virtual Disk Option

The MSM driver supports a virtual disk option on those device codes marked with an asterisk in Table 4-4. A virtual device is an area of contiguous cylinders on an MSM disk that appears to the system as a real physical disk. A virtual disk can be initialized and checked with FASTCHEK, dumped and restored using FASTBACK or BACKUP, marked on/off and used just as any other disk.

Virtual disk can be used for the following purposes.

- Virtual disk provides dedicated spool, temporary and roll volumes, without wasting an entire physical disk. This significantly reduces fragmentation on the permanent file storage volumes, and it also reduces allocation time for spool, roll and temporary files.

- Virtual disk provides additional smaller user volumes. This can reduce file allocation time by reducing the size of the directories to be searched. It also allows separate groups, departments or projects to be assigned private volumes.

- Virtual disk provides additional scratch volumes that do not need to be backed up. This helps reduce fragmentation on more permanent storage volumes.

- Virtual disk allocates relatively small dedicated volumes for fairly static usage such as the system volume/system account or the on-line source, documentation, etc., of a mature software project.

Virtual disk can provide significant performance improvements and operational convenience. However, there are several cautions, which are listed below.

- The amount of memory used increases, due to the fact that an additional disk DCB is included for each virtual and physical disk configured. Also, additional system space is required if the disk is marked on with core directory support.

- If an existing disk drive is reconfigured as multiple virtual disks, the existing data must be backed up and redistributed among the virtual volumes after booting the new system and initializing the virtual disks.

- Once a physical disk has been initialized as multiple virtual disks, that disk pack (if removable) is locked into that particular drive and sysgen or one configured exactly the same. Furthermore, the drive is restricted to disk packs for which the virtual disk layout matches the sysgened layout. For this reason, it is recommended that removable drives be sysgened as a whole disk in addition to several virtual disks.

- All virtual disks on a removable disk must be marked off before removing the disk platter. The operating system does not enforce this.

- Virtual disks must be checked and backed up individually.

- The loader storage unit (LSU) can only boot an operating system from a virtual disk that starts at cylinder zero.

Virtual disks are configured by specifying two or more disks at the same address and using the VIRT= parameter of the DEVICES...ENDD statement to specify the starting cylinder and number of cylinders for the virtual disk. Cylinders are counted starting at zero. The number of usable cylinders for each MSM disk is given in Table 4-4. Virtual disks must not overlap, nor is the last virtual disk permitted to extend beyond the end of the physical disk. Neither Sysgen/32 nor the operating system (OS) enforces these restrictions; therefore, the user must exercise caution in configuring the system.

To determine the number of cylinders to allocate, divide the desired amount of disk storage per volume by the cylinder size. Cylinder size can be calculated from the information given in Table 4-4 by the following formula:

Cylsize = (sectors/track)*(tracks/cylinder)*0.25kb/sector

The following are examples of MSM disks with virtual disk options:

```
D300:,EC,54,SELCH=F1,CONTR=EB                    Note 1

DV01:,EC,54,SELCH=F1,CONTR=EB,VIRT=(0,323)       Notes 2 and 3

DV02:,EC,54,SELCH=F1,CONTR=EB,VIRT=(323,250)     Note 4

DV03:,EC,54,SELCH=F1,CONTR=EB,VIRT=(773,250)     Notes 5 and 6
```

NOTES

1. D300 is an entire 256Mb removable disk at locations EC, EB and F1. This device must not be marked on while any of the virtual disks are on-line. The disk normally allows access to disk packs that have not been initialized with virtual disks.

2. Virtual volume number 1 on D300 (DV01) contains 323 cylinders starting at cylinder zero.

3. The LSU can boot from this volume because it starts at cylinder zero.

4. DV02 contains 250 cylinders starting at cylinder 323.

5. DV03 contains 250 cylinders starting at cylinder 773.

6. Note that the number of cylinders in the virtual disks total 823. This number of total cylinders can be located in Table 4-4 under device number 54.

I/O error recording is supported for MSM disks only. Tables 4-1 and 4-2 contain the device codes for MSM disks. Table 4-4 contains an expanded listing of MSM disks.

TABLE 4-4 DEVICE CODES FOR MSM

DEVICE CODE	DEVICE TYPE	SECTORS PER TRACK	TRACKS PER CYLINDERS	CYLINDERS ON DISK	MAINTENANCE CYLINDERS	RELATED DEVICES
44*	256Mb,	64	19	1022	2	-
45	1.5Mb, HPT	64	10	9.6	-	46
46*	160Mb, fixed	64	10	821	2	45
47	1.5Mb, HPT	64	40	2.4	-	63
53*	67Mb, removable	64	5	823	-	-
54*	256Mb, removable	64	19	823	-	-
56	68.5Mb, fixed	64	5	836.2	3	-
57	1.5Mb, HPT	64	5	19.2	-	58
58*	67Mb, fixed	64	5	820	3	57
59*	16Mb, removable	64	1	823	-	60,61,62
60*	16Mb, fixed	64	1	821	2	59
61*	48Mb, fixed	64	3	821	2	59
62*	80Mb, fixed	64	5	821	2	59
63*	675Mb, fixed	64	40	821	2	47

* Supports virtual disk

Examples of MSM disks:

```
MSM1:,FC,53,SELCH=F0,CONTR=FB 67Mb disk, removable
MSM2:,FD,53,SELCH=F0,CONTR=FB 67Mb disk, removable
MSM3:,FE,58,SELCH=F0,CONTR=FB 67disk, fixed
HPT3:,FE,57,SELCH=F0,CONTR=FB 1.5Mb HPT
MSX3:,FE,56,SELCH=F0,CONTR=FB 67 + 1.5Mb, fixed

D300:,EC,54,SELCH=F1,CONTR=EB 256Mb disk, removable
D301:,EC,54,SELCH=F1,CONTR=EB 56Mb disk, removable
```

4.11.3 Direct Memory Access (DMA) Coordination Nodes

It is possible to configure DMA devices whose total bandwidth would exceed the capacity of the EDMA bus. To prevent the resultant data overruns and inefficient operation, coordination of EDMA activity can be configured by using the EDMA node. Generally, the 6250 bpi, 125 inches per second (ips) and the MSM disks are considered for EDMA coordination. Only when present in extreme numbers on the Model 7/32 processor would the slower tapes and disks be considered for coordination.

If the number of channels is greater than the number of simultaneous data transfers allowed in the system, a DMA coordination node must be configured. However, if the number of channels is less than or equal to the number of simultaneous data transfers allowed, no coordination statements are required.

DMA coordination is specified in the COORDINATION statement.

Example:

```
DSC1:,FC,53,SELCH=F0,CONTR=FB *67Mb (MSM80) disk
DSC2:,EC,53,SELCH=F1,CONTR=EB *67Mb (MSM80) disk2
Mag1:,85,68,SELCH=F2,CONTR=0  *6250 bpi, 125 ips magnetic
                               tape driver
COORDINATION=TRANSFER=4,SELCH=(F0,F1,F2)
```

4.11.4 Floppy Disk Subsystems

A floppy disk subsystem consists of one controller with one-to-four spindles. The controller is transparent to the user. All spindles share the same controller, causing a shared-busy condition. The shared-busy conflict is resolved by SYSGEN/32 based on the device address configuration described in Section 2.2.11.

The device statement for the disk has a device mnemonic associated with the drive only when the disk is marked off-line.

Two algorithms are available for scheduling I/O queued to a floppy disk:

- Priority
- FIFO

The priority algorithm is selected by specifying the QUEUE option in the device statement. For priority scheduling, COMFIFO is changed to COMQ. Priority schedules by calling task priority and then FIFO within each priority level. The default algorithm is FIFO. FIFO schedules without regard to priority or disk position.

A second option available is elimination of the EDMA coordination overhead in systems that do not require EDMA coordination. The field DCB.EDMA should be left at zero. This is accomplished in the DCBINI macro in SYSGEN32.MLB by omitting EDMA=SEEKCHK.

If there is more than one floppy disk subsystem in the system, a DMA coordination node must be configured above the subsystems it controls. If a DMA coordination node is not defined, intermittent I/O errors, loss of data or system failure can occur. Although the floppy disk is not a DMA device, it uses the same coordination methods to operate within the bandwidth limitations of the controller.

The device statement for each spindle has:

- a device mnemonic associated with the spindle, and
- a spindle number (0, 1, 2 or 3) specified in the SPINDLE option of the device statement.

Examples:

```
    FLP1:C1,055,SPINDLE=0          *First floppy disk spindle
    FLP2:C1,055,SPINDLE=1          *Second floppy disk spindle
    FLP3:C1,055,SPINDLE=2          *Third floppy disk spindle

    FLP1:C1,055,SPINDLE=0          *First spindle for subsystem1
    FLP2:C1,055,SPINDLE=1          *Second spindle for subsystem1
    FLP3:C1,055,SPINDLE=2          *Third spindle for subsystem1
    FLPA:C2,055,SPINDLE=0          *First spindle for subsystem2
    FLPB:C2,055,SPINDLE=1          *Second spindle for subsystem2
COORD DEV=(FLP1,FLP2,FLP3,FLPA,FLPB),TRANSFER=1
```

4.11.5 50Mb Cartridge Disk Drive (CDD50) Subsystem

The CDD50 subsystem, supplied with the basic Model 3205 System, consists of one 50Mb CDD50, a 1 by 4 intelligent disk controller (IDC), power supply, rack-mounting hardware and necessary cables.

The CDD50 provides both removable and fixed storage. The unformatted capacity of the CDD50 is 50Mb. The removable disk cartridge provides 25Mb of storage with the remaining 25Mb being provided by the nonremovable disk. The CDD50 has a transfer rate of 1.2Mb per second, an average access time of 35ms and an average rotational latency of 8.55ms.

4.12 8-LINE INTERRUPT MODULE

Each 8-line interrupt module must be defined by a device statement. Trap-generating device (TGD) support must be included in the system at sysgen time for an 8-line interrupt module.

Example:

```
LIN0:,20,128
LIN1:,21,128
LIN2:,22,128
LIN3:,23,128
LIN4:,24,128
LIN5:,25,128
LIN6:,26,128
LIN7:,27,128
```

4.13 SYSGENING A SYSTEM WITH A COMMUNICATIONS MULTIPLEXOR (COMM MUX)

The COMM MUX interfaces a Perkin-Elmer Series 3200 processor, via a MUX bus, to various device controllers. It is available in the 2- and 8-line version. Each line can be strapped for one of four groups, with each group containing four program-selectable clock rates. The COMM MUX baud rate is made compatible to the device by selecting the clock that matches the desired baud rate. The selected clock is specified by the CLOCK option of the device statement (see Table 4-5).

TABLE 4-5 BAUD RATES WITHIN AN INSTALLED GROUP

STRAP OPTION	BAUD RATE	CLOCK
Group 1	50	XA
	110	XB
	1,800	XC
	2,400	XD
Group 2	75	XA
	134.5	XB
	2,000	XC
	3,600	XD
Group 3	150	XA
	600	XB
	4,800	XC
	9,600	XD
Group 4	300	XA
	1,200	XB
	7,200	XC
	19,200	XD

4.14 CONVERSION EQUIPMENT CONTROLLER (MINI INPUT/OUTPUT (I/O) SYSTEM)

These conversion equipment controllers can be configured in the system:

- Real-time analog system
- Analog input controller
- Analog output controller
- Digital I/O controller

Example:

```
RTAS:,83,136          *Real time analog system
AIC:,88,138           *Analog input controller
AOC1:,98,139          *Analog output controller1
AOC2:,99,139          *Analog output controller2
DIC:,A9,140           *Digital input controller
DOC:,A8,140           *Digital output controller
```

CHAPTER 5
SAMPLE SYSTEM GENERATION/32 (SYSGEN/32) SESSIONS

5.1 INTRODUCTION

This chapter is presented to guide the less experienced user through the initial Sysgen/32 session. It is assumed that the user has read the previous chapters (Chapter 1 in particular) in this manual before attempting to use Sysgen/32. This chapter identifies three general situations and then presents a sample Sysgen/32 session that can be used for each situation. One of these situations should apply to a user about to use Sysgen/32 for the first time. The sample Sysgen/32 sessions presented are not the only way Sysgen/32 can be used in these situations. As the user becomes more familiar with the Sysgen/32 process, these samples need not be used. However, for the first-time user, an easy-to-follow procedure for using Sysgen/32 is provided.

5.2 SYSTEM GENERATION/32 (SYSGEN/32) START-UP SITUATIONS

When the user is ready to perform a sysgen operation, one of the following three situations will likely exist:

- A configuration input file has been previously created and must now be processed.
- No previously created configuration input file exists. The user wishes to create a configuration input file in the conversational mode and then process it.
- No previously created configuration input file exists. The user wishes to create a configuration input file in the interactive mode and then process it.

In each situation, the user must first load and then start the Sysgen/32 program using the system LOAD and START commands, respectively. The following three sections provide a sample Sysgen/32 session for each of the three situations described above. The user should note that these are sample Sysgen/32 sessions and the user is not required to use these procedures in every case.

5.2.1 Processing a Previously Created Configuration Input File

The following commands will load and start the Sysgen/32 program and process a previously created configuration input file:

```
*LOAD SYSGEN32
*START, INPUT=FIGUR1.SYS, OUTPUT=MACRO1.MAC, LIST=PR:
-
-SYSGEN/32 R00-00
```

When these commands are entered, the configuration input file (FIGUR1.SYS) is processed by the Sysgen/32 program in batch mode. The macro calls generated by the Sysgen/32 program are written to the output file (MACRO1.MAC). List output is written to the list file (PR:).

The user can now follow the procedures outlined in Chapter 1 for expanding the macros, assembling the expanded macros and linking the object module to yield an operating system.

5.2.2 Creating a Configuration Input File Conversationally and Processing the Input File

The following commands will load and start the Sysgen/32 program and allow the user to create a configuration input file in the conversational mode and then process the newly created file:

```
*LOAD SYSGEN32
*START, LIST=PR:, COMMAND=CON:
-
-SYSGEN/32 R00-00
>INPUT=FIGUR2.SYS
>OUTPUT=MACRO2.MAC    <must not exist>
>CONVERSATIONAL
  .
  .
  .
```

At this point, a conversational prompt and response session begins. (Section 5.3.1 presents a sample conversational prompt session.) Each response entered by the user is translated into a configuration input statement and then written to the designated input file (FIGUR2.SYS). When the prompt session has completed, the user processes the input file by entering the following command:

```
.
.
.
>PROCESS
 .
 .
 .
-END OF TASK CODE=0
```

When the Sysgen/32 program has successfully completed processing the configuration input file, the output file contains the macro calls generated. The user can now follow the procedures outlined in Chapter 1 for expanding the macros, assembling the expanded macros and linking the object module to yield an operating system.

5.2.3 Creating a Configuration Input File Interactively and Processing the Input File

The following commands will load and start the Sysgen/32 program and allow the user to create a configuration input file in an interactive mode and then process the newly created file:

```
*ALLOCATE FIBUR3.SYS,IN,80
*LOAD SYSGEN/32
*START, INPUT=FIGUR3.SYS,OUTPUT=MACRO3.MAC,LIST=PR:
       ,COMMAND=CON:
-
-SYSGEN/32 R00-00
>PROCESS
 READY FOR CONFIGURATION INPUT
>.
 .
 .
```

At this point, the user can interactively enter the required sysgen configuration statements. Chapter 2 describes the format and content of these sysgen statements. It is important to note that the specified input file (FIGUR3.SYS) must be an empty file or the interactive prompt will not appear. When the sysgen statements have been entered, the following commands are entered to complete the input file and begin processing:

```
>ENDC
 READY FOR SYSGEN COMMANDS
>PROCESS
 .
 .
 .
-END OF TASK CODE=0
```

When the task has reached a successful completion, the macro calls generated by the Sysgen/32 program are located in the output file, MACRO3.MAC. The user can now follow the procedures outlined in Chapter 1 for expanding the macros, assembling the expanded macros and linking the object module to yield an operating system.

5.3 CREATING A CONFIGURATION INPUT FILE

As the examples in the previous section show, Sysgen/32 allows the user to create a configuration input file in two modes:

- conversational mode, and
- interactive mode.

For the less experienced user, the conversational mode is ideal, since the user merely has to respond to a series of prompts supplied by the Sysgen/32 program. The user responses are translated to create configuration input statements, which are written to the configuration input file. This file is then processed via the Sysgen/32 PROCESS command.

The interactive mode requires more knowledge on the part of the user because no Sysgen/32 prompts are issued. The user enters sysgen configuration statements interactively from the command device and, therefore, must know what statements to include and the format of each statement. The sections that follow provide examples of creating a configuration input file in both conversational and interactive modes.

5.3.1 Using the Conversational Mode

A conversational prompt and user response session allows the user to create a configuration input file from the command device. The responses are translated into configuration input statements and are written to the specified configuration input file. This file is specified with either the INPUT parameter of the START command or with the INPUT command. If a user response to a system prompt is not acceptable, the Sysgen/32 program continues issuing the prompt until an acceptable response is entered. All conversational prompts follow this general format:

```
QUESTION (POSSIBLE CHOICES) [DEFAULT CONDITION]
```

A conversational session does not end until the Sysgen/32 program has issued all of the prompts contained in the conversational file. When all of the prompts have been displayed, Sysgen/32 informs the user that the program is ready to accept Sysgen/32 commands.

At this point, the user specifies the name of the macro output file using the Sysgen/32 OUTPUT= command and then begins processing of the configuration input file by entering the Sysgen/32 PROCESS command. The Sysgen/32 END command terminates the session. A successful completion has been reached if an end of task code of zero is received. Following is an example of a conversational session.

```
*LOAD SYSGEN32
*START, INPUT= FIGURE1.SYS, COMMAND=CON:
-
-SYSGEN/32 R00-00
>CONVERSATIONAL
 PROCESSOR MODEL (7/32, 8/32, 3210, 3220, 3230, 3240, 3250
3200MPS, 3205) [3220]
>3220
 NUMBER OF REGISTER SETS (2 OR 8) [8]
>8
 O/S VERSION (8 CHAR. ALPHANUMERIC STRING) [BLANKS]
>3220.N13
 SIZE OF MEMORY IN KB (64-16384, DIVISIBLE BY 16) [256]
>CR
 DYNAMIC SYSTEM SPACE IN KB (1 TO TOP OF MEMORY) [25]
>CR
 MAX. NUMBER OF TASKS (1 TO 252) [32]
>35
 BACKGROUND MAX. PRIORITY (11 TO 248) [16]
>15
 BACKGROUND MAX. SYSTEM SPACE (0.25 KB INCREMENTS) [9]
>9
 TIME SLICE [0]
>40
 POWER FAIL DELAY [5]
>10
 POWER FAIL RESTART (MANUAL OR AUTO) [MANUAL]
>AUTO
 COMMAND BUFFER LENGTH (32 TO 1024 BYTES) [80]
>CR
 LOG MESSAGE BUFFER LENGTH (32 TO 132 BYTES) [72]
>CR
 CSS LEVELS (1 TO 249) [5]
>5
 NUMBER OF QUEUE ENTRIES (10 TO 64999) [TOTAL NUMBER OF DEVICES]
>CR
 NUMBER OF JOURNAL ENTRIES (0 TO 12999) [0]
>20
 MAC/MAT ADDRESS (300, 500, 900) [X"300"]
>CR
 LINE CLOCK FREQUENCY (50 OR 60) [60]
>60
 PHYS. ADDR. OF PRECISION INTERVAL CLOCK [6C]
>6C
 LINE FREQUENCY CLOCK ADDR. [6D]
>6D
```

```
 DATE FORMAT (USA OR EUROPEAN) [USA]
>USA
  .
  .
  .
 SECONDARY DIRECTORY SUPPORT (YES OR NO) [NO]
>YES
 INTERCEPT SUPPORT (YES OR NOT) [NO]
>YES
 MAX. FILE BLOCK SIZE (1 TO 255) [4]
>10
 DATA AND INDEXED BLOCKS FOR INDEX FILES (d/i) [1/1]
>4/6
 DATA AND INDEXED BLOCKS FOR SPOOL FILES (d/i) [1/]
>4/6
 DATA AND INDEXED BLOCKS FOR NON=BUFFERED FILES (d/i) [64/3]
>64/8
  .
  .
  .
 ENTER STARTUP PROCEDURES (COMMANDS OR CR) [NO STARTUP PROC.]
>CR
 MEMORY DIAGNOSTIC SUPPORT (YES OR NO) [NO]
>Y
 ERROR RECORDING SUPPORT (YES OR NO) [NO]
>CR
 TCOM SUPPORT (YES OR NO) [NO]
>Y
 ENTER TCOM NAME
>TCM1
 ENTER START ADDR OF TCOM
>10000
 ENTER TCOM SIZE (IN 0.25 KB INCREMENTS)
>.25
 COORDINATION (EDMA) SUPPORT (YES OR NO) [NO]
>Y
 MCONFIG (YES OR NO) [NO]
>NO
 NAME OF CONSOLE DEVICE (CON:)
>CON:
 DEVICE STATEMENT SECTION:
 NOTE: TO ADVANCE TO THE NEXT DEVICE TYPE, ENTER A "/"
  IN RESPONSE TO THE NAME PROMPT OF THE CURRENT DEVICE TYPE
 TTY/RP MODEL 33 (CURRENT LOOP INTERFACE),DCOD 16:
 DEVICE NAME [TTY]
>/
 TTY/RP MODEL 33 (CURRENT LOOP INTERFACE),DCOD 81:
 DEVICE NAME [TTY]
>TTY1
 DEVICE ADDRESS [002]
>002
  INTERRUPT LEVEL [0]
>0
  IOCLASS [2]
>2
```

```
 TTY/RP MODEL 33 (CURRENT LOOP INTERFACE),DCOD 81:
 DEVICE NAME [TTY]
>TTY2
 DEVICE ADDRESS [002]
>006
 INTERRUPT LEVEL [0]
>0
 IOCLASS [2]
>2
 TTY/RP MODEL 33 (CURRENT LOOP INTERFACE),DCOD 81:
>/
 TTY/KP MODEL 35 (CURRENT LOOP INTERFACE),DCOD 17:
 DEVICE NAME [TTY]
>/
 TTY/RP MODEL 35 (CURRENT LOOP INTERFACE),DCOD 82:
 DEVICE NAME [TTY]
>/
 NONEDITING CRT (Current LOOP INTERFACE), DCOD18:
 DEVICE NAME [TTY]
>/
 CAROUSEL 15, 30, 35 (80 CHARACTERS, CLI), DCOD21:
 DEVICE NAME [TTY]
>/
  .
  .
  .
CONVERSATIONAL PROCESSING COMPLETE
>OUTPUT=MACRO3.MAC
>PROCESS
>END
-END OF TASK CODE=0
```

NOTE

In a conversational session, there is no default for device names and addresses. The user must enter a response to the device prompt or the program will continue to issue the same device prompt until a response is received. A slash (/) bypasses a device prompt, indicating that the device displayed is not desired in the system being configured or that no further devices of this type are to be configured in the system.

Table 5-1 describes the parameters involved in device prompts.

TABLE 5-1 DEVICE PROMPT PARAMETER DESCRIPTIONS

PARAMETER	DESCRIPTION
DEVICE NAME	4-character mnemonic the system associates with a device
DEVICE ADDRESS	Hexadecimal number specifying the physical address of a device
INTERRUPT LEVEL	Decimal number from 0 (highest) to 3 (lowest) specifying the hardware interrupt level for device interruption
I/O CLASS	Decimal number from 0 to 31 specifying the device or file class
CLOCK A, B, C, D	Baud rate at which data can be transmitted to and from a terminal
SELECTOR CHANNEL	2-digit hexadecimal number specifying the selector channel (SELCH) address
CONTROLLER	2-digit hexadecimal number specifying the controller address
SPINDLE NUMBER	Decimal number from 0 to 3 specifying the floppy disk spindle number
TRANSFER RATE	Number of simultaneous transfers if coordination is specified

5.3.1.1 Accessing the HELP File in Conversational Mode

To access the HELP file during a conversational session, enter a question mark (?) in response to a program prompt. Information pertinent to the last prompt issued will be displayed to the command device.

Example:

```
 BACKGROUND MAX. SYSTEM SPACE (0.25 KB INCREMENTS) [9]
>?
 BACKGROUND:ESTABLISHES BACKGROUND PRIORITY AND SYSTEM SPACE
 COMMAND FORMAT: BACKGROUND [MAXPRI] [,MAXSIZE]
                 MAXPRI=DECIMAL NUMBER FROM 11
                        THROUGH 248
                        DEFAULT=16
                MAXSIZE=DECIMAL NUMBER, INCREMENTS OF
                        0.25 KB
                        DEFAULT=9
>CR
 BACKGROUND MAX. SYSTEM SPACE (0.25 KB INCREMENTS)  [9]
>10
```

5.3.2 A Configuration Input File Created Interactively

To create a configuration input file interactively, the user must load and start the Sysgen/32 program either with the name of an interactive device specified as the command device, or by taking the default CON:.

The user can use the START command format shown in the example in Section 5.2.3, or the user can specify only the command device in the START command and then specify the input and output files using the Sysgen/32 commands, as shown in the following example.

Example:

```
*LOAD SYSGEN/32
*START,COMMAND=CON:
-
-SYSGEN/32 R00-00
>INPUT = FIGUR1.SYS          <  must be an empty file  >
>OUTPUT= MACRO7.MAC          <  must not preexist  >
>PROCESS
 READY FOR CONFIGURATION INPUT
>.
 .
 .
>ENDC
 READY FOR SYSGEN/32 COMMANDS
>PROCESS
>END
-END OF TASK CODE=0
```

The user enters the sysgen configuration statements after the following message appears:

```
READY FOR CONFIGURATION INPUT
```

The ENDC command terminates entry of configuration statements and returns the user to the Sysgen/32 command mode. The PROCESS command initiates processing of the configuration input file. When an end of task code of zero is received, the program has reached a successful completion.

5.3.2.1 Accessing the HELP File in Interactive Mode

To access the HELP file while creating a configuration input file interactively, the user must exit the configuration input mode using the ENDC command. The Sysgen/32 HELP command can then be used to access the HELP file. When the HELP command is entered, the following message is displayed:

```
FOR A LIST OF COMMANDS TYPE HELP*
FOR HELP ON ANY COMMAND MNEMONIC, TYPE HELP <MNEMONIC>
```

When the HELP command is entered with an asterisk (*) all Sysgen/32 commands and configuration statements are displayed. When the HELP command is entered with the mnemonic of a configuration statement or Sysgen/32 command, information pertinent to that specific mnemonic is displayed.

Once the user has obtained the desired information from the HELP file, the Sysgen/32 EDIT command can be used to continue creating the configuration input file. Enter EDIT, get the input filename and position yourself to where you left off prior to using the HELP command.

Example:

```
*LOAD SYSGEN/32
*START,COMMAND=CON:
-
-SYSGEN/32 R00-00
>INPUT= FIGUR7.SYS          < empty input file >
>OUTPUT= MACRO7.MAC         < nonexistent output file >
>PROCESS
 READY FOR CONFIGURATION INPUT
>VERSION 3220
>CPU 3220
>MEM 1024
>DSYS 150
>ENDC
 READY FOR SYSGEN COMMANDS
>HELP ACCOUNTING
     ACCOUNTING: INCLUDES ACCOUNTING SUPPORT
          COMMAND FORMAT: ACCOUNTING [ N ] [,NO FILE ACCOUNTING]
          N= Decimal Value from 2 through 32, SPECIFYING THE
             MAXIMUM NUMBER OF ACCOUNTING CLASSES
             DEFAULT FOR N=4
             NO FILE ACCOUNTING SPECIFIES THAT FILE
             ACCOUNTING SUPPORT IS EXCLUDED
>EDIT
 READY FOR EDIT COMMANDS
>GET FIGUR7.SYS
 APPEND BOTTOM
 5  ENDC
 7  -1
 4  DSYS 150
 7  INS
 4.01 ACCOUNTING 4
 .
 .
 .
>DONE
 READY FOR SYSGEN COMMANDS
>PROCESS
>END
-END OF TASK CODE=0
```

5.3.3 Modifying a Configuration Input File Via Edit/32

Regardless of how it was created, a configuration input file that contains an error or an improperly specified configuration statement can be modified using the Sysgen/32 EDIT command. The EDIT command invokes Edit/32 and makes all of the Edit/32 commands available to the Sysgen/32 user. After modification, the input file can then be processed via the PROCESS command.

Example:

```
*LOAD SYSGEN32
*START,INPUT= FIGUR1.SYS, COMMAND=CON:
-
-SYSGEN/32 R00-00
>EDIT                          < invokes editor >
 READY FOR EDIT COMMANDS
>GET FIGUR1.SYS                < specifies file to be edited >
>T 1-4                         < displays lines 1-4 >
1 VERSION 3220
2 CPU 3220
3 MEM 1024
4 DSYS 200
>CHANGE /200/, /150/,4         < modifies line 4 >
4 DSYS 150
>DONE                          < saves file and exits Edit/32 >
 READY FOR SYSGEN COMMANDS
>OUTPUT= MACRO1.MAC
>PROCESS
>END
-END OF TASK CODE=0
```

APPENDIX A
COMPARISON OF SYSTEM GENERATION/32 (SYSGEN/32) AND OS/32 CONFIGURATION UTILITY PROGRAM (CUP) CONFIGURATION STATEMENT DEFAULTS

STATEMENT	SYSGEN DEFAULT	CUP DEFAULT
ACCOUNTING	4	4
BACKGROUND PRIO	16	16
BACKGROUND SIZE	9	9
CLOCK	60,6C,6D	60,6C,6D
CMDLEN	80	80
CPU	3220	7/32
REGISTERS	8 for all but 7/32	2
CSS	5	5
DATE	MMDDYY	MMDDYY
DEVADADS	0	0
DIRECTORY	OFF	OFF
D36CBLOCK	4	4
DSYS	25	25
ERRORREC	OFF	OFF
FLOAT*	N,N	N,N
FLOAT**	H,H	H,H
ILEVEL	0	0
INTERCEPT	OFF	OFF
IOCLASS	0,1,2,3	0,1,2,3
IREADER	NOIREADER	NOIREADER
ITAM	OFF	OFF
JOURNAL	0	0
LOGLEN	72	72
LPU	1	N/A
MAXAPU	9	N/A
MAXTASK	32	32
MCONFIG	OFF	OFF
MEMCHECK	OFF	OFF
MEMORY	256	128
PROCESSOR LIMITS:	16,384 for 3250, 3240, 3230 4,096 for 3210 1,024 for 3220, 7/32, 8/32	
NOSEG	ON	ON
TGD	NOTGD	TGD
QUEUE	NO. OF DEVICES	NO. OF DEVICES
RESPONSETIME	OFF	OFF
ROLL	OFF	OFF

STATEMENT	SYSGEN DEFAULT	CUP DEFAULT
SCREENTIME	OFF	OFF
SPOOL+	SPOOLER	SPOOLER
SPOOL++	N/A	SPL/32
SPL/32	SPL/32	N/A
SSTABLE	32	32
TEMP	BLANK	BLANK
TGD	OFF	ON
VERSION	BLANK	BLANK
VOLUME	BLANK	BLANK

* CPU, 7/32, 8/32
** Perkin-Elmer Series 3200
+ with parameter
++ without parameter

APPENDIX B
SYSTEM GENERATION/32 (SYSGEN/32) MESSAGES

ACCOUNTING ERROR

The maximum number of device or file classes to be supported was specified by a decimal number less than 2 or exceeding 32.

The ACCOUNTING statement was entered with a conflicting NOFILEACCOUNTING parameter.

ADDRESS ERROR address

Device address exceeds the maximum allowable range set in the DEVADS statement.

COMMAND DEVICE ERROR

Specified device or file is syntactically incorrect or failed to assign.

COMMAND NOT RECOGNIZED

An invalid command mnemonic was entered.

CONSOLE ERROR

A console device was not specified, or more than one device was specified as a console device.

CONVERSATIONAL PROCESSING COMPLETE

All conversational prompts have been displayed. The PROCESS command can now be entered.

COORDINATION ERROR name

Named device not recognized as a valid device.

DUPLICATE START OPTION: xxx

Start options were entered more than once.

ERRORS IN ASSEMBLY, SYSGEN ABORTED

A sysgen command substitution system (CSS) message indicating that errors were encountered in the macro assembly phase.

ERRORS IN MACRO EXPANSION, SYSGEN ABORTED

A sysgen CSS message indicating that errors were encountered in the macro expansion phase.

FILE DESCRIPTOR ERROR

A device or filename entered in a command was either invalid or omitted.

```
              INPUT:
              OUTPUT:
FILE ERROR ON LIST:              [Text]
              CONVERSATIONAL:
```

Text specifies one of the following:

```
ACCOUNT ERROR
ASSIGNMENT ERROR
BUFFER ERROR
FILE DESCRIPTOR ERROR
I/O ERROR
ILLEGAL FUNCTION
INVALID LU
NAME ERROR
PRIVILEGE ERROR
PROTECT ERROR
SIZE ERROR
SVC7 ERROR
TGD ASSIGNMENT ERROR
TYPE ERROR
VOLUME ERROR
```

An error was encountered during the allocation or assignment of the device or file identified by file descriptor (fd).

ILEVEL ERROR ilevel

An invalid ilevel value was issued, or an ilevel other than 0 was specified with only 2 register sets.

ILLEGAL SEPARATOR: xxx

A separator was omitted or incorrectly entered.

ILLEGAL TCOM: ADDRESS / NAME / SIZE

TCOM address is not in global memory, TCOMs overlap, the names of two TCOMs are the same or the name TSKCOM was specified.

ILLEGAL VALUE: xxxx

The characters replacing xxxx represent the flagged illegal value.

INPUT FILENAME OMITTED

A sysgen CSS message indicating the name of the configuration input file was not entered as input to the Sysgen/32 CSS procedure.

INPUT FILE filename.SYS DOES NOT EXIST, SYSGEN ABORTED

A sysgen CSS message indicating that the specified configuration input file does not exist.

INPUT MUST BE ENTERED

A CONVERSATIONAL or PROCESS command was entered before an input file was specified.

INVALID ARGUMENT: argument

An illegal argument was located while processing the configuration input file.

INVALID COMMAND IN BATCH: CONVERSATIONAL/EDIT

The EDIT or CONVERSATIONAL command was entered in batch mode.

INVALID START OPTION: option

The program does not recognize a START command option.

```
                INPUT:
                OUTPUT:
I/O ERROR ON    LIST:     text ON fd
                COMMAND:
```

Text specifies one of the following:

```
DEVICE UNAVAILABLE
END OF FILE
END OF MEDIUM
ILLEGAL OR UNASSIGNED LU
ILLEGAL FUNCTION
PARITY OR RECOVERABLE ERROR
UNRECOVERABLE ERROR
```

fd identifies the assigned file or device on which the error occurred.

LINE xxx ADDR xxxxx STACK OVERFLOW TASK PAUSED

The program does not have sufficient memory to process the input file. Reload the program with a larger segment size increment.

LIST DEVICE ERROR

The specified device or file is syntactically incorrect or failed to assign.

MACRO FILE filename.MAC DOES NOT EXIST, SYSGEN ABORTED

This is a sysgen CSS message output in macro expansion phase indicating that the filename of the macro output file does not exist.

MACRO LIBRARY filename.MLB NOT FOUND, SYSGEN ABORTED

This is a sysgen CSS message indicating a required macro library was not found.

MAX. NUMBER OF TCOMS EXCEEDED

Too many TCOMs were specified.

MIRROR DISK ERROR

Less than two direct access devices were included in the system.

MISSING ARGUMENT

A required argument was omitted.

MODULE NOT INCLUDED: module name

The module identified by module name was not included in the system. There might be a combination of invalid option settings.

NO DEVICES

No devices were specified in the input file.

NO DIRECT ACCESS SUPPORT

DIRECTORY, ROLL, SPOOL, TEMP or VOLUME statements were entered, but no disk devices were specified.

OBJECT LIB filename NOT FOUND, OS LINKING ABORTED

This is a sysgen CSS message indicating that a required system or driver object library was not found.

OS32MT rr-uu.vvvvvvvv

This is the 8-character operating system version with revision and update numbers displayed on the system console at system initialization.

OUTPUT FILE ERROR: xxx

The output file already exists, or the fd is incorrect.

OUTPUT MUST BE ENTERED

The output file was not specified.

READY FOR CONFIGURATION INPUT

The program is waiting for the user to interactively enter configuration input statements to be used to create the configuration input file.

READY FOR EDIT COMMANDS

The editor was invoked via the EDIT command, and the system is waiting for edit commands.

READY FOR SYSGEN COMMANDS

The OS/32 EDIT END or DONE command ended the edit session and control is returned to the sysgen program. The Sysgen/32 program is waiting for commands to resume processing the configuration input file.

SEQUENCE ERROR

The statement preceding the message was entered more than once, or a corresponding required statement was omitted.

The ILEVEL or IOCLASS statement was entered after all devices were processed; i.e., after the ENDD statement was read.

SPECIFIED FILE filename DOES NOT EXIST, LINK OF OS ABORTED

This is a sysgen CSS message indicating that a file required in the Link procedure cannot be found.

STATEMENT NOT RECOGNIZED

A statement in the input file is not recognized by the program.

SYSGEN32 ERROR, SYSGEN32 ABORTED

An error in running the Sysgen/32 program was detected. The sysgen procedure is aborted.

xxxx ERROR

The characters replacing xxxx specify ROLL, SPOOL, SPL32, TEMP, VOLUME, MEMORY, MODULE, FLOAT, ILEVEL, ERROR RECORDING, ACCOUNTING, NAME, ADDRESS, DCOD, SELCH or COORDINATION.

Invalid syntax was detected in the volume.

More than 16 pseudo devices were specified (SPOOL), or spool support was requested but no pseudo devices were found.

The memory specified exceeds the allowed range for the requested central processing unit (CPU).

A MODULE statement was entered with a name that could not be recognized by the program, or the MODULE name was syntactically incorrect.

Invalid floating point options or combinations were entered: no single precision floating point (SPFP), but hardware double precision floating point (DPFP), or, for the Perkin-Elmer Series 3200 CPUs, the options were H,S or S,H.

The SELCH, controller and connected devices have different ILEVELS.

A syntax error was detected in the name field of a device statement, the same name was entered twice, or a blank device name was found.

The device address was invalid, greater than the maximum physical address specified, or not 0 for spool devices. The device address was 0, indicating a spool device, but the device code was not 0 or one.

The specified device code was invalid, not a number from 16 to 254, or not 0 or 1 for spool devices.

More than one device specified the same controller with different SELCH specifications.

APPENDIX C
SYSTEM GENERATION/32 (SYSGEN/32) COMMAND AND STATEMENT SUMMARY

ACCOUNTING [=] [{ n / 4 }] [,NOFILEACCOUNTING]

BACKGROUND [=] [{ maxpriority / 16 }] [, { maxsize / 9 }]

CLOCK [=] [{ 50 / 60 }] [, { pic addr / 6C }] [, { lfc addr / 6D }] [,D]

CMDLEN [=] [{ n / 80 }]

CONVERSATIONAL

COORDINATION [=] [{ SELCH= { S_1 / (S_1, S_2, S_n) } / DEVICE= { $name_1$ / $(name_1, \ldots, name_n)$ } }] , TRANSFER=n

COPY

[$line_1$ . . . $line_n$]

ENDCOPY

CPU [=] [{ 3250 / 3240 / 3230 / 3220 / 3210 / 3200MPS / 3205 / 8/32 / 7/32 }] [, { 2 / 8 }]

CSS [=] [{ n / 5 }]

DATE [=] [{ DDMMYY / MMDDYY }]

DEVADS [=] [{ 3 / 1 / 0 }]

DEVICES

dev name:,dev address,dev dcode [,CLOCK={XA|XB|XC|XD}] [,CM=n]

[,CONSOLE] [,CONTROLLER=n] [,DISC] [,DUAL] [,EOV] [,ILEVEL=n]

[,INTIMER={n|10}] [,IOCLASS={n|global I/O class}] [,IOLIMIT={n|10}]

[,LEADCOUNT=n] [,LINESTATUS=n] [,MAXFRAMES=n]

[,MAXWRITEBUFF={n|1024}] [,MINREADBUFF={n|64}] [,MTO=n] [,N2={n|10}]

[,NCS={n|&SES}] [,NODISC] [,NONSHARED] [,OUTIMER={n|10}]

[,PADCOUNT=n] [,POLLDELAY=n] [,POLLIMIT=n] [,POLLTIME={n|3}]

[,QUEUE=name] [,READCONTROL=n] [,RECLEN=n]

[,RESPONSETIME={n|7FFF,3}] [,SCREENTIME=n] [,SELCH=n] [,SIZE=n]

[,SPINDLE=n] [,SSA=name] [,TO2={n|30}] [,TRANSLATE=name]

[,USCI=n] [,USCO=n] [,USER=(uparm[=parameter])]

[,VIRT=(startcyl,ncyl)] [,WAKEUP={n|30}] [,WRITECONTROL=n]

[,XDCOD=n]

.
.
.

ENDD

DISCBLOCK [=] [{ {n / 4} } [,MXBLKSZ= {n / 4}]]

[,INDEX= {bsize / 1} / {isize / 1}]

[,SPOOL= {bsize / 1} / {isize / 1}]

[,NONBUF= {bsize / 64} / {isize / 3}]

DSYS [=] [{n / 25}]

EDIT

END

ENDC

ERRORREC [=] fd,size,period

FLOAT = [{ S,S / S,H / H,H / N,N }]

HELP [{name / *}]

ILEVEL [=] [{ 3 / 2 / 1 / 0 }]

INPUT fd

INTERCEPT

IOCLASS [=] [{ cc / 0 / 1 / 2 / 3 }]

IREADER

ITAM

JOURNAL [=] [{ n / 0 }]

LOGLEN [=] [{ n / 72 }]

LPU [{ n / MAXAPU value }] [, { w / 1 }]

```
MAXAPU [ {n} ]
       [ {9} ]

[APU1 , contr1]
[APU2 , contr2]
[     .       ]
[     .       ]
[     .       ]
[APUn , contrn]
ENDAPU
```

```
MAXTASK [=] [ {n}  ]
            [ {32} ]
```

```
MCONFIG  BLOCK=nn,START=xx,RANGE=yy [,INTERL={2}]
                                       {4}
                                       {0}
         [,SHARED={NORECORD}]
                  {NVRECORD}
                  {RECORD  }
```

```
MEMCHECK
```

```
MEMORY [=] [ {n}   ]
           [ {256} ]
```

```
MODULE

  [new module name1]
  [new module name2]
  [       .        ]
  [       .        ]
  [       .        ]
  [new module namen]

ENDM
```

```
NOSEG
```

```
POWERFAIL [=] [ {n} ] [, {AUTO  } ]
              [ {5} ]    {MANUAL}
```

OUTPUT fd

PAUSE

PROCESS

QUEUE [=] [{n | total number of devices}]

ROLL [=] [rvoln]

SLICE [=][n]

SPOOL [=] [spvoln]

SPL32

SSTABLE [=] [{n | 32}]

START [,INPUT=fd_1] [,OUTPUT=fd_2] [,COMMAND=fd_3] [,LIST=fd_4]

[STARTUP
.
.
.
ENDS]

TCOM [=] $name_1$,$address_1$,$size_1$ [/.../$name_n$,$address_n$,$size_n$]

TEMP [=] [tvoln]

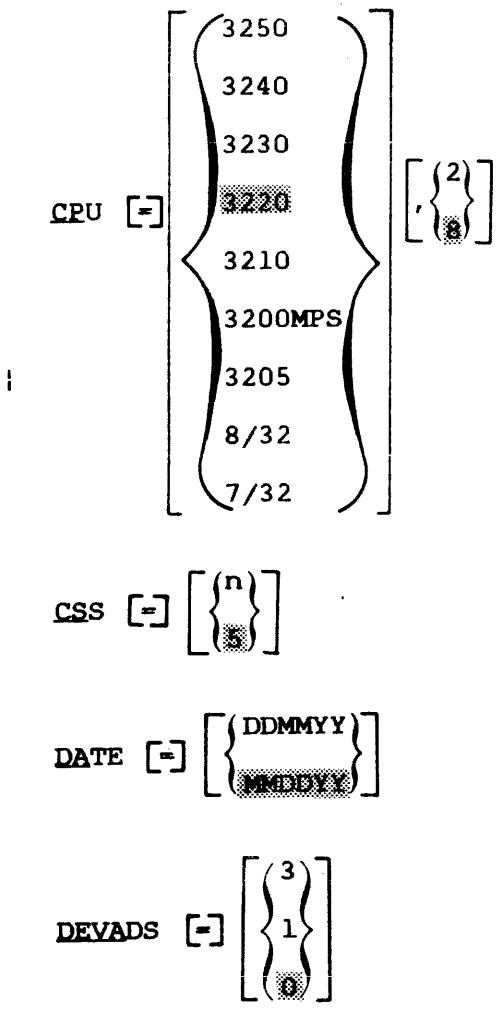
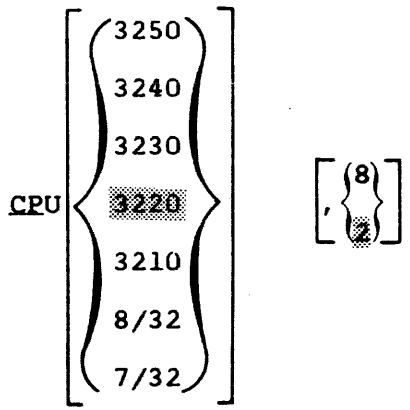
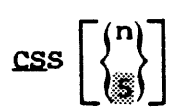
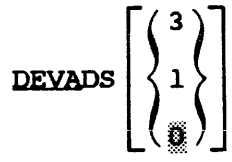
TGD

VERSION [=] [vvvvvvvv]

VOLUME [=] [voln]

APPENDIX D
SYSTEM GENERATION/32 (SYSGEN/32) AND OS/32 CONFIGURATION UTILITY PROGRAM (CUP) STATEMENT COMPARISONS

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
ACCOUNTING [=] [{n | 4}] [,NOFILEACCOUNTING]	ACCOUNTING [{nn | 4}]
BACKGROUND [=] [{maxpriority | 16}] [,{maxsize | 9}]	BACKGROUND [{maxpriority | 16}] [,{maxsize | 9}]
CLOCK [=] [{50 | 60}] [,{pic addr | 6C}] [,{lfc addr | 6D}] [,D]	CLOCK [{50 | 60}] ,[{pic addr | X'6C'}] ,[{lfc addr | X'6D'}] [,D]
CMDLEN [=] [{n | 80}]	CMDLEN [{n | 80}]
COORDINATION [=] [{SELCH= S_1 | SELCH= (S_1, S_2, S_n) | DEVICE= $name_1$ | DEVICE= $(name_1, \ldots, name_n)$}] , TRANSFER=n	
COPY [$line_1$ . . . $line_n$] ENDCOPY	

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
	
	
	
	

SYSGEN/32 STATEMENTS

DEVICES

```
dev name:,dev address,dev dcode [,CLOCK={XA|XB|XC|XD}] [,CM=n]
    [,CONSOLE] [,CONTROLLER=n] [,DISC] [,DUAL] [,EOV] [,ILEVEL=n]
    [,INTIMER={n|10}] [,IOCLASS={n|global I/O class}] [,IOLIMIT={n|10}]
    [,LEADCOUNT=n] [,LINESTATUS=n] [,MAXFRAMES=n]
    [,MAXWRITEBUFF={n|1024}] [,MINREADBUFF={n|64}] [,MTO=n] [,N2={n|10}]
    [,NCS={n|FFFF}] [,NODISC] [,NONSHARED] [,OUTIMER={n|10}]
    [,PADCOUNT=n] [,POLLDELAY=n] [,POLLIMIT=n] [,POLLTIME={n|3}]
    [,QUEUE=name] [,READCONTROL=n] [,RECLEN=n]
    [,RESPONSETIME={n|7FFF, 3}] [,SCREENTIME=n] [,SELCH=n] [,SIZE=n]
    [,SPINDLE=n] [,SSA=name] [,TO2={n|30}] [,TRANSLATE=name]
    [,USCI=n] [,USCO=n] [,USER=(uparm[=parameter])]
    [,VIRT=(startcyl,ncyl)] [,WAKEUP={n|30}] [,WRITECONTROL=n]
    [,XDCOD=n]
.
.
.
ENDD
```

OS/32 CUP STATEMENTS

DEVICES

```
{level|*} [dmnem]:dnum,dcod,{C|D|E|S} [,{xdcod|Xxdcod}] [,{recl|132}]
    [,{size|66}]
{level|*} [dmenm]:dnum,dcod,{C|D|E|S} [,{xdcod|Xxdcod}] [,{recl|132}]
    [,{size|66}]
.
.
.
ENDD
```

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
DIRECTORY	DIRECTORY
DISCBLOCK [=] [{ n / 4 } [, MXBLKSZ= { n / 4 }]]	DISCBLOCK [{ n / 4 }]
[, INDEX= { bsize / 1 } / { isize / 1 }]	
[, SPOOL= { bsize / 1 } / { isize / 1 }]	
[, NONBUF= { bsize / 64 } / { isize / 3 }]	
DSYS [=] [{ n / 25 }]	DSYS [{ n / 25 }]

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
ENDC	ENDC
ERRORREC [=] fd,size,period	ERRORREC fd,size,period
FLOAT = [{ S,S / S,H / H,H / N,N }]	FLOAT [{ S,S / S,H / H,H / N,N }]
HELP [{ name / * }]	
ILEVEL [=] [{ 3 / 2 / 1 / 0 }]	ILEVEL [{ 3 / 2 / 1 / 0 }]

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
INTERCEPT	
IOCLASS [=] [{ cc / 0 / 1 / 2 / 3 }]	IOCLASS { cc / 0 / 1 / 2 / 3 }
IREADER	
ITAM	ITAM
JOURNAL [=] [{ n / 0 }]	JOURNAL [{ n / 0 }]
LOGLEN [=] [{ n / 72 }]	LOGLEN [{ n / 72 }]

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
LPU [{n / MAXAPU value}] [, {w / 1}]	
MAXAPU [{n / 9}]	
[APU$_1$, contr$_1$ APU$_2$, contr$_2$ . . . APU$_n$, contr$_n$] ENDAPU	
MAXTASK [=] [{n / 32}]	MAXTASK [{n / 32}]
MCONFIG BLOCK=nn, START=xx, RANGE=yy [, INTERL={2 / 4 / 0}] [, SHARED={NORECORD / NVRECORD / RECORD}]	
MEMCHECK	MEMCHECK

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
MEMORY [=] [{ n / 256 }]	MEMORY [{ n / 128 }]
MODULE [new module name$_1$ / new module name$_2$ / . / . / . / new module name$_n$]	MODULE [new module name / new module name / . / . / . / new module name]
ENDM	ENDM
NOSEG	NOSEG
	NOTGD
POWERFAIL [=] [{ n / 5 }] [, { AUTO / MANUAL }]	
QUEUE [=] [{ n / total number of devices }]	QUEUE [{ n / total number of devices }]

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
ROLL [=] [rvoln]	ROLL [rvoln]
SLICE [=] [n]	
SPOOL [=] [spvoln]	SPOOL [spvoln]
	SPOOL
SPL32	
SSTABLE [=] [{n | 32}]	SSTABLE [{n | 32}]
[STARTUP . . . ENDS]	[STARTUP . . . ENDS]
TCOM [=] $name_1$,$address_1$,$size_1$ [/.../$name_n$,$address_n$,$size_n$]	TCOM $name_1$,$address_1$,$size_1$ [/.../$name_n$,$address_n$,$size_n$]

SYSGEN/32 STATEMENTS	OS/32 CUP STATEMENTS
TEMP [=] [tvoln]	TEMP [tvoln]
TGD	
VERSION [=] [vvvvvvvv]	VERSION vvvvvvvv
VOLUME [=] [voln]	VOLUME voln

INDEX

PERKIN-ELMER
Data Systems Group

D O C U M E N T A T I O N C H A N G E N O T I C E

The purpose of this documentation change notice (DCN) is to provide a quick and efficient way of making changes to technical manuals before they are formally updated or revised.

The manual affected by these changes is:

48-037 F00 R02	SYSTEM GENERATION/32 (SYSGEN/32) Reference Manual

- Page 4-2 and 4-9

The local device codes for the CDD50 disks are stated incorrectly. In Table 4-1 on page 4-2, and in Table 4-2 on page 4-9, the device codes are reversed.

The Tables should read:

LOCAL DEVICE CODE	DEVICE TYPE
42	50Mb CDD50, removable portion (25Mb)
43	50Mb CDD50, fixed portion (25Mb)

PERKIN-ELMER

PUBLICATION COMMENT FORM

We try to make our publications easy to understand and free of errors. Our users are an integral source of information for improving future revisions. Please use this postage paid form to send us comments, corrections, suggestions, *etc.*

1. Publication number_____

2. Title of publication_____

3. Describe, providing page numbers, any technical errors you found. Attach additional sheet if neccessary.

   _____

   _____

   _____

4. Was the publication easy to understand? If no, why not?

   _____

5. Were illustrations adequate? _____

   _____

   _____

6. What additions or deletions would you suggest? _____

   _____

   _____

7. Other comments: _____

   _____

   _____

From _____ Date _____

Position/Title _____

Company _____

Address _____

_____

_____

6417

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Data Systems Group
106 Apple Street
Tinton Falls, NJ 07724

ATTN:
TECHNICAL SYSTEMS PUBLICATIONS DEPT.

FOLD

FOLD

STAPLE

STAPLE

6433-1