



---

**MP-60  
COMPUTER SYSTEM**

**PRELIB/PRELIBOS  
REFERENCE MANUAL**

---

**REVISION RECORD  
( Standard Product Manuals )**

Revision	Description	Change Authority
---	Preliminary issue.	
4-04-74		
A	Final issue.	
2-15-75		
B	Manual revised; includes Engineering Change	
12-01-76	Order CB34611	
	Pages revised: Revision Record, v-vii, 1-1,	
	1-2, 2-2, 2-4, 2-5, 3-1, 3-3 - 3-9, 4-7 - 4-9,	
	5-1, 5-3 - 5-8, 6-4, I-1, and I-2. This edition	
	obsoletes all previous editions.	
C	Manual revised; includes Engineering Change Order	
7-31-77	CB37288. Pages revised: Cover, Title page,	
	Revision Record, v, 3-3, and Comment Sheet.	
D	Manual revised; includes Engineering Change	
2-01-79	Order CB42039. Pages revised: 4-2, 4-3, 4-6,	
	and 4-7.	
E	Manual revised and reprinted. This edition	ECO CB43152
2-15-79	obsoletes all previous editions.	

©1974, 1975, 1976, 1977, 1979  
by Control Data Corporation  
Printed in the United States of America

AA 6802 12/78

**REVISION LETTERS I, O, Q AND X ARE NOT USED**

Use COMMENT SHEET in the back of this manual

## LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Rev.	Page	Rev.	Page	Rev.	Page	Rev.
Cover	E						
Title Page	E						
ii	E						
iii/iv	E						
v/vi	E						
vii	E						
viii	E						
ix	E						
x	E						
1-1 and 1-2	E						
2-1 thru 2-6	E						
3-1 thru 3-10	E						
4-1	A						
4-2 thru 4-10	E						
5-1 thru 5-8	E						
6-1	A						
6-2	A						
6-3	A						
6-4	E						
6-5/6-6	E						
I-1 and I-2	E						
Comment Sheet	E						



## PREFACE

---

This reference manual describes the PREpare LIBrary (PRELIB) program for maintaining and updating MPX Operating System resident and library routines. The user's familiarity with the MPX Operating System is assumed.



# CONTENTS

1	INTRODUCTION .....	1-1
2	PRELIB FLOW .....	2-1
	PRELIB Control .....	2-1
	MICR Module .....	2-1
	BOOT Module .....	2-3
	MPX Module .....	2-3
	LIB Module .....	2-3
	ABS Module .....	2-4
	DUMPSYS Module .....	2-4
	CNTR Module .....	2-4
	DIAG Module .....	2-5
	CPUD Module .....	2-5
3	PRELIB CONTROL CARDS .....	3-1
	*ABS (NABS = lun) .....	3-1
	*ALT (ALT=lun) .....	3-2
	*AMICR (ID=id, FWA=fwa) or *AMICR (NH, FWA=fwa) .....	3-2
	*BOOT (FWA=fwa) .....	3-2
	*CNTR0, *CNTR1, *CNTR2, *CNTR3 .....	3-3
	*CPUD .....	3-3
	*DELETE (ID=id) .....	3-3
	*DIAG0, *DIAG1, *DIAG2, *DIAG3 .....	3-4
	*DIRCT (DEPT=ept1, ept2, ... eptn) .....	3-4
	*DIRCT (ABSL=PROGNAME, ALT=lu) .....	3-4
	*DUMPSYS (NSYS=lun1, DMPX=lun2, DLIB=lun3, MICR=lun4, BOOT=lun5, CNTR0=lun6, ..., DIAG0=lun10, ..., CPUD=lun14, ID=id) .....	3-4
	*ENDPLIB .....	3-6
	*ENDT .....	3-6
	*INSERT (ID=id) .....	3-6
	*JLDR(n) .....	3-6
	*JMTR(n) .....	3-7
	*LIB (OLIB=lun1, NLIB=lun2) .....	3-7
	*MICR .....	3-8
	*MPX (NMPX=lun) .....	3-8
	*ORIGIN (ID=id, FWA=fwa) .....	3-8
	*REPLACE (ID=id) .....	3-9
	*TEXT (ID=id) .....	3-9

## CONTENTS (CONT.)

4	PRELIB PROCESS DECK STRUCTURE .....	4-1
	MPX Control Cards .....	4-1
	*PRELIB .....	4-1
	Sample Deck Structures .....	4-2
	Generate New MPX System Tape .....	4-2
	Restore MPX System .....	4-3
	Update MPX Library .....	4-4
	Build New MPX Resident .....	4-4
	Generate New MP-60 Emulator and Copy MPX Boot Program From MPX System Tape .....	4-5
	Assemble and Add New Program to MPX Executive Routines .....	4-5
	Restore MPX System Using Alternate Input Unit .....	4-6
	Build MPX Installation Tape .....	4-7
	Generate New IOC Microcode .....	4-7
	Generate New IOC0 and IOC1 Microcode .....	4-8
	Generate New MPX System Tape .....	4-8
	Generate New DIAG0 Microcode .....	4-9
5	FILES AND FORMATS .....	5-1
	MPX System Device .....	5-1
	Absolute Header Records .....	5-3
	MPX Resident .....	5-3
	MPX Library .....	5-4
	MPX System Tape .....	5-5
	PRELIB Scratch File .....	5-7
6	PRELIB DIAGNOSTICS .....	6-1
	INDEX .....	I-1



# ILLUSTRATIONS

---

Figure		Page
1-1	PRELIB Functions .....	1-2
2-1	PRELIB Flow .....	2-2
5-1	MPX System Device Layout and Logical Sector Addresses .....	5-1
5-2	MPX Library File .....	5-4
5-3	MPX Sample System Tape (No Optional Fields) .....	5-5
5-4	MPX Sample System Tape (Two IOC, Two DIAG, and One CPUD Programs) .....	5-6

# TABLES

---

Table		Page
6-1	PRELIB Diagnostics .....	6-1

# INTRODUCTION

1

---

The PREpare LIBrary (PRELIB) program is an MPX library program which performs the following functions (Figure 1-1):

- 1) Builds an absolute record of the MP-60 emulator program and copies it to the MPX system device
- 2) Builds an absolute record of the MPX boot program and copies it to the MPX system device
- 3) Builds an absolute record of the MPX resident routines and copies it to a user-specified file
- 4) Creates a new library or updates an existing library of relocatable programs, writing the new library to a user-specified file
- 5) Builds absolute records of microcode and/or MP-60 relocatable programs and copies these to a user-specified unit
- 6) Builds an MPX system tape which can be used for installing the MPX operating system
- 7) Builds an absolute record of an IOC microcode program and copies it to the MPX system device
- 8) Builds an absolute record of the diagnostic microcode program for an IOC and copies it to the MPX system device
- 9) Builds an absolute record of the CPU diagnostic microcode program and copies it to the MPX system device.

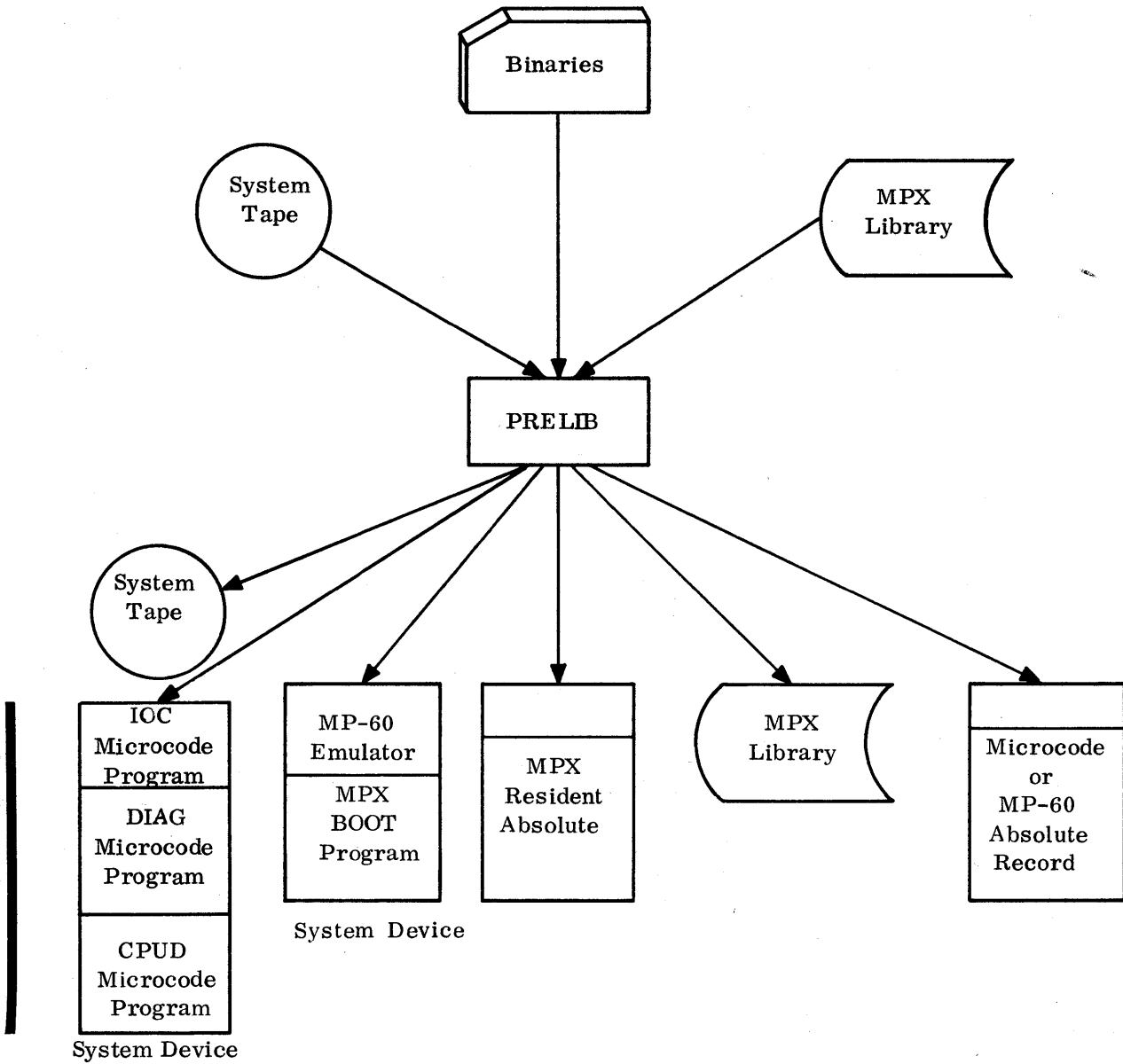


Figure 1-1. PRELIB Functions

---

The PRELIB program is separated into 11 modules, one module for each of the 10 functions of PRELIB plus a control module (Figure 2-1).

## PRELIB CONTROL

PRELIB Control receives control immediately after PRELIB is loaded. The basic function of PRELIB Control is to read control cards from the standard input unit and pass control to the proper module for further processing. All modules return to PRELIB Control when they are through processing. PRELIB Control recognizes 17 control cards:

- \*MICR - Passes control to the MICR module
- \*CNTR0 - Passes control to the CNTR module
- \*CNTR1 - Passes control to the CNTR module
- \*CNTR2 - Passes control to the CNTR module
- \*CNTR3 - Passes control to the CNTR module
- \*DIAG0 - Passes control to the DIAG module
- \*DIAG1 - Passes control to the DIAG module
- \*DIAG2 - Passes control to the DIAG module
- \*DIAG3 - Passes control to the DIAG module
- \*CPUD - Passes control to the CPUD module
- \*BOOT - Passes control to the BOOT module
- \*MPX - Passes control to the MPX module
- \*LIB - Passes control to the LIB module
- \*ABS - Passes control to the ABS module
- \*DUMPSYS - Passes control to the DUMPSYS module
- \*ALT - Specifies an alternate file/unit for input to PRELIB
- \*ENDPLIB - Completes processing and exits back to the MPX operating system

## MICR MODULE

The MICR module of PRELIB reads a microcode absolute binary deck from the input unit and builds an absolute record in core. If no error occurs, the MICR module writes the absolute record to the area of the MPX system device reserved for the MP-60 emulator. Subsequent deadstarting of the system loads this program as the MP-60 emulator.

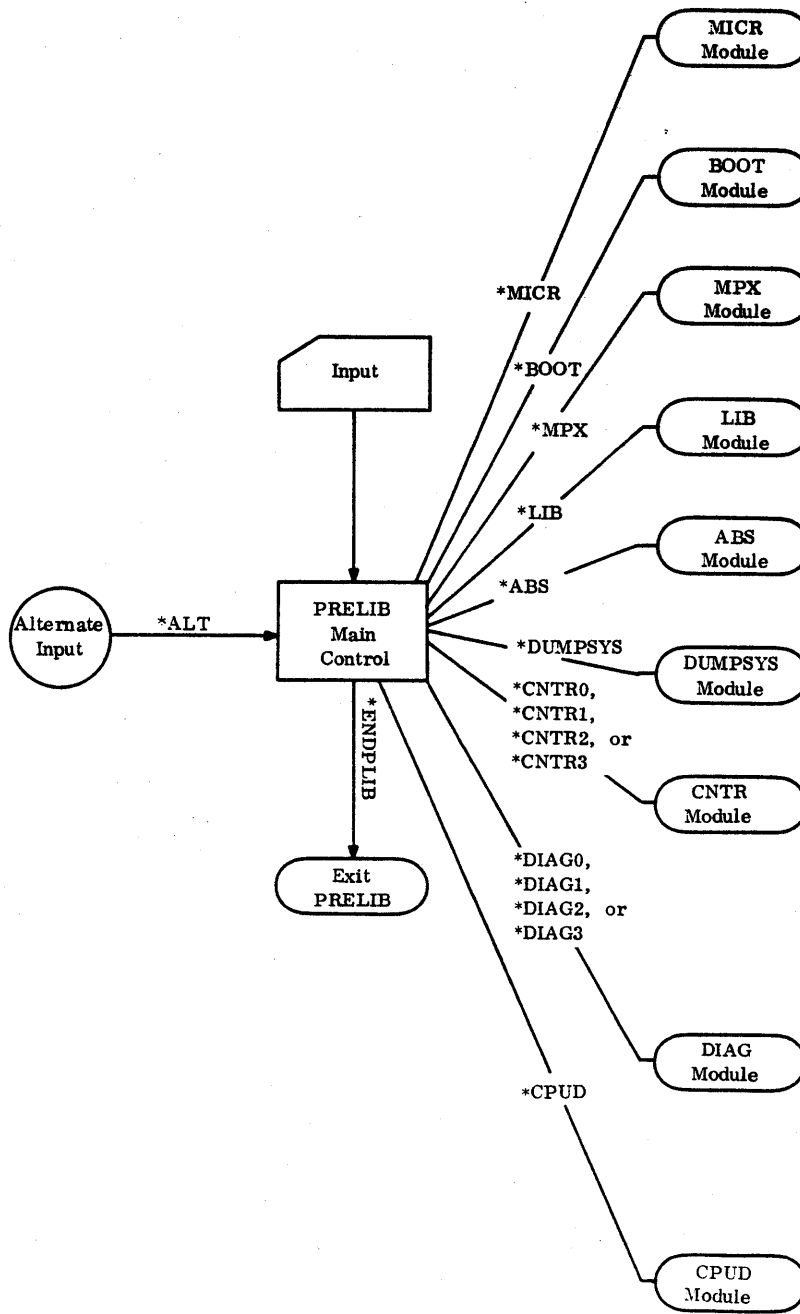


Figure 2-1. PRELIB Flow

The MICR module recognizes one control card:

- \*ALT - Specifies an alternate file/unit for input to MICR module

## **BOOT MODULE**

The BOOT module of PRELIB reads an MP-60 relocatable binary deck from the input unit and builds an absolute record in core. If no error occurs, the BOOT module writes the absolute record to the area of the MPX system device reserved for the MPX boot program. Subsequent autoloading of the system loads this program as the MPX boot program. The BOOT module recognizes one control card:

- \*ALT - Specifies an alternate file/unit for input to BOOT module

## **MPX MODULE**

The MPX module of PRELIB reads MP-60 relocatable binary decks and/or control cards from the input unit and builds an absolute record of the MPX resident routines. If no error occurs, the MPX module writes the absolute record to a user-specified file. The MPX module recognizes three control cards:

- \*JMTR - Indicates following relocatable binary decks are the job monitor programs
- \*JLDR - Indicates following relocatable binary decks are the job loader programs
- \*ALT - Specifies an alternate file/unit for input to MPX module

## **LIB MODULE**

The LIB module of PRELIB reads MP-60 relocatable binary decks, control cards, and/or source cards from the input unit and modifies an existing MPX library or creates a new MPX library. If no error occurs, the LIB module writes a new MPX library on a user-specified file. The LIB module recognizes seven control cards:

- \*REPLACE - Replaces a program on the existing library with a new program
- \*DELETE - Deletes a program from the existing library
- \*INSERT - Inserts a new program after a specified program on the existing library
- \*DIRCT - Builds a loader directory card on the new library with specified entry points

- \*TEXT - Copies source cards from the input unit to the new library
- \*ENDT - Stops copying source cards from the input unit to the new library
- \*ALT - Specifies an alternate file/unit for input to the LIB module

## ABS MODULE

The ABS module of PRELIB reads microcode absolute binary programs, MP-60 relocatable binary programs, and/or control cards from the input unit and builds absolute records, copying these absolute records to a user-specified unit. The ABS module recognizes three control cards:

- \*ORIGIN - Builds an absolute record of MP-60 relocatable binary decks which follow this card on the input unit
- \*AMICR - Builds an absolute record of the microcode absolute deck which follows this card on the input unit
- \*ALT - Specifies an alternate file/unit for input to the ABS module

## DUMPSYS MODULE

The DUMPSYS module of PRELIB writes a new MPX system tape on a user-specified unit. The DUMPSYS module recognizes no control cards.

## CNTR MODULE

The CNTR module of PRELIB reads a microcode absolute binary deck from the input unit and builds an absolute record in core. If no error occurs, the CNTR module writes the absolute record to the area of the disk reserved for the respective IOC firmware. Subsequent deadstarting of the IOC loads this program as the IOC firmware. The maximum number of IOCs a system can configure is four - IOC0, IOC1, IOC2, and IOC3.

The references to IOC denote an I/O controller, which can be an MPP configured as a mass storage controller or a unit record controller. The CNTR module recognizes one control card:

- \*ALT - Specifies an alternate file/unit for input to the module



## DIAG MODULE

The DIAG module of PRELIB reads a microcode absolute binary deck from the input unit and builds an absolute record in core. If no error occurs, the DIAG module writes the absolute record to the area of the disk reserved for the respective DIAG firmware. This provides the option of deadstarting an IOC with the diagnostic program.

When PRELIB encounters any of the control cards, \*DIAG0, \*DIAG1, \*DIAG2, or \*DIAG3, it passes control to the DIAG module. The DIAG module recognizes one control card:

- \*ALT - Specifies an alternate file/unit for input to the module

## CPUD MODULE

The CPUD module of PRELIB reads a microcode absolute binary deck from the input unit and builds an absolute record in core. If no error occurs, the CPUD module writes the absolute record to the area of the disk reserved for the CPUD firmware. This provides the option of deadstarting the CPU with the diagnostic program.

When PRELIB encounters this control card, it passes control to the CPUD module. This module recognizes one control card:

- \*ALT - Specifies an alternate file/unit for input to the module



## PRELIB CONTROL CARDS

3

---

All PRELIB control cards have the following format:

\*CCID (P<sub>1</sub>, P<sub>2</sub>, ..., P<sub>n</sub>)

PRELIB recognizes the following control card identifiers (CCID):

- |          |           |        |
|----------|-----------|--------|
| • ABS    | • DIAG3   | • TEXT |
| • ALT    | • DIRCT   |        |
| • AMICR  | • DUMPSYS |        |
| • BOOT   | • ENDPLIB |        |
| • CNTR0  | • ENDT    |        |
| • CNTR1  | • INSERT  |        |
| • CNTR2  | • JMTR    |        |
| • CNTR3  | • JLDR    |        |
| • CPUD   | • LIB     |        |
| • DELETE | • MICR    |        |
| • DIAG0  | • MPX     |        |
| • DIAG1  | • ORIGIN  |        |
| • DIAG2  | • REPLACE |        |

P<sub>1</sub>, P<sub>2</sub>, ..., P<sub>n</sub> are character strings separated by commas which define parameters for the particular control card identifier. All parameters are required unless specified as optional.

### \*ABS (NABS = lun)

The control card directs PRELIB Control to enter the ABS module, that is, build absolute records of microcode and/or MP-60 relocatable programs and copy these to a user-specified unit.

The term lun is one or two numeric characters identifying the logical unit on which the absolute records are to be written.

Following \*ABS on the input unit should be the control cards recognized by the ABS module with appropriate binary decks.

## **\*ALT (ALT=lun)**

This control card directs the currently operating module to read its input from an alternate unit. The alternate unit is read to end-of-file, end-of-allocated area, or an \*ALT control card. If end-of-file or end-of-allocated area is encountered, the standard input unit, INP, is read for the next control card. The term lun is one or two numeric characters identifying the logical unit from which alternate input is to be read.<sup>†</sup>

## **\*AMICR (ID=id, FWA=fwa) or \*AMICR (NH, FWA=fwa)**

This control card is recognized only by the ABS module of PRELIB. The \*AMICR control card directs the ABS module to build a microcode absolute record and output it to the unit specified on the \*ABS control card. If the NH parameter is specified, the absolute record is built without a header. If the NH parameter is not specified, then the absolute record is built with a header, specified by id.

The term id is one to eight alphanumeric characters identifying the absolute record.

The term fwa is one to four hex characters identifying the first word address of the absolute record. This parameter is strictly documentary and does not affect the loading of the microcode since it is already in absolute binary format.

Following \*AMICR on the input unit should be the absolute binary deck of the microcode program from which the absolute record is built or an \*ALT control card designating a unit from which the absolute binary microcode can be read.

## **\*BOOT (FWA=fwa)**

This control card directs PRELIB Control to enter the BOOT module, that is, build an absolute record of the MPX boot program and copy it to the MPX system device.

The term fwa is one to four hex characters identifying the first word address where the MPX boot program is to be relocated.

Following the \*BOOT control card on the input unit should be the MP-60 relocatable binary deck of the MPX boot program from which the absolute record is built, an \*ALT control card designating a unit from which the MP-60 relocatable binary deck can be read, or a PRELIB Control control card.<sup>††</sup>

<sup>†</sup> The alternate unit must be in MPX standard block format (that is 1920 characters per block).

<sup>††</sup> If a PRELIB Control control card follows the \*BOOT control card on the input unit, the MPX boot program is copied from the old MPX system tape, designated on the \*PRELIB control card, to the MPX system device.

## **\*CNTRO, \*CNTR1, \*CNTR2, \*CNTR3**

These control cards direct PRELIB Control to enter the CNTR module; that is, build an absolute record of the respective IOC microcode program and copy it to the MPX system device.

Following the \*CNTRn, n=0, 1, 2, 3 control cards, on the input unit, should be the microcode absolute binary deck of the respective IOC microcode program, an \*ALT control card designating an alternate unit from which the IOC microcode binary deck can be read, or a PRELIB Control control card. If a PRELIB Control control card follows the \*CNTR control card on the input unit, the IOC microcode program is copied from the old MPX system tape, designated on the \*PRELIB Control control card, to the MPX system device.

## **\*CPUD**

This control card directs PRELIB Control to enter the CPUD module; that is, build an absolute record of the CPUD microcode program and copy it to the system device.

Following the \*CPUD control card on the input unit should be the microcode absolute binary deck of the CPUD microcode program, an \*ALT control card designating an alternate unit from which the CPUD microcode binary deck can be read, or a PRELIB Control control card. If a PRELIB Control control card follows the \*CPUD control card on the input unit, the CPUD microcode program is copied from the old MPX system tape, designated on the \*PRELIB Control control card, to a predefined system device.

## **\*DELETE (ID=id)**

This control card is recognized only by the LIB module of PRELIB. The \*DELETE control card directs the LIB module to copy the old library to new library until program identifier id is encountered. The old library is then positioned after the program with identifier id.

The term id is one to eight alphanumeric characters identifying the program to be deleted from the old library. This identifier is the same as the identifier which appears on the IDC card in the MP-60 relocatable binary deck of this program.

Following the \*DELETE control card can be another LIB module control card, a PRELIB Control control card, or the MP-60 relocatable binary deck of programs to be copied on the new library.

## **\*DIAG0, \*DIAG1, \*DIAG2, \*DIAG3**

This control card directs PRELIB Control to enter the DIAG module; that is, build an absolute record of the diagnostic microcode program for the respective IOC and copy it to the MPX system device.

Following the \*DIAG control cards, on the input unit, should be the microcode absolute binary deck of the DIAG microcode program, an \*ALT control card designating an alternate unit from which the DIAG microcode binary deck can be read, or a PRELIB Control control card. If a PRELIB Control control card follows the \*DIAG control cards on the input unit, the respective DIAG microcode program is copied from the old MPX system tape, designated on the \*PRELIB Control control card, to the MPX system device.

## **\*DIRCT (DEPT=ept1,ept2,....,eptn)**

This control card is recognized only by the LIB module of PRELIB. The \*DIRCT control card directs the LIB module to build a loader directory header (LDC) containing nonprotected entry points (ept<sub>1</sub>, ept<sub>2</sub>, . . . , ept<sub>n</sub>). The loader uses these entry points in the LDC at load time for satisfying externals.

The term ept is one to eight alphanumeric characters identifying entry points which are to be entered in the LDC. A maximum of 13 entry points is allowed. All entry points must be contained within EPT cards of the MP-60 relocatable binary deck which is being copied to the new library. If no \*DIRCT card is on input, the LIB module enters the entry points (maximum of 13) from the EPT card of the MP-60 relocatable binary deck which is being copied.

Following the \*DIRCT control card on the input unit should be the MP-60 relocatable binary deck being copied to the new library or an \*ALT control card designating a unit from which the MP-60 relocatable binary deck can be read.

## **\*DIRCT (ABSL=PROGNAME,ALT=lu)**

This alternate format of the DIRCT CONTROL CARD is for MPX/OS only, and is used when adding absolute modules to the library. This control card directs PRELIB to copy the formerly absolutized module named "PROGNAME" found on alternate logical unit "lu" to the library.

## **\*DUMPSYS (NSYS=|un1,DMPX=|un2,DLIB=|un3,MICR=|un4,BOOT=|un5, CNTRO=|un6,....,DIAG0=|un10,....,CPUD=|un14,ID=id)**

This control card directs PRELIB Control to enter the DUMPSYS module, that is, build an MPX system tape.

The term un1 is one or two numeric characters identifying the logical unit for the new MPX system tape.

The term lun2 is one or two numeric characters identifying the logical unit of the MPX resident file to be dumped to the new MPX system tape. The term lun2 must refer to a disk file or an old MPX system tape.

The term lun3 is one or two numeric characters identifying the logical unit of the MPX library file to be dumped to the new MPX system tape. The term lun3 must be a disk file or an old MPX system tape.

The term lun4 is one or two numeric characters identifying the logical unit from which the CPU firmware microcode program is dumped.

The term lun5 is one or two numeric characters identifying the logical unit from which the MPXBOOT program absolute code is dumped.

The terms lun6, lun7, lun8, and lun9 are one or two numeric characters identifying the logical units from which the microcode programs for I/O controllers 0, 1, 2, and 3, respectively, are dumped.

The terms lun 10, lun 11, lun 12, and lun 13 are one or two numeric characters identifying the logical units from which the diagnostic microcode programs for I/O controllers 0, 1, 2, and 3, respectively, are dumped.

The term lun14 is one or two numeric characters identifying the logical unit from which the CPU diagnostic microcode program is dumped.

The term id is one to eight alphanumeric characters identifying the new MPX system tape. These characters are put into the MPX system tape header.

The fields NSYS, DMPX, DLIB, and id are required. The fields MICR and BOOT may be ignored; in that case, default disk addresses will be used. Lun1, lun2, and lun3 must also be supplied. If any of the optional fields are omitted, the corresponding microcode program will not be dumped to the new system tape. If any of the optional logical units are not supplied, then a default disk address will be used.

The following files are copied to the new MPX system tape.

- 1) MP-60 emulator program is copied from the MPX system device or the logical unit number supplied.
- 2) MPX boot program is copied from the MPX system device or the logical unit number supplied.
- 3) MPX resident is copied from the MPX resident file specified by the DMPX parameter.
- 4) The library is copied from the library specified by the DLIB parameter.
- 5) The capability of dumping up to four IOC microcode programs, up to four diagnostic IOC microcode programs, and the CPU diagnostics program is provided (they are optional parameters).

Following the \*DUMPSYS control card on the input unit should be the next PRELIB control card.

### **\*ENDPLIB**

This control card directs PRELIB Control to end processing.

### **\*ENDT**

This control card is recognized only by the LIB module of PRELIB. This control card directs the LIB module to stop copying source text cards from the input unit to the new library.

### **\*INSERT (ID=id)**

This control card is recognized only by the LIB module of PRELIB. The \*INSERT control card directs the LIB module to copy the old library to new library through program identifier id.

The term id is one to eight alphanumeric characters identifying the program through which the old library is to be copied. This identifier is the same as the identifier which appears on the IDC card in the MP-60 relocatable binary deck of this program.

Following the \*INSERT control card can be another LIB module control card, a PRELIB Control control card, or MP-60 relocatable binary decks of programs which are to be copied onto the new library.

### **\*JLDR(n)**

This control card is recognized only by the MPX module of PRELIB. The \*JLDR control card indicates to the MPX module that the MP-60 relocatable binary decks following it belong to the job loader portion of MPX resident.

The term n is a numeric character (1 through 9) indicating the length in memory pages of the job loader. The term n is optional and, if omitted, n is assumed to be 1.

The \*JLDR control card must follow the MP-60 binary relocatable decks of the MPX resident program or job manager program on the input unit or an \*ALT control card designating a unit from which the MP-60 relocatable binary decks of the MPX resident program or job manager program were read.

Following the \*JLDR control card on the input unit must be the MP-60 binary relocatable decks of the job loader programs or an \*ALT control card designating a unit from which the MP-60 binary relocatable decks of the job loader programs can be read.



### **\*JMTR(n)**

This control card is recognized only by the MPX module of PRELIB. The \*JMTR control card indicates to the MPX module that the MP-60 relocatable binary decks following it belong to the job manager portion of MPX resident.

The term n is a numeric character (1 through 9) indicating the length in memory pages of the job manager. The term n is optional and, if omitted, n is assumed to be 1.

The \*JMTR control card must follow the MP-60 binary relocatable decks of the MPX resident program or job loader program on the input unit or an \*ALT control card designating a unit from which the MP-60 relocatable binary decks of the MPX resident program or job loader program were read.

Following the \*JMTR control card on the input unit must be the MP-60 binary relocatable decks of the job manager programs or an \*ALT control card designating a unit from which the MP-60 binary relocatable decks of the job manager programs can be read.

### **\*LIB (OLIB=lun1, NLIB=lun2)**

This control card directs PRELIB Control to enter the LIB module, that is, create a new library or update an existing library of relocatable programs, writing the new library to a user-specified file (refer to Section 5).

The term lun1 is one or two numeric characters identifying the logical unit, either a mass storage file or the old MPX system tape, containing the old library. If none is specified, it is assumed that a creation of a new library is requested rather than an update of an existing library.

The term lun2 is one or two numeric characters identifying the logical unit on which the new library is written. This parameter is required.

Following the \*LIB control card can be a LIB module control card, a PRELIB Control control card, or MP-60 relocatable binary decks of programs which are to be copied onto the new library. If the card following the \*LIB control card is a PRELIB Control control card, the old library, specified by the OLIB parameter on the \*LIB control card, is copied to the new library.

## **\*MICR**

The control card directs PRELIB Control to enter the MICR module, that is, build an absolute record of the MP-60 emulator program and copy it to the MPX system device.

Following the \*MICR control card on the input unit should be the microcode absolute binary deck of the MP-60 emulator program, an \*ALT control card designating an alternate unit from which the MP-60 emulator binary deck can be read or a PRELIB Control control card. If a PRELIB Control control card follows the \*MICR control card on the input unit, the MP-60 emulator program is copied from the old MPX system tape, designated on the \*PRELIB control card, to the MPX system device.

## **\*MPX (NMPX=lun )**

The control card directs PRELIB Control to enter the MPX module, that is, build an absolute record of the MPX resident routines and copy it to a user-specified unit (refer to Section 5).

The term lun is one or two numeric characters identifying the logical unit on which the absolute record of the MPX resident routines is to be written. This parameter is required.

Following the \*MPX control card can be the MP-60 relocatable binary decks of the MPX executive routines, an \*ALT control card designating an alternate unit from which the MP-60 relocatable binary decks of the MPX executive routines can be read, or a PRELIB Control control card. If a PRELIB Control control card follows the \*MPX control card on the input unit, the MPX resident absolute record is copied from the old MPX system tape, designated on the \*PRELIB control card, to the user-specified unit. If the MP-60 relocatable binary decks of the MPX executive routines or the \*ALT control card follows the \*MPX control card, then, following these on the input unit must also be the \*JMTR control card and the \*JLDR control card (only \*JMTR is needed if the job manager and job loader are combined into one relocatable binary deck).

## **\*ORIGIN (ID=id, FWA=fwa)**

This control card is recognized only by the ABS module of PRELIB. The \*ORIGIN control card directs the ABS module to build an MP-60 absolute record header and output it to the unit specified on the \*ABS control card.

The term id is one to eight alphanumeric characters identifying the absolute record.

The term fwa is one to four hex characters identifying the first word address of the absolute record.

Following \*ORIGIN on the input unit should be the MP-60 relocatable binary decks of the MP-60 program from which the absolute record is built, or an \*ALT control card designating a unit from which the MP-60 binary relocatable binary decks of the MP-60 program can be read.

### **\*REPLACE (ID=id)**

The control card is recognized only by the LIB module of PRELIB. The \*REPLACE control card directs the LIB module to copy the old library to new library until program identifier id is encountered. The old library is then positioned after the program with identifier id.

The term id is one to eight alphanumeric characters identifying the program to be replaced on the old library. This identifier is the same as the identifier which appears on the IDC card in the MP-60 relocatable binary deck of this program.

Following the \*REPLACE control card must be a \*DIRCT control card, the MP-60 relocatable binary deck of the program(s) which are to replace the program with identifier id, or an \*ALT control card designating a unit from which the relocatable binaries can be read.

### **\*TEXT (ID=id)**

This control card is recognized only by the LIB module of PRELIB. The \*TEXT control card directs the LIB module to copy source text cards from the input unit to the new library until an \*ENDT control card is encountered on the input unit.

The term id is one to eight alphanumeric characters identifying the source text following the \*TEXT control card.

Following the \*TEXT control card on the input unit must be the source text to be copied to the new library followed by an \*ENDT control card.



---

## MPX CONTROL CARDS

The PRELIB program is an MPX library program. It is loaded and executed whenever the MPX operating system encounters a \*PRELIB control card in an input deck. All files used by PRELIB must have been previously opened or equipped. Refer to the MPX reference manual for a discussion of the control cards necessary for opening and equipping files.

### \*PRELIB

PRELIB is loaded from the MPX library and executed whenever the MPX operating system encounters a \*PRELIB control card in an input deck. There are two forms for this control card:

\*PRELIB

\*PRELIB (OSYS=lun)

The term lun is a logical unit number of an old MPX system tape. This parameter is necessary only if subsequent processing requires it.

The PRELIB control cards and/or binaries which are required for the run follow the \*PRELIB card in the input deck.

\*JOB(...)

\*SCHED(...)

\*PRELIB

(PRELIB control deck)

\*EOJ

## SAMPLE DECK STRUCTURES

### GENERATE NEW MPX SYSTEM TAPE

The MPX library which is currently operating is copied to logical unit 20, deleting the program deck with identifier PROG1.

The MPX boot program is absolutized relative to first word address F000 and written to the MPX system device.

The MP-60 emulator program is absolutized and written to the MPX system device.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*OPEN(20,...)  
*OPEN(30,...)  
*PRELIB  
*LIB(OLIB=58,NLIB=20)  
*DELETE(ID=PROG1)  
*BOOT(FWA=F000)  
    (MPX boot program binary)  
*MICR  
    (MP-60 emulator program binary)  
*MPX(NMPX=30)  
    (MPX executive binaries)  
*JMTR  
    (MPX job monitor binaries)  
*JLDR  
    (MPX job loader binaries)  
*DUMPSYS(NSYS=10, DMPX=30, DLIB=20, MICR, BOOT, ID=MPXSTAPE)  
*ENDPLIB  
*EOJ
```

The MPX executive, job monitor, and job loader binaries are absolutized and written on logical unit 30 as the MPX resident.

The MP-60 emulator program, MPX boot program, MPX resident from logical unit 30, and MPX library from logical unit 20 are copied to the new MPX system tape on logical unit 10.

### RESTORE MPX SYSTEM

The MP-60 emulator is copied from the MPX system tape on logical unit 10 to the MPX system device.

The MPX boot program is copied from the MPX system tape on logical unit 10 to the MPX system device.

The MPX resident is copied from the MPX system tape on logical unit 10 to the new MPX resident file on logical unit 30.

The MPX library is copied from the MPX system tape on logical unit 10 to the new MPX library on logical unit 20.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*ALLOCATE(MPX-RESIDENT,...)  
*ALLOCATE(MPX-LIBRARY,...)  
*OPEN(30,MPX-RESIDENT,...)  
*OPEN(20,MPX-LIBRARY,...)  
*PRELIB(OSYS=10)  
*MICR  
*BOOT(FWA=F000)  
*MPX(NMPX=30)  
*LIB(OLIB=10,NLIB=20)  
*ENDPLIB  
*EOJ
```

## UPDATE MPX LIBRARY

The MPX library which is currently operating is copied to logical unit 10, deleting the program deck with identifier PROG1. The program deck with identifier PROG2 is copied to logical unit 10 with a directory entry having entry point PROG2EPT. The rest of the library is copied to logical unit 10.

```
*JOB(...)  
*SCHED(...)  
*OPEN(10,...)  
*PRELIB  
*LIB(OLIB=58,NLIB=10)  
*DELETE(ID=PROG1)  
*DIRECT(DEPT=PROG2EPT)  
    (PROG2 binary)  
*ENDPLIB  
*EOJ
```

## BUILD NEW MPX RESIDENT

The MPX executive, job monitor, and job loader binaries are absolutized and written on logical unit 10 as MPX resident.

```
*JOB(...)  
*SCHED(...)  
*ALLOCATE(MPX-RESIDENT,...)  
*OPEN(10,MPX-RESIDENT,...)  
*PRELIB  
*MPX(NMPX=10)  
    (MPX executive binaries)  
*JMTR  
    (Job monitor binaries)  
*JLDR  
    (Job loader binaries)  
*ENDPLIB  
*EOJ
```



## GENERATE NEW MP-60 EMULATOR AND COPY MPX BOOT PROGRAM FROM MPX SYSTEM TAPE

The MP-60 emulator program is absolutized and written to the MPX system device. The MPX boot program is copied from the MPX system tape on logical unit 10 to the MPX system device.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*PRELIB(OSYS=10)  
*MICR  
    (MP-60 emulator binary)  
*BOOT(FWA=F000)  
*ENDPLIB  
*EOJ
```

## ASSEMBLE AND ADD NEW PROGRAM TO MPX EXECUTIVE ROUTINES

The new program is assembled by COMPASS and placed on the LGO file (logical unit 57).

The MPX executive binaries, including the new program from the alternate input unit 57, and the job monitor and job loader binaries are absolutized and written on logical unit 10.

```
*JOB(...)                                *JLDR  
*SCHED(...)                              (Job loader binaries)  
*OPEN(10,...)                            *ENDPLIB  
*CMP(X, L)                               *EOJ  
    (ASCII source of program)  
*REWIND(57)  
*PRELIB  
*MPX(NMPX=10)  
    (MPX executive binaries)  
*ALT(ALT=57)  
*JMTR  
    (Job monitor binaries)
```

## RESTORE MPX SYSTEM USING ALTERNATE INPUT UNIT

The control cards and binary decks necessary for restoring the MP-60 emulator, MPX boot program, MPX resident, and MPX library are on the alternate input unit 10.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*ALLOCATE(MPX-RESIDENT,...)  
*ALLOCATE(MPX-LIBRARY,...)  
*OPEN(30,MPX-RESIDENT,...)  
*OPEN(20,MPX-LIBRARY,...)  
*PRELIB  
*ALT(ALT=10)
```

The following control cards and binaries are on the alternate input unit 10.

```
*MICR  
    (MP-60 emulator program binary)  
*BOOT(FWA=F000)  
    (MPX boot program binary)  
*MPX(NMPX=30)  
    (MPX executive binaries)  
*JMTR  
    (MPX job monitor binaries)  
*JLDR  
    (MPX job loader binaries)  
*LIB(NLIB=20)  
    (MPX library binaries)  
*ENDPLIB
```

The following control card is on the standard input unit.

```
*EOJ
```

## BUILD MPX INSTALLATION TAPE

The magnetic tape boot program and MPX installation program are absolutized and written on logical unit 10.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*PRELIB  
*ABS(NABS=10)  
*ORIGIN(ID=TAPEBOOT, FWA=F000)  
    (Tape boot binary)  
*ORIGIN(ID=INSTALL, FWA=0000)  
    (Install binary)  
*ENDPLIB  
*EOJ
```

## GENERATE NEW IOC MICROCODE

The IOC microcode program is absolutized and written to the MPX system device.

```
*JOB(...)  
*SCHED(...)  
*PRELIB  
*CNTR0  
    (IOC0 microcode binary)  
*ENDPLIB  
*EOJ
```

## GENERATE NEW IOC0 AND IOC1 MICROCODE

The IOC0 and IOC1 microcode programs are absolutized and written to the MPX system device.

```
*JOB(...)  
*SCHED(...)  
*PRELIB  
*CNTR0  
    (IOC0 microcode binary)  
*CNTR1  
    (IOC1 microcode binary)  
*ENDPLIB  
*EOJ
```

## GENERATE NEW MPX SYSTEM TAPE

The MP-60 emulator program, MPX boot program, MPX resident from logical unit 20, MPX library currently operating, the IOC0 microcode program, IOC1 microcode program, DIAG0 microcode program from logical unit 30, CPUD microcode program, MP-60 emulator microcode program, and MPXBOOT program are copied to the new MPX system tape on logical unit 10.

```
*JOB(...)  
*SCHED(...)  
*EQUIP(10=MT)  
*OPEN(20, MPX-RESIDENT, MPX, 01, ****, R)  
*OPEN(30, IOC0FID, MPX, 01, ****, R)  
*PRELIB  
*DUMPSYS(NSYS=10, DMPX=20, DLIB=58, ID=MPXSTAPE, CNTR0, CNTR1,  
    DIAG0=30, CPUD)  
*ENDPLIB  
*EOJ
```

## GENERATE NEW DIAG0 MICROCODE

The DIAG0 microcode program is absolutized and written to the MPX system device.

\*JOB(...)

\*SCHED(...)

\*PRELIB

\*DIAG0

(DIAG microcode binary for IOC0)

\*ENDPLIB

\*EOJ



---

**MPX SYSTEM DEVICE**

The MPX system device for the MPX operating system is the mass storage unit which contains one or more IOC controller firmware and diagnostic microcode programs; the MP-60 emulator; MPXBOOT program, all in absolute form; and may also contain the CPU diagnostic microcode program. The MPX system device also contains the first segment of the label file for the MPX system. The label file must start on a logical track boundary. Figure 5-1 illustrates the general format for the layout of MPX system device. The logical addresses for different device types are listed in the columns to the right of the diagram. All microcode programs up to MPXBOOT are a standard disk configuration. If any are not applicable, these will be gaps on the disk. Any additional firmware will be added in between MPXBOOT and the label file. The address of the label file starts at the beginning of the next logical track following the last microcode program. The additional microcode programs are allocated at 4,096 32-bit word increments.

System Device Layout	<u>Logical Sector Address</u>				
	<u>9425 and 9427</u>	<u>844</u>	<u>858</u>	<u>MD640</u>	
Device label (80 words)	0	0	0	0	
IOC0 firmware (4K words)	1	1	1	1	
IOC0 diagnostics (4K words)	42	36	33	22	
MP-60 CPU emulator (4K words)	83	71	65	43	
MP-60 CPU diagnostics (4K words)	124	106	97	64	
MPXBOOT (4K words)	165	141	129	85	
Additional microcode programs	IOC1	206	176	161	106
	DIAG1	247	211	193	127
	IOC2	288	246	225	148
	DIAG2	329	281	257	169
	IOC3	370	316	289	190
DIAG3	411	351	321	211	
Label file	16 (n+1) <sup>†</sup> 80 (n+1) 32 (n+1) 36 (n+1)				
MPX resident file					
MPX library file					
User space					

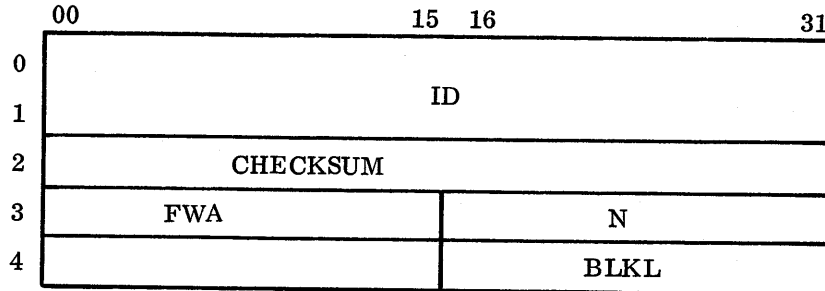
Figure 5-1. MPX System Device Layout and Logical Sector Addresses

<sup>†</sup> n is the track address for the last microcode program.



## ABSOLUTE HEADER RECORDS

Absolute records generated by PRELIB have header records with the following generalized format.



ID - One to eight character identifier for the absolute record

CHECKSUM - Complemented checksum of the absolute record

FWA - First word address of absolute record

N - Number of 4K blocks in the record

BLKL - Word length of the last block of the record

## MPX RESIDENT

MPX resident, that is, the MPX executive and job monitor program, resides as a file on mass storage in the MPX system. Before calling PRELIB to create an MPX resident file, the user must allocate and open the file with the following parameters:

File Name - MPX-RESIDENT

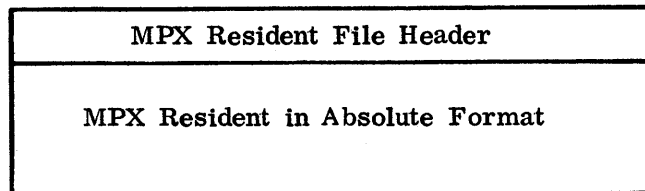
Owner - MPXR

Access Privacy - \*\*\*\*

Block Size - 1

Segmentation - NS

Since MPX resident resides as a file, a user may have several editions of MPX resident on mass storage at the same time. At autoloading time, the user specifies the edition number of the MPX resident file he wishes to execute.



The identifier in the MPX resident file header is \*MPXRESID.

## MPX LIBRARY

MPX library, that is, binary decks of user routines and programs, resides as a file on mass storage in the MPX system. Before calling PRELIB to create a MPX library file, the user must allocate and open the file with the following parameters:

File Name - MPX-LIBRARY

Owner - MPXL

Access Privacy - \*\*\*\*

Block Size - 480 words

The MPX library (Figure 5-2) is created using the MPX logical I/O routines, blocker/deblocker. The MPX library consists of a series of binary records. (Refer to the loader section of the MPX reference manual for a discussion of the records.) The MPX library is searched at load time by the system loader for unsatisfied externals.

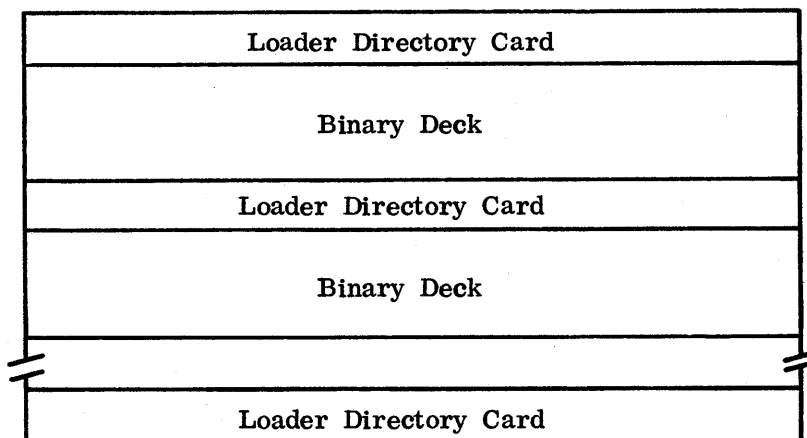


Figure 5-2. MPX Library File

Each loader directory card in the library has a pointer to the block containing the next loader directory card. The last loader directory card in the library has a pointer of zeros signifying end of the library.

## MPX SYSTEM TAPE

The MPX system tape (Figures 5-3 and 5-4) contains all of the information necessary for installing the MPX operating system. The system tape format depends upon the optional parameters on the \*DUMPSYS control card.

System Tape Header
EOF
MP-60 Emulator (4K)
EOF
MPX BOOT Header
MPX BOOT
EOF
MPX Resident Header
MPX Resident
EOF
MPX Library
EOF
EOF

Figure 5-3. MPX Sample System Tape  
(No optional fields)

<b>System Tape Header</b>
EOF
<b>MP-60 Emulator</b>
EOF
<b>IOC0 Header</b>
IOC0
EOF
<b>DIAG0 Header</b>
DIAG0
EOF
<b>CPUD Header</b>
CPUD
EOF
<b>MPXBOOT Header</b>
MPXBOOT
EOF
<b>IOC1 Header</b>
IOC1
EOF
<b>DIAG1 Header</b>
DIAG1
EOF
<b>MPX Resident Header</b>
MPX Resident
EOF
<b>MPX Library</b>
EOF
EOF

Figure 5-4. MPX Sample System Tape  
(Two IOC, two DIAG and one CPUD programs)

The first file on the MPX system tape consists of a header record with the following format.

0	*	M	P	X
1	S	Y	S	T
2	ID			
3				

ID - One- to eight-character identifier corresponding to the ID parameter on the \*DUMPSYS card.

The second file on the MPX system tape is the MP-60 emulator written in absolute format as one 4K word block.

The next group of files are variable and may include IOC0, IOC1, IOC2, IOC3, DIAG0, DIAG1, DIAG2, DIAG3, and CPUD (programs written to the devices to provide deadstarting of the IOCs and CPUs). Each of these files is written in absolute format as one 4K block plus a header record.

Following these files (if they exist), on the MPX system tape, is the MPX boot program in absolute format with a header record.

Following the MPX boot file on the MPX system tape is the MPX resident programs in absolute format with a header record.

Following the MPX resident file on the MPX system tape is the MPX library written as 480 word blocks.

### **PRELIB SCRATCH FILE**

PRELIB uses one scratch file, logical unit 59. The user should use caution when utilizing scratch in a job process which calls PRELIB.



# PRELIB DIAGNOSTICS

All PRELIB diagnostics appear on the standard output unit, OUT, and have the following format:

\*\*\*\*\* MODULE message

The module is 1 through 8 characters identifying the PRELIB module currently executing when the error occurred.

Table 6-1 lists PRELIB diagnostics, their causes, and action taken by PRELIB, if any.

TABLE 6-1. PRELIB DIAGNOSTICS

Message	Cause	Action
BINARY CARD	Binary card was encountered when ASCII card was expected	Processing continues with next valid control card
CHECKSUM ERROR XX PROGRAM XXXXXXXX	Checksum error occurred on card XX in binary deck of program XXXXXXXX	Processing continues
INVALID CONTROL CARD		Processing continues with next valid PRELIB Control control card
ILLEGAL HEX CHARACTER	Illegal hex character was encountered in absolute microcode deck	Processing continues with next valid control card
ILLEGAL PARAMETER		If on *PRELIB card, job aborts; otherwise, processing continues with next valid control card
LIBRARY CREATION RUN	LIB module encountered a LIB control card which references an old library when none was specified	Processing continues with next valid control card

TABLE 6-1. PRELIB DIAGNOSTICS (Cont.)

Message	Cause	Action
LOAD ERROR error message	Loader error occurred; error messages are MPX loader errors (Section 9 of MPX Reference Manual)	Processing continues with next valid control card
LOAD MEMORY EXCEEDED	Boot program larger than 4K. Absolute microprogram larger than 4K	Processing continues with next valid control card
LOADER CORE EXCEEDED	Not enough memory available to PRELIB to absolutize program	Processing continues with next valid control card
LUN INVALID		If *PRELIB card, job aborts; otherwise, processing continues with next valid control card
LUN XX ILLEGAL DEVICE		If *PRELIB card, job aborts; otherwise, processing continues with next valid control card
LUN XX ILLEGAL FORMAT	Logical unit is not in format expected	Processing continues with next valid control card
LUN XX NOT EQUIPPED OR OPEN		If *PRELIB card, job aborts; otherwise, processing continues with next valid control card
MORE THAN 13 ENTRY POINTS	LIB module was directed to enter more than 13 entry points on loader directory card	The first 13 entry points are entered and processing continues
NO BINARY	Binary deck was expected	Processing continues with next valid control card
NO ENDPLIB	End-of-file was encountered on standard input unit	PRELIB terminates normally



TABLE 6-1. PRELIB DIAGNOSTICS (Cont.)

Message	Cause	Action
NO ENDT	LIB module encountered another control card before encountering *ENDT control card	Processing continues with next valid control card
NO ENTRY POINT XXXXXXXXX IN PROGRAM XXXXXXXXX	Named entry point was on *DIRCT card but not on EPT in binary deck	Processing continues
NO MICRO DECK	Absolute microdeck was expected	Processing continues with next valid control card
NO OLD SYSTEM TAPE	Old system tape is required but was not specified on *PRELIB card.	Processing continues with next valid PRELIB Control control card
NO PROGRAM XXXXXXXXX	Program specified on LIB module control card is not on old library	Processing continues with next valid control card
NO JMTR	MPX module did not encounter a *JMTR control card	Processing continues with next valid PRELIB Control control card
NO TEXT	Lib module encountered an *ENDT control card when a *TEXT control card had never been read	Processing continues with next valid control card
PARAMETER APPEARS TWICE		Processing continues
PARAMETER REQUIRED	Parameter which is required is missing	Processing continues with next valid control card

TABLE 6-1. PRELIB DIAGNOSTICS (Cont.)

Message	Cause	Action
PACK } PACKD } ERR ON XX PACKD }  STATUS = XXXXXXXX		Job aborts
PICK } PICKD } ERR ON XX PICKI }  STATUS = XXXXXXXX		Job aborts
PROGRAM ERROR OCCURRED AT XXXX	Unexplained condition occurred in PRELIB program at address XXXX	Job aborts
READLU } WRITLU } ERR ON XX  STATUS = XXXXXXXX		Job aborts; if XX = 00, error was on the MPX system device
SEQUENCE ERROR, LAST CARD XXXXX CURRENT CARD XXXXX	Microcode absolute binary deck out of sequence	Processing continues with next valid PRELIB Control control card
VERIFY ERROR ON SYSTEM TAPE	MPX system tape which was just generated failed to verify	Processing continues with next valid control card
NO CPUD ON OLD SYSTEM TAPE  NO DIAG0 ON OLD SYSTEM TAPE  NO DIAG1 ON OLD SYSTEM TAPE  NO DIAG2 ON OLD SYSTEM TAPE  NO DIAG3 ON OLD SYSTEM TAPE  NO IOC0 ON OLD SYSTEM TAPE	The requested programs are not on the MPX system tape	Processing continues with next valid control card

TABLE 6-1. PRELIB DIAGNOSTICS (Cont.)

Message	Cause	Action
NO IOC1 ON OLD SYSTEM TAPE NO IOC2 ON OLD SYSTEM TAPE NO IOC3 ON OLD SYSTEM TAPE	The requested programs are not on the MPX system tape	Processing continues with next valid control card



# INDEX

---

Item	Page
*ABS	2-1, 3-1, 3-2, 3-8
ABS module	2-1, 2-4, 3-1, 3-2, 3-8
Absolute header records	5-3
*ALT	2-1, 2-3, 2-4, 3-2, 3-3, 3-6 3-7, 3-3, 3-9, 4-5
*AMICR	2-4, 3-2
*BOOT	2-1, 3-2, 4-6
BOOT module	2-1, 2-3, 3-2
*CNTR0	2-1, 3-3, 4-7, 4-8
*CNTR1	2-1, 3-3
*CNTR2	3-1, 3-3, 4-8
*CNTR3	2-1, 3-3
CNTR module	2-1, 2-4, 3-3
*CPUD	2-1, 3-3, 5-7
CPUD module	2-1, 2-5, 3-3
*DELETE	2-3, 3-3
*DIAG0	2-1, 3-4, 4-9, 5-7
*DIAG1	2-1, 3-4, 5-7
*DIAG2	2-1, 3-4, 5-7
*DIAG3	2-1, 3-4, 5-7
DIAG module	2-1, 2-5, 3-4
Diagnostics	6-1
*DIRCT	2-3, 3-4, 3-9
*DUMPSYS	2-1, 3-4, 5-7
DUMPSYS module	2-1, 2-4, 3-4
*ENDPLIB	2-1, 3-6
*ENDT	2-4, 3-6, 3-9

## INDEX(CONT.)

Item	Page
*INSERT	2-3, 3-6
*JLDR	2-3, 3-6, 3-8, 4-5, 4-6
*JMTR	2-3, 3-7, 3-8, 4-5, 4-6
Job loader	3-6, 3-7, 3-8, 4-5, 4-6
Job manager	3-6, 3-7, 3-8, 4-5, 4-6
*LIB	2-1, 3-7, 4-6
LIB module	2-1, 2-3, 3-4, 3-6, 3-7, 3-9
*MICR	2-1, 3-8, 4-6
MICR module	2-1, 3-6
MP-60 emulator	2-1, 3-6, 4-2, 4-3, 4-5, 4-6, 4-8 5-1, 5-7
*MPX	2-1, 3-8
MPX boot program	2-3, 3-2, 3-5, 4-2, 4-3, 4-5, 4-6, 4-8, 5-1, 5-7
MPX control cards	4-1
MPX executive	4-2, 4-4, 4-5, 5-4
MPX library	2-2, 4-1, 4-2, 4-3, 4-4, 4-6, 4-8, 5-4, 5-7
MPX module	2-1, 2-3, 3-6, 3-7, 3-8
MPX resident	2-3, 3-5, 3-7, 3-8, 4-2, 4-3, 4-4, 4-6, 4-8, 5-3, 5-4, 5-7
MPX system device	2-1, 2-3, 3-2, 3-4, 3-5, 3-8, 4-7, 4-8, 5-1
MPX system tape	2-4, 3-3, 3-4, 3-5, 3-7, 3-8, 4-2, 4-5, 4-8, 5-5
*ORIGIN	2-4, 3-8, 3-9
*PRELIB	3-6, 3-9, 4-1
PRELIB Control	2-1, 3-1, 3-2, 3-3, 3-4, 3-6, 3-7, 3-8
PRELIB Diagnostics	6-1
*REPLACE	2-3, 3-9
Scratch files	5-7
*TEXT	2-4, 3-9

# COMMENT SHEET

MANUAL TITLE MP-60 Computer System PRELIB/PRELIBOS Reference  
Manual

PUBLICATION NO. 14062200 REVISION E

**FROM:** NAME: \_\_\_\_\_  
BUSINESS ADDRESS: \_\_\_\_\_

## COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

PRINTED IN U.S.A.

AA3419 REV. 11/69

**NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.**  
FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

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

FIRST CLASS  
PERMIT NO. 8241  
MINNEAPOLIS, MINN.



POSTAGE WILL BE PAID BY  
**CONTROL DATA CORPORATION**  
Aerospace Division  
Box 609 HQG346B  
Minneapolis, Minnesota 55440

CUT ALONG LINE

FOLD

FOLD

