

SERIES 200

LIBRARY PROCESSOR C

GENERAL SYSTEM:

SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT:

Operational Characteristics and Operating Procedures for Library Processor C, a Program for Processing Macro Instructions and Inserting Macro Routines into the Symbolic Input Programs to EasyCoder Assemblers C and D.

SPECIAL
INSTRUCTION

This software bulletin supersedes the Honeywell Information Bulletin entitled Library Preprocessor (DSI-326), dated October 1, 1964.

Date: October 28, 1965

FILE NO.: 122.1605.001C.0-051

8514

51065

Printed in U.S.A.

FOREWORD

This software bulletin is a companion publication to the software bulletin entitled Easycoder Assemblers C and D, File Number 122.1105.001C.00.00. The reader should be thoroughly familiar with the information presented in the above publication. The reader should also be familiar with the information presented in the Honeywell Series 200 Programmers' Reference Manual, Models 200/1200/2200 (File Number 113.0005.0000.00.00) and the general operating characteristics of the applicable Series 200 equipment as explained in the Honeywell Series 200 Equipment Operators' Manual (Model 200) (DSI-294).

Copyright 1965
Honeywell Inc.
Electronic Data Processing Division
Wellesley Hills, Massachusetts 02181

TABLE OF CONTENTS

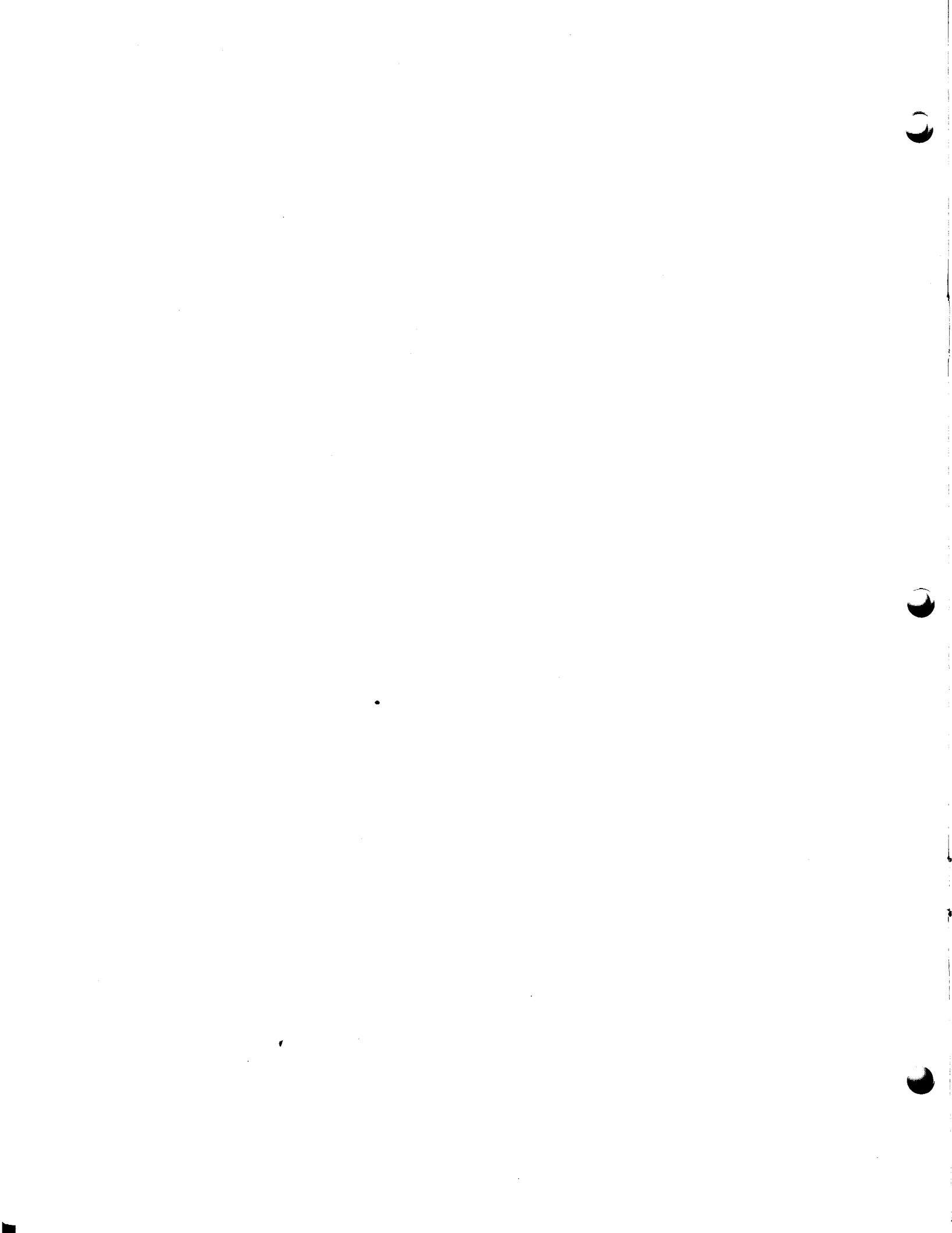
		Page
Section I	General Description of Library Processor C.....	1-1
	Equipment Requirements.....	1-2
Section II	Writing a Macro Routine.....	2-1
	Parameter Designation	2-1
	Selective Omission of Coding	2-1
	Tag Prefixes.....	2-3
	Adding Macro Routines to the SPT	2-4
Section III	Writing a Macro Instruction.....	3-1
	Continuation Cards	3-1
	Selective Omission of Parameters.....	3-1
	Card Numbers	3-3
	Using Library Processor C to Produce Symbolic Decks	3-4
Section IV	Input and Output Files	4-1
	Input File.....	4-1
	Console Call Card	4-1
	Equipment Configuration Director	4-1
	Standard Equipment Configuration Numbers.....	4-2
	Card Input/Card Output	4-2
	Card Input/Tape Output	4-2
	Tape Input/Tape Output	4-3
	Full Equipment Configuration Director	4-3
	Input Deck	4-5
	System Specific Header Card.....	4-5
	Assembly Input	4-5
	End-of-File Card	4-5
	Output File	4-6
Section V	Operating Procedures	5-1
	Initial Setup Procedures.....	5-1
	Bootstrap Procedures.....	5-1
	Error Conditions.....	5-1
Appendix A	Procedure for Changing Standard Equipment Configurations.....	A-1

LIST OF ILLUSTRATIONS

Figure 1-1	Functions of Library Processor C.....	1-2
Figure 4-1	Input and Output Deck Arrangements	4-6
Figure A-1	Sample Coding for Library Processor C Standard Configurations 2, 5, 6, and 8.....	A-1

LIST OF TABLES

Table 4-1	Format for File Media Fields	4-4
Table 5-1	Library Processor C Error Conditions.....	5-2



SECTION I
GENERAL DESCRIPTION OF LIBRARY PROCESSOR C

Library Processor C processes symbolic card-image files that contain input to EasyCoder Assembler C or D. Library Processor C is used prior to the assembly of a source program by EasyCoder Assembler C or D. This program inserts macro routines which exist on the symbolic program tape (SPT) into source programs in response to macro instructions (calls) within the source programs, and it specializes each routine to perform the specific function desired.

The macro routines are stored on the SPT where they are available to be incorporated into a program when and where desired. A macro routine is made up of instructions that perform a particular operation (e. g., open or close a tape file). This routine is considered generalized when it contains elements whose values are missing and/or when only certain of the included instructions may be used in its execution. The routine becomes specialized when the missing values are supplied and/or when the instructions which are not to be used are omitted.

A macro instruction (call) is an instruction written in a program, in-line, for the purpose of calling in a macro routine. Included within the macro instruction is the list of desired parameter values for use by the macro routine being called. A parameter is a variable element within a macro routine which must be assigned an explicit value in order to specialize the routine. A parameter designator is an expression used in the macro routine to identify a parameter of the routine.

Figure 1-1 illustrates, functionally, the use of the Library Processor C. The source program(s) includes macro instructions which call the macro routines from the SPT. The source programs and macro routines are processed by Library Processor C which is stored on the binary run tape (BRT). The processing reproduces the source program(s), inserting the specialized macro routines to form a new deck which is now ready for assembly by EasyCoder Assembler C or D. The concepts of "card" and "deck" are used in referring to card-image files, regardless of the medium on which the file exists.

The Library Processor C input file may include input to systems other than the EasyCoder Assemblers. Consequently, the card-image output tape prepared by the Library Processor may contain input to several systems.

A second function of the Library Processor C is reproducing symbolic decks of entire programs on the SPT.

EQUIPMENT REQUIREMENTS

Library Processor C requires the use of the following equipment:

1. Advanced Programming Instructions Feature (011).
2. 12,288 locations of memory.
3. Card reader (for card input), card punch (for card output), or 204B tape drive(s) (if card images are used).
4. Two 204B tape drives (for the BRT and the SPT).

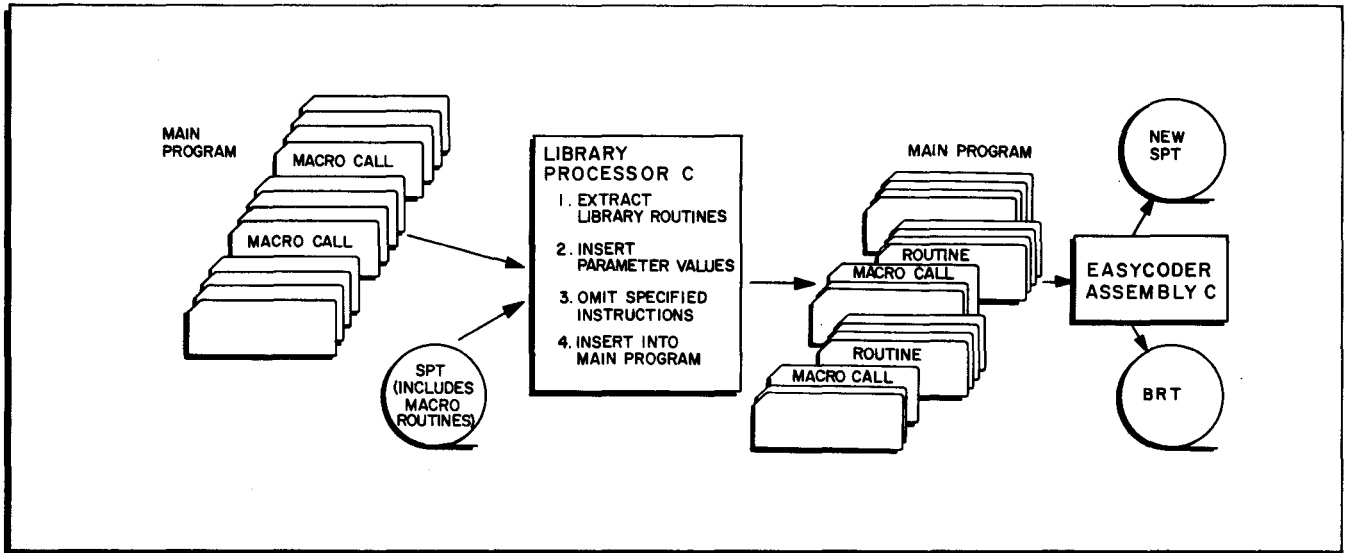


Figure 1-1. Functions of Library Processor C

SECTION II
WRITING A MACRO ROUTINE

Some routines are Honeywell-supplied (e. g., Input/Output Control programs) while others may be written by the user. This section explains how the user should write a generalized macro routine.

PARAMETER DESIGNATION

Parameter designators have the format pxy, where p is any alphanumeric character chosen by the programmer,¹ and x and y form the decimal parameter number from 00 to 63. A control instruction, Set Parameter Designator (SETP), is used to assign a value to p. SETP is written in the operation code field, and the desired value for p is written in column 21. The value of p may be changed at any time by writing another SETP instruction. If the SETP instruction is not used, the value of p is assumed to be octal 35, which prints as %. The following are possible assignments of p:

EASYCODER
CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____

CARD NUMBER	V	M	P	R	LOCATION	OPERATION CODE	OPERANDS		
							14 15	20 21	
1	2	3	4	5	6	7	8	62 63	80
						SETP			
						SETP	*		

Parameters are indicated by writing the currently assigned value of p, followed by a parameter number (xy) which the programmer assigns consecutively. When the routine is specialized by Library Processor C, the parameter designator is replaced by the explicit value supplied in the calling macro instruction. For example, assume that parameter 03 is an index register number. An indexed address using that register with an augment of one would appear within the macro routine as 1+Xp03. When the routine is specialized, the parameter value (e. g., 5) replaces the designator p03, creating the address 1+X5. Parameter 00 (p00) is always used to indicate the tag (if any) written in the location field of the macro instruction.

SELECTIVE OMISSION OF CODING

The programmer may desire that certain lines of coding be omitted from the macro

¹Although there is no restriction on the characters that can be assigned to p, it is the responsibility of the programmer to insure that the resulting parameter designators do not duplicate the form of any other language element, such as a symbolic tag.

routine. The zone portion of y in the parameter designator may be overpunched with R(+) or X(-). An R(+) overpunch indicates that if the value for parameter xy is blank or omitted in the calling macro instruction, this line of coding is omitted from the routine. An X(-) overpunch indicates that if the parameter value is included in the macro instruction, this line of coding is omitted from the routine.

Suppose, for example, that a macro routine computes a hash total of a particular field on an optional basis. The field to be totaled is parameter 01, and the field to contain the total is parameter 02. The instruction in the routine to update the total would be coded as follows:

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER		TYPE	LOCATION	OPERATION CODE	OPERANDS	
1	2				3	4
1	2	3	4	5	6	7
				A		P01,P0B
2						

NOTE: B is the result after the overpunching of 2 with R(+).

If the parameter value for the field which is to contain the total is omitted from the call, this instruction is also omitted from the specialized routine by Library Processor C. Instructions which do not explicitly address an optional parameter may also be selectively omitted by punching the parameter designator (with the appropriate overpunch) in the remarks portion of each such card.

Conditional (COND) control statements may also be used to omit lines of coding from the macro routine. The programmer writes COND in the operation code field. The operands field contains nnnnn, pxy, v, c, where:

- nnnnn - card number of the next statement not to be omitted when the condition is true
- v - literal value against which parameter xy is tested
- c - condition

The condition c is coded as follows:

<u>c</u>	<u>Condition:</u>
0	Never true
1	True if value of pxy > v
2	True if value of pxy = v
3	True if value of pxy ≥ v
4	True if value of pxy < v
5	True if value of pxy ≠ v
6	True if value of pxy ≤ v
7	Always true

The condition is tested by a binary Compare of the value of parameter xy (A address) against v (B address), followed by a Branch on Condition Test with a variant character of 4c (where c is interpreted as shown above). If c is true, all statements of the macro routine (including additional COND and SETP statements, if any) from this point up to, but not including, nnnnn are omitted. All rules of the Compare instruction apply to the condition function.¹ A void parameter produces an equal result when compared with a field of up to 40 blanks.

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER	T M A R K	L O C A T I O N	O P E R A T I O N C O D E	O P E R A N D S	
1			COND	00014, P02, KRAM, 1	(EXAMPLE)
2			COND	nnnnh, pxy, vc	(FORMAT)
3					

When the Conditional control statement illustrated in the example is processed, the value assigned to parameter 02 is compared with the literal value KRAM. If the parameter value is greater than KRAM (since c = 1), the following statements are omitted from the routine, up to statement 00014, which is included in the routine. If the parameter value is less than or equal to KRAM, no statements are omitted from the routine at this point.

TAG PREFIXES

Duplication of tags between the calling program and any of the macro routines called (or between two of the macro routines called) must be avoided. To avoid duplication, it is recommended that each macro routine be assigned a particular prefix and that each of its tags be preceded by this prefix. The length of the tag and its prefix must never exceed six characters. If the same macro routine is to be called more than once by a single main program, the tag prefix should be designated by means of a parameter to avoid duplication. Thus, the tag prefix will be different for each insertion of the routine.

¹The binary Compare reads the A- and B-address fields simultaneously from right to left starting at the end of the B address and ending with the word mark of the B address. (To avoid an equal comparison that may not be truly equal, an extra 0 character should begin the B-address field.) An example follows:

<u>A address</u>	<u>B address</u> (v values above)
nonstandard	(0) standard

(If the zero is present at the beginning of the B address, the comparison does not stop after reading "standard" in the two address fields but also compares the next character. When such possibilities do not exist, the additional character in the B address need not be used.)

ADDING MACRO ROUTINES TO THE SPT

A generalized macro routine is prepared in the same manner as any other program, except for the presence of parameter designators to indicate missing values. It is submitted to EasyCoder Assembler C or D, complete with its own PROG and END cards, for addition to the SPT. Since there is no distinction between a macro routine and any other program on the SPT and since Library Processor C searches the SPT by program name rather than by visibility, each macro routine must be assigned a program name which is unique within the entire SPT. Tape searching by Library Processor C will be minimized if macro routines are stored together in one particular portion of the SPT.

SECTION III
 WRITING A MACRO INSTRUCTION

The programmer writes a macro instruction at the point in his program where a macro routine is to be incorporated. The type field contains C when all the parameter values for a particular macro routine do not fit on one card and require continuation cards to follow; otherwise, it contains an L. The location field may contain a symbolic tag which, when written, is always interpreted as the value of parameter p00. The operation code field contains the name of the desired macro routine (which is also the name on the PROG card for the routine); the operands field contains the parameter values, written in order by parameter number, starting with the value of parameter p01.

CONTINUATION CARDS

A continuation card is used when a macro call can not contain all the parameters for a particular macro routine. Although the first card of a multiple-card call is not a continuation card, it indicates, with a C in the type field, that a continuation card follows. Each continuation card, except the last, contains a C. The last continuation card contains an L in the type field.

Although macro instruction cards and continuation cards are reproduced by Library Processor C, they are regarded as remarks cards by EasyCoder Assembler C or D. COND, SETP, and END cards in a macro routine are not reproduced by Library Processor C.

SELECTIVE OMISSION OF PARAMETERS

A parameter value may contain any character except the comma. The comma is used to follow each parameter value, including the last. The comma also serves as a method of omitting a parameter value from the macro instruction. Each missing parameter value is indicated by its comma. However, any number of values may be omitted without their terminating commas if no further values are needed.

Example: If a macro routine has ten parameters (p01 through p10) and the programmer wishes to omit values 3, 5, 6, and 8 - 10, he may code the instruction as follows:

EASYCODER
 CODING FORM

PROBLEM _____										PROGRAMMER _____										DATE _____										PAGE _____ OF _____									
CARD NUMBER		TYPE	LOCATION	OPERATION CODE		OPERANDS																																	
1	2			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30								
1		L	(TAG)	NAME	VAL1,	VAL2,	VAL4,	VAL7,																															
2																																							
3																																							

An alternative method of omitting parameter values is convenient for omitting several consecutive values when more values are to follow. Write the number of the next parameter not to be omitted in columns 15 and 16 of the next continuation card. Then write the actual value of this parameter in the operands field and continue as usual. To omit the first n values, do not write any values in the macro instruction card, and write the number of the first parameter whose value is not omitted in columns 15 and 16 of the first continuation card.

Example: If a macro routine has parameters p01 through p10 and values 2 and 6-9 are to be omitted, the programmer may write the following:

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____

CARD NUMBER	MARK	LOCATION	OPERATION CODE	OPERANDS	
				14 15	20 21
1	C	(TAG) NAME	VAL1, , VAL3, VAL4, VAL5,		
2	L	10	VAL10,		

To omit values 1-4, 6, 7, and 10, the programmer may write the following:

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____

CARD NUMBER	MARK	LOCATION	OPERATION CODE	OPERANDS	
				14 15	20 21
1	C	(TAG) NAME			
2	C	05	VAL5,		
3	L	08	VAL8, VAL9,		

The following example summarizes the complete relationship of the macro instruction, the generalized macro routine, and the macro routine after it has been specialized and incorporated into the main program. The macro routine is shown first in its generalized form as it exists on the SPT.

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____

CARD NUMBER	MARK	LOCATION	OPERATION CODE	OPERANDS	
				14 15	20 21
1			PROG	PROB	
2			SETP	@	
3		@00	SCR	@05ZEX+3, 70	
4		@05ZAD	A	@01+X@02, (@04)	
5			C	(@04), @05ZLM	
6			BCT	@05ZCN, 45	
7		@05ZEX	B	00	
8		@05ZCN	MCW	(@04), @03+15	
9					
10					
11					
12					
13		@05ZLM	DCW	@06	
14			END		

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER	MARK FIELD	LOCATION	OPERATION CODE	OPERANDS	
				14 15	20 21
1		MAIN PROGRAM			
2					
3					
4					
5					
6					
7					
8					
9	L	TAGER	PROB	AUG, 4, PRINT, SUM, MI, +100	Macro Instruction (call)
10		TAGER	SCR	MIZEX+3, 70	Specialized version of the same routine. This is inserted directly after the macro call in the output produced by Library Preprocessor.
11		MIZAD	A	AUG+X4, (SUM)	
12			C	(SUM), MIZLM	
13			BCT	MIZCN, 45	
14		MIZEX	B	00	
15		MIZCN	MCW	(SUM), PRINT+15	
16					
17					
18					
19					
20		MIZLM	DCW	+100	
21					
22		MAIN PROGRAM (UNTIL THE NEXT CALL)			
23					
24					
25					
26					
27					
28					
29					
30					

CARD NUMBERS

When a macro routine is inserted into a program, either of two types of card numbers are assigned. The type is assigned according to the entry in the mark field (column 7) of the first macro instruction (call) card. It should be noted that when a macro call contains more than one card, each card must have a different number. The two mark field entries are specified and interpreted as follows:

- Δ (blank): Generate card numbers by incrementing the card number of the macro call.
- D: Generate card numbers by adding the card number of each card in the generalized macro routine to the card number of the macro call.

If the mark field is blank (Δ), card numbers are assigned to the specialized macro routine cards by incrementing by four (octal 04) the low-order character of the card number of the last macro instruction card (L type). Thus, on the assembly listing, the low-order character of the card number column may be non-numeric, as shown by the following example using a single card macro instruction with a card number of 00100.

Macro Instruction (Call)	Specialized Macro Routine	
	Program Listing	Octal Value of Low-Order Character
00100LΔ	00100*	00
	00104	04
	00108	10
	0010:	14
	0010+	20
	0010D	24
	0010H	30
	.	.
	.	.
	.	.
	0010 (74
00110	00	

If the mark field contains a D, card numbers are assigned to the specialized macro routine cards by decimally adding the card number of each card in the generalized macro routine to the card number of the last card (L type) of the macro instruction. Therefore, when D is used, the card numbers of the generalized macro routine must be decimal numerals and their values should be as low as practicable. The program listing of the resulting program will contain numeric card numbers unless there is an error in the macro instruction, in which case the resulting error lines will be numbered as if the mark field had been blank (Δ).

Since improper use of card numbers may cause the "symbolic cards out of sequence" halt during assembly, the programmer should carefully allow for the card numbers assigned to the specialized macro routine when he assigns the card number to the line following the macro instruction (regardless of the method of card numbering). When unassigned card numbers are limited, it is advisable to use the first method of card numbering (blank in the mark field) to decrease the likelihood of introducing duplicate numbers in the specialized output.

USING LIBRARY PROCESSOR C TO PRODUCE SYMBOLIC DECKS

In order to reproduce a complete program from the SPT, a special macro instruction is issued to Library Processor C. This special macro instruction consists of a card containing a P in the type field (column 6) and the name of the macro routine (program to be duplicated) in the op code field (columns 15-20). This single special macro instruction usually comprises a unique program in the input file, but it may alternatively be included in another program which also contains standard macro instructions and continuation cards (C and L cards). When included in another program, the P type of macro instruction must not be immediately preceded by a C card. When Library Processor reproduces a program as a result of the P type macro instruction, any conditional statements and parameters in the program are not processed, the program line numbers are not renumbered according to the number of the macro instruction, any program END, COND, and SETP statements are reproduced, and the PROG statement is not reproduced. In addition, the P type macro instruction is itself eliminated from the Library Processor output.

SECTION IV
INPUT AND OUTPUT FILES

INPUT FILE

The input file for Library Processor C comprises the Equipment Configuration Director and the input deck. The file is described in punched-card format. If the file appears on card-image tape or paper tape, the coding format remains the same (assuming normal use of control frames on paper tape).

Although it is described here, the Console Call card is not considered part of the input file. When used, it precedes the input file.

Console Call Card

The following Console Call card entries represent a request for the loading of Library Processor C, which is identified by the name AACLIB.

Column(s)	Entry	Meaning
1-6	AACLIB	Program name of requested unit.
7-8	01	Segment name of requested unit.
9	d	Device number from which unit is to be loaded.
18	*	The asterisk in column 18 identifies the card as a Console Call card.

Equipment Configuration Director

The methods of specifying the equipment configuration are explained in detail in the operating procedures software bulletin Easycoder Assemblers C and D (File No. 122.1105.001C.00.00). These are:

- Method 1 - Standard ECD number residing in the Loader-Monitor communication area.
- Method 2 - Standard ECD number obtained from ECD image.
- Method 3 - Full Equipment Configuration Director as ECD image.

Under methods 2 and 3, an Equipment Configuration Director image precedes the input deck in the input file. This section defines the standard configurations for Library Processor C and explains the procedure for preparing a full Equipment Configuration Director. The specific information in this section and the detailed treatment of the three methods in the

aforementioned publication provide complete instructions for specifying equipment configurations for Library Processor C.

STANDARD EQUIPMENT CONFIGURATION NUMBERS

There are three standard operating modes for Library Processor C, each of which has its own complement of standard configurations:

1. Card input/card output (2 tapes required) - standard configurations 0, 1, 3, 4, and 7.
2. Card input/tape output (3 tapes required) - standard configurations 2, 5, 6, and 8.
3. Tape input/tape output (4 tapes required) - standard configuration 9.

As a system processing expedient, the standard configurations for Library Processor C are designed to coincide with the standard configurations for EasyCoder Assemblers C and D.

The description below shows the magnetic tapes and peripheral equipment used by each of the ten standard equipment configurations for Library Processor C. The tape numbers are the logical assignments. Tapes preceded by an asterisk (*) should be mounted in PROTECT status.

Card Input/Card Output

The first operating mode, card input and card output, is designed for use with those assembly configurations which do not include a card-image tape. Therefore, it is assigned standard configuration numbers 0, 1, 3, 4, and 7.

<u>Device</u>	<u>Use</u>	
* Tape 0	Program tape	
* Tape 6	SPT containing the library routines to be specialized	Standard Configurations 0, 1, 3, 4, 7
Card reader	Input to Library Processor C	
Card punch	Output from Library Processor C	

Card Input/Tape Output

The second operating mode employs card input and tape output. It is designed to be used with those assembly configurations which include a card image tape. Therefore, it has the standard configuration numbers of 2, 5, 6 and 8.

<u>Device</u>	<u>Use</u>
* Tape 0	Program tape
Tape 1	Card-image tape (output from Library Processor C)

<u>Device</u>	<u>Use</u>	
* Tape 6	SPT containing the Library routines to be specialized	Standard Configurations 2, 5, 6, 8
Card reader	(Input to Library Processor C)	

Tape Input/Tape Output

The third operating mode, tape input and tape output, is designed to allow card-image input to Library Processor C. Standard configuration 9 is used in this case.

<u>Device</u>	<u>Use</u>	
* Tape 0	Program tape	
Tape 1	Card-image tape (output from Library Processor C)	Standard Configuration 9
Tape 6	SPT containing the library routines to be specialized	
* Tape 2	Card-image tape (input to Library Processor C)	

FULL EQUIPMENT CONFIGURATION DIRECTOR

If it should be necessary to specify an equipment configuration which is not included among the standard configurations for Library Processor C, a full, user-constructed ECD image is required.¹

The format of the full Equipment Configuration Director is described below. Note that columns 1 through 5 must be blank to distinguish this from an Equipment Configuration Director specifying a standard equipment configuration.

Columns 1-5:	Contain Blanks.
Column 6:	Contains E to designate equipment configuration card.
Columns 8-10:	Contain read/write channel assignments for input, output, and SPT input, respectively; normally, read/write channels 1, 2, and 3 respectively. The coding for RWC assignments is as follows:
Column 8:	R X, 9 (51 ₈) for RWC1
Column 9:	X, 8, 2 (52 ₈) for RWC2
Column 10:	\$ 8, 3 (53 ₈) for RWC3

NOTE: When Library Processor C is run as a background program in conjunction with Interrupt Control D, one read/write channel must be reserved for the foreground program. For a processor with three RWC's, the same RWC code should appear in two columns. The third RWC will be free for the foreground program.

Columns 16-17:	Contain 00 to designate the lowest memory bank (4K module) available (always 00 for the first bank used).
----------------	---

¹If this full ECD is used frequently, one or more of the standard ECD's should be reassembled to reflect the needs of the installation. See Appendix A.

Columns 19-20: Contain the number of the highest memory bank available (must be greater than or equal to 02).

Columns 21-80: Contain the File Media fields, which designate the equipment configurations to be used. Columns 21 through 80 contain 20 of these File Media fields made up of three columns each. These three columns, in turn, contain three characters which completely specify the device to be used. The contents of each of the fields is listed in Table 4-1.

Table 4-1. Format for File Media Fields

Columns	Designate	First Character Device Type ¹	Second Char. PCU ²	Third Char. Tape Drive ³
21-23	Loading device	1	P	t
24-26	Console device	2 or 5	0 or P	0
27-29	Card input device	J or 1 or L	P	0 or t
30-32	Listing device (not used)	Δ	Δ	Δ
33-35	Card output	K or 1	P	0 or t
36-38	Library SPT	1	P	t
39-80	Not used with Library Processor C			

NOTES:

- The first character specifies the device type. The symbol for each device is as follows:

<u>Character</u>	<u>Octal Code</u>	<u>Device Type</u>
Δ	15	File Absent
0	00	Unspecified
1	01	Type 204B Magnetic Tape
2	02	Control Panel
3	03	Type 204A Magnetic Tape
4	04	Type 270 Drum
5	05	Type 220 Console
6	06	Main Memory
-	40	Type 206 Printer
J	41	Card Reader (Type 227)

<u>Character</u>	<u>Octal Code</u>	<u>Device Type</u>
K	42	Card Punch (Type 227)
L	43	Type 209 Paper Tape Reader
M	44	Type 210 Paper Tape Punch
N	45	Card Reader (Type 223, 214-2, or 224)
0	46	Card Punch (Type 214-1, 214-2, or 224)

2. The second character specifies the peripheral control number. This is control character C₂ of the PDT machine instruction. For tape files, Library Processor C will set bit 1 (I/O bit). For card and print files, this bit must be set correctly in the character as it is punched in the card. For the control panel and console, this character is 0 and 7, respectively.
3. The third character specifies the number of the tape drive to be used. This is the low-order octal digit of control character C₃ of the PDT machine instruction. For devices other than tape drives, this character is usually 0.

Input Deck

While the Equipment Configuration Director may or may not appear as a physical part of the input file (it is not used when a standard ECD number resides in the Loader-Monitor communication area), the input deck will always be present and must contain a System Specific Header card, the assembly input, and an End-of-File card, in that order. These cards are shown in Figure 4-1 and described below.

SYSTEM SPECIFIC HEADER CARD

Columns 1-5: Contain 1 HDRΔ
Columns 21-30: Contain MACROINPUT

ASSEMBLY INPUT

These cards include the System Specific Header for the EasyCoder Assembler, assembly directives, and symbolic program cards (including macro instructions), and the End-of-File card for the assembler (see Figure 4-1).

END-OF-FILE CARD

Columns 1-5: Contain 1 EOFΔ
Columns 21-30: Contain MACROINPUT

NOTE: The input file for Library Processor C may also include input to other system programs (e. g., Update and Select C). This is to allow input for several systems to be stored on one card-image output tape. All cards up to and including the End-of-File card for Library Processor C are included in the output. For example, if input to Update and Select C is placed after the assembly input and before the 1 EOFΔ MACROINPUT card. Library Processor C copies it unchanged to the output file. This output file becomes input to the Easy Assembler. After the assembler has processed the file, it is left positioned at the point where the input to Update and Select C begins. Update and Select C can be immediately loaded and executed.

THE OUTPUT FILE

The output file consists of all the cards which made up the input deck. In order it contains:

1. The Systems Specific Header card.
2. The input to the assembler with each macro instruction followed by the corresponding specialized macro routine.
3. Any input to other system programs.
4. The End-of-File card for Library Processor C.

Two End-of-Recorded-Information cards (1ERIA in columns 1-5) are generated following the last End-of-File card.

NOTE: When the output file is on tape an additional 80-character record precedes the System Specific Header record and has the following format:

Characters 1-5: 1HDRΔ
 Characters 21-30: CARDIMAGES
 Characters 41-44: 1A01
 Characters 78-80: 11A

The remaining characters are preserved from the previous contents of the tape record.

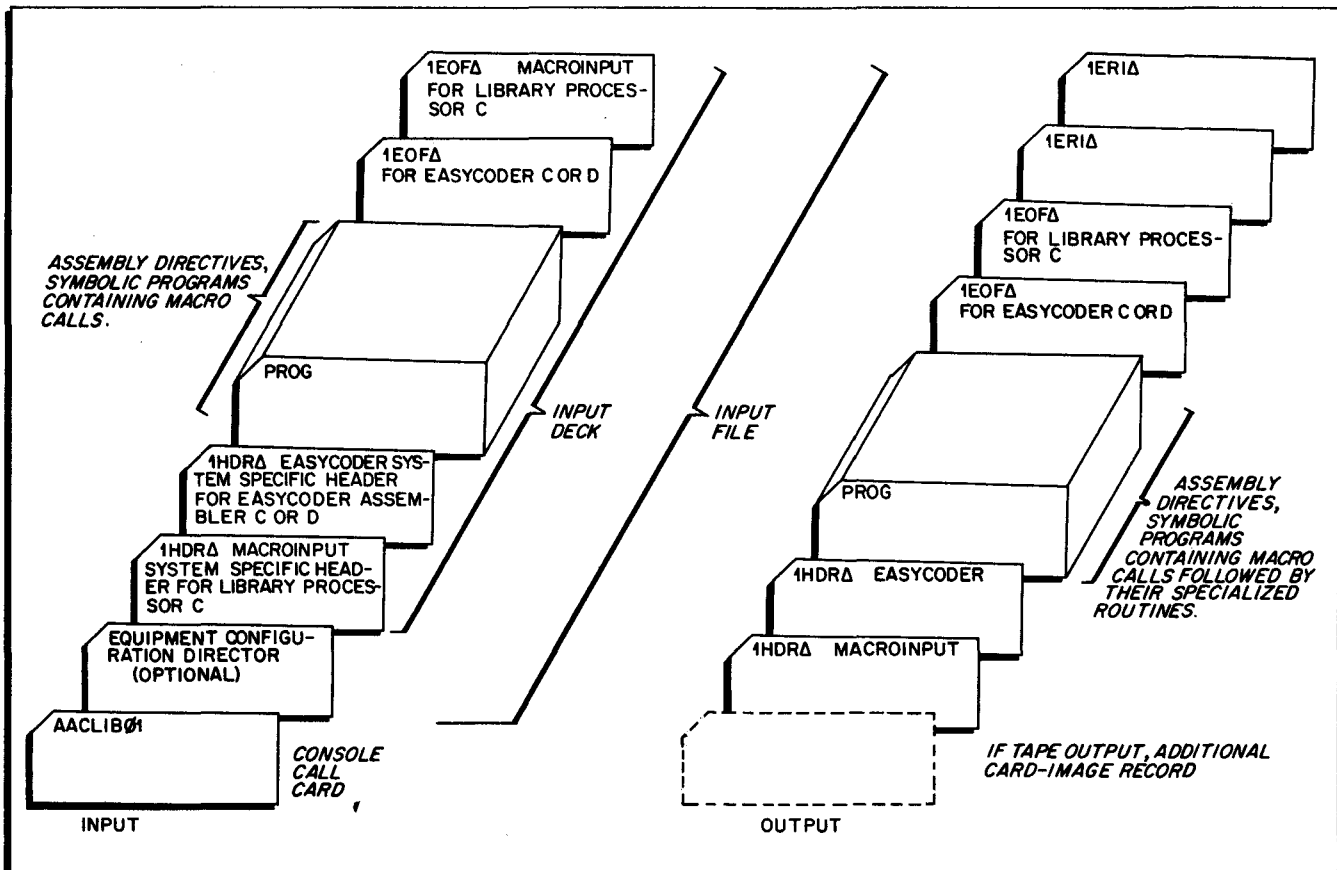


Figure 4-1. Input and Output Deck Arrangements

SECTION V
OPERATING PROCEDURES

INITIAL SETUP PROCEDURES

1. Depress the STOP button on the control panel if the machine is not already stopped.
2. Place the cards to be read into the card reader. Initialize the reader and perform the necessary steps to begin reading cards.

NOTE: If the card-reader is not to be used (i. e., when the input deck is on a card-image tape or on paper tape and the contents of the Console Call card are to be entered from the control panel), omit this step and proceed to step 3.

3. If the input deck is on paper tape, mount the paper tape reel or insert the paper tape strip in the paper tape reader. Initialize the paper tape reader.

NOTE: If the input deck is not on paper tape, omit this step and proceed to step 4.

4. Mount the program tape on the tape drive designated as logical 0. Re-wind the program tape and set the PERMIT-PROTECT switch to PROTECT.
5. Prepare the remaining tapes in accordance with the designated usage as outlined in "Standard Equipment Configuration Numbers" in Section IV. If a full ECD is used, refer to the file media fields in columns 27 through 38 of the ECD card.

BOOTSTRAP PROCEDURES

The bulletins PLUS-Tape Loader-Monitor (DSI-327) and Floating Tape Loader-Monitor C and Interrupt Control D (File No. 122.5005.001C.00.00) outline the procedures for bringing into memory the Tape Loader-Monitor C and Floating Tape Loader-Monitor C, respectively. The aforementioned publications also contain console call procedures which should be used for loading and executing Library Processor C.

ERROR CONDITIONS

For all halts, the B-address register should be displayed first to indicate the reason for the halt. In some cases, the A-address register contains supplementary information.

If Library Processor C is using the control panel, the following actions occur at an error condition:

1. Library Processor C halts.
2. Displaying the B-address gives the reason for the halt (see Table 5-1).
3. If the run is to be continued, the RUN button is depressed.

If Library Processor C is using the console typewriter, the following actions occur at error condition:

1. The console warning bell rings.
2. The error message is typed (see Table 5-1).
3. The TYPE Light is illuminated.
4. Library Processor C stalls.
5. If the run is to be continued, the G key is depressed.

Table 5-1. Library Processor C Error Conditions

Contents of B-Address Register	Console Typewriter Message	Cause and Action
07025	: NO E CARD□	<p>The card image that has been specified as the ECD does not contain an E in character 6.</p> <ol style="list-style-type: none"> 1. If the ECD is to be entered through the input device specified by the ECD parameter in the loader communication area, ascertain that the desired ECD is in the input device and continue the run. 2. If the ECD is <u>not</u> to be entered through the input device specified by the ECD parameter in the loader communication area, the ECD field, location 227g - 232g, may be changed to <ol style="list-style-type: none"> a. Accept the ECD from a different device <li style="text-align: center;">or b. Select one of the ten standard equipment configurations. <p>When this has been done, continue the run by depressing RUN button or G key.</p>
04007	: SPT DIRECTORY EXCEEDS MEMORY□	<p>This should occur only on a 12K memory configuration.</p> <p>Library Processor C stores in memory the name of each program on the SPT over which it passes. On a 12K machine, it can store over 300 program names.</p> <p>Continue the run. Any library routines located on the SPT from this point on will not be processed by Library Processor C.</p>
04010	: NO SYSTEM HEADER□	<p>Either the director file or the SPT was not found.</p> <p>Mount the correct input and start the run over.</p>

Table 5-1 (cont). Library Processor C Error Conditions

Contents of B-Address Register	Console Typewriter Message	Cause and Action
0culd	: RD ER cu d □	<p>If "cu" is the number of a tape control unit, an uncorrectable read error has occurred on tape "d" of that control unit.</p> <p>Continue the run to retry the correction procedure.</p> <p>If this action is not effective, corrective measures such as cleaning the tape and re-starting the run, recreating the tape which caused the error, etc., should be performed.</p>
0cull	: RD ER cu l □	<p>If "cu" is the control unit number of the card reader, a hole-count error has occurred.</p> <ol style="list-style-type: none"> 1. Remove the cards from the input hopper of the card reader. 2. Run-out the cards in the reader. 3. Place the cards from the run-out hopper back into the card reader followed by the remaining input cards. <p>Continue the run.</p>
0cul2	: RD ER cu 2 □	<p>If "cu" is the control unit number of the card reader, an illegal punch has been detected.</p> <p>The same action is taken in this case as is outlined for a hole-count error above.</p> <p>NOTE: The first card in the run-out hopper must be corrected to remove the illegal punch.</p>
0cu2d	: WR ER cu d □	<p>If "cu" is the number of a tape control unit, an uncorrectable write error has occurred on tape d of that control unit.</p> <p>Continue the run to retry the correction procedure.</p> <p>If this action is not effective, corrective measures such as cleaning the tape and re-starting, repeating the run which created the tape, etc., should be performed.</p>
0cu2l	: WR ER cu l □	<p>If "cu" is the control unit number of the card punch, a hole-count error has been detected.</p> <p>Continue the run to repunch the error cards.</p>

Table 5-1 (cont). Library Processor C Error Conditions

Contents of B-Address Register	Console Typewriter Message	Cause and Action
0cu10	: RD ER cu 0 □	<p>If "cu" is the number of the paper tape control unit, a parity error has been detected on paper tape.</p> <p>Continue the run to ignore the error.</p> <p>(The operator may make note of which program is currently being processed.)</p>

APPENDIX A

PROCEDURE FOR CHANGING STANDARD EQUIPMENT CONFIGURATIONS

Although the standard equipment configurations accommodate normal applications, a user may generate new standards by reassembling Library Processor C. Standard configurations are often changed to allow Library Processor C paper tape input, or to type messages on the console.

Each of the ten standard configurations is an 80-character area (ECD area) of Library Processor C. The contents of this area correspond exactly to a full ECD card image. The leftmost character of each ECD area is tagged ECDn, where n is the number of the standard configuration corresponding to that ECD image.

Each ECD area is coded as shown in Figure A-1, which illustrates the ECD area for Library Processor C configurations 2, 5, 6 and 8.

- To allow Library Processor C, or any system program, to use a console typewriter, change the CONSOLE DEVICE field of each ECD area to

DC @ 5u0 @ CONSOLE DEVICE

where u is the address of the typewriter control unit.

- To allow input to be read from the paper tape reader, change the CARD INPUT DEVICE field to

DC @ Lu0 @ CARD INPUT DEVICE

where u is the address of the paper tape reader control unit.

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER	OPERATION CODE	LOCATION	OPERANDS	
			14 15	20 21
1	DC	AEC02	@AAAAEA@	
2	DC		#3C515253	RWC Assignment
3	DC		@AAAAA@	
4	DC		@62@	Highest Bank Available
5	DC		@10@	Loading Device
6	DC		@20@	Console Device
7	DC		@JJ@	Card Input Device
8	DC		@181@	Listing Device (Not Used)
9	DC		@18@	Card Output Device
10	DC		@AAA@	File 1: Library SPT
11	DC		@AAA@	
12	DC		@AAA@	
13	DC		@AAA@	
14	DC		@AAAC	
15	DC		@AAA@	
16	DC		@AAA@	
17	DC		@AAA@	
18	RESV,		28	
19	DCW	L	#1C@	

Figure A-1. Sample Coding for Library Processor C Standard Configurations 2, 5, 6, and 8

COMPUTER-GENERATED INDEX

ADDING MACRO ROUTINES TO THE SPT, 2-4
 ARRANGEMENTS
 INPUT AND OUTPUT DECK ARRANGEMENTS, 4-6
 ASSEMBLY INPUT, 4-5
 BOOTSTRAP PROCEDURES, 5-1
 CALL CARD
 CONSOLE CALL CARD, 4-1
 CARD
 CONSOLE CALL CARD, 4-1
 CONTINUATION CARDS, 3-1
 END-OF-FILE CARD, 4-5
 " INPUT/CARD OUTPUT, 4-2
 " INPUT/TAPE OUTPUT, 4-2
 " NUMBERS, 3-3
 SYSTEM SPECIFIC HEADER CARD, 4-5
 CHANGING STANDARD EQUIPMENT CONFIGURATIONS
 PROCEDURE FOR CHANGING STANDARD EQUIPMENT
 CONFIGURATIONS, A-1
 CODING
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 SELECTIVE OMISSION OF CODING, 2-1
 CONDITIONS
 ERROR CONDITIONS, 5-1
 LIBRARY PROCESSOR C ERROR CONDITIONS, 5-2
 CONFIGURATION
 " DIRECTOR,
 EQUIPMENT CONFIGURATION DIRECTOR, 4-1
 FULL EQUIPMENT CONFIGURATION DIRECTOR, 4-3
 " NUMBERS,
 STANDARD EQUIPMENT CONFIGURATION NUMBERS, 4-2
 CONFIGURATIONS
 PROCEDURE FOR CHANGING STANDARD EQUIPMENT
 CONFIGURATIONS, A-1
 " 256,
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 CONSOLE CALL CARD, 4-1
 CONTINUATION CARDS, 3-1
 DECK
 " ARRANGEMENTS,
 INPUT AND OUTPUT DECK ARRANGEMENTS, 4-6
 INPUT DECK, 4-5
 USING LIBRARY PROCESSOR C TO PRODUCE SYMBOLIC DECKS,
 3-4
 DESCRIPTION
 GENERAL DESCRIPTION OF LIBRARY PROCESSOR C, 1-1
 DESIGNATION
 PARAMETER DESIGNATION, 2-1
 DIRECTOR
 EQUIPMENT CONFIGURATION DIRECTOR, 4-1
 FULL EQUIPMENT CONFIGURATION DIRECTOR, 4-3
 END-OF-FILE CARD, 4-5
 EQUIPMENT
 " CONFIGURATIONS,
 PROCEDURE FOR CHANGING STANDARD EQUIPMENT
 CONFIGURATIONS, A-1
 " REQUIREMENTS, 1-2
 EQUIPMENT CONFIGURATION
 " DIRECTOR, 4-1
 FULL EQUIPMENT CONFIGURATION DIRECTOR, 4-3
 " NUMBERS,
 STANDARD EQUIPMENT CONFIGURATION NUMBERS, 4-2
 ERROR CONDITIONS, 5-1
 LIBRARY PROCESSOR C ERROR CONDITIONS, 5-2
 FIELDS
 FORMAT FOR FILE MEDIA FIELDS, 4-4
 FILE
 INPUT AND OUTPUT FILES, 4-1
 INPUT FILE, 4-1
 " MEDIA FIELDS,
 FORMAT FOR FILE MEDIA FIELDS, 4-4
 OUTPUT FILE, 4-6
 FORMAT FOR FILE MEDIA FIELDS, 4-4
 FUNCTIONS OF LIBRARY PROCESSOR C, 1-2
 GENERAL DESCRIPTION OF LIBRARY PROCESSOR C, 1-1
 HEADER CARD
 SYSTEM SPECIFIC HEADER CARD, 4-5
 INITIAL SETUP PROCEDURES, 5-1
 INPUT
 " AND OUTPUT DECK ARRANGEMENTS, 4-6
 " AND OUTPUT FILES, 4-1
 ASSEMBLY INPUT, 4-5
 " DECK, 4-5
 " FILE, 4-1
 INPUT/CARD OUTPUT
 CARD INPUT/CARD OUTPUT, 4-2
 INPUT/TAPE OUTPUT (CONT.)
 INPUT/TAPE OUTPUT
 CARD INPUT/TAPE OUTPUT, 4-2
 TAPE INPUT/TAPE OUTPUT, 4-3
 INSTRUCTION
 WRITING A MACRO INSTRUCTION, 3-1
 LIBRARY PROCESSOR
 " C ERROR CONDITIONS, 5-2
 FUNCTIONS OF LIBRARY PROCESSOR C, 1-2
 GENERAL DESCRIPTION OF LIBRARY PROCESSOR C, 1-1
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 USING LIBRARY PROCESSOR C TO PRODUCE SYMBOLIC DECKS,
 3-4
 MACRO
 " INSTRUCTION,
 WRITING A MACRO INSTRUCTION, 3-1
 " ROUTINE,
 ADDING MACRO ROUTINES TO THE SPT, 2-4
 WRITING A MACRO ROUTINE, 2-1
 MEDIA FIELDS
 FORMAT FOR FILE MEDIA FIELDS, 4-4
 NUMBERS
 CARD NUMBERS, 3-3
 STANDARD EQUIPMENT CONFIGURATION NUMBERS, 4-2
 OMISSION
 SELECTIVE OMISSION OF CODING, 2-1
 SELECTIVE OMISSION OF PARAMETERS, 3-1
 OPERATING PROCEDURES, 5-1
 OUTPUT
 CARD INPUT/CARD OUTPUT, 4-2
 CARD INPUT/TAPE OUTPUT, 4-2
 " DECK ARRANGEMENTS,
 INPUT AND OUTPUT DECK ARRANGEMENTS, 4-6
 " FILE, 4-6
 INPUT AND OUTPUT FILES, 4-1
 TAPE INPUT/TAPE OUTPUT, 4-3
 PARAMETER DESIGNATION, 2-1
 PARAMETERS
 SELECTIVE OMISSION OF PARAMETERS, 3-1
 PREFIXES
 TAG PREFIXES, 2-3
 PROCEDURE
 BOOTSTRAP PROCEDURES, 5-1
 " FOR CHANGING STANDARD EQUIPMENT CONFIGURATIONS, A-1
 INITIAL SETUP PROCEDURES, 5-1
 OPERATING PROCEDURES, 5-1
 PROCESSOR
 FUNCTIONS OF LIBRARY PROCESSOR C, 1-2
 GENERAL DESCRIPTION OF LIBRARY PROCESSOR C, 1-1
 LIBRARY PROCESSOR C ERROR CONDITIONS, 5-2
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 USING LIBRARY PROCESSOR C TO PRODUCE SYMBOLIC DECKS,
 3-4
 REQUIREMENTS
 EQUIPMENT REQUIREMENTS, 1-2
 ROUTINE
 ADDING MACRO ROUTINES TO THE SPT, 2-4
 WRITING A MACRO ROUTINE, 2-1
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 SELECTIVE OMISSION
 " OF CODING, 2-1
 " OF PARAMETERS, 3-1
 SETUP PROCEDURES
 INITIAL SETUP PROCEDURES, 5-1
 SPECIFIC HEADER CARD
 SYSTEM SPECIFIC HEADER CARD, 4-5
 SPT
 ADDING MACRO ROUTINES TO THE SPT, 2-4
 STANDARD
 " CONFIGURATIONS 256,
 SAMPLE CODING FOR LIBRARY PROCESSOR C STANDARD
 CONFIGURATIONS 2,5,6, AND, A-1
 " EQUIPMENT CONFIGURATION NUMBERS, 4-2
 " EQUIPMENT CONFIGURATIONS,
 PROCEDURE FOR CHANGING STANDARD EQUIPMENT
 CONFIGURATIONS, A-1
 SYMBOLIC DECKS
 USING LIBRARY PROCESSOR C TO PRODUCE SYMBOLIC DECKS,
 3-4
 SYSTEM SPECIFIC HEADER CARD, 4-5
 TAG PREFIXES, 2-3
 TAPE INPUT/TAPE OUTPUT, 4-3
 WRITING
 " A MACRO INSTRUCTION, 3-1
 " A MACRO ROUTINE, 2-1

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: LIBRARY PROCESSOR C

DATED: OCTOBER 28, 1965

FILE NO: 122.1605.001C.0-051

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME _____

DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

Cut Along Line

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.



BUSINESS REPLY MAIL
No postage stamp necessary if mailed in the United States
POSTAGE WILL BE PAID BY

HONEYWELL
ELECTRONIC DATA PROCESSING DIVISION
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATT'N: PUBLICATIONS AND WRITING SERVICES DEPARTMENT

Honeywell
ELECTRONIC DATA PROCESSING