HONEYWELL EDP

SOFTWARE MANUAL

SERIES 200

TAPE LOADER-MONITOR C

GENERAL SYSTEM: SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT: Program Operating Procedures for the Tape Loader-Monitor C Program.

SPECIAL INSTRUCTIONS: This software bulletin supersedes the information bulletin entitled Plus—Tape Loader-Monitor (DSI-327A).

DATE: April 25, 1966

FILE NO. 123.5305.021C.0-221*

9663
8M
3467
Printed in U. S. A.

*When ordering this publication please specify Title and Underscored portion of File Number.

FOREWORD

Tape Loader-Monitor C is one of several loaders available in the Mod 1 Operating System. Others are: Card Loader-Monitor B, Floating Tape Loader-Monitor C, and Drum Monitor C.

This manual consists of four main sections. Section I provides basic information that is necessary for the simplest operation of the Loader-Monitor. Section II describes the operation of the Loader-Monitor in detail - its various functions, the communication area, and the parameters and returns that control operation. Section III is programmer-oriented, presenting the considerations and procedures he needs for sequencing programs and segments with examples that outline certain optional methods of operation. Section IV is operator-oriented, providing complete procedures for all methods of operation.

The reader is assumed to be familiar with the Honeywell Series 200 Programmers' Reference Manual, Order No. 139 and the Honeywell Series 200 Equipment Operators' Manual, Order No. 040. Additional related information is presented in the Easycoder Assemblers C and D Manual, Order No. 041.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

LIST OF TABLES

SECTION I
INTRODUCTION

GENERAL DESCRIPTION

Tape Loader-Monitor C is an operation control program in the Mod 1 Operating System that loads program units from binary run tapes (BRT). It is the only program permanently in memory during a run, and as such it has a central function in the system. Whenever another program unit on the BRT is desired, a call (by the operator or the program currently in control) is made to the Loader-Monitor. The Loader-Monitor then searches the tape until it finds the desired unit, loads it, and either starts that unit, returns to the program that made the call, or transfers control to some other predetermined routine.

Tape Loader-Monitor C is available in two versions: a three-character version, Tape Loader-Monitor (3C), and a four-character version, Tape Loader-Monitor (4C). Tape Loader-Monitor (3C) can load and start programs only in the first 32K characters of memory, whereas Tape Loader-Monitor (4C) may load and start units anywhere up to 262K - in all other respects they are the same. They both have the same program and segment names (AAAMONS1). When stored on the same BRT, they are distinguished by visibility, with the standard version usually assigned visibility A.

EQUIPMENT REQUIREMENTS

Minimum Equipment Required

1. Main memory locations 64 through 1,339 (decimal).
2. Index registers X5 and X6.
3. Advanced Programming Instructions (Feature 1011, 010 or 011).
4. One half-inch tape drive.

Additional Equipment Usable

A card reader is useful for entering parameters with a call card.

STANDARD PROCEDURES

The Tape Loader-Monitor is a flexible program that may be operated in a number of different ways. To fully use all its various options and features, you must have a detailed knowledge of the material presented in later sections. But it is also possible, with the following procedures for standard operation, to use the Loader-Monitor easily and simply without going into the complexities of the various options.

The Loader-Monitor is stored at the beginning of any BRT. To load it, you mount the BRT on logical drive 0, bootstrap twice into location 0, and press RUN twice. It is then ready to load other programs desired for the run. Next, you put call cards for the desired programs in the card reader,[1] and press RUN. The Loader-Monitor will read the first call card, search the tape by program and segment name until it finds the requested program unit, load it into the locations specified by assembly, and then start execution of this unit by branching to its normal starting address.

When this program unit has completed its processing, it should branch back to the Loader-Monitor. If this is an indirect branch to location 139, the Loader-Monitor will halt (Halt 3). The operator presses RUN. The Loader-Monitor reads the next call card and proceeds as before. Alternatively, the program currently in control may call for another program unit without a halt or call card. To do this, it moves the program name of the desired unit into locations 68 through 73 and the segment name into locations 74 and 75. It then returns control to the Loader-Monitor with a direct branch to location 130. See Example 1 on page 3-3.

Operating Procedure

1. Initialize the central processor and activate the peripheral devices to be used.
2. Mount the program BRT on logical drive 0. Be sure that the tape is rewound. Remove the write-enable ring. Set the PERMIT-PROTECT switch to PROTECT.
3. Bootstrap the BRT (pcu address = 40_8) into location 0, twice.
4. Press RUN. When Halt 1 (B-Address = 17001) occurs, press RUN again.
5. When Halt 3 (B-Address = 17002) occurs, the Loader-Monitor is in memory and awaiting a call for the first program.
6. Activate the card reader. Arrange the call cards for the desired programs, with any control and data cards that are needed, in the order these programs will be wanted during the run. Put the correctly arranged deck in the card reader. Press RUN.
7. When Halt 3 occurs again, press RUN to load and start the next program.

   If other halts occur, refer to Table 4-1 and/or the procedures for the program that has just been called.

Programming Procedure

1. ORG to 1340 or above. Do not store anything in locations 64 through 1339. This area is reserved for the Loader-Monitor.

[1] The format and contents of a call card are presented on page 3-2. Any control or data cards for a particular program should come immediately after its call card.

2. Do not load into or clear index registers X5 or X6. The Loader-Monitor uses X5 and X6 as distribution registers.

3. The last executable instruction in any program unit must be a branch to the loader.

   a. An indirect branch to 139 will cause a console call halt (Halt 3). The operator then presses RUN, which reads the next call card and loads and starts the requested program unit.

   b. A direct branch to 130 will start the loader without halting or reading a call card. Before the branch is made, the program must move the program and segment name of the next program unit desired into locations 68-73 and 74-75. See Example 1 on page 3-3.

SECTION II
DETAILED DESCRIPTION

GENERAL

Tape Loader-Monitor C performs three related functions:

1. Searching the BRT for the requested program unit.
2. After finding the program unit, loading it.
3. Starting the loaded unit, the program that made the call, or some other routine in memory.

These three functions are controlled through a communication area (see Table 2-1), which is located in main memory locations 64 through 155. This communication area contains:

1. Parameters that control the searching, loading, and starting operations of the Loader-Monitor.
2. Returns for transferring control back to the Loader-Monitor.
3. Exit and Return points for own-coding associated with the loading operation.
4. Parameters that are provided for use by other programs.

All the parameters in the communication area, except for Program Name, Segment Name, and Halt Name, are initially set to certain standard values. Some of these are automatically reset by a Console Call, a Special Call, or after loading a program unit. "Initial Value," as used in the following paragraphs, refers to the contents of a parameter field within the communication area which are established when the Loader-Monitor itself is initially loaded into memory. "Reset" refers to a value entered by the Loader-Monitor into a parameter field within the communication area at a Console Call, Special Call, or after loading a program unit.

A complete sequential layout of the communication area is presented in Table 2-1. Following that, the searching, loading, and starting functions of the Loader-Monitor are described including the parameters that control each function. Then follows an explanation of the various return points for returning control to the Loader-Monitor. Additional features in the communication area are described at the end of the section. The own-code exit and returns are explained in Appendix B.

Table 2-1. Sequential Layout of the Loader-Monitor Communication Area

Location		Function	Initial Value	Reset		
Decimal	Octal			Console Call	After Loading	Special Call
64	100	Method of Console Call Parameter Entry (card or manual)	0			
65-67	101-103	Revision Number of Unit Last Loaded	Δ			
68-73	104-111	Program Name	Δ			
74-75	112-113	Segment Name	Δ			
76	114	Tape Drive Number	0			
77-84	115-124	Halt Name	Δ			
85	125	ID Character from Column 18 of Call Card	0			
86-89	126-131	Fixed Start 0 (Manual Return for Console Call)				
90-93	132-135	Fixed Start 1 (Not used by Loader-Monitor)				
94-97	136-141	Fixed Start 2 (Not used by Loader-Monitor)				
98-101	142-145	Fixed Start 3 (Not used by Loader-Monitor)				
102-105	146-151	Exit to Own-Code Routine	*	*	*	
106	152	Search Direction	22	22	22	
107-109	153-155	Relocation Augment	0	0	0	
110	156	Relative Position	01	01	01	
111	157	Search Mode	20	20		01
112	160	Start Mode	N	N		N
113-118	161-166	Visibility Mask	40 00 00 00 00 00			
119-121	167-171	Special Start Location	0			
122-125	172-175	Own-Code Return 1 (before distribution)				
126-129	176-201	Own-Code Return 2 (after distribution)				
130-138	202-212	Return for Normal Call				
139-141	213-215	General Return Address (Halt for Console Call)				
142-146	216-222	Current Date	Δ			
147	223	Trapping Mode	0			

Table 2-1 (cont). Sequential Layout of the Loader-Monitor Communication Area

Location		Function	Initial Value	Reset		
Decimal	Octal			Console Call	After Loading	Special Call
148-150	224-226	Alternate Return Address (Read another Call Card)				
151-154	227-232	ECD Field	JJ0#			
155	233	Console Typewriter Availability	item mark			

NOTES: All fields in the communication area except 155 (decimal) have a word mark in their leftmost location — punctuation in the communication area must not be changed.

* - The Exit to Own-Code parameter initially and after reset assumes that there is no own-coding.

SEARCHING

Programs and segments may be arranged on a BRT in any order. Obviously, the first thing the Loader-Monitor must do is to search the designated tape until it finds the desired program unit. Each unit on the tape is identified by a Program Name, a Segment Name, and a Visibility Key. Using these identifiers, the Loader-Monitor searches for a requested unit in one of five ways:

1. By program and segment name.
2. By segment name (within a given program).
3. By visibility and relative position.
4. By program name, segment name, and visibility.
5. By segment name and visibility (within a given program).

The Loader-Monitor can search either forward or backward - it is normally set to search forward. If the requested unit is not found in a forward search of the tape, the Loader-Monitor will search the tape backward. If the unit is not found in a backward search, the loader will halt and await operator action. A forward search may be initiated with the BRT positioned anywhere between the header label record and the trailer label record. A backward search may be initiated only if the BRT is positioned immediately after the unit last loaded.

Search Parameters

Parameter	Location		Values	Initial Value	Reset at Console Call	Reset After Loading	Reset at Special Call
	Decimal	Octal					
Search Mode	111	157	01: vis. & rel. pos. 20: prog. & seg. 00: seg. (within prog.) 60: prog., seg., vis. 40: seg. & vis. (within prog.)	20	20	-	01
Search Direction	106	152	22: forward 23: backward	22	22	22	-
Program Name	68-73	104-111	-	Δ	-	-	-
Segment Name	74-75	112-113	-	Δ	-	-	-
Visibility Mask	113-118	161-166	-	40 00 00 00 00 00	-	-	-
Relative Position	110	156	1 (back.): prog. last loaded. 1 (fwd): next prog.	1	1	1	-
Tape Drive Number	76	114	-	0	-	-	-

SEARCH MODE (Location 111)

The search mode is established by entering one of the five search mode designations into location 111 (octal 157). The octal values of the five search mode designations and the meaning of each are provided in Table 2-2, together with the identifying parameters used when searching in the designated mode.

Table 2-2. Search Mode Designations

Search Mode (Octal)	Parameters Used	Meaning
01	Visibility and Relative Position	Search in the specified direction, and load the nth unit of specified visibility (n is the binary value of the relative position parameter).
20	Program and Segment Names	Search for and load the unit with the specified program and segment names, irrespective of visibility.
00	Segment Name	Search for and load the unit with the specified segment name within the current program. (Similar to search mode 20 except that the machine is directed to halt upon encountering a segment header record with a program name different from that of the unit last loaded. A search in the opposite direction can then be initiated by depressing the RUN button on the operator's control panel.)

Table 2-2 (cont). Search Mode Designations

Search Mode (Octal)	Parameters Used	Meaning
60	Program Name, Segment Name, Visibility	Search for and load the unit with the specified program name, segment name, and visibility.
40	Segment Name, Visibility	Search for and load the unit with the specified segment name and visibility within the current program. (This mode is similar to 00 with the addition of the visibility key.)

The initial value of the Search Mode parameter is 20, and it is reset to 20 by a console call. However, upon receipt of a special program call (Section III), the Search Mode parameter is reset to 01.

SEARCH DIRECTION (Location 106)

The search direction is established by entering octal 22 (search forward) or octal 23 (search backward) into location 106 (octal 152). The initial value of the Search Direction parameter is octal 22. It is reset to 22 by a console call, and it is reset to 22 after loading each unit.

PROGRAM NAME (Locations 68-73)

The Program Name is one of the search parameters (keys) used in search modes 20, 00, 60, and 40. Thus, in all search modes except 01, the calling unit (or the calling or loading operation of a previously loaded unit) enters the program name of the specified unit into locations 68 to 73 (octal 104 to 111). However, regardless of the search mode used, the program name of the applicable program unit is always entered into locations 68 to 73 by the Loader-Monitor during the loading operation, thus enabling subsequent use of search modes 00 and 40 without the necessity of entering the program name.

SEGMENT NAME (Locations 74-75)

Similar to the Program Name, the Segment Name is one of the search parameters used in search modes 20, 00, 60, and 40; thus, for these modes, the segment name is entered by the calling unit into locations 74 and 75 (octal 112 and 113). Similarly, the segment name is always entered into locations 74 and 75 by the Loader-Monitor during the loading operation.

VISIBILITY MASK (Locations 113-118)

Visibility is one of the search parameters (keys) used in search modes 01, 60, and 40. The initial value of the visibility mask in the communication area (as established by the Loader-

Monitor) is octal 40 00 00 00 00 00 (visibility A). When searching is by visibility, a visibility match must be obtained to identify the desired program unit prior to loading. A visibility match occurs when at least one bit position in the visibility key of the program unit and the corresponding bit position of the visibility mask both contain a 1. When it is desired to search for a program unit by a visibility other than A, the desired visibility code must be entered into locations 113 to 118 (octal 161 to 166) by either manual or program entry as described in Section III.

RELATIVE POSITION (Location 110)

The Relative Position parameter is a binary number (n) that is used in conjunction with search mode 01. The initial value of the Relative Position parameter is 1. The unit last loaded may be reloaded with a relative position value of 1 if the Search Direction parameter is set for backward search. The Relative Position parameter is reset to 1 by a console call, and it is reset to 1 after loading each unit. The value 0 for the Relative Position parameter is undefined.

TAPE DRIVE NUMBER (Location 76)

The Tape Drive Number parameter identifies the tape drive from which a unit is to be loaded. The low-order three bits of the character which is entered into location 76 (octal 114) specify the number of the tape drive containing the desired program unit. (The peripheral instructions are specialized from this character before executing each call.) In contrast to the tape drive, the tape control is not designated by a search parameter; if the tape control number is not 0, the appropriate number must be entered through the operator's control panel when the bootstrap procedure is performed (Section IV). The bootstrap routine and the Loader-Monitor both direct the use of tape control 0, unless another tape control number is entered during the bootstrap procedure. Thereafter the tape control used during the bootstrap procedure is automatically addressed, and the particular type drive addressed is determined by the value of the low-order three bits in the Tape Drive Number parameter location.

LOADING

After finding the requested program unit, the Loader-Monitor will automatically start loading it into main memory.[1] Loading consists of reading and then distributing and punctuating each successive record of the unit. Successively, each record is read into a buffer. From there, as specified by control characters in the record, the instructions and constants are distributed to specified memory locations and punctuated.

[1] Before starting to load, the Loader-Monitor enters the program name, segment name, and revision number of the unit into the communication area.

Between the reading and distributing phases of loading, it is possible to execute own-coding routines. After reading a record into the buffer, the Loader-Monitor will branch to the own-coding routine. This routine may return to use the Loader-Monitor's own distribution routine (Own-Code Return 1), or it may do its own distribution (if the tape does not have a standard BRT format) and return to the read routine of the Loader-Monitor (Own-Code Return 2). With both versions (3C and 4C) of the Loader-Monitor, exits and returns must be made in three-character mode and below 32K. For a more detailed discussion of own-coding considerations, refer to Appendix B.

The Loader-Monitor, using the Relocation Augment parameter, can load a unit into higher locations than the ones specified at assembly. However, it does not modify DSA's or address fields of instructions, so this feature is useful only if indexed addressing has been used throughout the program.

Version (3C) will load as high as location 32,767. Version (4C) will load as high as 262,143.

The Loader-Monitor uses and does not restore index registers X5 and X6. These registers have word marks in their leftmost character locations at the completion of loading. Index register X6 is the distribution counter which contains an address one higher than that into which the last character of the called unit was loaded. The Loader-Monitor does not use or disturb any locations below location 64 (octal 100), other than index registers X5 and X6.

After a program unit has been loaded, the Loader-Monitor resets the Relative Position parameter to 01, Relocation Augment to 0, Search Direction to 22, and Exit to Own-Coding to assume no own-coding. It will then either halt (Halt 6) or proceed to the starting operation.

Load Parameters

Parameter	Location		Values	Initial Value	Reset at Console Call	Reset After Loading	Reset at Special Call
	Decimal	Octal					
Relocation Augment	107-109	153-155	-	0	0	0	-
Halt Name	77-84	115-124	-	-	-	-	-

RELOCATION AUGMENT (Locations 107-109)

The Relocation Augment parameter is a three-character (18-bit) binary number that is added to:

1. The locations of the loading data that have been specified by assembly.

2. The high- and low-order addresses of any area to be cleared.

3. The loaded unit's normal starting address.

The initial value of the Relocation Augment is 0; it is reset to 0, both after a program unit has been loaded and at console call.

HALT NAME (Locations 77-84)

The Halt Name parameter is used to halt operation after a specified program unit has been loaded. The program and segment names of that program are entered into locations 77 to 84 (octal 115 to 125) of the communication area. This combined program and segment name constitutes the "halt name"; and locations 77 to 84 are treated as one field with only one word mark set in the field at location 77. The Loader-Monitor compares this parameter with the name on the segment header record of each program unit that is loaded. If the comparison indicates that this is the specified program unit, the machine halts (Halt 6) after loading it. Depressing the RUN button on the operator's control panel causes the Loader-Monitor to continue as directed by the start parameters.

STARTING

After a program unit has been loaded, the Loader-Monitor will do one of three things.

1. Start the program unit just loaded, by branching to the normal starting location. (Normal Start)

2. Transfer control to some predetermined routine in memory, by branching to the special starting location specified in locations 119 through 121. (Special Start)

3. Return control to the program unit that made the call, by branching to the location immediately following the location from which the call to the Loader-Monitor was made. (Return)

Start Parameters

Parameter	Location		Values	Initial Value	Reset at Console Call	Reset After Loading	Reset at Special Call
	Decimal	Octal					
Start Mode	112	160	N: normal S: special R: return	N	N	-	N
Special Start Location	119-121	167-171	-	0	-	-	-
Trapping Mode	147	223	00: off 04: on	00	-	-	-

START MODE (Location 112)

One of the three start mode parameter values is entered intc location 112 (octal 160) of the communication area to specify the method of initiating the start phase of the Loader-Monitor program. The value and meaning of each of the three start modes are listed in Table 2-3.

Table 2-3. Start Mode Designations

Parameter Value		Meaning
Alphabetic	Octal	
N	45	Branch to the normal starting location in the program unit just loaded. The relocation augment is added to this address before the branch is executed. Version (4C) will execute this in three-character mode if the starting address is below 32,768 and in four-character mode if 32,768 or above.
S	62	Branch to the location specified by the "special start location" parameter value entered in locations 119 to 121. (The relocation augment is not added to this address.)
R	51	Branch to the location in the calling program (unit) immediately following the location from which the call to the Loader-Monitor was made. Note that this mode can be used only when control is returned to the Loader-Monitor via a branch to location 130. Neither version (3C or 4C) can return to a location higher than 32,767.

The initial value of the Start Mode parameter (as established by the Loader-Monitor program) is N; it is reset to N by a console call, and it is also reset to N by a special call (Section III).

SPECIAL START LOCATION (Locations 119-121)

The Special Start Location parameter is used in conjunction with the Start Mode parameter value S. Specifically, the Special Start Location parameter value is a three-character address, entered into locations 119 to 121 (octal 167 to 171) of the communication area. The Loader-Monitor program branches to the special start location to start the called program when the Start Mode parameter value entered in location 112 is S. The initial value of the Special Start Location parameter is 0.

TRAPPING MODE (Location 147)

The Trapping Mode parameter may be used in conjunction with any of the three start mode parameter values N, S, or R. The trapping mode is established by entering the octal value 04 into location 147 (octal 223). Once the trapping mode has been entered, the value 00 must be entered into location 147 if it is desired to return to the non-trapping mode. Immediately before starting a called program, the Loader-Monitor references the trapping mode parameter value and establishes either the non-trapping mode (00) or the trapping mode (04). Thus, a calling

program, or the operator, can designate whether the trapping mode will be in effect when the called unit is started. The initial value of the Trapping Mode parameter is 00.

RETURNS TO THE LOADER-MONITOR

The return points in the communication area provide several methods of returning control to the Loader-Monitor. These returns are described below. Methods for using them are described in Section III.

Return Points

RETURN FOR NORMAL CALL (Locations 130-138)

This return to the Loader-Monitor is used to load another program unit automatically without any operator action. The program unit making the call changes the appropriate parameter values in the communication area and then branches to the Return for Normal Call, location 130. When this return is used, the Loader-Monitor does not reset any parameter values - any changes in parameter values must be made by the program in control before it branches to 130.[1]

Locations 130 through 134 contain an SCR instruction that stores the address used for the Return start mode. Locations 135 through 138 contain a branch to the search routine of the Loader-Monitor.

GENERAL RETURN ADDRESS (Locations 139-141)

The General Return Address is used as a program return to set up a console call. The program unit in control, without changing values in the communication area, branches indirectly to location 139 - B/(139). This resets the Start Mode parameter to N, Search Mode to 20, Relative Position to 01, Relocation Augment to 0, Search Direction to 22, and Exit to Own-Code to assume no own-coding; and then causes a console call halt (Halt 3) for changing parameter values. These values may be entered by the operator either with a call card or manually through the control panel.

Locations 139 through 141 normally contain the address of the console call routine in the Loader-Monitor. But when a series of programs are to be executed as a system, with a user-written control program, the control program may change the contents of the General Return Address to the address of some routine within itself. In this case, all of the program units in

[1] Note that the Search Direction, Relocation Augment, and Relative Position parameters were reset when the calling unit was loaded. See Table 2-1.

the series terminate with an indirect branch to location 139 - B/(139). This has the effect of returning control to the system's control program, allowing it to determine which unit it wants to be loaded next and to make the appropriate call. After a systems run, the control program should restore the General Return Address to its initial value.

ALTERNATE RETURN ADDRESS (Locations 148-150)

The Alternate Return Address may be used, if the Entry Method parameter (location 64) contains 00, to read the next call card and load the unit it specifies. The program unit in control branches indirectly to location 148 - B/(148). If location 64 contains 00, the Loader-Monitor reads the next call card without resetting any parameter values or halting. The only parameter values changed are those specified on the call card.

FIXED START 0 (Locations 86-89)

Fixed Start 0 provides, for the operator, a manual return to the Loader-Monitor to set up a console call. Using the procedure given on page 4-2, the operator executes Fixed Start 0. This has the same effect as an indirect program branch to the General Return Address (location 139). It resets the Start Mode parameter to N, Search Mode to 20, Relative Position to 01, Relocation Augment to 0, Search Direction to 22, and Exit to Own-Code to assume no own-coding; and then causes a console call halt (Halt 3).

ADDITIONAL FEATURES

Console Fixed Starts

The communication area contains four Branch instructions that may be used for console starts. The first Branch instruction is designated as Fixed Start 0, and it is stored in locations 86 to 89 (octal 126 to 131). The execution of Fixed Start 0 is described in the preceding paragraph. The other three Branch instructions (locations 90 through 101) are available for use by the object programs. The Branch op code is located in the leftmost location of each four-character field, followed by the three-character A-address portion. The initial value in each is a Branch to location 0.

Parameters

Several parameters in the Loader-Monitor communication area are provided for the use and convenience of other programs and are not used by the Loader-Monitor itself.

Parameter	Location		Values	Initial Value	Reset at Console Call	Reset After Loading	Reset at Special Call
	Decimal	Octal					
Revision Number	65-67	101-103	-	Δ	-	-	-
Current Date	142-146	216-222	-	Δ	-	-	-
ECD Field	151-154	227-232	-	JJ0#	-	-	-
Console Typewriter Availability	155	233	Item Mark: no console typewriter Word Mark: console typewriter	Item Mark	-	-	-

REVISION NUMBER (Locations 65-67)

After a called unit has been found on the tape, before starting to load, the Loader-Monitor enters the revision number of this unit into the Revision Number parameter, locations 65 through 67. This is provided for use or reference by the operator or other programs.

CURRENT DATE (Locations 142-146)

The operator may enter the current date into locations 142 through 146, for reference by other programs. The standard format is for locations 142 and 143 to specify the year (00 to 99) and for locations 144 through 146 to specify the day of the year (001 to 366).

ECD FIELD (Locations 151-154)

The ECD Field is used by other systems programs in the Mod 1 Operating System. It should contain either a standard equipment configuration number or a device address from which to read an Equipment Configuration Descriptor. The initial value is JJ0# for Card Reader entry of the ECD. Any changes must be made manually - it is never automatically reset.

Device Address

Character	1	2	3	4
Contents	J	J	0	#

ECD Field to Obtain ECD Image from Card Reader:

Character Number	Location (Octal)	Contents
1	227	Device Type (J)
2-3	230-231	Control Unit and Device (J0)
4	232	Read/Write Channel (#)

Standard ECD Number

Character	1	2	3	4
Contents	Δ	X	t(Δ)	t(Δ)

Character Number	Location (Octal)	Contents
1	227	Blank (Δ)
2	230	Standard Configuration Number (X) $0 \leq X \leq 9$
3-4	231-232	Highest memory bank (octal) available to the system program. If these characters are blank, the memory size in the standard configuration is used. (tt) or (ΔΔ).

CONSOLE TYPEWRITER AVAILABILITY (Location 155)

The Console Typewriter Availability parameter is provided for reference by other programs. An item mark indicates that no console typewriter is available. A word mark indicates that a console typewriter is available for use by the program. The initial value is an item mark. Any changes must be made manually - it is never automatically reset.

SECTION III
CALLING PROCEDURES

To call a program unit, two operations are required:

1. Setting up the communication area with the desired parameter values.
2. Transferring control to the Loader-Monitor.

This may be done with a console call, where the operator enters the desired parameter values into the communication area; or it may be done with a program call, where the program unit initiating the call moves in the desired values.

CONSOLE CALLS

For a console call, the first step is to give control to the Loader-Monitor. This may be done by the operator, using fixed start 0, or it may be done by the program unit currently in control, using an indirect branch to the general return address (location 139). The Loader-Monitor, then, resets the start mode parameter to N, search mode to 20, relative position to 01, relocation augment to 0, search direction to 22, and exit to own-code to assume no own-coding, and halts at Halt 3.

The next step is for the operator to enter any desired changes of parameter values into the communication area, either with a call card or manually through the control panel. The method of entry chosen must be specified in location 64 of the communication area: 00 = card entry, 01 = manual entry. The initial value is 00. Any changes to the entry method parameter must be made manually - it is never automatically reset.

The last step is for the operator to start the Loader-Monitor.

Alternatively, if control is returned to the Loader-Monitor by branching indirectly to the alternate return address (location 148), the Loader-Monitor will automatically read the next call card and load that unit without resetting any parameter values or halting. In this case, location 64 must contain 00 for card entry.

Card Entry

A call card (Figure 3-1) can enter only four parameter values: program name, segment name, tape drive number, and halt name. Usually, in making a call, these are the only parameters that need to be changed - the Loader-Monitor automatically resets other parameters to standard values. The card, punched in special Honeywell code, is

read directly into the communication area (into locations 68 through 85), overlaying the previous contents. If any field on the card is left blank, blanks will be read into the corresponding locations in the communication area. Therefore, all of the parameter values, even ones that have not changed, must be punched on each card.

When looking for a call card, the Loader-Monitor will keep reading cards until it finds one with an asterisk in column 18 - if call cards are missing, the loader will read through the entire input deck with no result. A call card is read only through column 18 - anything punched in subsequent columns is ignored.

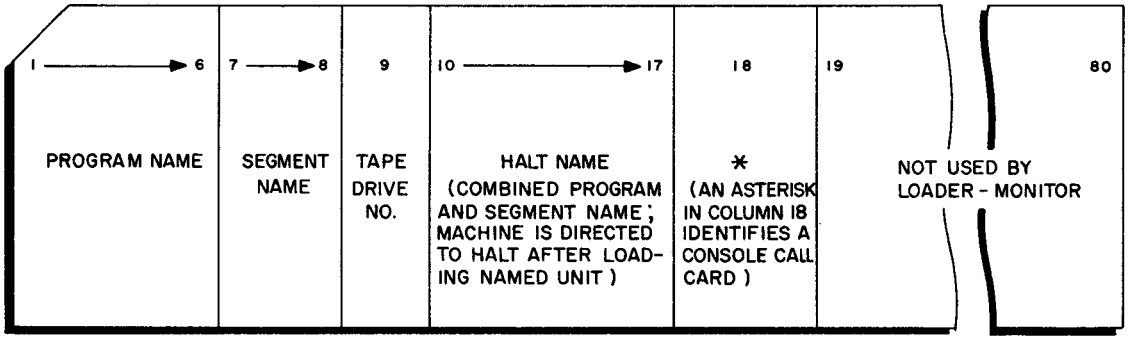

Figure 3-1. Call Card

Manual Entry

The value of any parameter in the communication area may be changed by the operator through the control panel. This method is normally used to change parameter values that cannot be entered with a call card, or at installations without a card reader. However, it can be used at any time as desired or needed. If no call card is to be read, the entry method parameter (location 64) must be set to 01.

PROGRAM CALLS

Once the first program unit has been loaded and started using a console call, subsequent program units may be loaded and started without operator action, using a program call. This may be either a normal program call or a special program call.

Normal Program Call

A Normal Program Call is performed by instructions in the current program. The program unit making the call first moves the desired parameter values into the communication area and then transfers control to the Loader-Monitor. The return branch to the Loader-Monitor is made to the return for normal call (location 130). This loads the requested unit without resetting any parameter values or halting.

When setting up parameters, the program unit making the call must not alter any punctuation in the communication area. All fields in the communication area, except 155, have a word mark in their leftmost location - and this punctuation must not be changed.

EXAMPLES

There are a number of different ways to code a normal call, depending on the sort of operation desired. Examples covering six different situations are described below. A seventh example is given for returns to the Loader-Monitor from above 32K.

Example 1 - Program and Segment Names

Call the program unit named PROCES AA, search Tape 0 in the forward direction by program name and segment name, and start PROCES AA at its normal starting location (see Figure 3-2). Note that the coding in Figure 3-2 does not include entries for Tape Drive Number, Search Mode, Search Direction, or Start Mode, since the desired values for these parameters are the initial values established by the Loader-Monitor.

EASYCODER
CODING FORM

PROBLEM EXAMPLE I dept. no. PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MAR	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 14	15 20	21 62 63 80
				MCW	PRNAME,73
				MCW	SGNAME,75
				B	13Ø
			PRNAME	DCW	@PROCES@
			SGNAME	DCW	@AA@

Figure 3-2. Symbolic Coding for Example 1

Example 2 - Visibility

Call the next program unit on tape 0 that is identified by visibility B, search in the forward direction, and start the specified unit at its normal starting location (see Figure 3-3). Note that the search mode and visibility are non-standard, so the desired values of these parameters must be entered by the program making the call, as in Figure 3-3.

EASYCODER
CODING FORM

PROBLEM EXAMPLE II PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MAR	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 14	15 20	21 62 63 80
				MCW	VIS,118
				MCW	MODE,111
				B	13Ø
			VIS	DCW	#6C2ØØØØØØØØØØØ
			MODE	DCW	#1CØ1

Figure 3-3. Symbolic Coding for Example 2

Example 3 - Visibility and Relative Position, Backward Search

Call the third program unit on tape 0 that is identified by visibility C, search in the backward direction, and start the specified unit at its normal starting location (see Figure 3-4). The Loader-Monitor directs a backward search of the BRT and increments a counter by one each time a program unit of visibility C is encountered. When the count reaches three, the specified unit has been located and is subsequently loaded and started at its normal starting location.

EASYCODER
CODING FORM

PROBLEM EXAMPLE III PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 ... 14	15 ... 20	21 ... 62
3				MCW	POS,110
4				MCW	VIS,118
5				MCW	REV,106
6				MCW	MODE,111
7				B	130
9			POS	DCW	#1C03
10			VIS	DCW	#6C10000000000
11			REV	DCW	#1C23
12			MODE	DCW	#1C01

Figure 3-4. Symbolic Coding for Example 3

Example 4 - Program and Segment Name and Visibility

Call the program unit named INITPR NN that also is identified by either visibility C or visibility D, search tape 0 in the forward direction, and start the specified unit at its normal starting location (see Figure 3-5).

EASYCODER
CODING FORM

PROBLEM EXAMPLE IV PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 ... 14	15 ... 20	21 ... 62
3				MCW	PR,73
4				MCW	SG,75
5				MCW	SRCH,111
6				MCW	VISIB,118
7				B	130
10			PR	DCW	@INITPR@
11			SG	DCW	@NN@
12			SRCH	DCW	#1C60
13			VISIB	DCW	#6C140000000000

Figure 3-5. Symbolic Coding for Example 4

Example 5 - Return Start, Backward Search

Call the program unit named PROCES AA from tape 0, search in the backward direction

and do not start execution after loading PROCES AA, but return to the next instruction in the program that made the call (the instruction which immediately follows B/130). See Figure 3-6.

EASYCODER
CODING FORM

PROBLEM EXAMPLE V PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 14	15 20	21 62 63 80
				MCW	PRNAME,73
				MCW	SGNAME,75
				MCW	REV,106
				MCW	NOSTRT,112
				B	130
			PRNAME	DCW	@PROCES@
			SGNAME	DCW	@AA@
			REV	DCW	#1C23
			NOSTRT	DCW	@R@

Figure 3-6. Symbolic Coding for Example 5

Example 6 - Relocation, Special Start

Call the program unit on tape 0 named AAAMEM S1, search in the forward direction, relocate the program unit 2500 octal locations higher, and start AAAMEM S1 at octal location 2510 (see Figure 3-7). (DSA's and the operand addresses of instructions are not altered by the Relocation Augment.)

EASYCODER
CODING FORM

PROBLEM EXAMPLE VI PROGRAMMER DATE PAGE OF

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5	6	7	8 14	15 20	21 62 63 80
				MCW	PN,73
				MCW	SN,75
				MCW	RELOC,109
				MCW	STMODE,112
				MCW	SPST,121
				B	130
			PN	DCW	@AAAMEM@
			SN	DCW	@S1@
			STMODE	DCW	@S@
			RELOC	DCW	#3C002500
			SPST	DCW	#3C002510

Figure 3-7. Symbolic Coding for Example 6

Example 7 - Return from Above 32K

When a program operating above 32K branches to the Loader-Monitor, it must ensure that the A-address register contains an address less than 32,768. Figure 3-8 illustrates one way this

can be accomplished (the SW is executed in the four-character addressing modes and is used only to put an address lower than 32K (64) into the A-address register).

EASYCODER
CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE ____ _____ PAGE ____ OF ____

	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1					SW	64	
2					CAM	ØØ	
3					B	13Ø	
4							
5							
6							

Figure 3-8. Symbolic Coding for Example 7

Special Program Call

The special program call is used only by programs operating in the Mod 1 environment that were originally written to be run in the Basic Programming System. It requires no parameter entry — the special call, by resetting the Search Mode parameter to 01 and the Start Mode parameter to N, loads the next program unit on the tape of the specified visibility and uses a normal start.

It is programmed as follows: In the program unit making the call, the first instruction must be an SCR instruction which moves the contents of the B-address register into the A-address portion of a return Branch instruction at the end of the unit. This returns control to the Loader-Monitor at the instruction that follows the Loader-Monitor starting Branch to the program unit, which is always interpreted by the Loader-Monitor as a special call (see Figure 3-9).

EASYCODER
CODING FORM

PROBLEM Special Program Call PROGRAMMER _____ DATE ____ _____ PAGE ____ OF ____

	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1				SEGX	SCR	EXIT,7Ø	
2							
3							
4							
5				EXIT	B	Ø	
6					EX	SEGX	
7							
8							
9							
10							
11							
12							
13							

Figure 3-9. Symbolic Coding for Special Program Call

CODING CONSIDERATIONS

1. Always ORG to 1340 or above. Do not store anything in locations 64 through 1339. This area is reserved for the Loader-Monitor.

2. Do not load into or clear index registers X5 or X6. The Loader-Monitor uses X5 and X6 as distribution registers (X5 addresses data in the buffer and X6 addresses the locations where the data is to be stored). To use X5 and X6, move the desired contents into the index register during execution with an LCA rather than storing there during the loading process with a DC or DCW.

3. The last executable instruction of any program unit should be a Branch to one of the Loader-Monitor returns - 130, (139), or (148).

4. A return start can be executed only if the previous return to the Loader-Monitor was made to location 130 (the Return for Normal Call).

5. When calling another program unit, you may initiate a forward search with the BRT positioned anywhere between the header label and the trailer label. A backward search may be initiated only if the BRT is positioned immediately after the unit last loaded.

6. All returns to the Loader-Monitor must be made in three-character mode. If the return is made from above 32K, before branching you must clear the sector bits (high-order bits 16 through 18) in the A-address register to zeros. One procedure for doing this is given in Example 7 on page 3-5.

7. If the starting address of a program is below 32K, the program will be started in 3-character mode. Thus, any program with a starting address below 32K that is to be run in 4-character mode must immediately in its first executable instruction CAM to 4-character mode. Any program with a starting address above 32K will be started in 4-character mode.

8. Special starts and return starts are made only in 3-character mode, and cannot be made to locations above 32K.

9. Any own-coding used with the Loader-Monitor should be stored below 32K and executed in 3-character mode.

SECTION IV
OPERATING PROCEDURES

GENERAL

There are two basic procedures involved in using the Tape Loader-Monitor: (1) Loading the Loader-Monitor itself into memory, and (2) Using the Loader-Monitor to load other programs. The first procedure, the bootstrap, is used only at the beginning of a run. Once the Tape Loader-Monitor has been loaded, it is usually left resident in memory and rarely needs to be reloaded. The other procedure, the console call, is used whenever it is necessary for the operator to call some other program unit from the tape. Usually during a run all programs are loaded from a program tape on logical drive 0. For detailed procedures for operating the central processor and peripheral devices, refer to the Honeywell Series 200 Equipment Operators' Manual, Order No. 040.

BOOTSTRAP PROCEDURE

To load the Tape Loader-Monitor, you must first bootstrap into memory the first record of the self-loading tape bootstrap routine. The tape bootstrap routine always precedes the Loader-Monitor[1] on any BRT. (It immediately follows the tape header label.) After the first record is bootstrapped, the tape bootstrap routine loads itself into memory, halts (Halt 1),[2] and then loads the Tape Loader-Monitor. When Halt 3 occurs, the Loader-Monitor has been loaded and is ready for a console call.

1. Initialize the central processor and activate the peripheral devices to be used.
2. Mount the program BRT on logical drive 0. Be sure that the tape is rewound. Remove the write-enable ring. Set the PERMIT-PROTECT switch to PROTECT.
3. Set the CONTENTS buttons to the peripheral control unit (pcu) address of the tape (usually 40_8). Depress BOOTSTRAP. (This positions the tape past the header label.)
4. Again, set the CONTENTS buttons to the pcu address of the tape. Depress BOOTSTRAP. (This reads the second record on the tape, i.e., the first record of the tape bootstrap routine.)
5. If the pcu address used for the tape (in steps 3 and 4) was 40_8, proceed to step 6. Otherwise, manually enter the pcu address into locations 65_8, 73_8, and 101_8.

---

[1] If there is more than one Loader-Monitor on the BRT, they all have the same name, AAAMON, and are distinguished by visibility.

[2] Halt 1 provides the option of loading some other Loader-Monitor.

6. Depress RUN. (This loads the tape bootstrap routine.)
7. When Halt 1 (B-Address = 17001) occurs, if the visibility of the Tape Loader-Monitor is other than "A," enter it into locations 124_8 through 129_8.
8. Depress RUN. (This loads the Tape Loader-Monitor.)
9. When Halt 3 (B-Address = 17002) occurs, the Loader-Monitor is in memory awaiting entry of console call parameters.
10. If other halts occur, refer to Table 4-1.

CONSOLE CALL PROCEDURES

At console call, the Loader-Monitor is ready to load another program unit. It is set to search forward by program and segment name, and execute a normal start. Using a call card or keying in through the control panel, the operator enters the program and segment name of the desired program unit, and other parameter values if desired. He then starts the Loader-Monitor.

Obtaining the Console Call Halt

Parameter values for a console call may be entered only when the Loader-Monitor is at the console call halt (Halt 3). This halt usually occurs automatically when the system is ready for another program unit to be loaded — either just after the Loader-Monitor itself is loaded, or whenever a program branches indirectly to the General Return Address (location 139_{10}).

In certain abnormal conditions, however, it is necessary for the operator to initiate the Console Call Halt. In such cases, Halt 3 is obtained by executing the branch instruction in Fixed Start 0 (location 126_8). This has the same effect as an indirect program branch to the General Return Address.

To execute Fixed Start 0, perform the following steps:

1. Enter "126_8" into the sequence register.
2. Set ADDRESS MODE switch to "3."
3. Be sure that the sector bits (high-order bits 16 through 18) in the A-address register are zero.
4. Press RUN.

Entering Parameter Values

You may enter parameter values into the communication area with a call card or through the control panel. Using a card is easier and faster than keying in through the control panel. A call card, however, can enter only four parameter values: program name, segment name, tape drive number, and halt name. But usually, these are the only parameters that need to be changed — others are reset to standard values. When it is necessary to change other param-

eter values, you must do this through the control panel. In cases where both methods of entry are used, manual entrys must be made before the call card is read.

CARD ENTRY

1. Punch call cards for the desired programs. (See Figure 3-1 on page 3-2.)
2. Be sure that the Loader-Monitor is at Halt 3 and that card entry is specified by "00" in location 100_8.
3. Activate the card reader. Arrange the call cards for the desired programs, with any control and data cards that are needed, in the order these programs will be wanted during the run. Put the correctly arranged deck in the card reader.

MANUAL ENTRY

1. Be sure that the Loader-Monitor is at Halt 3.
2. Designate manual entry by entering "01" into location 100_8.
3. Enter the program name of the requested unit into locations 104_8 through 111_8.
4. Enter the segment name of the requested unit into locations 112_8 and 113_8.
5. If the logical drive number of the tape that contains the requested program unit is not "0," enter it into locations 114_8.
6. If a halt is desired after loading the unit before starting execution, enter the program and segment names of that unit into locations 115_8 through 124_8.

Loading and Execution

After performing parameter entry procedures, the operator starts the Loader-Monitor. It then reads a call card (if any) and finds, loads, and (if not directed to halt) starts the requested unit.

1. Press RUN.
2. If halts occur during loading, refer to Table 4-1. If halts occur during execution, refer to the procedures for the program that has just been called.

PROGRAMMED HALTS

Table 4-1 lists the halts for the tape bootstrap routine and the Tape Loader-Monitor. Halts 1 and 3 have been described above. Halt 6 occurs after any program unit specified by the Halt Name parameter is loaded. The other six halts are error halts, for which the appropriate operator action is indicated. Contents of the A- and B-address registers are given in octal.

Table 4-1. Coded Halts for Tape Bootstrap Routine and Tape Loader-Monitor C

Halt No.	Cause	A Address	B Address	Operator Action
1	Bootstrap routine requests loader visibility if non-standard	00124	17001	If a loader of non-standard visibility (not visibility A) is desired, perform steps 1 and 2 below; if the standard Tape Loader-Monitor is to be used, proceed to step 2. 1. Enter six-character visibility code of desired loader into memory beginning at octal location 124. 2. Depress RUN.
2	Loader of requested visibility missing	00000	14002	Check mounted BRT and requested visibility. Correct error, rewind BRT, and start again from beginning of bootstrap procedure.
3	Loader requests console call	00100	17002	Enter necessary parameter values into communication area, using either card entry or manual entry as outlined in "Console Call Procedures"; then depress RUN.
4	Uncorrectable read error on BRT. p: tape control d: tape drive	00000	10pld	Check tape for dirt and damage, and correct as necessary. (Refer to the Honeywell 200 Equipment Operators' Manual.) Start again with console call procedure to try to reread.
5	Illegal punch on call card	00000	10110	Correct call card, obtain halt number 3 as outlined in "Obtaining the Console Call Halt" and reread the call card as outlined in "Card Entry."
6	"Halt name" unit has been loaded	00000	14000	Perform action requested by programmer; then depress RUN to continue.
7	Called unit not found on backward search	00000	14010	Depress RUN to search forward.
8	Called segment not found within current program searching in specified direction	00000	14012	Depress RUN to search in opposite direction.

APPENDIX A
USE OF VISIBILITY

Since the only realistic test environment for new programs is operation with other systems programs, the possibility of a new program causing problems in other parts of the system increases as the number of individual programs grows. Malfunctions in unchecked programs can cause errors which affect other programs and cause a programmer to waste time searching for errors in one program, only to find that they originated in another.

Ideally, each programmer should have his own system tape for test purposes. Such a tape would consist of all the checked-out programs plus the new programs of his own. In this way, new programs can be tested without creating problems in any other part of the system also under test. Any improper output could be assumed to originate in the new program or programs of a particular system tape. In effect, the visibility code system provides for such separate tapes.

As an example, consider that in a system under development there are three checked-out programs. For convenience, these will be called P-ONE, P-TWO, and P-THREE. A fourth program, P-FOUR, is still under test; however, it gives proper output within known limits which can be used as input to a new program P-FIVE. For a given computer run, it is desired to test a new version of P-FOUR which incorporates new coding (this version will be called P-FOUR-M) and also to test P-FIVE. Obviously, there is a possibility that P-FOUR-M can introduce new errors, making the output completely useless for any realistic test of P-FIVE. On the other hand, the coding patches may be correct, allowing P-FOUR-M to be the final version of the program that is to be included as part of the system. In this case, P-FIVE should be tested with P-FOUR-M. By use of the visibility codes, a binary run tape could be produced using the following visibility coding:

PROGRAM	VISIBILITY		
P-ONE	A	B	C
P-TWO	A	B	C
P-THREE	A	B	C
P-FOUR		B	
P-FOUR-M	A		C
P-FIVE		B	C

In this way, the three versions of the same system exist on the same tape. The first version called for could be version A. If version-A output were not satisfactory (i.e., P-FOUR-M did not produce the desired results), P-FIVE could still be tested by calling in version B which

includes P-FOUR without the corrective coding included in version A. If version-A output did include the desired output, version C could be called for and executed. In either case, the testing of P-FIVE could proceed independently of the work being done on P-FOUR. Improper output from any version could be isolated to a specific program with a high degree of certainty.

Note that the visibility assignment given above was only one of the many ways in which the desired results could be obtained.

Table A-1. Visibility Codes

Visibility Code	Octal Visibility Key	Visibility Code	Octal Visibility Key
A	40 00 00 00 00 00	T	00 00 00 20 00 00
B	20 00 00 00 00 00	U	00 00 00 10 00 00
C	10 00 00 00 00 00	V	00 00 00 04 00 00
D	04 00 00 00 00 00	W	00 00 00 02 00 00
E	02 00 00 00 00 00	X	00 00 00 01 00 00
F	01 00 00 00 00 00	Y	00 00 00 00 40 00
G	00 40 00 00 00 00	Z	00 00 00 00 20 00
H	00 20 00 00 00 00	0	00 00 00 00 10 00
I	00 10 00 00 00 00	1	00 00 00 00 04 00
J	00 04 00 00 00 00	2	00 00 00 00 02 00
K	00 02 00 00 00 00	3	00 00 00 00 01 00
L	00 01 00 00 00 00	4	00 00 00 00 00 40
M	00 00 40 00 00 00	5	00 00 00 00 00 20
N	00 00 20 00 00 00	6	00 00 00 00 00 10
O	00 00 10 00 00 00	7	00 00 00 00 00 04
P	00 00 04 00 00 00	8	00 00 00 00 00 02
Q	00 00 02 00 00 00	9	00 00 00 00 00 01
R	00 00 01 00 00 00	*	00 00 00 00 00 00
S	00 00 00 40 00 00		

APPENDIX B
OWN-CODING

INTRODUCTION

The Tape Loader-Monitor program's own-coding provisions are intended primarily for the use of Honeywell systems programs. However, although the use of own-coding is not recommended for the average user, the own-coding provisions may be required by the user concerned with the development of programming systems.

OWN-CODE EXECUTION DURING LOADING

Functions Provided

The Loader-Monitor provides an own-coding exit after each record of a program unit is read into the input buffer, thereby enabling the execution of own-code routines during loading. The own-code routine may return to the same point and use the distribution process of the Loader-Monitor to distribute the data portion of the program unit records, or it may bypass the distribution process of the Loader-Monitor and distribute the data portions of the program unit records itself.

Own-Code Exit

A calling unit may execute own-coding during the loading of a called unit by setting up an appropriate Branch instruction in the Loader-Monitor communication area. Specifically, the starting address of the own-code routine must be entered into locations 103 to 105 (octal 147-151) of the communication area. The starting address of the own-code routine is then the A-address of the Branch instruction whose op code is stored in location 102. Locations 103 to 105 contain no punctuation, and the calling program must not place punctuation in these locations. The branch to the starting address of the own-code routine is made immediately after reading each record. However, before the branch is executed, the Loader-Monitor sets index register X5 to the address of the first data character in the record. The own-code routine must expect to be started in the three-character addressing mode, and the return to the Loader-Monitor must also be in the same mode. The own-code exit branch is reset after loading each unit.

Own-Code Return Points

The own-code routine must conclude with a branch to location 122 or 126 in the Loader-Monitor communication area.

When the return branch is to location 122, the Loader-Monitor performs distribution in the normal manner. Use of this option requires that the setting of index register X6 (the index register used by the Loader-Monitor for distribution) must not be altered by the own-code routine.

When the return branch is made to location 126, the Loader-Monitor bypasses the distribution process and reads the next record. Use of this option requires that the own-code routine must be able to identify the last record of the called unit. When this record is identified, the own-code routine must not return to location 126; instead, it must set index register X5 to the address of a location in which the control character 61 is stored, followed by the three-character starting address of the unit just loaded. The own-code routine must then provide a branch to location 122 in the communication area.

LOADING DATA DISTRIBUTED BY OWN-CODING

When the data portions of BRT records are always distributed by an own-code routine (rather than by the distribution process of the Loader-Monitor), they need not conform to the pattern described in Appendix C in the paragraph entitled "Data Fields"; thus, they may have any form interpretable by the own-code routine. However, the identification and control portion must be constructed as specified in Appendix C under "Identification and Control Fields."

APPENDIX C
FORMAT OF THE BINARY RUN TAPE

GENERAL

A Series 200 binary run tape is classified as a bannered file. All BRT records (with the exception of the header label record, the first record of the tape bootstrap routine, the trailer label record, and the two end-of-reserved-information records) are classified as unblocked, variable-length data records (see Figure C-1). Thus, each data record on a BRT contains one variable-length item, and the record may vary in length up to a maximum of 250 characters. The additional bootstrap records, which appear after the first record of the tape bootstrap routine, are treated as a subset within the data record category, because they each have a "1" in the high-order bit positions of their banner characters. The trailer label record identifies the "end of file" and is followed by two ERI records which signify the "end of reserved information" to terminate the reel properly.

The portion of code which is found and loaded as the result of a single call to the Loader-Monitor is referred to as a program unit (or unit). A single program unit may represent one object program, or several units may constitute one object program. As recorded on a BRT, a unit appears as one or more consecutive records, the first of which is a segment header record. The segment header record contains identification and control information pertaining to the unit it represents, as shown in Figure C-1; it also contains data to be loaded. Subsequent records (non-header records) within the unit contain a minimum of control information immediately followed by loading data.

The first character of every BRT data record is a banner character that identifies the BRT record type (see Figure C-1). Table C-1 contains a brief description of the various types of BRT data records, together with their respective octal banner characters. Although the records of the tape bootstrap routine do not have the standard BRT program unit format, they may also be identified by their banner characters. The first record of the tape bootstrap routine has an octal 22 in its first location, and the remaining 14 records are identified by octal 42 in their first (banner) character locations, as shown in Figure C-1.

RECORD FORMATS OF PROGRAM UNITS

Types of Information

A program unit record contains two categories of information:

1. Identification and control; and
2. Data.

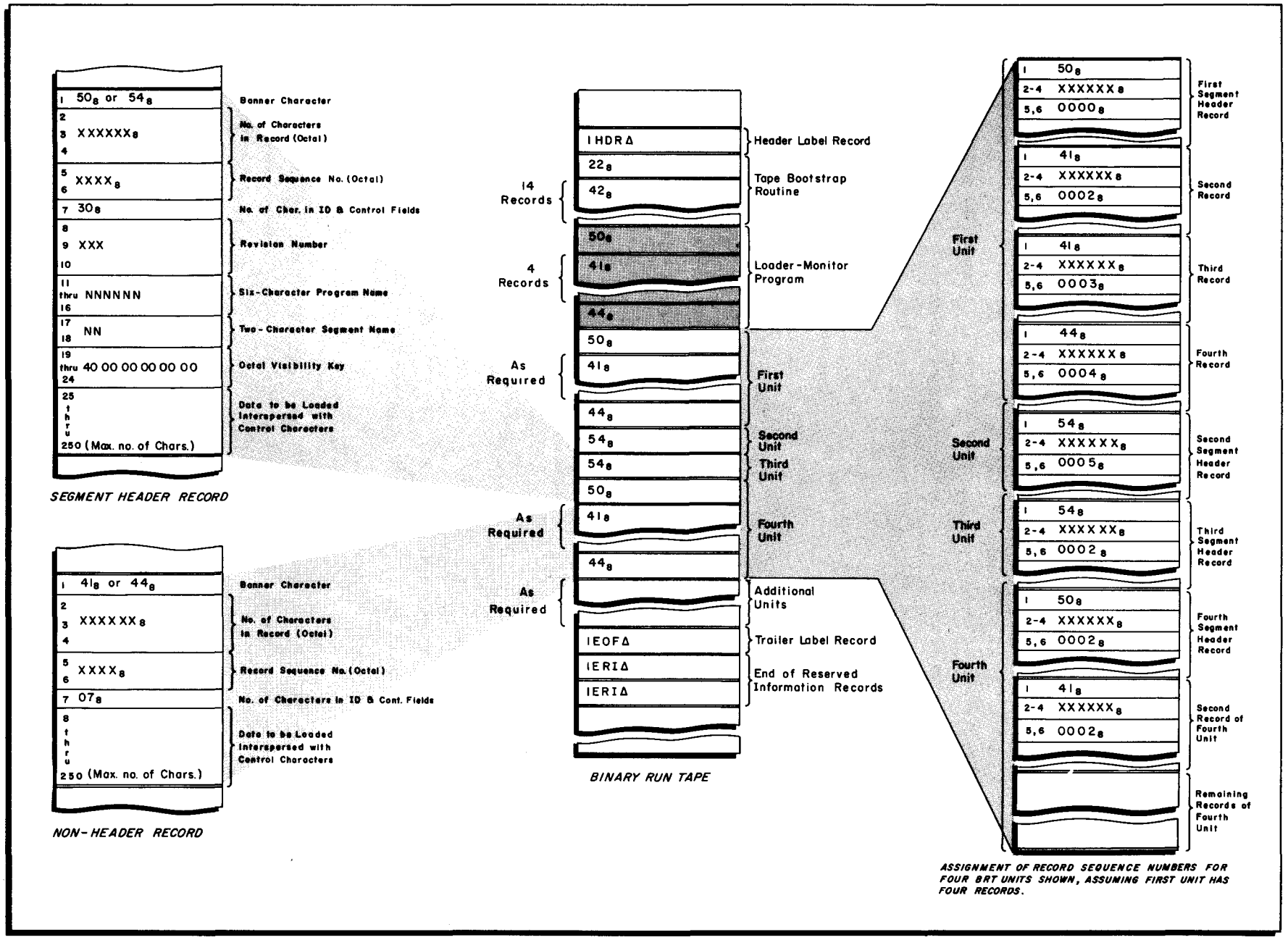

Figure C-1. Format of a Binary Run Tape (BRT)

Identification and Control Fields

The first portion of each program unit data record contains identification and control information. For a segment header record, this portion occupies the first 24 character locations; for a non-header record, it occupies the first seven character locations. As mentioned previously, the first character of each data record is a banner character which identifies the record type (see Figure C-1). Character locations two through four contain an octal number which indicates the number of characters in the record. Character locations five and six contain the record sequence number, which is an octal number used by the Loader-Monitor in performing efficient backward searching over a BRT. Specifically, the record sequence number specifies the number of backspaces to be executed to position the tape for reading the preceding segment header record, and its value is established by the following criteria:

1. The record sequence number of the first segment header record on the BRT is always 0;
2. The record sequence number of subsequent segment header records is equal to the number of records in the previous unit plus one; and
3. The record sequence number of non-header records is equal to the number of records which precede it in its own unit plus one.

The seventh character designates the number of characters in the identification and control fields, which is 24 (octal 30) for segment header records and seven (octal seven) for non-header records.

The format for the first seven character locations is the same for both segment header records and non-header records; however, the segment header record contains additional identification and control fields. The first seven character locations of both segment header records and non-header records contain information that was generated during assembly, whereas the information in the remaining identification and control fields of the segment header record is specified by the programmer in his symbolic program. The identification and control fields are tabulated by record type and character locations in Table C-1.

Table C-1. Identification and Control Fields of BRT Program Unit Records

Record Type	Character		Function
	Location	Name	
Segment—header and non-header records (50_8, 54_8, 41_8, 44_8)	1	Banner	Identifies record type (octal designations below): 50 - segment header record, not last record of a unit; 54 - segment header record, last record of a unit;

Table C-1 (cont). Identification and Control Fields of BRT Program Unit Records

Record Type	Character Location	Character Name	Function
Segment—header and non-header records (50_8, 54_8, 41_8, 44_8) (cont)	1	Banner	41 - not segment header record, not last record of a unit; 44 - not segment header record, last record of a unit.
	2-4	Record Length	Designates number of characters in record in octal.
	5-6	Record Sequence Number	Specifies number of backspaces (in octal) to position BRT for reading previous segment header record.
	7	Length of ID and Control Fields	Designates the number of characters in the identification and control fields: octal 30 for segment-header records, 7 for non-header records.
Segment-header (50_8, 54_8)	8-10	Revision Number	Three-character number assigned by the programmer in his symbolic program. If unassigned, the assembly program assigns zero.
	11-16	Program Name	Six-character program name assigned by programmer.
	17-18	Segment Name	Two-character segment name assigned by programmer.
	19-24	Visibility Key (octal)	Six-character loading key assigned by the programmer and used by the Loader-Monitor when searching for a unit. (It may be used to correlate two or more units as a system subset to be run together or to distinguish between different versions of the same program.)

Data Fields

Immediately following the control information on both segment header and non-header records is the data portion of the record which is variable in length up to the maximum record length of 250 characters. The data portion of each record consists of strings of data characters to be loaded interspersed with control characters. The control characters control the loading of the data characters by specifying such information as loading locations and applicable punctuation. The formats and description of the nine control characters are provided in Table C-2. The data portion of each record within a BRT unit has the following characteristics:

1. It begins with a control character;
2. Every record except the last record of a unit terminates with the control character octal 77;

3. The last record of a unit terminates with control character octal 61, followed by a three-character address;

4. The "nnnn" digits of control characters 1 through 3 in Table C-2 are never 0000.

Table C-2. Data Field Control Characters

Control Character			Meaning
No.	Octal	Binary	
1	Variable from 01 to 17	00nnnn	Interpret the nnnn digits as a binary number. Move the following nnnn characters to successive locations, placing the leftmost character in the location specified by the current setting of the distribution counter (in X6). Clear punctuation in locations into which the characters are moved. Advance the distribution counter by nnnn.
2	Variable from 21 to 37	01nnnn	Perform same functions as control character no. 1, and set a word mark in the leftmost character location loaded.
3	Variable from 41 to 57	10nnnn	Perform same functions as control character no. 1, and set an item mark in the leftmost character location loaded.
4	60	110000	Place the following three characters into the distribution counter. (The next string will be loaded with its leftmost character at this address.)
5	61	110001	Terminate loading. Interpret the following three characters as the normal starting location for the unit just loaded.
6	62	110010	Clear an area of memory, using the following seven characters to identify the area to be cleared and the character with which to clear it. (Characters 1 through 3 are interpreted as the lowest address of the area to be cleared; characters 4 through 6 are interpreted as the highest address; and character seven is transferred to every location in the cleared areas with punctuation marks cleared.)
7	63	110011	Set a word mark in the location whose address is one less than the current setting of the distribution counter.
8	64	110100	Set an item mark in the location whose address is one less than the current setting of the distribution counter.
9	77	111111	Read the next record.

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200
TAPE LOADER-MONITOR C
SOFTWARE MANUAL

DATED: APRIL 25, 1966

FILE NO: 123.5305.021C.0-221

ERRORS NOTED:

Fold

Cut Along Line

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME _____ DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

_____

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: TECHNICAL COMMUNICATIONS DEPARTMENT

Cut Along Line

Honeywell
ELECTRONIC DATA PROCESSING