

SERIES 200

PROGRAM TEST SYSTEM C ADDENDUM NUMBER 1

GENERAL SYSTEM:

SERIES 200/OPERATING SYSTEM - MOD 1
(TAPE RESIDENT)

SUBJECT:

Corrections and additions to the manual entitled
Program Test System C, dated December 15, 1965.

SPECIAL
INSTRUCTIONS:

The attached pages should be used to replace the
corresponding pages in the manual. They will be
incorporated in the next edition of the manual.

DATE: May 31, 1966

FILE NO.: 123.2005.001C.0-^{*}049

8969

5566

Printed in U. S. A.

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

Copyright 1966

Honeywell Inc.

Electronic Data Processing Division

Wellesley Hills, Massachusetts 02181

SERIES 200

PROGRAM TEST SYSTEM C

GENERAL SYSTEM:

SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT:

Programming and Operating Procedures for Program Test System C, an Automatic Testing System for Mod 1 Programs.

SPECIAL
INSTRUCTIONS:

The information herein reflects the software released as revision 0.

DATE: December 15, 1965

FILE NO.: 123.2005.001C.0-049^{*}

51265
8551

Printed in U. S. A.

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

FOREWORD

Program Test System C is an integral part of the SERIES 200/OPERATING SYSTEM - MOD 1. The Program Test System comprises a set of control and utility programs encompassing the functions of obtaining dynamic and terminal memory dumps, emergency memory dumps, tape dumps, automatic patches, and operator instruction generation. The use of these programs in the system, as described herein, presupposes that they are stored on a binary run tape (BRT); hence, they are loaded into memory by the Tape Loader-Monitor C program which is described in detail in the Honeywell Information Bulletin entitled PLUS-Tape Loader-Monitor.

The reader is assumed to be familiar with the information presented in the PLUS-Tape Loader-Monitor bulletin (DSI-327) and the Honeywell Series 200 Equipment Operators' Manual (Model 200), Order Number 040.

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

TABLE OF CONTENTS

		Page
Section I	Introduction.....	1-1
	Equipment Requirements	1-1
	Required Equipment	1-2
	Optional Equipment.....	1-2
Section II	Program Description and Preparation	2-1
	Addressing Mode	2-1
	System BRT.....	2-2
	Test Director Deck	2-3
	Console Call Cards	2-4
	Program Test System Console Call Card	2-5
Object Program Console Call Card.....	2-5	
Section III	System Programs	3-1
	Initializer.....	3-1
	Operating Considerations	3-1
	List Comments.....	3-2
	Comment Card Format	3-2
	Operating Considerations	3-4
	Console Call Card.....	3-4
	Card Read Error.....	3-4
	Print Error.....	3-4
	Program Halts.....	3-4
	Test Data Generator	3-5
	Parameter Card Processing.....	3-5
	Parameter Cards	3-6
	File Card.....	3-6
	Item Card	3-10
	Data Card	3-10
	Post Card	3-12
	End Card.....	3-13
	Operating Considerations	3-13
	Halts	3-14
	Data Card Error.....	3-14
	Item Card Error.....	3-14
	Card Read Error	3-15
	Tape Write Error.....	3-15
	Unidentified Card Error.....	3-15
	Memory Dump Control	3-15
	Addressing Modes.....	3-15
	Programmed Instruction Dumps	3-16
	Format 1.....	3-17
	Format 2.....	3-17
	Format 3.....	3-17
	Format 4.....	3-17
Item Mark Trapping.....	3-17	
Console Call Card.....	3-18	
Operating Considerations	3-20	

TABLE OF CONTENTS (cont)

	Page
Section III (cont)	
Memory Dump	3-21
Addressing Modes	3-21
Functional Description	3-21
Operating Considerations	3-21
Patch.....	3-22
Addressing Modes	3-22
Functional Description	3-23
Format of the Required Cards.....	3-23
Console Call Card (CH).....	3-23
Console Call Card (Used by S2).....	3-24
Octal Correction Card.....	3-25
Termination Card.....	3-26
Operating Considerations	3-26
Tape Dump	3-27
Parameter Entry Cards.....	3-27
Operating Considerations	3-29
Format of a Tape Dump Listing.....	3-30
Emergency Memory Dump.....	3-30
Addressing Modes.....	3-32
Parameters	3-32
Operating Considerations	3-33
Console Call Card.....	3-33
Console Entry.....	3-33
Technical Note.....	3-34
End	3-34
Console Call Card	3-34
Section IV	
Operating Procedures for Program Test System C.....	4-1
Initial Setup Procedures.....	4-1
Tape Mounting Procedures.....	4-1
Bootstrap Procedures.....	4-1
Console Call Card Procedure	4-1
Recovery Procedures	4-2
Emergency Memory Dump	4-2
Emergency Terminal Memory Dump	4-3
Appendix A	
Sample Run of Program Test System C	A-1
Initialization of the Program Test System	A-2
Processing with the Program Test System.....	A-3
Test Director Deck Number One.....	A-3
Test Director Deck Number Two.....	A-5
Test Director Deck Number Three.....	A-6
Appendix B	
Provision for Additional Utility Programs.....	B-1

LIST OF ILLUSTRATIONS

		Page
Figure 2-1.	Program Test System BRT	2-2
Figure 2-2.	Sample Test Director Decks	2-3
Figure 2-3.	Program Test System Console Call Card.....	2-5
Figure 3-1.	Console Call Card for Loading Initializer.....	3-1
Figure 3-2.	Sample Format for a Comment Card.....	3-3
Figure 3-3.	Console Call Card for Loading List Comments.....	3-4
Figure 3-4.	Sample Coding Entries for Header and Trailer Records.....	3-7
Figure 3-5.	File Card Format.....	3-8
Figure 3-6.	Item Card Format	3-10
Figure 3-7.	Data Card Format	3-11
Figure 3-8.	Post Card Format	3-12
Figure 3-9.	End Card Format.....	3-13
Figure 3-10.	Console Call Card for Loading Test Data Generator	3-13
Figure 3-11.	Console Call Card for Loading Memory Dump Control	3-19
Figure 3-12.	Memory Dump Control Console Call Card Placement	3-20
Figure 3-13.	Format of a Memory Dump	3-21
Figure 3-14.	The Function of Patch.....	3-22
Figure 3-15.	Console Call Card for Loading Patch.....	3-23
Figure 3-16.	Console Call Card for Loading an Object Program to be Patched	3-25
Figure 3-17.	Octal Correction Card	3-26
Figure 3-18.	Deck Arrangement for Using Patch	3-26
Figure 3-19.	Tape Dump Parameter Entry Card.....	3-27
Figure 3-20.	Last Card of Tape Dump Parameter Entry Card Deck.....	3-27
Figure 3-21.	Console Call Card for Loading Tape Dump.....	3-29
Figure 3-22.	Sample Tape Dump Listing.....	3-31
Figure 3-23.	Console Call Card for Loading Emergency Memory Dump.....	3-33
Figure 3-24.	Console Call Card for Loading End	3-34
Figure A-1.	Sample Testing Decks.....	A-1
Figure A-2.	Test Director Deck Number One.....	A-2
Figure A-3.	Test Director Deck Number Two	A-3
Figure A-4.	Test Director Deck Number Three.....	A-3

LIST OF TABLES

Table 2-1.	Three- and Four-Character Addressing Mode Utility Program Names	2-1
Table 3-1.	List Comments Line Spacing Codes	3-3
Table 3-2.	Parameter Fields Used in Conjunction with the Four Function Codes.....	3-28
Table 3-3.	Coded Halts for Tape Dump.....	3-29
Table 4-1.	Programs of Program Test System C.....	4-2
Table 4-2.	Error Halts.....	4-4



SECTION I

INTRODUCTION

Program Test System C is an automatic checkout system for all Series 200 programs operating with a machine configuration of 12,288 memory locations or greater. It is an integral part of the Series 200/Operating System - Mod 1. The system operates in conjunction with the Tape Loader-Monitor program to provide automatic run-to-run sequencing, in which, test data generation, program patching, and production of diagnostics for program checkout are all accomplished with little or no operator intervention. The Program Test System comprises a series of utility programs which are called using console call cards. Only those functions needed for a particular checkout run need be called. The system is completely open-ended in that additional functions can be added in accordance with the conventions outlined in Appendix B.

System utility programs are varied in function and size and may or may not occupy the memory storage area when the object program is in memory. Depending on the function involved, the utility programs may be executed before, during, or after the user's object program. The maximum number of storage locations occupied by the Program Test System (when the object program is in memory) is 2,045 characters for programs using three-character addressing and 2,585 characters for programs using four-character addressing in addition to the area used by the Tape Loader-Monitor program.

The utility programs can be instructed to perform the following functions:

1. Automatic, sequential testing of one program after another,
2. Printouts of messages and operator instructions,
3. Test data generation,
4. Octal program patching for a specific run,
5. Dynamic, terminal and emergency memory dumping, and
6. Terminal and emergency tape dumping. In addition, the system has facilities for the incorporation of additional features and/or specialized routines.

The specific functions and capabilities of each of the nine utility programs are described in detail in Section III.

EQUIPMENT REQUIREMENTS

Program Test System C runs on a Series 200 computer having the minimum required equipment outlined below.

Required Equipment

- 12,288 characters of memory
- One card reader (control unit 41)
- One printer (control unit 02)
- One magnetic tape drive

Optional Equipment

One additional tape drive may be used to contain the object programs on a binary run tape (BRT). Use of this second tape will increase the speed of the Program Test System operations.

SECTION II
PROGRAM DESCRIPTION AND PREPARATION

Program Test System C consists of nine utility programs placed on a binary run tape (BRT) as shown in Figure 2-1. Any number of these programs may be utilized by placing an appropriate console call card in the "test director" input card deck. The Tape Loader-Monitor program is used to read the test director deck and load the proper utility and object programs from the BRT.

The ability to test a series of object programs by using the utility programs in the system is possible because all utility programs and object programs under test return control to the Tape Loader-Monitor program. When running in conjunction with the Program Test System, the loader does not halt between program executions; but always reads the next console call card to determine the next program to be loaded and executed.

ADDRESSING MODE

Utility programs which are independent of the object program (e.g., List Comments, Test Data Generator) operate in the three-character addressing mode, regardless of the mode and size of the object program for which they are performing their function. Utility programs which operate directly on the object program's memory (e.g., Patch, Memory Dump Control) are supplied in both a three-character and a four-character version. The four-character version is needed whenever the object program and the utility program do not reside completely within the area between location 0 and 32,767. Table 2-1 gives the program names to be used in calling the utility programs in both three and four-character modes.

Table 2-1. Three- and Four-Character Addressing Mode Utility Program Names

Utility Program	Name on Console Call Card	
	3-Character	4-Character
Initializer	AAATST	AAATST
List Comments	AAASPE	AAASPE
Test Data Generator	AAAG12	AAAG12
Memory Dump Control	AAADUM	AAAD4C
Memory Dump	AAAMEM	AAAM4C
Patch	AAAPAT	AAAP4C
Tape Dump	AAATAP	AAATAP
Emergency Dump	AAAS0S	AAAS4S
End	AAAEND	AAAEND

SYSTEM BRT

The system BRT contains the utility programs as they are shown in Figure 2-1. This sequence follows the logical order of program execution that is most often taken when testing an object program. While utility programs and object programs may be used in any combination, for maximum efficiency, ease, and uniformity of operation, the sequence and procedures given in this section are recommended. The Initializer program is included twice on the tape to reduce searching time.

Object programs to be tested may be included on the system BRT, in which case all such programs should be inserted (in any order) between the Patch and Tape Dump programs. All object programs should terminate by branching indirectly to the Tape Loader-Monitor general return address (location 139).

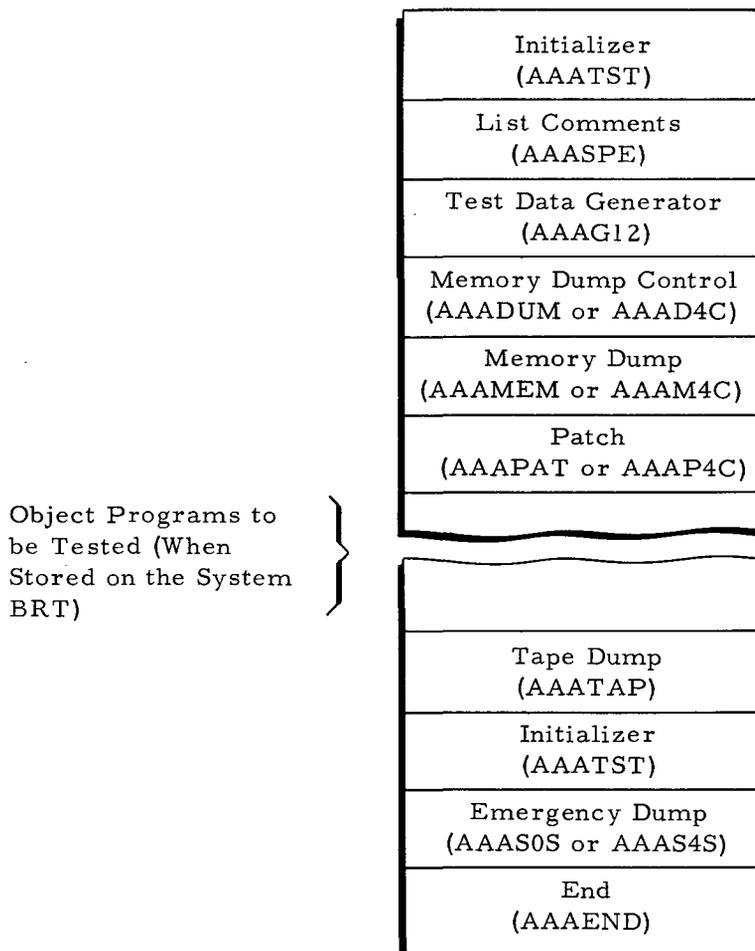


Figure 2-1. Program Test System BRT

Alternatively, a separate BRT may be created to contain the object programs to be tested. In this case, the programs should be in the order in which they are to be executed and must each terminate by branching indirectly to the Tape Loader-Monitor general return address (Location 139).

TEST DIRECTOR DECK

A test director deck can contain the following types of cards:

1. Utility program console call cards,
2. Utility program detail cards,
3. Object program console call cards, and
4. Object program input cards.

A test director deck should be made up for each object program to be tested. Each deck is required to contain a console call card for the Initializer program to initialize the Program Test System and a console call card for the object program to be tested. In addition, each deck should contain the console call cards and their corresponding detail cards for each utility program requested by the programmer.

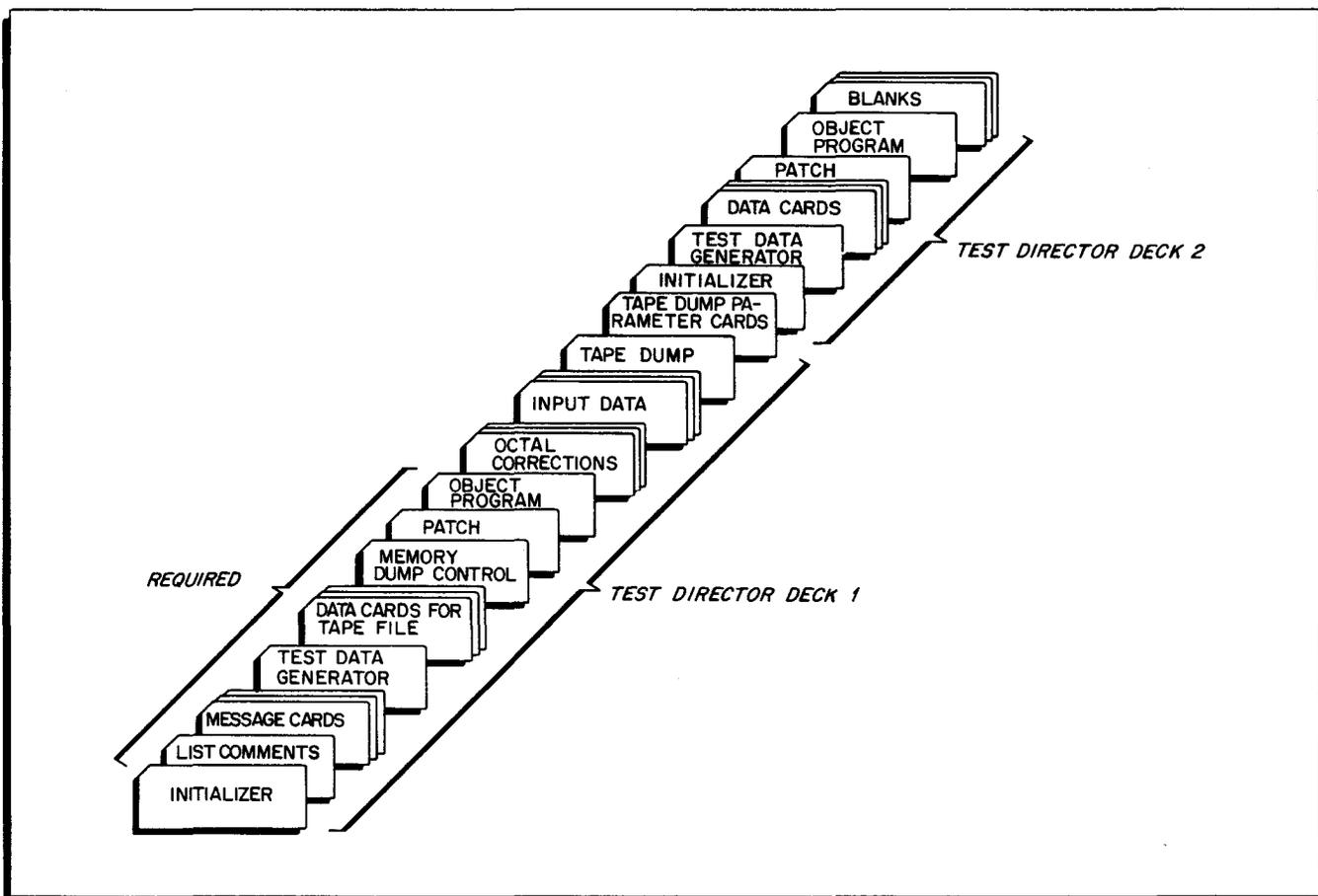


Figure 2-2. Sample Test Director Decks

A test director deck for the testing of an object program is constructed from required and optional console call cards. The first sample test director deck shown in Figure 2-2 contains each of the required console call cards and several optional console call cards which should be ordered as follows:

Initializer (AAATST) - required to initialize the Program Test System.

List Comments (AAASPE) - and corresponding detail cards if message or instruction printouts on the printer or typewriter console are desired. A halt may be indicated to allow for operator preparation.

Test Data Generator (AAAG12) - and corresponding detail cards if test data is to be generated.

Memory Dump Control (AAADUM or AAAD4C) - three- or four-character addressing modes are available. AAADUM inserted in the first six columns of the console call card indicates three-character addressing, AAAD4C indicates four-character addressing will be used. The addressing mode may not be changed during program operation.

Patch (AAPAT or AAP4C) - three- or four-character addressing modes are available. AAPAT inserted in the first six columns of the console call card indicates three-character addressing, AAP4C indicates four-character addressing will be used. The addressing mode may not be changed during program operation.

Object Program Console Call Card - required to call the program to be tested.

Octal Correction Cards - if a console call card for the Patch program is in the test director deck, the octal correction cards should be placed in this position followed by a blank card.

Object Program Input Cards - the program being tested should be inserted in this position.

Tape Dump (AAATAP) - and corresponding detail cards if terminal tape dumps or tape positioning is desired.

CONSOLE CALL CARDS

The console call cards include those for loading the Program Test System utility programs and those for loading the object program to be tested. Both types have the standard console call card format described in the PLUS Tape Loader-Monitor bulletin, DSI-327.

Program Test System Console Call Card

The Program Test System console call card directs the loader to load the utility program specified in columns 1-6. The card contains a utility program name, its segment name, and the tape drive to be used. All Program Test System console call cards must contain a 0 in column 9 to designate the tape address of the system BRT and an asterisk (*) in column 18 to indicate to the loader that this is a console call card. As illustrated in Figure 2-3, the console call card is read only through column 18 leaving the remaining columns available for any purpose.

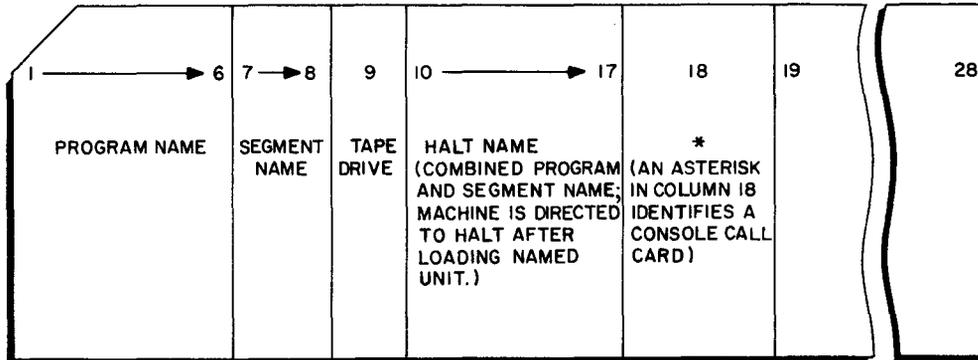


Figure 2-3. Program Test System Console Call Card

Columns 1-6: Contain the program name

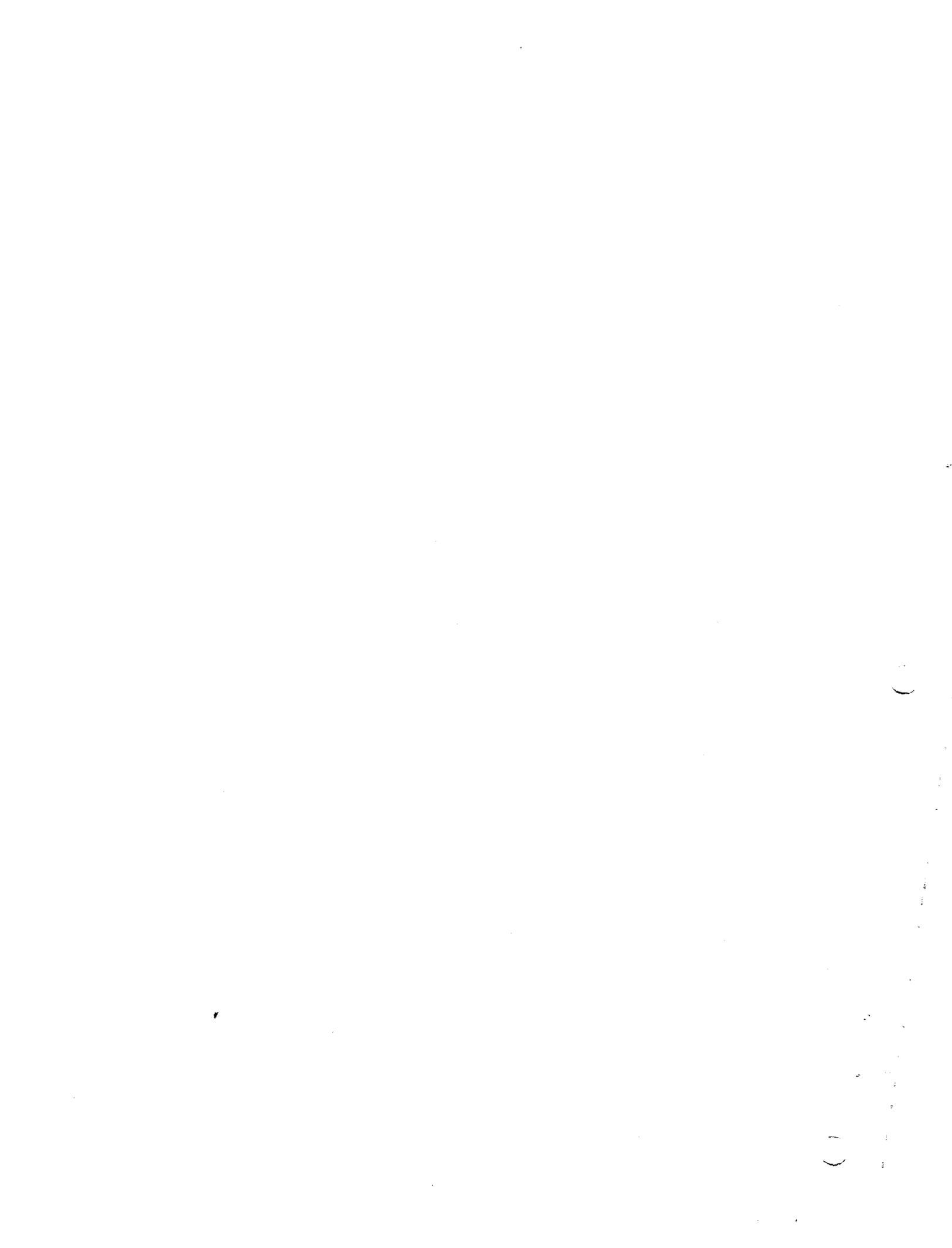
Columns 7-8: Contain the segment name

Column 9: Designates the device from which the utility program is to be loaded.

Column 18: Contains an asterisk

Object Program Console Call Card

The object program console call card directs the Loader to load the object program specified in columns 1-6. The card is identical in format to the console call card described in the preceding paragraphs with the exception that an entry of 0 in column 9 is required only if the programs reside on the system BRT. If the object programs are contained on a separate BRT, column 9 should contain the tape address of the object program BRT.



SECTION III
SYSTEM PROGRAMS

Program Test System C consists of a series of utility programs with which to test the user's object programs. These utility programs are discussed in this section in the order in which they would normally be called from the system binary run tape (BRT). Operating instructions for the system are contained in Section IV.

INITIALIZER

Initializer is a utility program which prepares the Program Test System for automatic reading of console call cards from the test director deck (see Figure 2-2) and loading of utility programs from the BRT.

The program performs the following functions:

1. Sets the computer to the non-trapping mode,
2. Rewinds the system BRT so the system programs may be called in sequential order, and
3. Suppresses the Tape Loader-Monitor console call halt (halt number three) to enable automatic loading of programs as directed by console call cards in the test director deck.

Operating Considerations

Normally the Initializer program is the first program to be executed for each program checkout. This requires a console call card for Initializer to be placed at the beginning of each object program checkout deck.

The console call card (see Figure 3-1) must be punched with the appropriate program name (AAATST) and the segment name (01), etc.

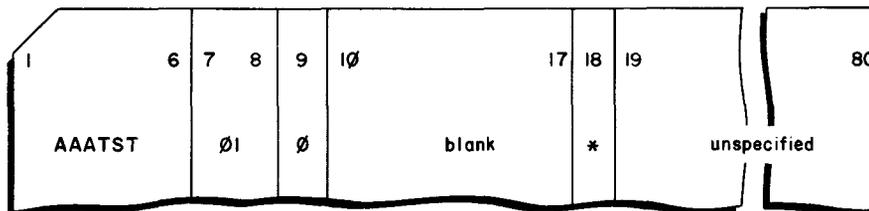


Figure 3-1. Console Call Card for Loading Initializer

After the program has been loaded, it rewinds the system BRT (to allow for sequential execution of the utility programs) and returns control to the loader.

LIST COMMENTS

List Comments is a utility program which reads cards punched with comments and prints or types out the comments on the printer or the console typewriter. Each comment card may contain up to 72 alphanumeric characters for a printed comment or up to 64 characters for a typed comment. A console typed or printed line consists of information from only one comment card, and the comment appears left-justified on the output listing.

There are a variety of applications for the List Comments program; for instance, it may be desired to change SENSE switch settings before executing a program. In this case, List Comments may be called in to read a comment card containing the new SENSE switch settings and to list these settings so that the operator will know exactly what to do for preparation of the next program. Another application arises when a user must create headings for a report. List Comments can be used to write the heading and any sub-heading(s) desired, spacing up to nine lines between headings.

Comment Card Format

A comment card is made up of control fields and a comments field. The following paragraphs explain the use of each field on a comment card.

IDENTIFICATION FIELD (CARD COLUMNS 1-3)

Columns 1 through 3 may be punched COM, GO Δ , or they may be left blank. COM is required for cards containing comments. GO Δ will cause the List Comments program to terminate by returning directly to the loader without halting. If this field is blank, the routine will encounter a coded halt (05000 in the B-address register) in order to allow the operator to comply with the typed or printed instructions. Depressing RUN will cause the program to return to the loader.

END CHARACTER (CARD COLUMN 4)

Column 4 contains an alphanumeric character which specifies the end of the comments field on the card. The end character may be any character not used as part of the comment itself. If no end character is specified, or if the specified end character is not found within the comments field, the entire contents of the comments field are listed. End characters are used chiefly with typed comments to save the time that would otherwise be required for typing blanks.

SPACING CHARACTER (CARD COLUMN 5)

Column 5 indicates the desired spacing to be applied after listing the current comment. The table below defines the character codes which designate the number of lines to be spaced between comments.

Table 3-1. List Comments Line Spacing Codes

Character Code	Spacing
Δ	Space one line
1-9	Space the specified number of lines (printer only)
L	Space to the head of the form (printer only)
0	Do not return carriage (console only)

DEVICE TYPE (CARD COLUMN 6)

Column 6 indicates the output device to be used. "C" is punched to specify the console typewriter; a blank in this column designates the high-speed printer.

DEVICE CONTROL FIELD (CARD COLUMNS 7-8)

Card columns 7 and 8 contain the peripheral address (in octal) of the output device. If this field is blank, 02 is assumed as the printer address and 07 as the console typewriter address.

COMMENTS FIELD

Console Typewriter (card columns 9 through 72): This field contains the comment made up of alphanumeric characters. The characters must be punched exactly as they are to appear in the comment. The character in column 9 appears in print position 1.

Printer (card columns 9 through 80): The comment is punched as for the console typewriter, except that 72 characters can be printed.

Note that List Comments does not edit or position the contents of the comments field, except that the end character, if specified, will not appear on the listing. Thus, for example, heading lines printed in the form of comments are limited to the leftmost 72 (or 64) print positions of a report. Figure 3-2 shows a sample format for a comment card.

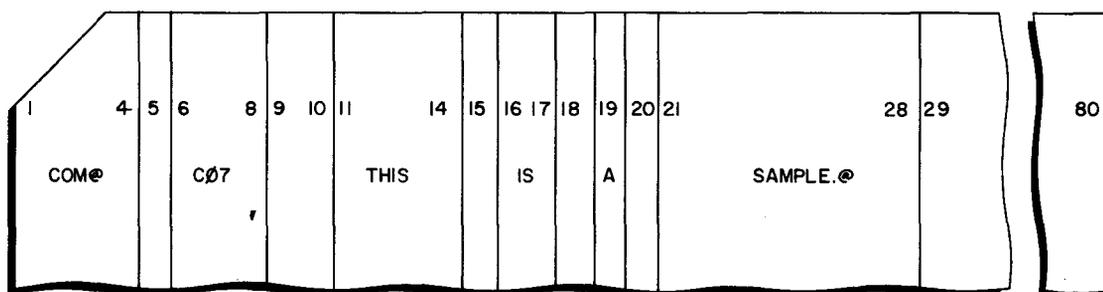


Figure 3-2. Sample Format for a Comment Card

Operating Considerations

To load and execute the List Comments program, a console call card (Figure 3-3) is included in the test director deck. The comment cards immediately follow the console call card.

CONSOLE CALL CARD

The console call card used to load List Comments must be punched with the appropriate program name (AAASPE), segment name (AK), and an asterisk (*) as described in Section II.

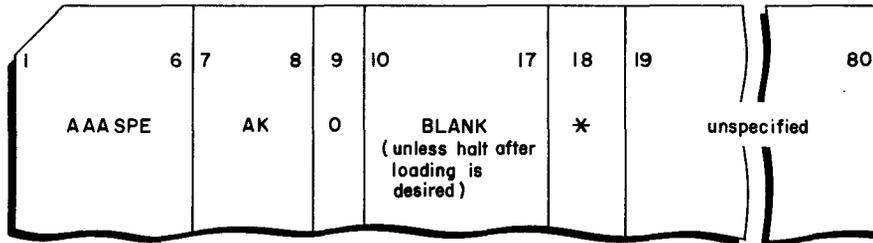


Figure 3-3. Console Call Card for Loading List Comments

After the program has been loaded, it reads comments cards and reproduces the comments that serve as directions for the operator. The directions may specify that the operator dismount and label tapes, change SENSE switch settings, mount a particular tape on a specified drive, etc.

CARD READ ERROR

If the List Comments program registers a card read error, the machine halts with the B-address register containing either of the following:

1. 10111 - for a hole-count error; or
2. 10112 - for an illegal punch error.

The operator may correct the card if illegally punched, reposition the cards in the card reader hopper, and press RUN; the routine will read the card and continue processing. This procedure is to be followed for each card read error.

PRINT ERROR

If a print error is sensed, the machine halts and the B-address register contains lpp20, where pp is the peripheral address of the printer or console typewriter. To correct the print image and continue processing, the cards must be repositioned in the card reader hopper and the RUN button pressed.

PROGRAM HALTS

If in columns 1 through 3 of a comments card there are blanks, then the B-address register contains 05000. When RUN is depressed, the program will exit to the general return address of the loader.

TEST DATA GENERATOR

The Test Data Generator is a utility program to create test data on 1/2-inch magnetic tapes. The Test Data Generator program is of particular use in the development and testing of user-written programs. It generates the required type of data and provides a full range of conditions against which the program under development can be tested.

Parameter Card Processing

The test data and the format in which the data is to be laid out on tape is specified by the programmer on parameter control cards which are placed directly behind the console call card in the test director deck. The Test Data Generator program acts upon these set parameters and creates data on magnetic tapes for use by the program under test.

There are five parameter cards in the Test Data Generator program, each of which has a specific job to perform in order to create the necessary "test" output for the users object program. The cards are listed with their respective functions below. More detailed descriptions of these cards begin on page 3-6.

1. File Card: Contains the overall specifications of the desired test data. These specifications include the type of items and records in a file, the output device to be used e.g., the Type 204B Magnetic Tape Unit), and a notation as to whether a printed listing of the test data generated output file is required.
2. Item Card: Specifies the length of the item desired and the number of items to be contained in a record.
3. Data Card: Contains the actual contents of the item. This data may be printed as the test material to be used by the users object program.
4. Post Card: Transfers an item a specified number of times to the output buffer and writes the contents of the buffer on the output device the specified number of times.
5. End Card: Terminates processing and returns control from the Test Data Generator program to the Tape Loader-Monitor program.

The Test Data Generator program deals with items, records, and files. The lowest units of information in the Test Data Generator program are characters. These characters, or data, are entered into units called items. Items make up a record and records make up a file.

The term "item area" is used to define an area set aside in main memory which will contain the longest item within a specific file. The parameters used to define the maximum item length are specified on the file card and will not change until redefined by another file card. The program can then create an item image which corresponds to these parameters. The item card is used to specify the actual number of characters in the succeeding item(s). Test Data

Generator regards an attempt to update the areas outside of the current item image being processed as an error. The method for recovering from such an error is covered under Test Data Generator "Halts." To enter actual data into the item image, a data card is used. When the contents of the item are completely specified the item is moved to an output buffer for writing on the output device. This transfer is initiated by a post card.

The maximum number of items in a record and the maximum number of characters in a record are specified on the file card mentioned above. During processing, if the program determines that either of these values will be exceeded when a transfer of data to the buffer is made, the items in the buffer at that time will be written as a record on the output device. The overflow item will then become the first item in the next record.

Each of the input cards is interpreted and acted upon in the preceding manner until an end card is recognized. At that time execution of the Test Data Generator program will terminate.

Parameter Cards

FILE CARD

The file card defines the type of file to be constructed, the device on which it is to be written, the size of the item area, the size of the buffer, and a notation as to whether a printed listing of the file to be generated is needed. These parameters will remain constant until another file card redefines them. The file type field on the file card designates the format of a file, viz., blocked or unblocked, fixed or variable, bannered or bannerless.

The Test Data Generator can create files in any one of six basic formats. Three of the formats require a "banner character" as the first character of each record; a file in one of these formats is called "bannered." The other three formats require no banner characters, and are called "bannerless." Items within files may be of fixed or variable length, and may be blocked one or more items to a record. Bannered and bannerless records within a file may be written in even or odd parity. The six file formats are:

1. Bannerless Fixed — Each record is made up of one or more items fixed length. Each record contains data only; no control information is necessary.
2. Bannerless Unblocked Variable — Each record is made up of one item, whose length may vary from record to record within the file. Each record contains data only; no control information is necessary.
3. Bannerless Blocked Variable — Each record is made up of a character-count field and one or more items of variable length. The count field is the first four characters of the record; it contains a decimal entry which designates the number of characters in the record. Each item must contain a three-character decimal item character-count field,

specifying the number of characters in the item. The position of this field within an item may vary from file to file, but must remain constant within a file.

4. Bannered Fixed — This is identical to bannerless fixed, except that a banner character is appended as the first character of each record.
5. Bannered Unblocked Variable — Each record is made up of a banner character, a character-count field, and one item, whose length may vary from record to record within the file. The count field contains a binary entry which specifies the number of characters in the record. The count field occupies locations two through four of the record and, optionally, may appear also as the last three characters of the record.
6. Bannered Blocked Variable — Each record is made up of a banner character, a character-count field, and one or more items of variable length. The count field contains a binary entry which specifies the number of characters in the record. It appears as characters two through four of the record and, optionally, may also appear as the last three characters of the record. Each item contains, as its first two characters, a count field which specifies, in binary, the number of characters in the item.

EASYCODER

CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER	TYPE	LOCATION	OPERATION CODE	OPERANDS	
1	FILEID	PR0080	080	010062	
2	DATAA	0500505	1,HDR		
3	DATAA	1003010	FILE NAME		For Header Label
4	POST	01A00			
5					
6	FILEHV	PA0500	1,00	990062	
7		:			
8		:			For Data Items
9		Data, and Post Cards			
10		:			
11		:			
12	FILEID	PF0080	080	010062	
13	DATAA	0500505	1,EOF		For Trailer Label
14	POST	01A01			

Figure 3-4. Sample Coding Entries for Header and Trailer Records

The Test Data Generator does not write header, trailer, or tape mark records automatically. Instead, the user specifies these records as he would his data, using the appropriate file parameters. This gives maximum flexibility in the arrangement of these records. For example, a file preceded by a header label and followed by a trailer label could be generated as shown in Figure 3-4. (Fixed-length formats, either bannered or bannerless, are convenient for specifying labels.)

The file card shown in Figure 3-5 is supplemented by the following description of the file card format.

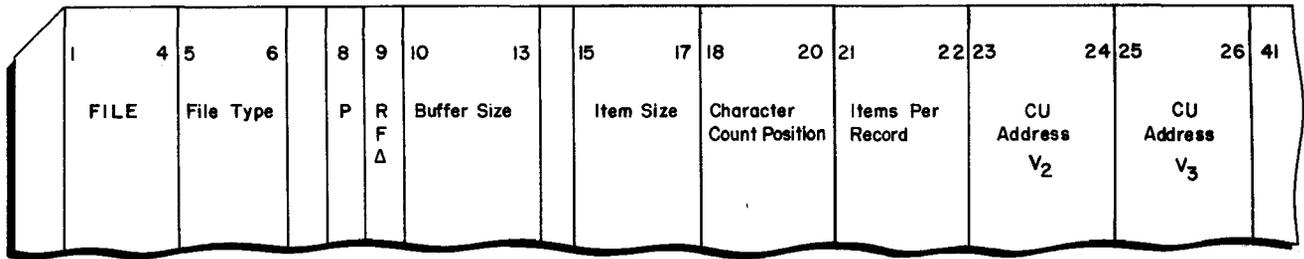


Figure 3-5. File Card Format

Card Columns

Contents

Columns 1-4

Contain the four letters "FILE" to identify the type of card.

Columns 5-6

Contain one of the following file type entries to designate the format of a file:

1. The entry IV or an entry of a Δ indicates a bannerless blocked variable file.
2. The entry ID indicates either a bannerless unblocked variable or bannerless fixed file.
3. The entry HV indicates a bannered blocked variable file.
4. The entry HD indicates a bannered fixed file.
5. The entry HU indicates a bannered unblocked variable file.

Column 8

Contains the letter P to designate that a printed listing of the generated data file is desired. If a listing is not required, this column may be left blank.

Column 9

May contain a rewind condition or indicate that record and item counts are to be put into the item image.

R - Rewind tape and read label into the image area. This option is used only when the next record written is to be a header label.

F - Put record and item counts into the image area. This option is used when the next record written is to be a trailer label.

Δ - Use for all non-label records.

Card Columns

Contents

Columns 10-13

Contain a four-character designation (actual or maximum) of the buffer size to be used. Actual record length is recorded in these columns when fixed-length records (including labels) are used. Maximum record length must be specified when variable-length records are used.

Columns 15-17

Contain a three-character designation (actual or maximum) of the item size to be used. The actual item length is to be recorded in these columns when fixed-length records are used. For fixed-length, unblocked records (including labels), the item length equals the record length in a bannerless file; whereas, in a bannered file, the item length equals the record length minus 1.

The maximum item length is to be recorded in the instance of variable-length records.

Columns 18-20

Contain identification of the character count position, for those file types requiring a count field. For bannerless, blocked, variable-length-record file types (more than one item per record), this field defines the position in the item image of the rightmost character of the item character count. The position may range from 003 to the maximum item size. This field may also remain blank in which event 003 is assumed as the rightmost character position.

For bannered variable files in either blocked or unblocked format (one item per record or multiple items per record), the entry of number 3 in column 18 will specify that the record character count is to be placed at the end of each record.

The entry of any other character in column 18 signifies that no character count will appear at the end of the record.

Columns 21-22

Contain the figures which indicate the number (actual or maximum) of items per record (01 to 99). The actual number of items per record is to be specified when fixed-length records are used. Note that unblocked records (including labels) should be specified by the numbers 01.

The maximum number of items per record is to be specified when variable-length records are used. Just as for fixed-length records, unblocked records are to be designated by the numbers 01.

Columns 23-24

Contain the peripheral control unit address as it appears in the second I/O control character of a PDT instruction.

Card Columns

Contents

Columns 25-26

Contain the device address as it appears in the third I/O control character of a PDT instruction. For example, the characters 6X specify odd parity and standard gap, where X is the logical tape drive number.

Column 41

Contains any character as this location is to be used as an initial filler for the item area.

ITEM CARD

The item card (see Figure 3-6) specifies the item length and/or the number of items required per record. In the case of blocked, variable files, which always contain a character count field within each item, this field is modified to contain the desired parameter. If a field is left blank, there is no change in that parameter from its previously set value.

Item cards are not necessary for fixed-length records (including labels), since the item information is specified on the file card. Item cards are used for variable-length items in order to change item length from item to item.

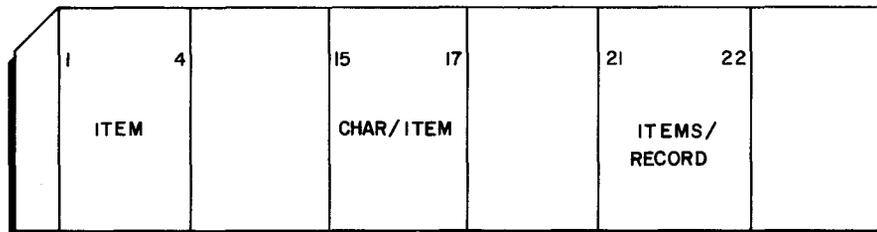


Figure 3-6. Item Card Format

Card Columns

Contents

Columns 1-4

Contain the letters "ITEM" to identify the type of card.

Columns 15-17

Contain the number of characters in the item(s) to follow.

Columns 21-22

Contain the number of items in the record(s) to follow.

DATA CARD

The data card changes the contents of the item image to reflect the specific data parameters required by the user's program. It contains the actual data to be used and specifies the conversion of this data to a usable form.

The data card shown in Figure 3-7 is supplemented by the following description of the data card format.

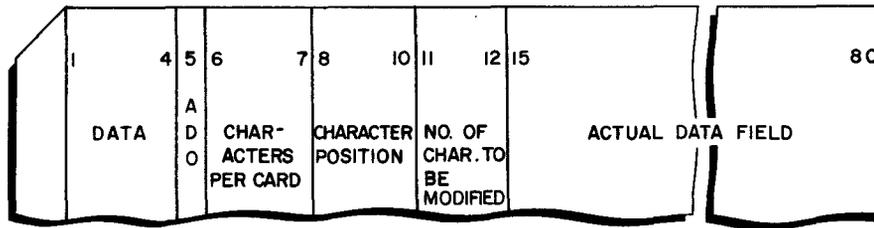


Figure 3-7. Data Card Format

Card Columns

Contents

Columns 1-4	Contain the letters "DATA" to identify the type of card.
Column 5	Contains one of the following three entries to designate mode: A = Alphanumeric, D = Decimal, O = Octal.
Columns 6-7	Contain the number of characters entered in columns 15 to 80 of the data card. This entry may range between 01 and 66.
Columns 8-10	Contain the character position in the item image of the rightmost character to be changed. This entry may range between 001 and 999.
Columns 11-12	Contain the number of 6-bit characters in the item image that are to be changed. This entry may range between 00 and 66.
Columns 15-80	Contain actual data to be inserted in the image area. The data is inserted in the mode specified by Column 5 of the Data Card. Alphanumeric: Characters to be modified are moved, without change, from the card to the item image. In general, the "number of characters to be modified" (columns 11 and 12) is equal to the "number of characters per card" (columns 6 and 7). If, however, the "number of characters to be modified" is greater, leading spaces are inserted as necessary to create a field the length specified in columns 11 and 12. If the "number of characters to be modified" is less than the "number of characters per card" a programmed halt occurs and the rightmost characters are dropped.

Card Columns

Contents

Columns 15-80
(cont)

Decimal: Decimal data is converted to binary form and is stored, right-justified, in a field the length specified in columns 11 and 12, in the item image. A maximum of 12 decimal digits can be converted.

Octal: Each two characters beginning with the rightmost pair are combined to form one character, the result of which is moved to the item image. In general, the "number of characters per card" is even and equal to twice the "number of characters to be modified." If, however, the "number of characters per record" is odd, one character (zero) is appended to the left to make it even. If the "number of characters per card" is less than twice the "number of characters to be modified," leading zeros are inserted as necessary to create a field the length specified in columns 11 and 12. If the "number of characters per card" is greater than twice the "number of characters to be modified," a programmed halt occurs and the rightmost characters are dropped.

POST CARD

The post card controls the transfer of the item image to the buffer area and the writing of all records on tape. The post card is shown in Figure 3-8 and described below.

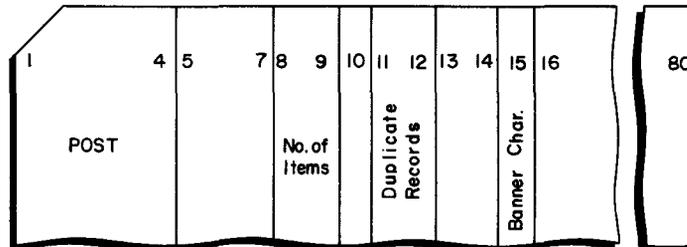


Figure 3-8. Post Card Format

Card Columns

Contents

Columns 1-4

Contain the letters "POST" to identify the type of card.

Columns 8-9

Identify the number of times the current item image is to be transferred to the buffer. This entry may range from 00 to 99.

Columns 11-12

Identify the number of duplicate records to be written following the posting of items. This entry may range from 00 to 99.

Card Column

Contents

Column 11-12
(cont)

If the last item posted produced a "write" on tape, the entire record is repeated. If, however, the last item posted did not cause a "write," only those items posted since the preceding "write" are written on the tape. The buffer is cleared each time a "write" on tape occurs.

Column 15

Specifies the banner character to be automatically inserted in each bannered record written following the post card. As stated in the file card description, the file card will set this banner character back to 40₈ each time the card is encountered. If column 15 is left blank, the last banner character used will be assumed for current use.

END CARD

The last program card contains the entry ENDA (see Figure 3-9), which indicates to the Test Data Generator program that the end of the card input deck has been reached.

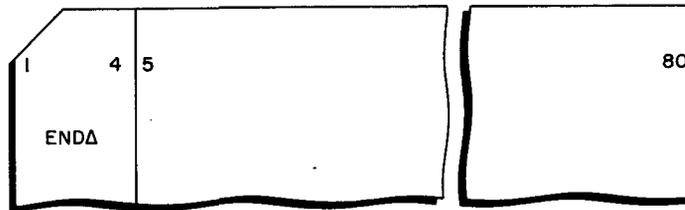


Figure 3-9. End Card Format

Operating Considerations

The Test Data Generator program is used with the Program Test System and stored as a loading unit on the system BRT to be loaded by the Tape Loader-Monitor program. A call to search for and load the Test Data Generator program is issued by placing the Test Data Generator console call card (see Figure 3-10) in the test director input card deck.

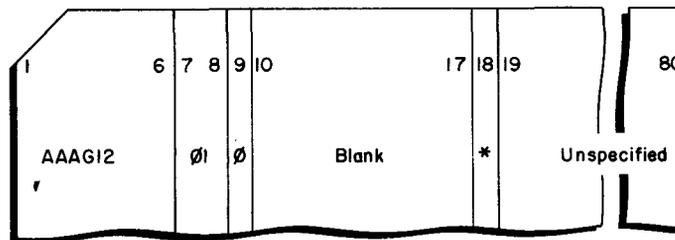


Figure 3-10. Console Call Card for Loading Test Data Generator

HALTS

Test Data Generator program halts may occur for two general reasons: (1) logical errors and halts which are generated by input control card data and (2) peripheral errors. For all halts, the A- and B-address registers (which indicate the cause of the halt) should be displayed and handled as follows:

Data Card Error

Test Data Generator will halt with 0041XX in the B-address register when data cards are found to contain inconsistent parameters. An illegal data card is not easily correctable during operation of the program and it is therefore suggested that the contents of the B-address register be recorded on a visually distinguishable card and reinserted in front of the next to the last card read (the illegal card). Depress RUN. Test Data Generator will assume maximum consistent parameter values and continue processing.

This procedure allows the user to identify the invalid data card and distinguish the type of error by interpreting the low order six bits comprising XX in the B-address register (0041XX) as follows:

<u>B-Address Register</u>	<u>Reason for Halt</u>
004120	The character position specified in columns 8 through 10 is greater than the maximum number of characters specified per item.
004110	The number of characters on the card as specified in columns 6 and 7 is larger than necessary for generating a field of the length specified in columns 11 and 12.
004104	The number of characters to be modified (columns 11 and 12) and the character position (columns 8 through 10) are such that moving the field would exceed the leftmost character of the item.

Item Card Error

An item card error is indicated by a halt with 004100 in the B-address register. This halt occurs if the item size specified on the item card is greater than the maximum item size specified on the file card.

The operating procedures to correct an item card error are the ones used for data card errors. Test Data Generator will assume the maximum item size specified on the file card and continue processing.

Card Read Error

A card read error is indicated by a halt with 000110 in the B-address register. The operator should correct the card and insert it in front of the unread cards in the card reader. Depress RUN.

Tape Write Error

A tape write error is indicated by a halt with 00cu2d in the B-address register. The letters cu represent the control unit number and d represents the drive number.

Unidentified Card Error

<u>B-Address Register</u>	<u>Reason for Halt</u>
004140	Columns 1 through 4 of a parameter card are not FILE, ITEM, DATA, POST, or ENDA. Depress RUN to bypass the illegal card.

MEMORY DUMP CONTROL

The Memory Dump Control program (AAADUM or AAAD4C) facilitates the taking of print-outs (memory dumps) of the core storage area. It is loaded by the Loader-Monitor program and uses the Memory Dump program (AAAMEM or AAAM4C) for the editing and printing of memory.

The Memory Dump Control program provides two methods of obtaining memory dumps. The first is by programmed instruction and the second is accomplished by item mark trapping. Programmed instruction allows for the taking of memory dumps by use of symbolic coding within the program. Item mark trapping may be accomplished by octal patching of the object program directly preceding the execution of that program, or by assembling the appropriate item marks into the program. Both methods of obtaining memory dumps may be used during the execution of an object program.

Addressing Modes

Three-character or four-character addressing is available with the Memory Dump Control program by use of the appropriate program name (see Table 2-1). The Memory Dump Control program may be called in either one or the other of the addressing modes, and the mode used to call the program is the mode in which both dynamic and terminal dump instructions must be assembled and executed and the mode in which the program returns to the object program, regardless of where the calling sequence is located.

The required sizes for the three-character version of the Memory Dump control program are 755 characters of memory for programmed instruction memory dumps and 795 characters of

memory for memory dumps obtained through item mark trapping. Because the Memory Dump program (AAAMEM) is called in to edit and print the locations specified by Memory Dump Control, 625 characters are added to each of the above figures to obtain the total number of locations required. An object program using both programmed instruction and item mark trapping to obtain memory dumps would require the maximum of 1,420 character locations.

The required sizes for the four-character version of Memory Dump Control (AAAD4C) are 980 memory locations for programmed instruction memory dumps and 1,085 memory locations for item mark trapping dumps. An additional area of 780 locations to be used by Memory Dump (AAAM4C) must be added to each of the above figures to obtain the total area required by the Memory Dump Control program.

The Memory Dump Control program is self-initializing and relocatable in memory with the relocation address being specified on the console call card. The fixed start areas in the loader are used for working storage and must not be altered by the user's object program. (Fixed start areas use locations 90 through 101.)

When item mark trapping is to be utilized, the machine is set to the "trapping mode" as Memory Dump Control is loaded. The user's object program must not alter the mode.

While Memory Dump (AAAMEM or AAAM4C) uses but does not restore index register 6, Memory Dump Control saves and restores that register, so that the object program may use X6 as desired even when taking dynamic dumps.

Programmed Instruction Dumps

Programmed dump instructions may be coded in the user's program to produce dumps at given points in that program. However, programmed dumps are optional at execution time and will not be performed if the Memory Dump Control console call card is not present.

There are four different instruction formats which may be used to obtain a dynamic object program dump. These instruction formats may be utilized in the user's program coding or for manual entry as described in Section IV under "Recovery Procedures."

Each of the formats which are discussed below must be executed in the addressing mode in which the program was called and control will always return to the object program in the same mode.

FORMAT 1

BCE (091), 090, *
NOP FROM, TO

The BCE instruction initiates a dynamic dump if the Memory Dump Control program has been loaded. The area to be dumped is specified by the FROM and TO parameters of the A and B addresses in the NOP instruction. Control is returned to this instruction.

FORMAT 2

BCE (091), 090 *
Any instruction other
than NOP

The second memory dump format is like format 1 with the exception that no new dump parameters are specified. The Dump control program uses the FROM and TO addresses as specified in the previous memory dump. If no previous addresses have been specified, the address locations 0000 to 8,191 (decimal) are assumed by the program and that area is dumped. Control is returned to the instruction following the BCE instruction.

FORMAT 3

BCE (095), 090, *
NOP FROM, TO

A terminal dump is produced as a result of the third format. The area in memory to be dumped is specified in the A- and B-addresses of the NOP instruction. Control is returned to the Loader-Monitor.

FORMAT 4

BCE (095), 090, *
Any instruction other
than NOP

Format 4 is like format 3 with the exception that no new dump parameters are specified. The Dump Control program uses the FROM and TO addresses as specified in the previous memory dump. If no previous addresses have been specified, the address locations 0000 to 8,191 (decimal) are assumed by the program and that area is dumped. Control is returned to the Loader-Monitor.

Item Mark Trapping

Item mark trapping¹ provides another means for taking a programmed memory dump. Placing an item mark over the operation code of an instruction will effect the programmed memory dump: no memory locations are used.

¹ Any object program that makes use of item mark trapping (as do the programs produced by Easytran and Bridge) should not utilize the item mark trapping feature of the Program Test System to initiate memory dumps. Such programs can perform the dumping function through the "programmed instruction" memory dump method.

Memory dumps taken by item mark trapping are similar to those taken by programmed instructions. The following paragraphs detail the capabilities of item mark trapping and note the requirements for both three-character and four-character addressing.

When memory dumps are taken by item mark trapping, dump parameters cannot be set dynamically. Alterations to the memory dump parameters can be made via the operator's console or control panel by patching the program or by reassembling the program. If the memory dump parameters are modified by patching, the patches should be read in after the first segment of the object program is loaded.

The memory dump parameters are located in the program at the relative octal locations specified below should alterations need to be made. To compute the absolute addresses of the from and to parameters, the addresses given below must be added to the base address of the program being used (AAADUM or AAAD4C). Each parameter is a three-character (18-bit) address with a word mark on the leftmost character.

<u>Three-Character Address (AAADUM)</u>	From 1500-1502
	To 1503-1505

<u>Four-Character Address (AAAD4C)</u>	From 2056-2060
	To 2061-2063

Instructions that depend on the previous contents of the A-address, B-address or variant register, or that depend on previous indicator settings (such as compare and overflow), should not be used for item mark trapping because these registers are used but not restored by the Memory Dump program. The change sequence register is used in the trapping mode and must not be altered by the object program.

Instructions that set the A-address, B-address, or variant register for other instructions should not be used for item mark trapping because the trapped instruction is executed out of sequence.

Console Call Card

The console call card for loading the Memory Dump Control program is in the standard loader format (see Figure 3-11) with the exception that the base location for loading the Memory Dump Control program and certain optional parameters are entered in columns 12 through 17.

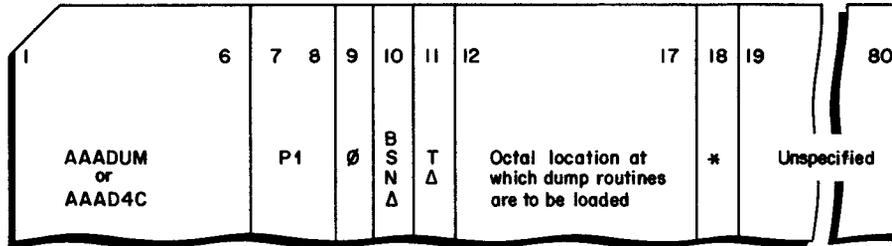


Figure 3-11. Console Call Card for Loading Memory Dump Control

Card Columns

Contents

Columns 1-6

Contain the letters AAADUM or AAAD4C, dependent on the addressing mode.

Columns 7-8

Contain the segment name, P1.

Column 9

Contains the number of the device from which the program is to be loaded (logical tape drive assignment).

Column 10

Specifies background loading and the object program "load-and-go" options: Δ (blank) - indicates that (a) "background" is to be loaded. (The term background refers to a halt character accompanied by a record mark.) It is loaded starting at location 1,373 (decimal) and ending at the location specified as the base location for the Memory Dump Control program, and (b) the object program is to be started at its starting address without operator intervention. (After initializing itself, the Memory Dump Control program reads the next console call card into the communication area of the Loader-Monitor program and makes a call to the loader.)

N - indicates that neither background nor object loading is desired.

B - indicates that background loading only is desired.

S - indicates that object program loading only is desired.

Column 11

Specifies whether or not the machine is to be set to the trapping mode.

T - indicates the machine is to be set in the trapping mode and printouts are to be taken when an item mark is sensed on an operation code (item mark trapping).

Δ - blank or any other character will leave the machine in a non-trapping mode.

Card Columns

Contents

Columns 12-17

Specify the octal location of the area into which the routines are to be loaded. All card columns should be punched in this field. For use with the three-character addressing mode, the loading area requires 1,380 consecutive memory locations when the programmed instruction method of obtaining a dump is used and 1,420 consecutive memory locations when item mark trapping is used. For use with the four-character addressing mode, this area must consist of 1,760 consecutive memory locations when the programmed instruction method of obtaining a dump is used and 1,865 consecutive memory locations when item mark trapping is used. In either mode, these locations must not be used by the program being dumped. In the three-character addressing mode, the base address of the area must be equal to or greater than 3,441₈ (1,825₁₀). In the four-character addressing mode, the base address of the area must be equal to or greater than 3,745₈ (2,021₁₀).

Column 18

An asterisk must appear in this column for identification at the time of loading.

Operating Considerations

Using the Memory Dump Control program to obtain dynamic dumps as described on page 3-16 involves the insertion of the proper instructions in the object program and the placement of the proper Dump Control console call card directly in front of the object program card deck. Dumps to be obtained through "Item Mark Trapping" require the proper insertion of item marks in the object program and the placement of the Dump Control console call card in front of the object program card deck (see Figure 3-12).

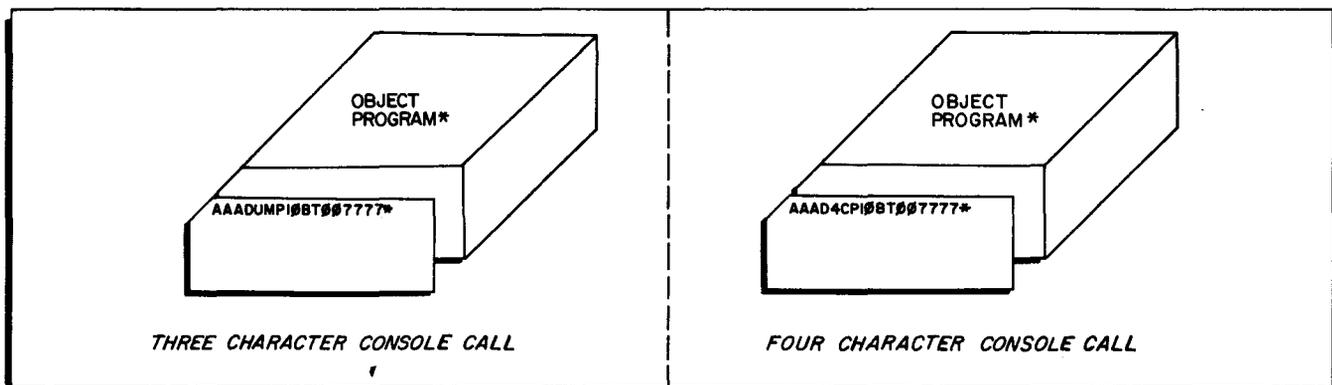


Figure 3-12. Memory Dump Control Console Call Card Placement

MEMORY DUMP

Memory Dump is a utility program that edits and prints data and punctuation in core memory. Specifically, it is used in conjunction with Memory Dump Control (AAADUM or AAAD4C) and Emergency Dump (AAASOS or AAAS4C) to edit and print the contents of memory as specified by the parameters given in those two programs. The Memory Dump program is stored on the Program Test System BRT from which it is loaded by either of the two companion programs mentioned above.

Addressing Modes

Three-character or four-character addressing modes are available in the Memory Dump program. For example, when the four-character mode of Memory Dump Control (AAAD4C) or Emergency Dump (AAAS4S) is called by the Loader-Monitor, the four-character mode of Memory Dump (AAAM4C) is loaded as part of the program.

Functional Description

The Memory Dump program consists of two separate parts: the main body of the routine and a print buffer. The print buffer is a fixed area which is located within the buffer of the Loader-Monitor. The coding of the main body of the program is relocatable at load time as determined by AAADUM or AAASOS. It may be loaded into any memory area where the required number of consecutive memory locations are available.

Figure 3-13 is an example of an object program memory dump as edited and printed by the Memory Dump program. The header line is produced by Memory Dump Control (AAADUM). The dump itself is produced by Memory Dump (AAAMEM).

Operating Considerations

The Memory Dump program is not used independently under the Program Test System, but only in conjunction with the Memory Dump Control and Emergency Dump programs. For independent use of Memory Dump, the reader is referred to the Honeywell Information Bulletin entitled PLUS - Memory and Tape Dump Routines, DSI-341.

TERMINAL											
DUMP 01 FROM LOC 07650 DTESTR 01 00003450 00003450 00000000 00000000 03002203 45006777 OFF OFF OFF OFF											
	ALPHA				OCTAL						
06640	JOHN	JOS	EPH	DOE	2396	NOR	TH	MADIS			
	3										
06700	ON	STR.	PHI	ADEL	PHIA	25	PENNSYLV				
	1		1			2					
06740	ANIA	C	5	034-26-1	652	L14	7.37	010			
	3		1					2			
	4 GROUPS OF 8 ALPHA CHARACTERS				4 GROUPS OF 16 OCTAL CHARACTERS						

Figure 3-13. Format of a Memory Dump

PATCH

Patch is a utility program which at object-program execution time makes programmer-specified octal corrections or changes to an object program in main memory. The corrections occur only in main memory (see Figure 3-14) and do not affect the object program stored on the binary run tape (BRT).

The corrections to be made are punched one-per-card in octal format. When Patch is loaded into memory by its console call card, it causes the loading of the object program and the reading of the octal correction cards. The corrections or changes indicated on these cards are then made to the specified locations of the object program, and the object program is automatically started.

Patch is stored on the Program Test System BRT and is loaded by the Tape Loader-Monitor. Since the program makes no provision for storing octal corrections, only segments (units) loaded by console call card may be "patched" (i.e., modified). Corrections to subsequent segments loaded under program control require programmed or operator intervention.

Addressing Modes

Three-character or four-character addressing is available with the Patch program by use of the appropriate program name (see Table 2-1).

<u>Addressing Mode</u>	<u>Program Name</u>
Three-character	AAAPAT
Four-character	AAAP4C

Four-character addressing must be applied when relocation of the Patch program is above memory location 32,768.

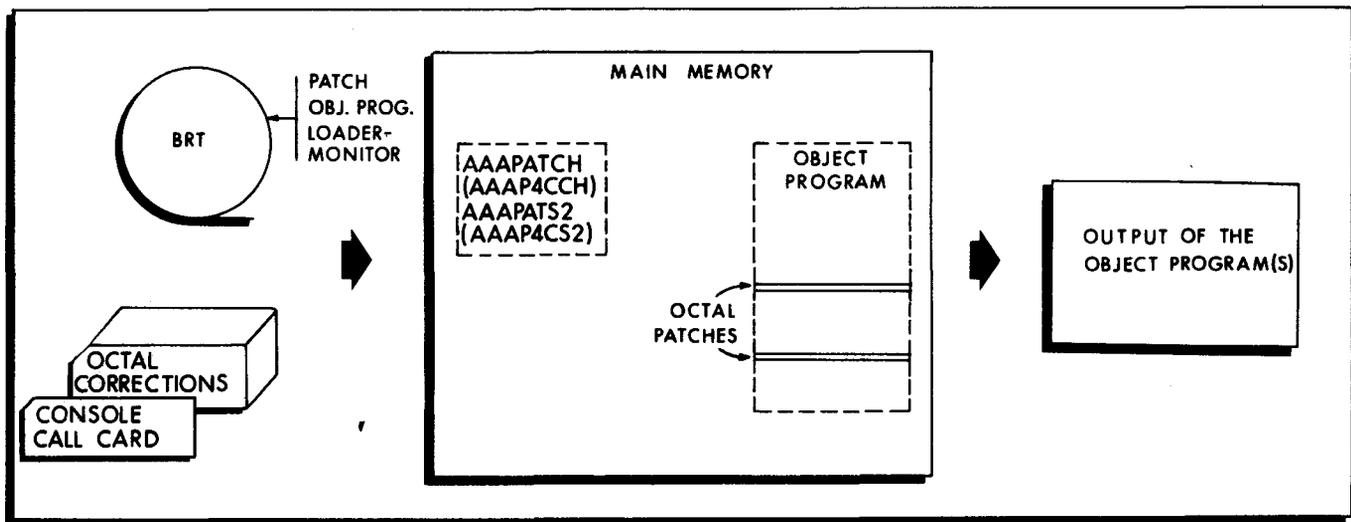


Figure 3-14. The Function of Patch

Functional Description

Patch is loaded by the Patch console call card and consists of two self-relocating segments, CH and S2, which require that 500 locations of memory for three-character addressing and 600 locations for four-character addressing be undisturbed by the loading of the object program. Index register six (X6) is also used by Patch and not restored. The locations will be occupied by the second segment of the octal correction routine but will be available to the object program after it has been patched and started. The starting location of this area is determined by a relocation augment specified in the object program's console call card, which is read by the first segment of Patch rather than by the loader.

Patch may be loaded by a programmed call to the loader or through a console call. The program to be patched, however, must have a console call card with additional required fields (columns 19 to 24 and 25 to 30). Peripheral arrangements needed for the object program (i. e., mounting its necessary tapes, etc.) should be completed before the octal correction program is loaded and started, because there are no operational halts in Patch. The object program is automatically loaded, patched, and started.

Format of the Required Cards

CONSOLE CALL CARD (CH)

A console call card is made up of control fields and two additional control fields for the program to be patched. The following paragraphs explain the use of each field on a console call card (see Figure 3-15). The format of this card is as follows:

Columns 1-6	AAAPAT	Program name.
Columns 7, 8	CH	Segment name.
Column 9	d	Device number (d) from which the unit is to be loaded.
Column 18	*	The asterisk identifies this card as a console call card.

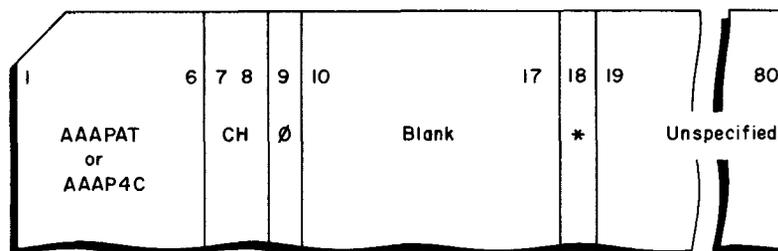


Figure 3-15. Console Call Card for Loading Patch

CONSOLE CALL CARD (USED BY S2)

The console call card used by Patch to call in the object program (see Figure 3-16) requires two additional fields (columns 19 to 30) in the console call card.

Card Columns

Contents

Columns 1-6	Program name of the unit ¹ to be loaded. When columns 1 through 6 are left blank, Patch reads and enters all patches without loading any object program. After all patches have been entered, control transfers to the specified address (columns 25 to 30). This operation allows the direct entry of a program on octal cards.
Columns 7, 8	Segment name of the unit to be loaded.
Column 9	Device number (d) from which the unit is to be loaded.
Columns 10-17	Program and segment name. After this program has been loaded, the loader halts. This program and segment may or may not be the same as in columns 1 through 8 above, or may be blank (depending on whether or not the programmer wishes to halt at the end of the first segment loaded, at a later point, or not at all).
Column 18	An asterisk (*) identifying this card as a console call card.
Columns 19-24	The relocating augment to be applied to the second segment of Patch, expressed in octal (must be > 3221 ₈). The second segment is assembled with the origin at location 0. Therefore, any relocation augment applied is the actual address for loading the second segment. Since the segment processes the octal correction cards after the object program has been loaded, the area in main memory which contains S2 of Patch must not be disturbed by the loading of the object program (which is accomplished by CH of Patch).
Columns 25-30	The start address of the object program, expressed in octal. This is the address at which the object program is to start after all corrections have been made. If this field is blank, the normal assembled start address is used.

¹A loading unit is a portion of code found and loaded as a result of a single call to the loader. This unit appears on the BRT as one or more consecutive records, the first of which is a segment-header record.

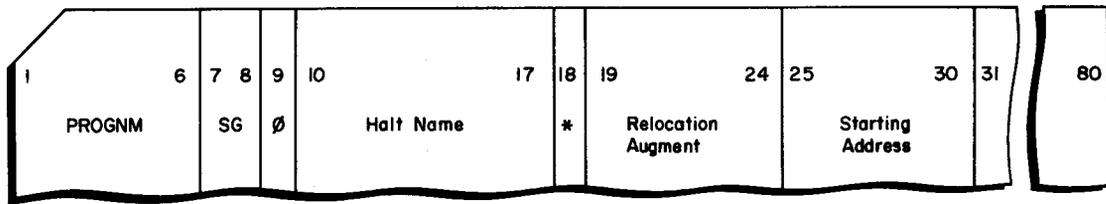


Figure 3-16. Console Call Card for Loading an Object Program to be Patched

OCTAL CORRECTION CARD

The octal correction facility enables corrections to be made to specified units without the need for submitting these units to the assembler. The op codes, addresses, and variant characters of these cards are written in octal notation, with two card columns representing one machine character.

The format of the octal correction card is as follows:

Card Columns

Contents

Columns 1-3

Contain the letters OCT.

Column 5

Contains the letter which specifies the punctuation to be placed at the left end of the string of characters. The codes employed are as follows:

- Δ No punctuation
- W Word mark
- I Item mark
- R Record mark

Column 6

Contains the letter which specifies the punctuation to be placed at the right end of the string of characters. The codes employed in this column are exactly the same as those used in column 5. If a single-character correction is made, the punctuation must be set at the right end.

Columns 7-12

Specify the location into which the leftmost character is to be loaded. This location must be punched as a three-character address, regardless of the addressing mode of the unit.

Columns 15, 16

Contain the octal command code which must be punched in these columns for a machine instruction. For a constant, this field must remain blank.

Columns 21-78

Contain a machine instruction or a constant. When a machine instruction is punched, the A address, B address, and variant characters are punched in discontinuous fields as follows:

Card Columns
Columns 21-78
(cont)

Contents

Field	Columns		
	Two-char.	Three-char.	Four-char.
A address	21-24	21-26	21-28
B address	31-34	31-36	31-38
Variants	41-78	41-78	41-78

Variants must be punched, without imbedded blanks, starting in column 41. The first blank in an odd-numbered column terminates the field.

A constant appearing in the operands field must be punched in octal without imbedded blanks, starting in column 21. The first blank in an odd-numbered column terminates the field. Data constants and address constants receive identical processing.

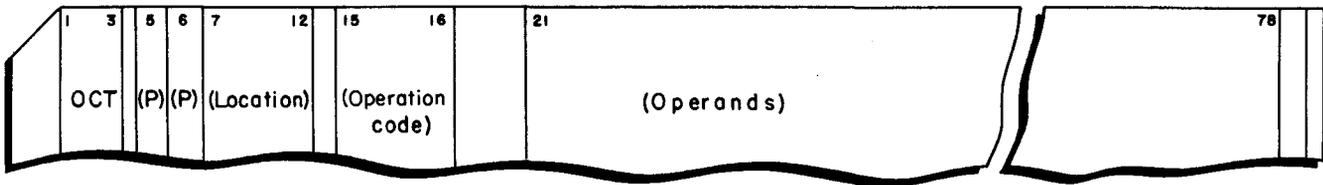


Figure 3-17. Octal Correction Card

TERMINATION CARD

A blank card must follow the octal correction cards to terminate Patch and to begin (or continue) execution of the object program. The deck arrangement is illustrated in Figure 3-18.

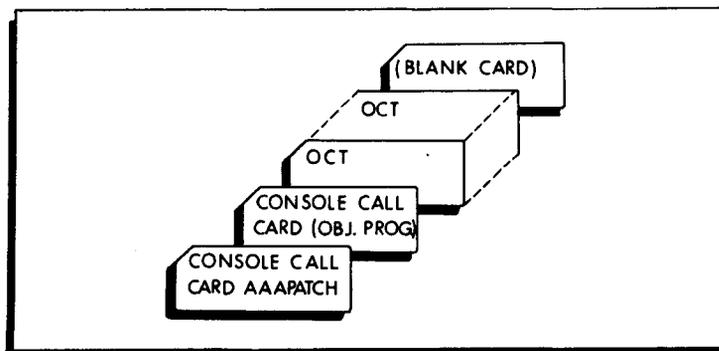


Figure 3-18. Deck Arrangement for Using Patch

Operating Considerations

To load and execute the Patch program, a console call card is placed in the test director deck. This is followed in turn by the object program console call card and the correction cards (see Figure 3-18).

TAPE DUMP

Tape Dump (AAATAP) is a utility program used to position magnetic tape files mounted on Model 204B (half-inch) tape drives and to edit and print their contents. During a single execution, the Tape Dump program can process up to 99,999 records having any record length up to a maximum size of 1,100 characters per record.

Parameter Entry Cards

When the Tape Dump program has been loaded, it is ready to accept parameter entry cards (see Figure 3-19). One parameter entry card is required for each separate dump function to be executed: rewinding, tape positioning (forward or backward), or editing and printing. Any number of Tape Dump functions (equal to the number of parameter entry cards used) may be performed automatically under control of the Tape Dump program. Regardless of the number of parameter entry cards used, the last one must be followed by an End card (see Figure 3-20) to properly terminate the last function. Once operation has been terminated, the Tape Dump program returns control to the loader for further Program Test System processing. The various fields of the parameter entry cards are described below as they may be used with the Tape Dump program.

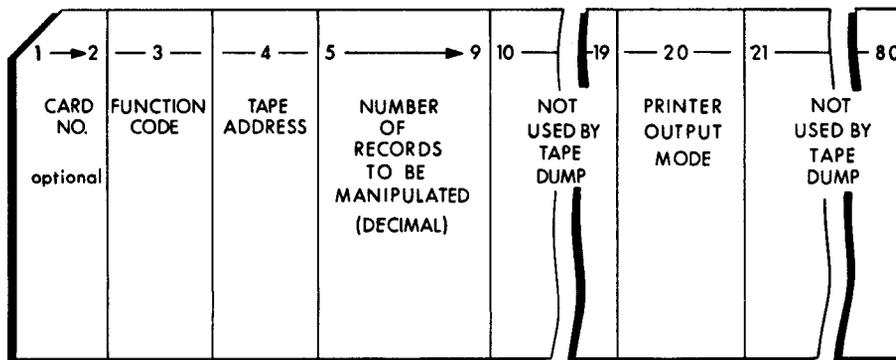


Figure 3-19. Tape Dump Parameter Entry Card

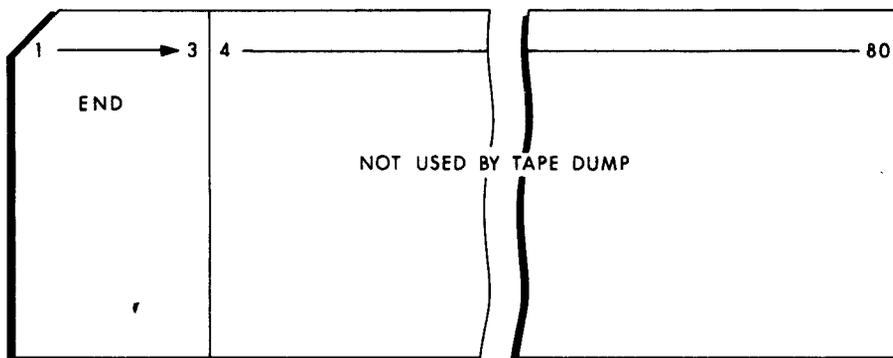


Figure 3-20. Last Card of Tape Dump Parameter Entry Card Deck

CARD NUMBER (COLUMNS 1-2)

The card number field provides a convenient means of identifying parameter entry cards; however, since the operation of the Tape Dump program is not contingent upon the card number, the use of this field is optional.

FUNCTION CODE (COLUMN 3)

The specific function to be performed by the Tape Dump program is designated by entering one of the four function code parameter values into column three. The function of moving tape forward is designated by the value F, backspacing is designated by B, rewinding by R, and editing by the value E. The parameter fields requiring entries to implement the use of each of the four function codes are designated in Table 3-2 in conjunction with the permissible range of values which may be used in these fields.

Table 3-2. Parameter Fields Used in Conjunction with the Four Function Codes

Function Code (Field 2)	Tape Address (Field 3)	No. of Records (Field 4)	Printer Output Mode (Field 5)
F	0-7	00001-99,999	-
B	0-7	00001-99,999	-
R	0-7	-	-
E	0-7	00001-99,999	An A in field 5 designates an alphabetic listing; any other entry designates an octal listing.

TAPE ADDRESS (COLUMN 4)

The logical address of the tape unit to be used in the function specified in field 2 must be entered into column four. As indicated in Table 3-2, the permissible values for the tape address parameter are 0 through 7. A parameter value that has been entered into a parameter field remains there until explicitly changed; thus, once a tape address has been entered into parameter field 3, subsequent Tape Dump functions using the same tape do not require reentry of the tape address.

NUMBER OF RECORDS (COLUMNS 5-9)

The "number of records" parameter field is a five-character field which specifies the number of records involved in the requested function. A value from 00,001 to 99,999 (decimal) must be entered into columns five through nine for three of the functions, as indicated in Table 3-2. Only the rewind (R) function does not require an entry in the "number of records" field.

PRINTER OUTPUT MODE (COLUMN 20)

The "printer output mode" parameter is used only in conjunction with the edit (E) and print function. When an alphabetic listing is required, the value A is entered in column 20. Any other character (or a blank) in column 20 designates an octal listing.

Operating Considerations

The procedures involved in operating the Tape Dump program are under automatic control. The user should concern himself with the proper placement of the Tape Dump console call cards and the parameter entry cards in the test director deck only. These cards are usually placed directly behind the object program console call cards.

The Tape Dump console call card is in the standard console call card format described in Section II and shown in Figure 3-21.

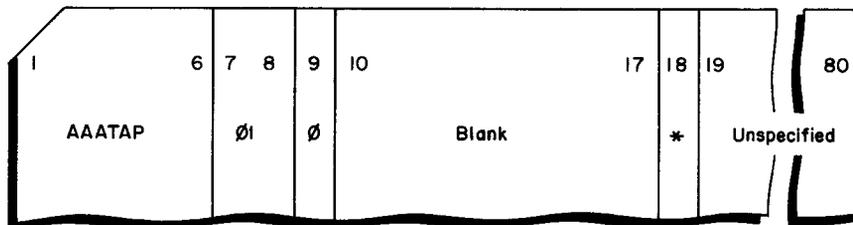


Figure 3-21. Console Call Card for Loading Tape Dump

The Tape Dump program has five programmed halts which are identified by displaying the A-address and B-address registers. Table 3-3 lists the five programmed halts with the applicable coding for each halt, the reason for halting (cause), and the appropriate operator action.

Table 3-3. Coded Halts for Tape Dump

A Address	B Address	Reason for Halt	Operator Action
00xxxx	*00cu10	Hole-count error	Depress START to ignore parameter last read. Otherwise, runout cards in feed hopper and place card in eject stacker in front of the deck. After placing deck in feed hopper and cycling up the card reader, depress RUN button.
00xxxxx (card #)	004201	Invalid tape request	Check parameter field 3 (parameter entry card, column 4) for illegal tape address. After parameter is corrected, depress RUN.

Table 3-3 (cont). Coded Halts for Tape Dump

A Address	B Address	Reason for Halt	Operator Action
00xxxx	004203	Invalid function request	Check parameter field 2 (parameter entry card, column 3) for illegal function request (not F, B, R, or E). Make correction and depress RUN.
00xxxx (card #)	004204	Number of records not consistent with function request	Check parameter field 2 (parameter entry card, columns 5-9) for incorrect "number of records" entry (i. e., a value of 0 for other than a rewind request, or a value not 00001-99,999 or F, B, and E requests). Correct error via the operator's control panel (or console); then depress RUN.
*cu = Control Unit			

Format of a Tape Dump Listing

The printed listing produced by the Tape Dump program may be in either alphabetic or octal format. As may be observed from Figure 3-22, an alphabetic listing prints 100 characters per line and the octal equivalent prints 50 characters per line. In addition to the features shown in Figure 3-22, the Tape Dump program provides two error flags which are printed in the left-hand margin when the designated error condition exists. The two error flags are PLE (print error) and RDE (tape read error).

Each edit and print operation (function code E) directs the printing of a seven-character header line at the left-hand margin of the Tape Dump listing. The seven characters are the "function code," "tape address," and the five-character "number of records" parameter values used to direct the dumping of the record(s) which are printed immediately below the header line. Thus, the header line E100003, which immediately precedes both the alphabetic listing (records designated A, B, and C) and the octal listing (records designated D, E, and F) in Figure 3-22, identifies a tape dump that was specified by a function code value of E, a tape address value of 1, and a number of records value of 00003.

The header line is followed by the lines of dumped information. Each line is preceded by a four-digit number which indicates the relative position within the record of the leftmost character in the line.

EMERGENCY MEMORY DUMP

The Emergency Memory Dump utility program provides a means of taking an emergency printout of core storage if difficulty is encountered during object program execution.

```

E100003
A → TAPE1 0001 0002AAATAP010
B → { 0001 0002AAATAP010          *V02CV000V000V000V06 B0001+N-00000000V06 V05RD0+:YV03B02C2 0-080+001 /N0101Y2"0
      0101 3B0101U02#RJGFW014RJJU02YRJ8U05#RJ
C → { 0001 0002AAATAP010          *V02CV000V000V000V06 B0001+N-00000000V06 V05RD0+:YV03B02C2 0-080+001 /N0101Y2"0
      0101 3B0101U02#RJGFW014RJJU02YRJ8U05#RJK"03B01E*V02580+J0+ RVO %01:05W701:064701:04X701:06%701:06Q7 0
      0201 +L00H:01#04DB04JB06PC0 ,=V05R+0+.0+4V056KW0+ *-0%0+)0+4.0+.0+4V05RKV04/804JN0001-8V04-N000118V02S:0+
      0301 0+6W0+ *- U05X*-8U0680--"06"0+/E:0+V0+4B0+W 0+L00D)0+W00D-V0*CV010*08*0003:Q000+50+50796BQ01:00H0+
      0401 98Q01 00 000HYD00DXC X BY+7*06 0+55*0820+54+

E100003
D → TAPE1 0001 0000000221212163214700010015151515151515
E → { 0001 000000022121216321470001001515151515151554650002236500000065000000650000065000657220000001204540
      0051 0000000000000000650006576500055124002014706500032200022302150040001000203600016061450001000170025500
      0101 032200010001640002525141272666000104514164000270514110640005365141
F → { 0001 000000022121216321470001001515151515151554650002236500000065000000650000065000657220000001204540
      0051 00000000000000000650006576500055124002014706500032200022302150040001000203600016061450001000170025500
      0101 0322000100016400025251412726660001045141640002705141106400053651414255000322000125546500026210002041
      0151 0020155165001635320001140005660732000114000604073200011400046707320001140017350732000114001750071500
      0201 2043000030140001520004242200044122000647230015734065000551330020330020046500051742660020601240003500
      0251 2034002004330020330020046500055142650004612200044145000000014010650004404500000001011065000262140020
      0301 5600200666002060124060640005671240106400171000404055001755002061251400206500200422002066150020430000
      0351 2434002066000024406500122365000146540010125000000314500000002005320020050007111722500001140000300020
      0401 1110500001600000372400003070240000246723577777677777227020075400065700200505540010020020050420

```

NOTE:

CALLOUTS A, B, AND C IDENTIFY THREE MAGNETIC TAPE RECORDS DUMPED IN ALPHABETIC FORMAT;

CALLOUTS D, E, AND F IDENTIFY THE SAME THREE RECORDS IN OCTAL FORMAT.

Figure 3-22. Sample Tape Dump Listing

Emergency Memory Dump calls in the Memory Dump program and sets the desired emergency memory dump parameters as specified by the program. Program control is then transferred to the Memory Dump program to edit and print the specified area, after which control is returned to the Loader-Monitor. The monitor ignores all succeeding cards until the next console call card is read as shown in the following example.

Addressing Modes

Three-character or four-character addressing is available with the Emergency Memory Dump program. Four-character addressing must be used when the Emergency Memory Dump program is assembled to run above memory location 32,768. All operating characteristics of the four-character addressing version of Emergency Memory Dump are like those for three-character addressing with the exception of the program console call name (AAASOS for three-character and AAAS4S for four-character addressing).

The memory requirement for the three-character addressing version of the Emergency Memory Dump program is 125 locations plus the 625 locations to be used by its companion program, Memory Dump. The memory requirement for the four-character version is 340 locations plus the 780 locations to be used by Memory Dump. The standard procedure for program allocation of memory area is to specify through installation convention the locations to be used by the Emergency Memory Dump program. These locations should not be available for use in processing object programs.

Parameters

The Emergency Memory Dump program indicates "FROM" and "TO" emergency memory dump parameters in the program for a machine configuration of 12,287 memory locations. These parameters may be changed to accommodate the user's machine configuration at any time by reassembling the program. The following example indicates the parameters as they are currently set in the Emergency Memory Dump program.

EASYCODER
CODING FORM

PROBLEM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____

CARD NUMBER	OPERATION CODE	LOCATION	OPERANDS	
			14 15	20 21
1	ORIGIN	ORG	11538	
2	FROM	DSA	0	
3	TO	DSA	12287	
4				
5				
6				
7				
8				
9				

Operating Considerations

To load and execute the Emergency Memory Dump program, a call must be issued for the Tape Loader-Monitor to initiate a search for Emergency Memory Dump on the Program Test System BRT. The call may be issued by a console call card or via an operator's call using the console control panel.

The console call card procedure is suggested as the most flexible method for calling and loading Emergency Memory Dump. This procedure requires a console call card to direct the Loader-Monitor to search for Emergency Memory Dump. After the program has been loaded, the program rewinds the system BRT, prints an emergency headerline and calls the Memory Dump program (AAAMEM) to overlay the Emergency Memory Dump program and edit and print the memory area specified by the parameters in the Emergency Memory Dump program.

CONSOLE CALL CARD

When a console call card is used to load Emergency Memory Dump, the console call card (see Figure 3-23) must be punched with the appropriate program name, segment name, and tape drive number (tape 0).

<u>Addressing Mode</u>	<u>Program Name</u>	<u>Segment Name</u>
Three-character	AAASOS	01
Four-character	AAAS4S	01

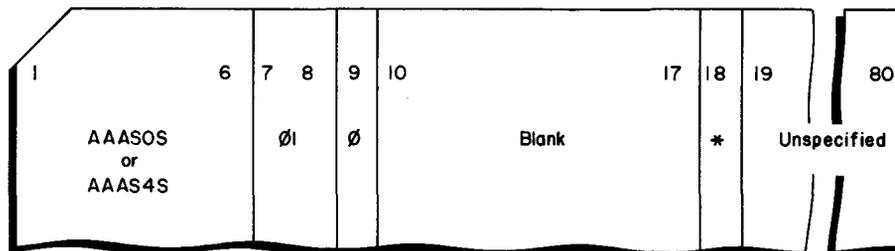


Figure 3-23. Console Call Card for Loading Emergency Memory Dump

The console call card should be inserted in front of the unread cards remaining in the card reader. Before executing Emergency Memory Dump, the operator may record pertinent register settings on the Emergency Memory Dump console call card and mark the card so that it is easily distinguishable. Such precautions are valuable for determining the point in the input deck as well as the register settings at the time a program hang-up occurs.

CONSOLE ENTRY

If an emergency dump of core storage is requested by the operator using the console control panel, the following search parameters must be entered to search for and load the Emergency Memory Dump program.

Three-Character Addressing:		Four-Character Addressing:	
<u>Location (Octal)</u>	<u>Content</u>	<u>Location (Octal)</u>	<u>Content</u>
100	01g	100	01
104-111	AAASOS	104-111	AAAS4S
112-113	01	112-113	01

TECHNICAL NOTE

In the event the object program destroys the Loader-Monitor in memory, the operator may follow the installation conventions that provide for a manual memory dump. After a manual dump, the Loader-Monitor must be re-initialized before processing can continue.

END

End (AAAEND) is a utility program which terminates the Program Test System's automatic checkout run after all designated object programs have been tested. The End program performs the following functions:

1. Rewinds the system BRT.
2. Prints a message indicating the last object program has been tested.
3. Returns control to the Tape Loader-Monitor at halt number three.

Console Call Card

The console call card is in the standard Loader-Monitor console call card format described in Section II and illustrated in Figure 3-24. It should be placed at the end of the test director deck followed by at least three blank cards.

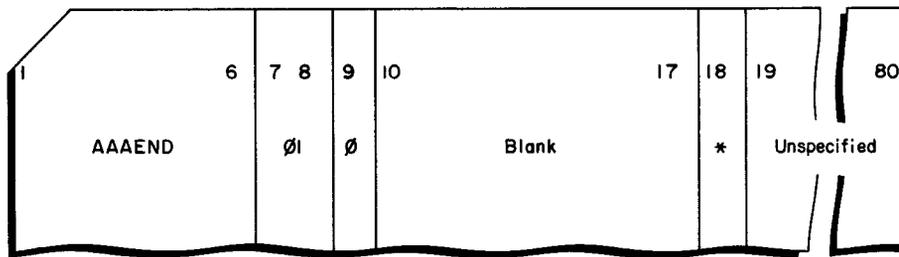


Figure 3-24. Console Call Card for Loading End

SECTION IV
OPERATING PROCEDURES FOR PROGRAM TEST SYSTEM C

INITIAL SETUP PROCEDURES

SENSE switches are not used by the programs of the Program Test System but are left for use by the object programs.

Place the Program Test System run deck in the card reader. Initialize the reader, ascertaining that it is ready to accept programmed commands.

Initialize the printer, ascertaining that it is ready to accept programmed commands.

NOTE: References to card readers and printers imply control units 41 and 02 respectively.

TAPE MOUNTING PROCEDURES

Mount the BRT containing the Loader-Monitor and the Program Test System on the tape drive designated as logical 0. Set the PERMIT-PROTECT switch to PROTECT.

If an object program BRT exists, mount the tape involved on the tape drive designated in column 9 of the object program console call card. This tape must be on the same tape control unit as the system BRT.

NOTE: Input tapes should be mounted in PROTECT status, while output tapes should be in PERMIT status. The DENSITY switch for an input tape must be in agreement with the density in which the tape was produced.

BOOTSTRAP PROCEDURES

Bootstrap procedures are detailed in the Tape Loader-Monitor bulletin, DSI-327.

CONSOLE CALL CARD PROCEDURE

The eight-character program and segment names for calling each utility program are listed in Table 4-1 in the order in which they may be called. If manual entry of the Memory Dump Control or Emergency Dump program is required, refer to "Recovery Procedures" on page 4-2.

Table 4-1. Programs of Program Test System C

No.	Program	Program Name	Segment Name
1	Initializer	AAATST	01
2	List Comments	AAASPE	AK
3	Test Data Generator	AAAG12	01
4	Memory Dump Control		
	Three-character addressing	AAADUM	P1
	Four-character addressing	AAAD4C	P1
5	Memory Dump		
	Three-character addressing	AAAMEM	01
	Four-character addressing	AAAM4C	01
6	Patch		
	Three-character addressing	AAAPAT	CH
	Four-character addressing	AAAP4C	CH
7	Tape Dump	AAATAP	01
8	Emergency Memory Dump		
	Three-character addressing	AAASOS	01
	Four-character addressing	AAAS4S	01
9	End	AAAEND	01

RECOVERY PROCEDURES

Difficulties encountered during the running of an object program checkout are most expediently handled by recording the pertinent register settings at the time of the hang-up and then using one of the following recovery methods to secure emergency memory dumps and to resume operation.

Emergency Memory Dump

The Emergency Memory Dump program (AAASOS or AAAS4S) may be requested by entering the search parameters for this program on a console call card or via the console.

It will be helpful to the programmer in determining the cause of the difficulty if an easily distinguishable card (for example, a colored card) is used as the console call card for the emergency dump. The register settings can be recorded on this card before inserting it in front of the unread cards in the reader. Then, when the card deck is returned to the programmer, he

will know at what point in the processing of his input cards the trouble occurred. After the dump has been taken, the monitor will bypass the remaining data cards in searching for the next console card. A supply of emergency dump console call cards, kept in a convenient place near the card reader, will make it easy for the operator to follow this recommended procedure.

If, instead, it is desired to request the dump parameters from the console, the search parameters as given below must be entered, without disturbing the existing punctuation, before resuming operation.

<u>Location (octal)</u>	<u>Contents (alpha)</u>
100	1
104-111	AAASOS or AAAS4S
112-113	01

In both instances, the following procedure should be taken after parameter entry:

- Set the sequence register to 00126_8 .
- Depress the RUN button.

The program will provide an emergency memory dump and exit to the general return address of the loader (139_8). Program Test System operations will resume automatically.

NOTE: In the event the Loader-Monitor is overlaid, it must be manually re-initialized by the operator before the Emergency Dump program can be called.

Emergency Terminal Memory Dump

The Memory Dump Control program (AAADUM or AAAD4C), if undisturbed in memory at the time of the hang-up, may be requested via the console:

- Set the sequence register to 00136_8 .
- Depress the RUN button.

NOTE: The range of the terminal dump is the same as that set by the last dump taken.

The program will provide a terminal emergency dump and exit to the general return address of the loader (139_8). Program Test System operations will resume automatically.

Table 4-2. Error Halts

B Address	Significance	Operator Action
005000	Message has been printed.	Look at printer or console typewriter for instructions. Depress RUN to continue.
0041xx	Incorrectable input card format.	Record A-address and B-address registers on an easily identifiable card. Insert this card as the last card in the output stacker of the card reader. Depress RUN to continue. The program will assume consistent values for parameters in error on the card.
0042xx	Correctable input card format.	<p>Correct the card, refeed it, and depress RUN. The incorrect card will be ignored and the corrected card used.</p> <p>If correction is not possible, record the A-address and B-address registers on an easily identifiable card. Insert this card as the last card in the output stacker of the card reader. Depress RUN (the incorrect card will be ignored).</p>
014010	Program not found on the BRT when backward search was made.	Record the A-address and B-address registers on an easily identifiable card. Insert this card as the last card in the output stacker of the card reader. Set the sequence register to 00126 ₈ . Depress RUN to read the next console call card and continue operations.
017002	End of Program Test System run.	Terminate operation. (NOTE: This halt also occurs during initialization. Depress RUN at that time to continue program operation.)
<p>NOTE: For unspecified object program halts and other non-peripheral halts, record the contents of the A-address and B-address registers on an easily identifiable card. Insert this card as the last card in the output stacker of the card reader. Set the sequence register to 00126₈. Depress RUN to continue.</p>		

APPENDIX A
SAMPLE RUN OF PROGRAM TEST SYSTEM C

The examples shown on the following pages comprise a sample Program Test System run. Figure A-1 illustrates the format of the three sample test director decks and Figures A-2 through A-4 show the formats of each card in the three test director decks.

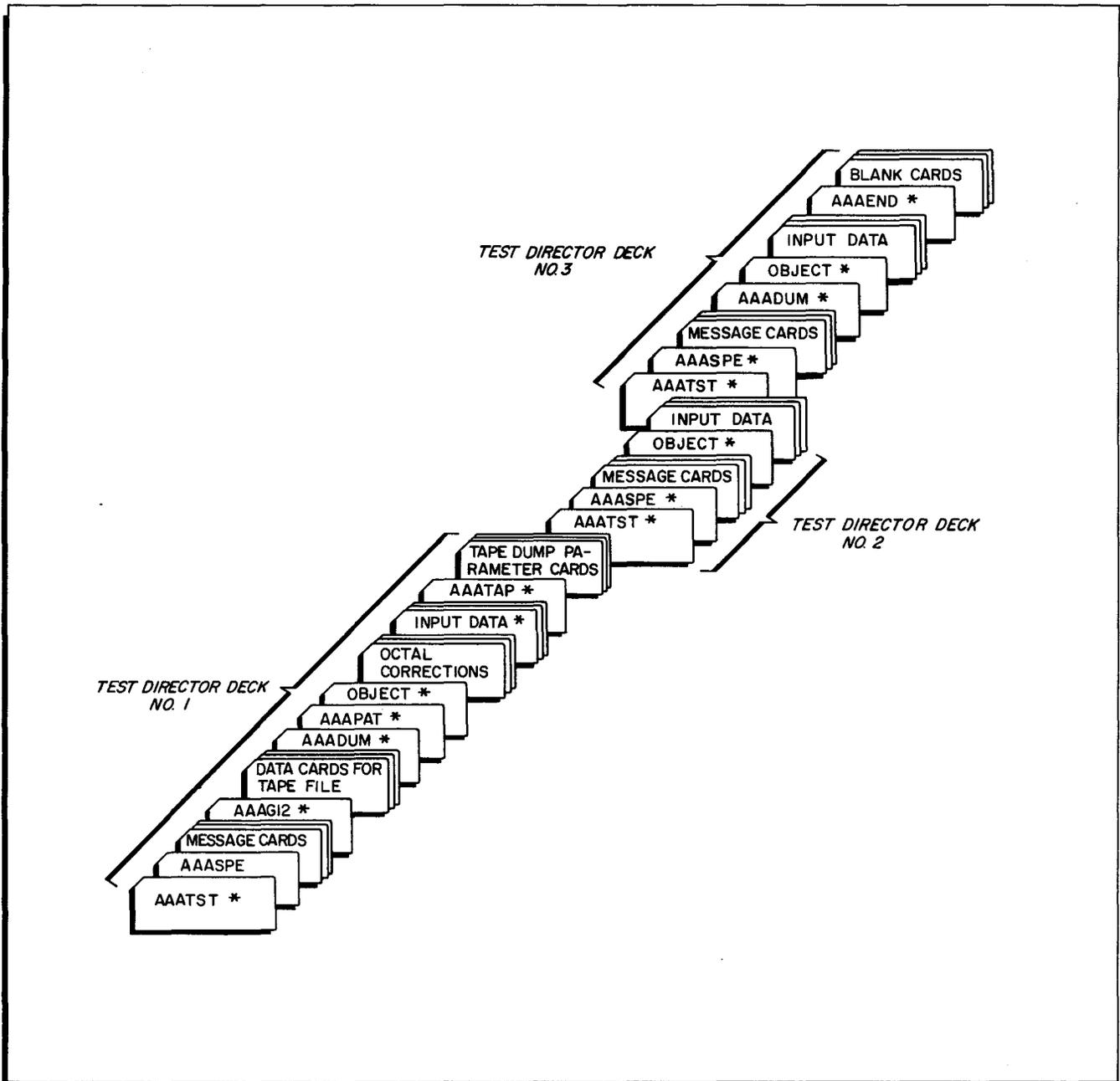


Figure A-1. Sample Testing Decks

With halt number three removed, automatic sequencing from one program to the next is possible. After performing the above functions, Initializer returns to the loader to read the next console call card from the test director deck.

The printout program, List Comments (AAASPE), is called next to list messages on the printer. The messages to be printed are contained on the message cards in the test director deck. When a blank card is sensed the program will halt with 005000 in the B-address register. After complying with the message which has been printed, the operator will depress the RUN button and List Comments will return to the loader to read the next console call card.

The Test Data Generator program is called in to create a specified test data tape file(s) from the detail cards that follow the program's console call card (AAAG12). Upon sensing an END Δ card in the program input deck, data generation is terminated and the program returns to the loader to read the next console call card.

The Memory Dump Control program, AAADUM, is the next program requested in the input deck. This program is loaded into memory beginning at the location specified in columns 12 to 17 (025000) of the console call card, and requires 1,420 memory locations. (A base location of 3,440₈ or greater is required.) The machine is then set to the trapping mode as trapping mode is specified in column 11 of the console call card. Memory Dump Control then loads the Memory Dump program, AAAMEM, into the memory area immediately following its own allocated memory area. The total area used by these two programs must not include memory locations used by the object program. Memory Dump Control returns control to the loader after the above processing to read the next console call card. The Memory Dump program is not executed at this time, but remains in memory ready to be called by the object program.

There are two segments to the Patch (AAPAT) program, CH and S2. The first segment (CH) of the program is loaded next into the memory area directly following the loader (octal locations 2,506 to 3,221). This segment reads the object program console call card and loads the second segment of the Patch program (S2) into the 500 memory locations beginning at the location specified in columns 19 to 24 (024000) of the object program console call card. Patch must not be loaded in the same area as the object program. The second segment of Patch then calls upon the loader to load the object program into memory and complies with the various octal corrections and punctuation formats contained on the detail cards that follow the object program console call card.

The first two octal correction cards used in this sample test (see Figure A-2) specify dump parameters of 000000 to 000030g (columns 21 to 26) with left justified word marks in the dump

parameter locations utilized by the Memory Dump Control program (see columns 5 and 7 through 12 respectively). The third octal correction card causes a record mark to be set (see column 6) on the operation code of an object program instruction stored in memory location 004,250. A printing of the contents of memory locations 00 to 30₈ (index registers) results each time the instruction in location 4,250 is executed during object program processing. The above entries accomplish what is referred to as item mark trapping (see page 3-17). When Patch senses a blank card, it terminates processing and transfers control to the object program.

The object program in turn reads and processes the input data cards (in the test director deck) and the generated data tape file. Upon termination of processing, the object program returns to the loader to read the next console call card. If desired, the object program may branch to Dump Control to take a terminal dump of memory, e.g., by executing the following instructions before returning to the Loader.

```
BCE    (095) , 090 , *  
NOP     0 , 12287
```

The Tape Dump program (AAATAP) is the last program requested in sample test deck number one. This program is called and reads the parameter cards which follow it to produce the specified terminal tape dumps. The tape dump parameter cards in this example (see Figure A-2) specify that tape 2 (column 4) is to be rewound (R in column 3) and that the first 55 records (columns 5-9) of this tape are to be edited (E in column 3) in alphanumeric format (column 20).

The END Δ card in Tape Dump terminates the program and control is returned to the loader to read the next console call card. In this instance, the next card will be the first card of test director deck number two.

Test Director Deck Number Two

The Initializer program is called for the second time in the Easytest Program Test System run and again reinitializes the system without halting. It rewinds the system BRT and resets the machine to the non-trapping mode before returning control to the loader to read the next console call card.

List Comments is called next to print (on the printer) the information contained on the message cards in the test director deck. When the GO Δ card is sensed, printing terminates and the program returns to the loader without halting. The next console call card is read.

The object program is called in and executed. During object program execution, the input data cards in the test director deck are read and processed. When processing is completed, the object program returns control to the loader to read the next console call card. The next card in the deck is the first card of test director deck number three.

Test Director Deck Number Three

The Initializer program is called and again performs the following:

1. Rewinds the system BRT,
2. Sets the machine to the non-trapping mode, and
3. Transfers control to the loader.

List Comments is the second program to be called and it lists operator instructions on the printer from the information contained on the message cards in the test director deck. Upon sensing a blank card, the program halts with 005000 in the B-address register. The operator, after complying with the printed instructions, depresses the RUN button to return program control to the loader to read the next console call card.

The Memory Dump Control program is then loaded into memory beginning at the location specified in columns 12 to 17 of the console call card (location 013000). Column 10 of this card indicates that background loading has been requested and column 11 specifies item mark trapping is not desired. When item mark trapping is not requested, it is assumed the object program contains programmed instructions for taking memory dumps.

Memory Dump Control then loads Memory Dump (AAAMEM) into the area immediately following the Memory Dump Control program's allocated area. The total area used by these two programs must not include any memory locations used by the object program. Memory Dump Control returns to the loader for the reading of the next console call card.

The object program console call card causes the object program to be called in and executed. Processing includes the reading of input data cards and the taking of dynamic dumps at specified points in the object program. When processing is terminated, control is transferred to the loader to read the next console call card.

The End program is the last program in the Program Test System input file to be called and executed. This program terminates the system run by:

1. Rewinding the system BRT,
2. Restoring loader halt number three,
3. Printing the message "END OF JOB" on the printer, and
4. Returning to the loader and halting at the halt number three (B-address register: 017002).

APPENDIX B
PROVISION FOR ADDITIONAL UTILITY PROGRAMS

The modular design of Program Test System C allows for the incorporation of additional utility programs. To maintain continuity throughout the system, programs and specialized routines which are to be added to the system should adhere to the following conventions.

1. Programs are to be written in either three- or four-character addressing mode.
2. Programs should terminate by (1) branching indirectly to the general return address of the Loader-Monitor, or (2) transferring control to a program that does return to the Loader-Monitor general return address.
3. Programs are to be inserted on the system BRT in the logical order of execution.
4. Programs to be executed when the Memory Dump Control program (AAADUM or AAAD4C) is in memory must not use the three fixed start areas in the loader communication area (octal locations 90 through 101).
5. Program halts, other than the conventional peripheral halts, should contain 041xx in the B-address register for incorrectable conditions and 042xx in the B-address register for correctable conditions. Advisably, the six bits comprising xx in these halt codes should differ from the existing Program Test System halt codes.

COMPUTER-GENERATED INDEX

ADDITIONAL UTILITY PROGRAMS
 PROVISION FOR ADDITIONAL UTILITY PROGRAMS, B-1

ADDRESSING MODE, 2-1
 ADDRESSING MODES, 3-15, 3-21, 3-22, 3-32
 * UTILITY PROGRAM NAMES,
 THREE- AND FOUR-CHARACTER ADDRESSING MODE
 UTILITY PROGRAM NAMES, 2-1

ARRANGEMENT
 DECK ARRANGEMENT FOR USING PATCH, 3-26

BOOTSTRAP PROCEDURES, 4-1

BRT
 PROGRAM TEST SYSTEM BRT, 2-2
 SYSTEM BRT, 2-2

CALL CARD
 CONSOLE CALL CARD, 3-4, 3-18, 3-33, 3-34
 CONSOLE CALL CARD (CH), 3-23
 CONSOLE CALL CARD (USED BY S2), 3-24
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM TO
 BE PATCHED, 3-25
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP,
 3-33
 CONSOLE CALL CARD FOR LOADING END, 3-34
 CONSOLE CALL CARD FOR LOADING INITIALIZER, 3-1
 CONSOLE CALL CARD FOR LOADING LIST COMMENTS, 3-4
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP CONTROL,
 3-19
 CONSOLE CALL CARD FOR LOADING PATCH, 3-23
 CONSOLE CALL CARD FOR LOADING TAPE DUMP, 3-29
 CONSOLE CALL CARD FOR LOADING TEST DATA GENERATOR,
 3-13
 CONSOLE CALL CARDS, 2-4
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5
 OBJECT PROGRAM CONSOLE CALL CARD, 2-5

* PLACEMENT,
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT,
 3-20

* PROCEDURE,
 CONSOLE CALL CARD PROCEDURE, 4-1
 PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5

CARD
 COMMENT CARD,
 SAMPLE FORMAT FOR A COMMENT CARD, 3-3
 CONSOLE CALL CARD, 3-4, 3-18, 3-33, 3-34
 CONSOLE CALL CARD (CH), 3-23
 CONSOLE CALL CARD (USED BY S2), 3-24
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM TO
 BE PATCHED, 3-25
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP,
 3-33
 CONSOLE CALL CARD FOR LOADING END, 3-34
 CONSOLE CALL CARD FOR LOADING INITIALIZER, 3-1
 CONSOLE CALL CARD FOR LOADING LIST COMMENTS, 3-4
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP CONTROL,
 3-19
 CONSOLE CALL CARD FOR LOADING PATCH, 3-23
 CONSOLE CALL CARD FOR LOADING TAPE DUMP, 3-29
 CONSOLE CALL CARD FOR LOADING TEST DATA GENERATOR,
 3-13
 CONSOLE CALL CARDS, 2-4
 DATA CARD, 3-10

* DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD
 DECK, 3-27

EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5
 END CARD, 3-13

* ERROR,
 DATA CARD ERROR, 3-14
 ITEM CARD ERROR, 3-14
 UNIDENTIFIED CARD ERROR, 3-15

FILE CARD, 3-6

* FORMAT,
 COMMENT CARD FORMAT, 3-2
 DATA CARD FORMAT, 3-11
 END CARD FORMAT, 3-13
 FILE CARD FORMAT, 3-8
 ITEM CARD FORMAT, 3-10
 POST CARD FORMAT, 3-12

FORMAT OF THE REQUIRED CARDS, 3-23

ITEM CARD, 3-10

LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD DECK,
 3-27

OBJECT PROGRAM CONSOLE CALL CARD, 2-5

OCTAL CORRECTION CARD, 3-25, 3-26

PARAMETER CARDS, 3-6

PARAMETER ENTRY CARDS, 3-27

* PLACEMENT,
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT,
 (CONT.)

CARD (CONT.)
 3-20

* POST CARD, 3-12
 PROCEDURE,
 CONSOLE CALL CARD PROCEDURE, 4-1

* PROCESSING,
 PARAMETER CARD PROCESSING, 3-5
 PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5

* READ ERROR, 3-4, 3-15
 TAPE DUMP PARAMETER ENTRY CARD, 3-27
 TERMINATION CARD, 3-26

CH
 CONSOLE CALL CARD (CH), 3-23

CODED HALTS FOR TAPE DUMP, 3-29

CODES
 FUNCTION CODES,
 PARAMETER FIELDS USED IN CONJUNCTION WITH THE
 FOUR FUNCTION CODES, 3-28
 LIST COMMENTS LINE SPACING CODES, 3-3

CODING ENTRIES
 SAMPLE CODING ENTRIES FOR HEADER AND TRAILER
 RECORDS, 3-7

COMMENT CARD
 * FORMAT, 3-2
 SAMPLE FORMAT FOR A COMMENT CARD, 3-3

COMMENTS
 * LINE SPACING CODES,
 LIST COMMENTS LINE SPACING CODES, 3-3
 LIST COMMENTS, 3-2
 LOADING LIST COMMENTS,
 CONSOLE CALL CARD FOR LOADING LIST COMMENTS, 3-4

CONJUNCTION
 PARAMETER FIELDS USED IN CONJUNCTION WITH THE FOUR
 FUNCTION CODES, 3-28

CONSIDERATIONS
 OPERATING CONSIDERATIONS, 3-1, 3-4, 3-13, 3-20,
 3-21, 3-26, 3-29, 3-33

CONSOLE CALL CARD, 3-4, 3-18, 3-33, 3-34
 * (CH), 3-23
 * (USED BY S2), 3-24
 CONSOLE CALL CARDS, 2-4
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5
 * FOR LOADING AN OBJECT PROGRAM TO BE PATCHED, 3-25
 * FOR LOADING EMERGENCY MEMORY DUMP, 3-33
 * FOR LOADING END, 3-34
 * FOR LOADING INITIALIZER, 3-1
 * FOR LOADING LIST COMMENTS, 3-4
 * FOR LOADING MEMORY DUMP CONTROL, 3-19
 * FOR LOADING PATCH, 3-23
 * FOR LOADING TAPE DUMP, 3-29
 * FOR LOADING TEST DATA GENERATOR, 3-13
 OBJECT PROGRAM CONSOLE CALL CARD, 2-5
 * PLACEMENT,
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT,
 3-20

* PROCEDURE, 4-1
 PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5

CONSOLE ENTRY, 3-33

CONTROL
 * CONSOLE CALL CARD PLACEMENT,
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT,
 3-20
 LOADING MEMORY DUMP CONTROL,
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP
 CONTROL, 3-19
 MEMORY DUMP CONTROL, 3-15

CORRECTION CARD
 OCTAL CORRECTION CARD, 3-25, 3-26

DATA CARD, 3-10
 * ERROR, 3-14
 * FORMAT, 3-11

DATA GENERATOR
 CONSOLE CALL CARD FOR LOADING TEST DATA GENERATOR,
 3-13
 TEST DATA GENERATOR, 3-5

DECK
 * ARRANGEMENT FOR USING PATCH, 3-26
 * NUMBER,
 TEST DIRECTOR DECK NUMBER ONE, A-2, A-3
 TEST DIRECTOR DECK NUMBER THREE, A-3, A-6
 TEST DIRECTOR DECK NUMBER TWO, A-3, A-5
 SAMPLE TEST DIRECTOR DECKS, 2-3
 SAMPLE TESTING DECKS, A-1
 TAPE DUMP PARAMETER ENTRY CARD DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD
 DECK, 3-27
 TEST DIRECTOR DECK, 2-3

DESCRIPTION (CONT.)

COMPUTER-GENERATED INDEX

DESCRIPTION
 FUNCTIONAL DESCRIPTION, 3-21, 3-23
 PROGRAM DESCRIPTION AND PREPARATION, 2-1

DIRECTOR DECK
 " NUMBER,
 TEST DIRECTOR DECK NUMBER ONE, A-2, A-3
 TEST DIRECTOR DECK NUMBER THREE, A-3, A-6
 TEST DIRECTOR DECK NUMBER TWO, A-3, A-5
 SAMPLE TEST DIRECTOR DECKS, 2-3
 TEST DIRECTOR DECK, 2-3

DUMP
 " CONTROL,
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP CONTROL, 3-19
 MEMORY DUMP CONTROL, 3-15
 " CONTROL CONSOLE CALL CARD PLACEMENT,
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT, 3-20
 EMERGENCY MEMORY DUMP, 3-30, 4-2
 EMERGENCY TERMINAL MEMORY DUMP, 4-3
 " LISTING,
 FORMAT OF A TAPE DUMP LISTING, 3-30
 SAMPLE TAPE DUMP LISTING, 3-31
 LOADING EMERGENCY MEMORY DUMP,
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP, 3-33
 LOADING TAPE DUMP,
 CONSOLE CALL CARD FOR LOADING TAPE DUMP, 3-29
 MEMORY DUMP, 3-21
 FORMAT OF A MEMORY DUMP, 3-21
 " PARAMETER ENTRY CARD,
 TAPE DUMP PARAMETER ENTRY CARD, 3-27
 " PARAMETER ENTRY CARD DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD DECK, 3-27
 PROGRAMMED INSTRUCTION DUMPS, 3-16
 TAPE DUMP, 3-27
 CODED HALTS FOR TAPE DUMP, 3-29
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5

EMERGENCY
 " MEMORY DUMP, 3-30, 4-2
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP, 3-33
 " TERMINAL MEMORY DUMP, 4-3

END, 3-34
 " CARD, 3-13
 " CARD FORMAT, 3-13
 LOADING END,
 CONSOLE CALL CARD FOR LOADING END, 3-34

ENTRIES
 CONSOLE ENTRY, 3-33
 SAMPLE CODING ENTRIES FOR HEADER AND TRAILER RECORDS, 3-7

ENTRY CARD
 " DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD DECK, 3-27
 PARAMETER ENTRY CARDS, 3-27
 TAPE DUMP PARAMETER ENTRY CARD, 3-27

EQUIPMENT
 OPTIONAL EQUIPMENT, 1-2
 REQUIRED EQUIPMENT, 1-2
 " REQUIREMENTS, 1-1

ERROR
 CARD READ ERROR, 3-4, 3-15
 DATA CARD ERROR, 3-14
 " HALTS, 4-4
 ITEM CARD ERROR, 3-14
 PRINT ERROR, 3-4
 TAPE WRITE ERROR, 3-15
 UNIDENTIFIED CARD ERROR, 3-15

FIELDS
 PARAMETER FIELDS USED IN CONJUNCTION WITH THE FOUR FUNCTION CODES, 3-28

FILE CARD, 3-6
 " FORMAT, 3-8

FORMAT
 COMMENT CARD FORMAT, 3-2
 DATA CARD FORMAT, 3-11
 END CARD FORMAT, 3-13
 FILE CARD FORMAT, 3-8
 ITEM CARD FORMAT, 3-10
 " OF A MEMORY DUMP, 3-21
 " OF A TAPE DUMP LISTING, 3-30
 " OF THE REQUIRED CARDS, 3-23
 POST CARD FORMAT, 3-12
 SAMPLE FORMAT FOR A COMMENT CARD, 3-3
 (CONT.)

FORMAT (CONT.)
 " 1, 3-17
 " 2, 3-17
 " 3, 3-17
 " 4, 3-17

FOUR-CHARACTER ADDRESSING MODE UTILITY PROGRAM N
 THREE- AND FOUR-CHARACTER ADDRESSING MODE UTILITY PROGRAM NAMES, 2-1

FUNCTION
 " CODES,
 PARAMETER FIELDS USED IN CONJUNCTION WITH THE FOUR FUNCTION CODES, 3-28
 " OF PATCH, 3-22

FUNCTIONAL DESCRIPTION, 3-21, 3-23

GENERATOR
 LOADING TEST DATA GENERATOR,
 CONSOLE CALL CARD FOR LOADING TEST DATA GENERATOR, 3-13
 TEST DATA GENERATOR, 3-5

HALTS, 3-14
 CODED HALTS FOR TAPE DUMP, 3-29
 ERROR HALTS, 4-4
 PROGRAM HALTS, 3-4

HEADER
 SAMPLE CODING ENTRIES FOR HEADER AND TRAILER RECORDS, 3-7
 INITIAL SETUP PROCEDURES, 4-1
 INITIALIZATION OF THE PROGRAM TEST SYSTEM, A-2
 INITIALIZER, 3-1
 LOADING INITIALIZER,
 CONSOLE CALL CARD FOR LOADING INITIALIZER, 3-1

INSTRUCTION DUMPS
 PROGRAMMED INSTRUCTION DUMPS, 3-16

INTRODUCTION, 1-1

ITEM CARD, 3-10
 " ERROR, 3-14
 " FORMAT, 3-10

ITEM MARK TRAPPING, 3-17

LINE SPACING CODES
 LIST COMMENTS LINE SPACING CODES, 3-3

LIST COMMENTS, 3-2
 CONSOLE CALL CARD FOR LOADING LIST COMMENTS, 3-4
 " LINE SPACING CODES, 3-3

LISTING
 SAMPLE TAPE DUMP LISTING, 3-31
 TAPE DUMP LISTING,
 FORMAT OF A TAPE DUMP LISTING, 3-30

LOADING
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM TO BE PATCHED, 3-25
 " EMERGENCY MEMORY DUMP,
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP, 3-33
 " END,
 CONSOLE CALL CARD FOR LOADING END, 3-34
 " INITIALIZER,
 CONSOLE CALL CARD FOR LOADING INITIALIZER, 3-1
 " LIST COMMENTS,
 CONSOLE CALL CARD FOR LOADING LIST COMMENTS, 3-4
 " MEMORY DUMP CONTROL,
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP CONTROL, 3-19
 " PATCH,
 CONSOLE CALL CARD FOR LOADING PATCH, 3-23
 " TAPE DUMP,
 CONSOLE CALL CARD FOR LOADING TAPE DUMP, 3-29
 " TEST DATA GENERATOR,
 CONSOLE CALL CARD FOR LOADING TEST DATA GENERATOR, 3-13

MARK TRAPPING
 ITEM MARK TRAPPING, 3-17

MEMORY DUMP, 3-21
 CONSOLE CALL CARD FOR LOADING EMERGENCY MEMORY DUMP, 3-33
 " CONTROL, 3-15
 CONSOLE CALL CARD FOR LOADING MEMORY DUMP CONTROL, 3-19
 " CONTROL CONSOLE CALL CARD PLACEMENT, 3-20
 EMERGENCY MEMORY DUMP, 3-30, 4-2
 EMERGENCY TERMINAL MEMORY DUMP, 4-3
 FORMAT OF A MEMORY DUMP, 3-21

MODE
 ADDRESSING MODE, 2-1
 ADDRESSING MODES, 3-15, 3-21, 3-22, 3-32
 " UTILITY PROGRAM NAMES,
 THREE- AND FOUR-CHARACTER ADDRESSING MODE UTILITY PROGRAM NAMES, 2-1

MOUNTING PROCEDURES (CONT.)

COMPUTER-GENERATED INDEX

MOUNTING PROCEDURES
 TAPE MOUNTING PROCEDURES, 4-1

NAMES
 FOUR-CHARACTER ADDRESSING MODE UTILITY PROGRAM
 NAMES, 2-1

NUMBER
 TEST DIRECTOR DECK NUMBER ONE, A-2, A-3
 TEST DIRECTOR DECK NUMBER THREE, A-3, A-6
 TEST DIRECTOR DECK NUMBER TWO, A-3, A-5

OBJECT PROGRAM
 " CONSOLE CALL CARD, 2-5
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM TO
 BE PATCHED, 3-25

OCTAL CORRECTION CARD, 3-25, 3-26

OPERATING
 " CONSIDERATIONS, 3-1, 3-4, 3-13, 3-20, 3-21, 3-26,
 3-29, 3-33
 " PROCEDURES FOR PROGRAM TEST SYSTEM C, 4-1

OPTIONAL EQUIPMENT, 1-2

PARAMETER
 " CARD PROCESSING, 3-5
 " CARDS, 3-6
 " ENTRY CARD,
 PARAMETER ENTRY CARDS, 3-27
 TAPE DUMP PARAMETER ENTRY CARD, 3-27
 " ENTRY CARD DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD
 DECK, 3-27
 " FIELDS USED IN CONJUNCTION WITH THE FOUR FUNCTION
 CODES, 3-28

PARAMETERS, 3-32

PATCH, 3-22
 DECK ARRANGEMENT FOR USING PATCH, 3-26
 FUNCTION OF PATCH, 3-22
 LOADING PATCH,
 CONSOLE CALL CARD FOR LOADING PATCH, 3-23

PATCHED
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM TO
 BE PATCHED, 3-25

PLACEMENT
 MEMORY DUMP CONTROL CONSOLE CALL CARD PLACEMENT,
 3-20

POST CARD, 3-12
 " FORMAT, 3-12

PREPARATION
 PROGRAM DESCRIPTION AND PREPARATION, 2-1

PRINT ERROR, 3-4

PROCEDURE
 BOOTSTRAP PROCEDURES, 4-1
 CONSOLE CALL CARD PROCEDURE, 4-1
 INITIAL SETUP PROCEDURES, 4-1
 OPERATING PROCEDURES FOR PROGRAM TEST SYSTEM C, 4-1
 RECOVERY PROCEDURES, 4-2
 TAPE MOUNTING PROCEDURES, 4-1

PROCESSING
 PARAMETER CARD PROCESSING, 3-5
 " WITH THE PROGRAM TEST SYSTEM, A-3

PROGRAM
 ADDITIONAL UTILITY PROGRAMS,
 PROVISION FOR ADDITIONAL UTILITY PROGRAMS, B-1
 " CONSOLE CALL CARD,
 OBJECT PROGRAM CONSOLE CALL CARD, 2-5
 " DESCRIPTION AND PREPARATION, 2-1
 " HALTS, 3-4
 " NAMES,
 THREE- AND FOUR-CHARACTER ADDRESSING MODE
 UTILITY PROGRAM NAMES, 2-1

OBJECT PROGRAM,
 CONSOLE CALL CARD FOR LOADING AN OBJECT PROGRAM
 TO BE PATCHED, 3-25
 PROGRAMS OF PROGRAM TEST SYSTEM C, 4-2
 SYSTEM PROGRAMS, 3-1

" TEST SYSTEM,
 INITIALIZATION OF THE PROGRAM TEST SYSTEM, A-2
 OPERATING PROCEDURES FOR PROGRAM TEST SYSTEM C,
 4-1
 PROCESSING WITH THE PROGRAM TEST SYSTEM, A-3
 PROGRAMS OF PROGRAM TEST SYSTEM C, 4-2
 SAMPLE RUN OF PROGRAM TEST SYSTEM C, A-1

" TEST SYSTEM BRT, 2-2
 " TEST SYSTEM CONSOLE CALL CARD, 2-5
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD,
 2-5

PROGRAMMED INSTRUCTION DUMPS, 3-16

PROVISION FOR ADDITIONAL UTILITY PROGRAMS, B-1

READ ERROR
 CARD READ ERROR, 3-4, 3-15

RECORDS
 TRAILER RECORDS, 3-7

RECOVERY PROCEDURES, 4-2

REQUIREMENTS
 EQUIPMENT REQUIREMENTS, 1-1

RUN
 SAMPLE RUN OF PROGRAM TEST SYSTEM C, A-1

SAMPLE
 " CODING ENTRIES FOR HEADER AND TRAILER RECORDS, 3-7
 " FORMAT FOR A COMMENT CARD, 3-3
 " RUN OF PROGRAM TEST SYSTEM C, A-1
 " TAPE DUMP LISTING, 3-31
 " TEST DIRECTOR DECKS, 2-3
 " TESTING DECKS, A-1

SETUP PROCEDURES
 INITIAL SETUP PROCEDURES, 4-1

SPACING CODES
 LIST COMMENTS LINE SPACING CODES, 3-3

SYSTEM
 " BRT, 2-2
 PROGRAM TEST SYSTEM BRT, 2-2
 " CONSOLE CALL CARD,
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD,
 2-5
 PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5
 PROGRAM TEST SYSTEM,
 INITIALIZATION OF THE PROGRAM TEST SYSTEM, A-2
 OPERATING PROCEDURES FOR PROGRAM TEST SYSTEM C,
 4-1
 PROCESSING WITH THE PROGRAM TEST SYSTEM, A-3
 PROGRAMS OF PROGRAM TEST SYSTEM C, 4-2
 SAMPLE RUN OF PROGRAM TEST SYSTEM C, A-1

" PROGRAMS, 3-1

S2
 CONSOLE CALL CARD (USED BY S2), 3-24

TAPE
 " MOUNTING PROCEDURES, 4-1
 " WRITE ERROR, 3-15

TAPE DUMP, 3-27
 CODED HALTS FOR TAPE DUMP, 3-29
 CONSOLE CALL CARD FOR LOADING TAPE DUMP, 3-29
 " LISTING,
 FORMAT OF A TAPE DUMP LISTING, 3-30
 SAMPLE TAPE DUMP LISTING, 3-31
 " PARAMETER ENTRY CARD, 3-27
 " PARAMETER ENTRY CARD DECK,
 LAST CARD OF TAPE DUMP PARAMETER ENTRY CARD
 DECK, 3-27

TECHNICAL NOTE, 3-34

TERMINAL MEMORY DUMP
 EMERGENCY TERMINAL MEMORY DUMP, 4-3

TERMINATION CARD, 3-26

TEST
 " DATA GENERATOR, 3-5
 CONSOLE CALL CARD FOR LOADING TEST DATA
 GENERATOR, 3-13
 " DIRECTOR DECK, 2-3
 SAMPLE TEST DIRECTOR DECKS, 2-3
 " DIRECTOR DECK NUMBER ONE, A-2
 TEST DIRECTOR DECK NUMBER ONE, A-3
 TEST DIRECTOR DECK NUMBER THREE, A-3, A-6
 TEST DIRECTOR DECK NUMBER TWO, A-3, A-5
 " PROGRAM TEST SYSTEM CONSOLE CALL CARD,
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD,
 2-5
 " SYSTEM,
 INITIALIZATION OF THE PROGRAM TEST SYSTEM, A-2
 OPERATING PROCEDURES FOR PROGRAM TEST SYSTEM C,
 4-1
 PROCESSING WITH THE PROGRAM TEST SYSTEM, A-3
 PROGRAMS OF PROGRAM TEST SYSTEM C, 4-2
 SAMPLE RUN OF PROGRAM TEST SYSTEM C, A-1

" SYSTEM BRT,
 PROGRAM TEST SYSTEM BRT, 2-2
 " SYSTEM CONSOLE CALL CARD,
 EASY TEST PROGRAM TEST SYSTEM CONSOLE CALL CARD,
 2-5
 PROGRAM TEST SYSTEM CONSOLE CALL CARD, 2-5

TESTING DECKS
 SAMPLE TESTING DECKS, A-1

THREE- AND FOUR-CHARACTER ADDRESSING MODE UTILITY
 PROGRAM NAMES, 2-1

TRAILER RECORDS
 SAMPLE CODING ENTRIES FOR HEADER AND TRAILER
 RECORDS, 3-7

TRAPPING
 ITEM MARK TRAPPING, 3-17
 UNIDENTIFIED CARD ERROR, 3-15

UTILITY
 " PROGRAM NAMES,
 THREE- AND FOUR-CHARACTER ADDRESSING MODE
 UTILITY PROGRAM NAMES, 2-1
 " PROGRAMS,
 PROVISION FOR ADDITIONAL UTILITY PROGRAMS, B-1

WRITE ERROR
 TAPE WRITE ERROR, 3-15

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200 PROGRAM TEST SYSTEM C
SOFTWARE MANUAL

DATED: DECEMBER 15, 1965
FILE NO: 122.2005.001C.0-049

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME _____

DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

Cut Along Line

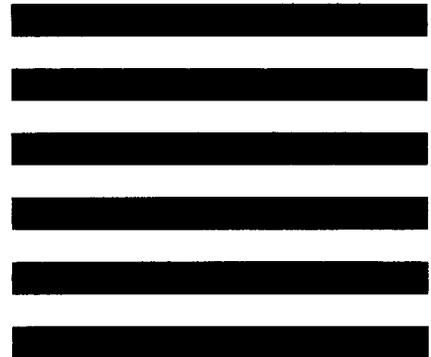
BUSINESS REPLY MAIL

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

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.



Cut Line

Honeywell
ELECTRONIC DATA PROCESSING