

## Systems Reference Library

### **Autotest for IBM 1401/1460 with IBM 1311: Specifications and Operating Procedures**

*Program Number 1401-AT-081 (8K)  
1401-AT-082 (16K)*

This publication describes the specifications and operating procedures for the IBM 1401/1311 *Autotest 8K* and the 1401/1311 *Autotest 16K* programs. Included are sections that define the control cards required, specify the arrangement of cards in the test deck, specify the autpatch card format and autpatching procedures, and describe the operating procedures for testing 1401/1311 Autocoder object programs.

Third Edition, June 1966

This is a reprint of C24-3195-1 incorporating changes released in the following Technical Newsletter:

<u>Form No.</u>	<u>Pages Affected</u>	<u>Date</u>
N24-0272	7, 8, 15, 16, 16.1, 16.2, 17, 18, 18.1, 18.2, 21, 22, 23, 25, and 26	October 15, 1964

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Dept. 425, Rochester, Minn. 55901.

## Contents

<b>Specifications</b> .....	5
Machine Requirements .....	5
Autotest Specifications .....	6
Autotest Standard Features .....	6
Identification Card (IC) .....	7
Console Status Card .....	7
Program Control Card (PCC) .....	7
Autotest Optional Features .....	8
Autopatch Feature .....	21
80/80 Listing of Output Cards .....	26
<b>Operating Procedures</b> .....	27
Program Deck Description .....	27
Autotest Setup Instructions .....	27
System Preparation .....	28
Autopatch Error Messages .....	30
Halts and Messages During Disk Utility Programs .....	30
Restarts .....	34
Autotest Halts and/or Messages .....	34
<b>Index</b> .....	40



Autotest, a testing program for the IBM 1401 and 1460 Data Processing Systems, controls the use of utility programs to provide complete documentation for program evaluation. It is useful for remote testing because it provides the programmer with an opportunity to pre-plan his test run with a minimum of operator intervention.

For a better understanding of the IBM 1401/1311 Disk Utility Programs used in Autotest, the reader should review the Systems Reference Library (SRL) publications, *Disk Utility Programs for IBM 1401/1311: Specifications*, Form C24-1484 and *Disk Utility Programs for IBM 1401/1311: Operating Procedures*, Form C24-3105.

There are two versions of the Autotest program available to the user, 8K and 16K, depending on the minimum size of the object machine used. Both versions perform exactly the same functions and are designed so the core-storage area occupied by Autotest during the running of the object program occurs in the upper core-storage positions of the object machine (8K or 16K) used. With the exception of some core-storage considerations, the specifications and operating procedures for the two versions of Autotest are identical. Where differences in the versions occur in this publication, the address for 16K Autotest appears first, without brackets, and the address for 8K Autotest follows, included within brackets. Example: 15909 [7909].

The Autotest program:

- Clears core storage before loading each object program.
- Loads the object program.
- Executes the object program.

Also, Autotest provides the following program features, which may be selected by the user.

1. Clear Disk Storage — Selected areas of disk storage can be cleared before or after execution of the object program.
2. Disk Record Load — Provides for the creation of disk records from card input prior to the test of each individual program.
3. Print Disk — Data from selected areas of disk storage can be printed before or after execution of the object program.

4. Disk-Label Program — Performs all necessary maintenance operations on the label track of a disk pack.
5. Tape-File Generator — Provides for the creation of tape files prior to the test of each individual program.
6. Tape-to-Printer — Prints fixed- or variable-length, blocked or unblocked tape records.
7. Snapshot — Data from a selected area of core storage can be printed at specified times during the execution of the object program.
8. Disk Trace — Data that is read from, or written on, the disk during execution of the object program is traced and printed with the control field of the read or write instruction.
9. Core-Storage Printout — The contents of core storage can be printed at the end-of-job or when a *hang-up* condition occurs.
10. Autopatch — Allows the programmer to correct the object program without reassembling or computing 3-position machine addresses. Patch instructions can have indexed operands.
11. 80/80 List — An 80/80 list of all punched-card input and output can be obtained at the end of the complete test session.

Any number of these optional features can be used when testing an object program.

The IBM 1401/1311 Autotest program requires that the assembled program be in the same format as the condensed Autocoder card output.

### Machine Requirements

The Autotest program can test programs written for all IBM 1401/1460 systems. The machine used to test the object programs must have the following configuration.

IBM 1401 Systems:

- 16,000 [8,000] positions of core storage
- An IBM 1311 Disk Storage Drive
- An IBM Disk Pack
- An IBM 1402 Card Read-Punch
- An IBM 1403 Printer, Model 2, or an IBM 1404 Printer
- Advanced Programming feature
- High-Low-Equal-Compare feature

## IBM 1460 Systems:

- 16,000 [8,000] positions of core storage
- An IBM 1311 Disk Storage Drive
- An IBM Disk Pack
- An IBM 1402 Card Read-Punch
- An IBM 1403 Printer, Model 2 or 3
- Indexing-and-Store-Address-Register feature

## Autotest Specifications

### Program Requirements

1. If control cards occur after the last data card in a test deck, any last-card test must be changed to a BRANCH IF CHARACTER EQUAL instruction. This can be done with a replace-type autopatch card. (See *Autopatch Feature*.) The last-card indication character is punched in the last data card of the program to be tested. If the program does not use cards as input, this Autotest requirement can be ignored.
2. A snapshot routine can occur at any four-or-more character instruction whose operands are not address-modified. It must not be a chained instruction. When it encounters the first snapshot, autotest automatically stores this instruction and then replaces it with an unconditional branch to the snapshot routine. After snapshot is executed, the original instruction is executed and the program continues. Additional snapshots are accomplished by using autopatch cards to cause a branch to the snapshot routine located at 14907 [6907].
3. The segment-mark special character (0-7-8) should not be used in the object program if autopatch is to be used. Autopatch uses this special character internally as a group-mark with a word-mark replacement character when loading the object program into disk storage. If the segment-mark special character (0-7-8) must be used, the programmer has the option of supplying another replacement character to the Autotest program. It is punched in the program control card. If this other character is specified, it must be excluded from the object program.
4. To decrease the time required for testing, the final halt (if a four-or-more character instruction) should be changed to an unconditional branch (not a halt and branch) to an Autotest linkage routine. This can be done by using autopatch. If a final halt in the user's program is less than a 4-character instruction, the console operator must, after the program reaches this end-of-job halt, manually adjust the console to branch to the linkage routine. The address of the Autotest linkage routine is 15909 (IOI) [7909 (IOZ)].

5. Object programs using the punch-feed-read special feature cannot be tested by Autotest.
6. If the disk-trace program that does not require a tape drive is used, disk read or write instructions located in positions 200-332 cannot be traced and/or any data read or written by disk instructions, and located in positions 200-332, cannot be displayed.

### Autotest Standard Features

For every object program to be tested, Autotest automatically generates instructions to clear core storage, load the object program, and execute the object program. If the programmer desires, it also prints the contents of core storage after execution or when the machine stops because the program cannot be executed.

#### Clear Core Storage and Load Routine

Because Autotest automatically clears core storage and then loads the object program, the programmer should remove the loader (clear core storage and load routine) from the deck produced by the Autocoder processor. However, if these cards are not removed, Autotest automatically bypasses them.

After loading, Autotest executes the object program.

#### Core-Storage Printout

Autotest prints the contents of core storage at the conclusion of the object program unless the programmer specifies otherwise on the program control card. The last instruction in the program to be tested branches to the Autotest linkage routine, which tests whether or not the core-storage printout program is to be executed. If it is to be executed, the routine loads the core-storage printout program from disk to core storage, and executes it immediately after loading.

The programmer can also specify that the contents of core storage are to be printed if the object program *hangs up*. After the machine halts, the operator can make the proper adjustments on the console to give control to the Autotest linkage routine in core storage. This routine loads the core-storage printout program from disk storage into core storage and executes it. If the Autotest linkage routine is destroyed by execution of the object program, a restart deck that reestablishes linkage is supplied by IBM.

In both cases, after execution of the core-storage printout program, Autotest continues reading cards until it encounters a clear-disk-storage (CDS) or print-disk (PRD) utility call card, a halt (HLT) card, a tape-print (T/P) call card, a disk-label-program utility call card, or the identification (IDC) card for the next object program.

UNPRINTABLE CHARACTERS	SUBSTITUTION CHARACTERS
( (12-5-8)	\$ (11-3-8)
< (12-6-8)	* (11-4-8)
# (12-7-8)	□ (12-4-8)
) (11-5-8)	@ (4-8)
; (11-6-8)	Y (0-8)
△ (11-7-8)	# (3-8)
∨ (0-5-8)	V (0-5)
## (0-7-8)	/ (0-1)
> (6-8)	X (0-7)
√ (7-8)	% (0-4-8)
\ (0-6-8)	
: (5-8)	. (12-3-8)

Figure 1. Substitution Characters for Core-Storage Printout and Snapshot Programs

1. The core-storage printout program prints the contents of core storage available to the user. It prints from position 001 to position 15500 [7500], in lines of 100 characters, with every ten characters followed by a blank. Each line of 100 characters is preceded by a key line that makes interpretation of the data easier. Blocks of 100 positions that are completely blank are bypassed as far as printing is concerned. The program always prints from position 101 to position 199 even if they are blank. Position 200 is never printed.
2. All unprintable characters are replaced with standard printable characters (Figure 1). *Example:* Group marks (12-7-8) are replaced with the printable-character lozenge □ (12-4-8).
3. All word marks are indicated by 1's printed on the next line directly under the characters associated with the word mark.
4. For storage considerations, positions 201-332 and positions 14601-15500 [6601-7500] of core storage are printed first, with word marks but without character substitution. However, group marks (12-7-8) are replaced with a printable □ (12-4-8).
5. Each line is labeled on the right-hand side with the appropriate 4- or 5-character address, which designates the lowest core-storage location of that line. The word AREA is printed after the address for easy recognition of the address by the programmer (that is, XXX01-AREA).

### Identification Card (IDC)

This must be the first card in the Autotest deck. Punch IDC in columns 1-3 and program identification data in any of the remaining columns.

Any number of additional IDC cards containing comments can accompany an object program. These must be placed directly behind the initial IDC card, at the beginning of the test deck.

If any input data cards for a program are to be stacked in stacker 1, an IDC card with 1 instead of C punched in column 3 must follow the IDC cards and precede any of the OBJ, SNF, and/or DTR cards.

### Stacker-Identification Cards

Stacker-identification cards are used to identify the punched-card output of the user's program. Autotest punches one or more stacker ID cards and directs the cards into the correct punch stacker for each program to be tested. Columns 1 and 2 of each stacker-identification card contain ID and column 3 contains a punch-stacker identification as follows:

Column 3	Stacker
8	8/2 stacker
4	4 stacker
N	Normal punch

Columns 4-80 of the IDC card for the program being tested are reproduced in columns 4-80 of each stacker-identification card.

### Console-Status Card

A console-status card must follow the last IDC card. The card is used to record the status of the console if a program process error occurs. It is directed into read stacker 1 and is identified by CCC in columns 1-3. Columns 4-80 contain the information punched in columns 4-80 of the last IDC card.

*Note:* Use of the console-status card is optional.

### Program Control Card (PCC)

A program control card must be used with every object deck to be tested or Autotest assumes the following:

1. The object program requires no sense-switch control.
2. Condition of the write-address key-light (ON or OFF) is immaterial during the running of the object program.
3. There are sufficient unused upper core-storage positions for loading the snapshot and/or disk-trace programs if called for.

If part of an object program is stored in these upper positions of core storage, it could overlay the snapshot and/or disk-trace program. Autotest does not warn the programmer of insufficient core storage before loading the object program unless there is a program control card.

4. Autopatch is not to be used.
5. The core-storage printout program is executed automatically if an end-of-job condition is reached. If the program cannot be executed and the operator manually restarts the machine at the Autotest linkage routine, the core-storage printout program is

also executed. In both cases, after the printout is completed, Autotest continues to read cards, searching for a print-disk (PRD) utility call card, a tape-print (T/P) call card, a disk-label-program utility call card, a clear-disk-storage (CDS) utility call card, a halt (HLT) card, or the next IDC card.

6. Tape drives are not to be used for testing.

When a program control card is used, it is placed immediately following the IDC cards when the console-status card is not used. Otherwise, it follows the console-status card. It is punched in the following format:

Columns	Contents
1-3	PCC
4-9	Punch from one to six of the alphabetic characters, B, C, D, E, F, G, in any order, to indicate which sense switches must be on during the running of the object program.
10-14	Punch the 5-character address of the highest core-storage position required by the object program. This address is used to test for sufficient core-storage positions for the snapshot, disk-trace, and autopatch routines. If a snapshot (SNP) or disk-trace (DTR) control card is subsequently encountered and the core-storage area is insufficient, Autotest prints a message and does not load or execute the routines. If this field is left blank, Autotest assumes there is sufficient unused upper core storage for the snapshot and/or disk-trace program. Because autopatch stores patch instructions beginning above the upper core-storage limit specified in these columns, a program control card is required if autopatch is to be used. If a program control card is not present, Autotest causes any autopatch cards to be bypassed.
15	Specifies write-address key-light requirement for object program. <ol style="list-style-type: none"> <li>1. Punch A if the write-address key-light is to be initially on and remains on during the execution of the entire object program.</li> <li>2. Punch S if the write-address key-light is to be initially off and remains off during the execution of the entire object program.</li> <li>3. Punch I if the write-address key-light is to be initially on, but the setting may be changed during execution of the object program.</li> <li>4. Punch 2 if the write-address key-light is to be initially off, but the setting may be changed during the execution of the object program.</li> <li>5. Leave field blank if setting of the write-address key-light is immaterial for running of the object program (no disk write instructions).</li> </ol> <p><i>Note:</i> Autotest will halt and print a message before running the object program if the write-address key-light setting must be changed before object program execution. However, when combining modes during object program execution, the user must program halts and messages when the write-address key setting is to be changed.</p>
16	A special character is punched if both of the following conditions hold:

Columns

Contents

1. Autopatch is used.
2. The segment-mark special character (0-7-8) is used in the object program.

If a special character is punched, it must be excluded from the object program (see *Autotest Specifications — Replacement Character*). If one or both of these conditions are not met, the field is left blank.

17

Punch N if a core-storage printout is not desired, either when an end-of-job condition is reached or when the program execution cannot be completed. When the last instruction (an unconditional branch to the Autotest linkage routine) is encountered or when the machine is manually restarted at the linkage routine (15909 [7909]), the core-storage printout program is bypassed and autotest continues to read cards, searching for a halt (HLT) card, a print-disk (PRD) utility call card, a tape-print (T/P) call card, a disk-label-program utility call card, a clear-disk-storage (CDS) utility call card, or the next IDC card.  
Leave the field blank if the core-storage printout program is to be executed.

*Note:* If the program hangs up (execution cannot be completed) and a manual restart at the Autotest linkage routine is unsuccessful (the linkage routine has been erroneously destroyed), linkage is re-established with a restart deck, deck C. (See *Core-Storage Printout*.) In this case, a core-storage printout is provided regardless of the contents of column 17 of the PCC.

18-23

Punch the numbers of the tape drives to be used during object program execution. Each of these columns can contain any number from 1-6. The tape drives do not have to be specified in any particular order and the punching can begin in any one of the 6 columns. Use these columns to specify the numbers of the drives on which are mounted: the object program tapes, any tapes used by the tape utility programs, the disk-trace tape, and any precreated tape specified in a HLT card. If tapes are used during object program execution, a PCC card with these columns correctly punched must be provided.

**Autotest Optional Features**

In addition to the standard functions described in the previous section, the Autotest program provides optional features, that can be selected by the programmer to obtain more complete documentation of all object programs. There are two types of optional features, or routines, used by Autotest:

1. *Disk Utility and Tape Utility Programs* for IBM 1401/1311, which are supplied by IBM for use with Autotest or with other programs:

- Clear Disk Storage
- Disk Record Load
- Print Disk
- Disk Label
- Tape-File Generator
- Tape-to-Printer



To use one or more of these utility programs in testing with Autotest, the programmer must generally place a utility call card before the corresponding control cards and a utility end card after them. These cards identify the particular utility within the Autotest program to which the control information applies.

2. *Autotest Features*, which can be used only with the Autotest program:
  - Snapshot feature
  - Disk Trace feature
  - Autopatch.

These features do not require special utility call cards. Control information is placed on the same card that identifies the particular feature. However, snapshot and disk-trace require separate execute cards, which are placed before the normal execute card in the object program or overlay segment to which the feature applies.

### Clear Disk Storage (CDS)

The clear-disk-storage program can be used before or after program execution if it is desirable to set certain areas to an initial condition. Punch CDS in columns 1-3 of a card and place behind it the necessary control cards described in this section.

To fill specified areas of disk storage with blanks or any desired IBM 1401/1460 character, place a CDS utility call card in the test deck after the IDC cards and before any of the snapshot (SNP), disk-trace (DTR), or object (OBJ) cards. The CDS card is placed after the data cards following the object program to restore to normal the disk-addressing scheme after modifying it during the execution of the object program.

Normally, before loading a file into disk storage, the file area is filled with blanks. In some cases, however, it is desirable to use 9's or some other character. When a track or record is subsequently read from the disk, unused locations are easily identified as fields containing only the clearing character.

It is usually desirable to restore the normal addresses to a disk pack if they have been modified by the object program. Because the modified addressing scheme must be known to restore the normal addresses, it is important that the programmer use the CDS utility program after program execution when he has modified disk addresses.

An area can be cleared in either the move or load mode, and in either 20-sector or track-record format. The write-address key-light must be ON for execution of the CDS program. It is important that an area be cleared in the mode and format that will be used subsequently to read from that area.

### CDS Addresses

The clear-disk-storage program writes the sector addresses in the address portions of the cleared areas. The user has three options from which to choose in writing the addresses.

1. Sequential addresses can be written, beginning with the address given as the lower limit of the area, and continuing through the track containing the upper limit. If the track-record format is used, the addresses written are computed from the upper-limit indication. If the upper limit is given as the zero sector of the track, the addresses written are the zero-sector addresses of each track. If the upper limit is given as the 16th sector of the track, the 16th-sector address of each track is written.

For example, if the limits on the control card are given as 000000-000335, the address written for the first track is 000015, the second track is 000035, the third track is 000055, and so forth. Neither the mode nor the format used need be that in which the area was previously written.

2. The user has the option of writing a set of sequential addresses that are compatible with the addressing structure on other disk drives in the system. For example, he may clear the area limited by addresses 000000-000219, which refer to the first disk drive in the system, and write sequential addresses 020000-020219, which refer to the second disk drive. In this case, the lower limit specified on the control card (see *CDS Control Cards*) is 020000; the upper limit is 020219. The reference number on the card is 0 because the first disk-drive addresses are presently on the pack.

Again, if track-record format is used, the address written for each track is computed from the upper-limit indication. Using track-record format in the preceding example instead of 20-sector format, the address written for the first track is 020019, the second track is 020039, and so forth. Neither the mode nor the format used need be that in which the area was previously written.

3. The same addresses as those originally in the area can be rewritten. If this option is taken, the area must be cleared in the same mode (M or L) and the same format (20-sector or track-record) as that in which it was previously written.

### CDS Control Cards

Any number of control cards can follow a CDS utility call card in the test deck. From one to four disk areas can be specified on one control card. (A disk area can be from one track to an entire disk pack.)

The cards are punched in the following format:

Columns	Contents
1	<i>M</i> or <i>L</i> to indicate move or load mode of operation.
2-7	6-digit address of the lower limit of the disk-storage area to be cleared. If the addressing structure is to be modified (see <i>CDS Addresses</i> , step 2), the lower limit reflects the lower address desired on the pack after clearing. <i>Note:</i> The lower limit must be the zero sector of the first track, even if the track does not physically contain a zero sector.
8-13	6-digit address of the last sector on the last track to be cleared. (Although any one of the 20 valid addresses of the last track to be cleared can be used here, the address of the 20th sector provides the greatest time efficiency.) If the addressing structure is to be modified, the upper limit reflects the upper address desired on the pack after clearing. The upper and lower limit of a disk area must be on the same pack. <i>Note:</i> If track-record format is being used, for time efficiency the upper limit should reflect the one sector that is used as the track address.
14	0, 2, 4, 6, or 8 to indicate disk-storage drive containing the area to be cleared.
15	0, 2, 4, 6, or 8 to indicate disk-storage drive to which the addresses now on the pack refer. (If retaining same address, punch \$.)
16	The character to which the data portion of the track is to be cleared.
17	Track format to be used: <i>blank:</i> 20-sector <i>I:</i> track record
18-20	Blank — not used.
21-40	} Control information for up to three additional disk areas to be cleared is punched in the preceding format.
41-60	
61-80	

*Note:* Although the upper and lower limits for a disk area must be in the same pack, the disk areas specified on one control card can be on separate packs.

#### CDS End Card

Immediately following the control cards is a utility end card. Columns 1-4 of this card are punched ECDS.

#### Disk Record Load (DRL)

A disk-record-load utility call card follows the IDC card(s) and occurs before any of the SNP, DTR, or OBJ cards. It is used if the programmer wishes to load addresses, single records, or entire tracks into IBM 1311 disk storage before program-execution time. Punch DRL in columns 1-3 of the card and place it ahead of the control-data cards described subsequently. The disk-record-load program reads data from the specified

track or sector into core storage, inserts the information from the cards into the desired locations, and writes the data on the disk.

In program testing, this option is often used to set up a sample file of records. It can also be used to modify existing disk records or to change addresses or data on a track.

If one record is placed in several locations on a track, a sector-scan indication in the control card causes the program to change all the sectors with the same address as that specified. If such a sector scan is required, or if an alteration is to be made to a disk address, the write-address key-light must be ON during execution of the DRL program. Otherwise, it must be OFF.

#### DRL Input Cards

The disk-record-load program does not require separate control and data cards. Instead, each input card contains both the data to be inserted and the control information. The first 20 columns of each card contain the required control information. Data to be inserted is punched in the remaining 60 columns.

Any number of input cards can follow a DRL utility call card in the test deck.

The cards are punched in the following format:

Columns	Contents
1	<i>M</i> or <i>L</i> to indicate move or load mode.
2-7	6-digit address of the track or sector to be operated upon.
8-11	4-digit location of the data within either the sector or track record, depending on which format is used (first position is 0001). If changing a disk address, leave columns 8-10 blank and punch \$ in column 11. See <i>Note</i> at the end of this section.
12-13	2-digit length of the data field (maximum is 60). If \$ in column 11, leave 12 and 13 blank.
14	0, 2, 4, 6, or 8 to indicate disk-storage drive on which the pack is mounted.
15	Blank — not used by Autotest.
16	Identical disk address option: <i>I:</i> specifies scan of entire track to alter all records with identical addresses (the write-address key-light must be on if a scan is specified). <i>blank:</i> if no identical addresses.
17	Word-mark option: <i>I:</i> specifies, if in the load mode, that a word mark is to accompany the first character of data. <i>blank:</i> if word mark is not desired or if in move mode.
18	Track format to be used: <i>blank:</i> 20-sector <i>I:</i> track record
19	Blank — not used by Autotest.

Columns	Contents
20	Blank — not used.
21-80	The data or address to be inserted on the disk pack.

*Note:* If changing an address, the new address is punched in columns 21-26 and must be a valid one for the track operated on. Columns 2-7 contain the address to be changed. Column 11 contains a \$. Columns 8-10, 12, and 13 are left blank. The write-address key-light must be ON when changing disk addresses.

#### DRL End Card

A utility end card must immediately follow the DRL control cards. Punch EDRL in columns 1-4 of this card.

#### Print Disk (PRD)

The print-disk program produces a printed copy of an area of disk storage. This program can be executed either before or after execution of the object program, or both. It can print tracks written in either the move mode or load mode and in either 20-sector or track-record format. If desired, special characters that are not printable with the standard 52-character set can be replaced with a printable alphameric or special character.

If a printed copy of a disk area is desired before object-program execution, place a print-disk utility call card in the test deck after the IDC cards and before any of the SNP, DTR, or OBJ cards. For a disk print after execution, place the print-disk card after the data following the object program. Punch PRD in columns 1-3 of the card and place it ahead of the control cards described subsequently.

#### PRD Control Cards

Any number of control cards can follow a PRD utility call card in the test deck. From one to four disk areas can be specified on one control card. (A disk area can be from one track to an entire disk pack.)

The cards are punched in the following format:

Columns	Contents
1	M or L to indicate move or load mode of operation.
2-7	6-digit address of the lower limit of the disk-storage area to be printed. <i>Note:</i> The lower limit must be the zero sector of the first track even if the track does not physically contain a zero sector.
8-13	6-digit address of the last track to be printed. Although any one of the 20 valid addresses of the last track can be used to obtain a disk-storage print through the 20th sector, the address of the 20th sector provides the greatest time efficiency. <i>Note:</i> If track-record format is being used, for time efficiency the upper limit should reflect the one sector that is used as the track address.
14	0, 2, 4, 6, or 8 to indicate disk-storage drive containing the area to be printed.

Columns	Contents
15	Character to be substituted for all special characters unprintable with the standard 48-character print chain. Leave blank if no character is to be substituted. This option allows faster processing.
16	Word-mark option: <i>blank:</i> specifies, if in the load mode, that word-mark indications are to be printed. <i>I:</i> specifies that word-mark printout is not desired in load mode.
17	Track format to be used: <i>blank:</i> 20 sector <i>I:</i> track record
18-20	Blank — not used
21-40	} Control information for up to three additional disk areas to be printed is punched in the preceding format.
41-60	
61-80	

*Note:* Although the upper and lower limits for a disk area must be in the same pack, the disk areas specified on one control card can be on separate packs.

#### PRD End Card

Immediately following the PRD control cards must be a utility end card. Columns 1-4 of this card are punched EPRD.

#### PRD Print Format

An index, which aids in determining character positions, is printed at the top and bottom of each page. In the 20-sector format, the program prints one sector per line. Each line contains an M or L to indicate the mode used to write the sector, followed by the sector address and the record.

In the track-record format, the first line of each track begins with an M or L and the address. The record is printed beginning after the address. If a new page is started during the printing of a track, the mode and address are reprinted with the first data line on the new page. Each line except the last contains 100 data characters.

When printing areas that are written in the load mode, the user has the option of printing an indication of word marks. A word mark is indicated by printing a one (1) under the character with which it is associated.

#### Disk-Label Program

The disk-label program performs all necessary maintenance operations on the label track of a disk pack. The program is used to:

1. Set up the initial header-label track on a disk pack.
2. Remove the entire label track by clearing it and restoring the addresses to the original range.
3. Enter a new label.
4. Delete an existing label.

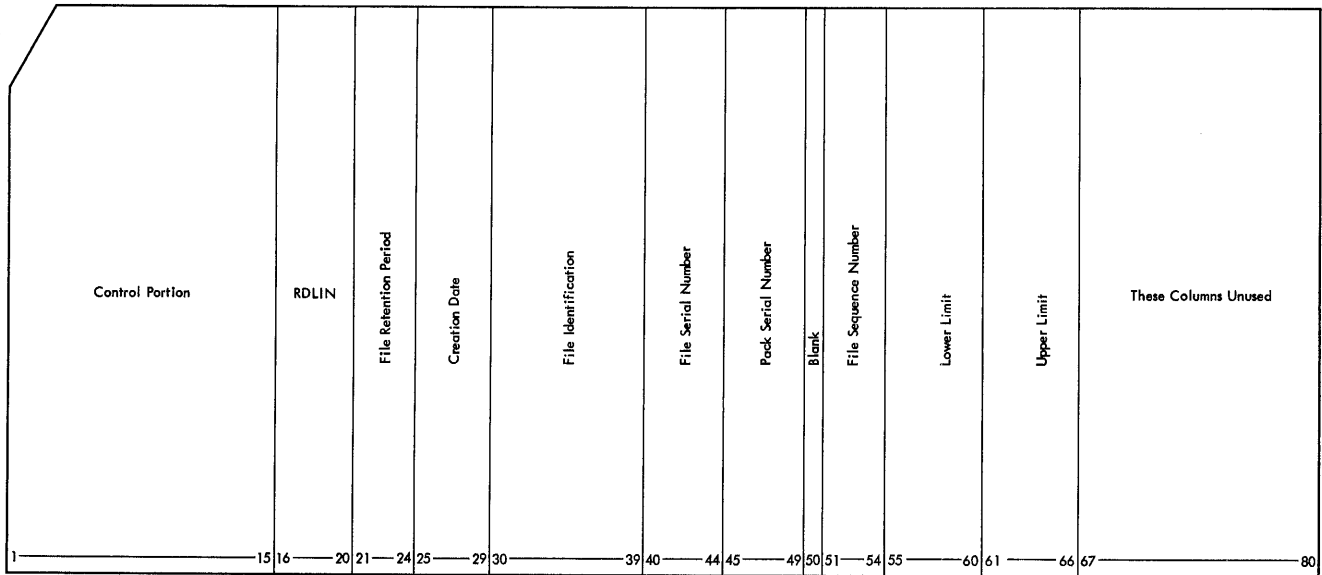


Figure 2. RDLIN Card

5. Make changes to an existing label.
6. Print labels.
7. Punch and print labels.

**RDLIN Cards**

RDLIN cards are used as input to this program. Normally, the standard RDLIN card shown in Figure 2 is used. When it is necessary to operate on fields not contained in the standard RDLIN card, an expanded

format is used (Figure 3). The expanded format consists of two cards. The first five columns of the first card are punched as in the standard RDLIN card. Columns 6-9 of the first card contain XPN1. Columns 1-5 of the second card are left blank, and columns 6-9 of the second card contain XPN2.

Columns 16-20 of both cards contain RDLIN.

The header label fields are punched, as shown in Figure 3, in the same order as they appear in the 100-character disk label.

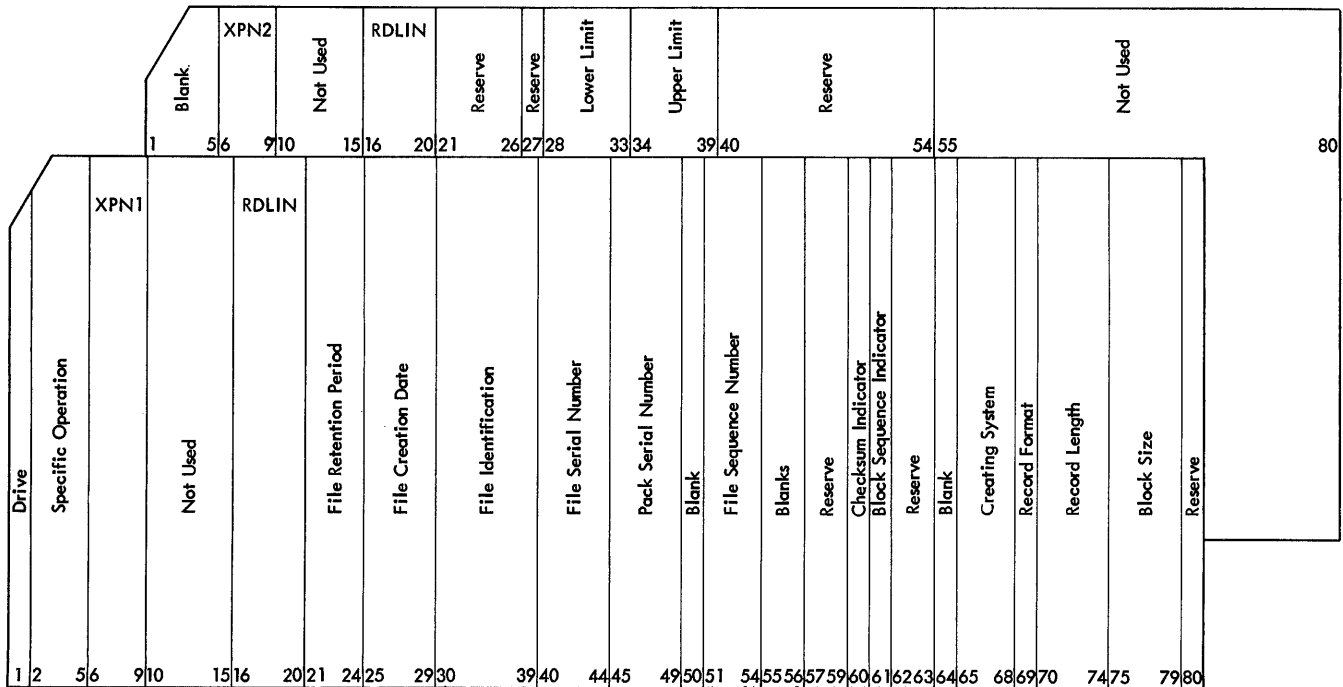


Figure 3. Expanded RDLIN Card Format

### Date Card

When the Autotest pack is built, the label track of the pack to be built is checked for unexpired files and a label is generated which protects the area of the pack used for Autotest routines. This label has today's date as a creation date, and a retention period of one day.

A date card must be provided with the required RDLIN cards when a label utility is being executed, when building the pack with deck A, and when checking the header label with deck B. This card, if required, must be the first RDLIN card of the label utility control cards. The placement of this card in deck A or deck B is explained in *Operating Procedures*.

Columns	Contents
1-5	08205
6	Word-separator character (0-5-8 punches)
7-11	Today's Date. Punch the year in columns 7, 8 and the day of the year in columns 9-11.
16-20	RDLIN

### Set Up Initial Header-Label Track (NEW)

The disk pack on which the initial header-label track is to be set up may be previously labeled, contain active labels, or be unlabeled. If the operator does not know the condition of the label track he should use the NEW operation to remove all active labels.

When setting up the initial header-label track, the program performs the following operations:

1. Seeks address 0X9980 and changes the addresses on the label track from 0X9980-0X9999 to 000180-000199. If address 0X9980 is not found, a label track with addresses 000180-000199 is assumed to be on the pack. All previous labels, active or non-active, are eliminated and a new label track is written.
2. Clears all 20 sectors to blanks.
3. Enters the label identifier (IHDRb) and the pack serial number in the first 19 sectors. Only the pack serial number is entered in the twentieth sector. The standard, single-card RDLIN card must be used for this operation. It is punched as follows:

Columns	Contents
1	Disk drive being used
2-4	NEW
5	Disk drive to which the addresses on the pack are referenced
6-15	Not used
16-20	RDLIN
21-44	Not used
45-49	Pack serial number
50-80	Not used

The date card is not used with this operation.

### Restore Normal Addresses (RNA)

This operation clears the entire label track to blanks and restores the addresses to the range: 0X9980-0X9999. The label track on drive 0 cannot be restored by using RNA.

The standard, single-card RDLIN card must be used for this operation. It is punched as follows:

Columns	Contents
1	Disk drive being used
2-4	RNA
5	Disk drive to which the addresses on the last track are to be referenced
6-9	Blank
10-15	Not used
16-20	RDLIN
21-80	Not used

The date card is not used with this operation.

### Enter a New Label (ENTR)

This operation is used to enter a complete new header label. The program first searches the label track to insure that there are no unexpired header labels that pertain to files within the limits given in the RDLIN card. Labels are checked for files within the same relative limits with addresses referenced to other disk drives. Any expired labels pertaining to this area are printed and then deleted. The program then enters the new label from the RDLIN card. The pack serial number entered in the NEW card is retained, regardless of whether it is punched in the RDLIN card.

Either the standard or the expanded RDLIN card format can be used. If the standard format is used, all fields not contained in the RDLIN card are blanked. The RDLIN card is punched as follows:

Columns	Contents
1	Disk drive being used
2-5	ENTR
6-9	Blank, XPN1, or XPN2
10-15	Not used
16-20	RDLIN

The rest of the fields in the RDLIN card, with the exception of the pack serial number (columns 45-49), are entered in the label. If any fields are left blank, blanks are written in the corresponding positions of the disk label. The lower and upper limits of the area in disk storage to which the label applies must be punched in the RDLIN card.

The date card is required for this operation.

### Delete Labels (DELT)

This operation is used to delete either all of the expired header labels pertaining to files within specified limits or to delete any specific label. If an unexpired label is to be deleted, it must be deleted by specific file identification. All labels deleted are printed.

The standard, single-card RDLIN card is used for this operation. The RDLIN card is punched as follows:

Columns	Contents
1	Disk drive being used
2-5	DELT
6-9	Blank
10-15	Not used
16-20	RDLIN
21-29	Not used
30-39	File identification, if a specific header label is to be deleted. Otherwise, these columns are blank.
40-54	Not used
55-66	Lower and upper limits of the areas to which the labels apply, if all labels within the limits are to be deleted. Otherwise, these columns are blank. If limits are to be used, columns 30-39 must be blank.
67-80	Not used

The date card must be supplied for this operation if all labels within specified limits are to be deleted.

### Change a Label (CHNG)

This operation is used to make changes to individual fields within a header label. The program finds and prints the header label specified in the file-identification field of the RDLIN card. All non-blank fields in the RDLIN card are then substituted for the corresponding fields in the header label. The changed label is then printed and written back onto the label track.

Either the standard or the expanded RDLIN card format can be used. The card is punched as follows:

Columns	Contents
1	Disk drive being used
2-5	CHNG
6-9	Blank, XPN1 or XPN2
10-15	Not used
16-20	RDLIN
30-39	File identification (must be punched)

All other fields are entered in the corresponding fields of the label if punched.

The date card is not used with this operation.

### Print Labels (PRNT)

This operation prints either all of the header labels on a disk pack or only specifically requested labels. The standard, single-card RDLIN card is used with this operation. The RDLIN card is punched as follows:

Columns	Contents
1	Disk drive being used
2-5	PRNT
6-9	Blank
10-15	Not used
16-20	RDLIN
21-23	ALL, if all header labels are to be printed.
24-29	Not used
30-39	File identification, if specific label is to be printed
40-80	Not used

The date card is not used with this operation.

### Punch and Print Labels (PNCH)

This operation punches and prints either all of the header labels on a disk pack or only specifically requested labels. The labels are punched in either standard or expanded RDLIN card format. The standard, single-card RDLIN card is used with this operation. The RDLIN card is punched as follows:

Columns	Contents
1	Disk drive being used
2-5	PNCH, if labels are to be punched in the standard RDLIN card format PNCX, if labels are to be punched in the expanded RDLIN card format
6-9	Blank
10-15	Not used
16-20	RDLIN
21-23	ALL, if all header labels are to be punched and printed
24-29	Not used
30-39	File identification, if specific label is to be punched and printed
40-80	Not used

The date card is not used by this operation.

### Tape-File Generation — Variable-Length Records

The Autotest tape-file generator provides for the creation of tape files, consisting of fixed-length or variable-length records, prior to the test of each individual program. This section explains the tape-file generator for variable-length records. The tape files can be created with header and trailer labels. Data tape records in variable-length format can be created without word marks in blocked or unblocked format. The

maximum record length that can be created is 13,000 [5,000]. Control cards and data cards are required to generate tape files.

### Control Cards

There are two types of control cards for variable-length records:

1. The `TFV` control card calls in the object program and prepares for file generation on the new tape unit. It is punched in the following format:

<i>Columns</i>	<i>Contents</i>
1-3	<code>TFV</code>
4	Tape unit number on which the file is to be created (1-6).
5	Parity. A <i>B</i> indicates odd parity. Any other punch indicates even parity.
6	Blank
7-80	Tape-file description

2. The `ETFV` control card signifies the end of the card-to-tape program. The `ETFV` card must be included. It is punched in the following format:

<i>Columns</i>	<i>Contents</i>
1-4	<code>ETFV</code>
5-6	Blank
7-80	Tape-file description

### Data Cards

The data cards are punched in the following format:

<i>Columns</i>	<i>Contents</i>
1-4	May contain numeric information for sequencing, but this information will not be checked.
5-6	<code>WT</code> — Write tape record <code>RM</code> — Record mark <code>TM</code> — Write tape mark Blank — Record length exceeds this card capacity.
7-10	Blank
11-80	Data to be created on the tape

These cards are used to create the tape file specified by the `TFV` control card. The data to be created on tape is punched in columns 11-80 in the tape image format. Multiple data cards are used to create larger tape records. Each new record must start in column 11 of a new data card. The end of each record is specified by punching an 11-8-4 (asterisk) one position to the right of the last data character. The use of the asterisk in the control card does not prevent its use within the record as a character or part of the tape record to be created. `WT` (write tape) must also be punched in

columns 5-6 of the last data card for each record. This causes the record to be written on the tape unit specified in the `TFV` control card.

A tape mark may be generated on the tape by punching `TM` (tape mark) in columns 5-6 and leaving the rest of the card blank. For example, if a tape mark is desired between the data file and a trailer label, a `TM` card would be inserted before the first data card of the trailer label.

Multifile tapes can be created by arrangement of the control and data cards as shown in the following example, which would generate three files on one tape:

```

TFV card
Data card for file 1
TM card
Data cards for file 2
TM card
Data cards for file 3
ETFV card

```

When generating blocked records that contain record marks (0-8-2) as record separators, the user can begin each individual record in the block on a new data card starting in column 11, rather than punch the complete block continuously. Punching `RM` (record mark) in columns 5-6 indicates that the record mark used as the record separator is the last character contained in this card. All card columns to the right of the last record mark are ignored and only the record mark and the characters to the left are used to build the block. The next individual record of the block must begin in column 11 of the next data card. The last card of the block, as in single records, must have the `WT` in columns 5-6 and an 11-8-4 (asterisk) punch following the last character of the block.

### Tape-File Generation — Fixed-Length Records

The Autotest tape file generator can also create files that consist only of fixed-length records. The program transfers data from punched cards to magnetic tape. Although the basic data records, when processed from tape, may remain the same as they were when in cards, it is often desirable to modify the format of the records. Fields that were required in the card file may not be required on tape. The information in master and detail cards may be rearranged and consolidated in the tape records. Exception procedures and field-selection ability are provided in this program to facilitate such modification.

For a better understanding of the fixed-length tape-file generator, the user should read the card-to-tape section of the Systems Reference Library publication: *Tape Utility Programs with 120-Character Label Capa-*

*bility for IBM 1401 and 1460: Specifications*, Form C24-3156. There are four exceptions to the specifications presented in this publication:

1. Multifile tapes can be created by Autotest only if the files have identical control-card specifications.
2. The maximum output-tape record size is:  
4,000 characters for an 8K machine.  
12,000 characters for a 16K machine.
3. The tape rewind and unload option at end-of-job is not provided by Autotest.
4. Only sense switch A is used by Autotest when generating a fixed-length tape file. The settings of the other switches are indicated in the TFF card.

#### Input Files

Input files must consist of fixed-length records. Each record must begin on a new card. The number of cards per data record must be the same for each record and is limited only by the amount of core storage available for the output tape record (see *Output Files*), except when performing field selection. In this case, the input record cannot consist of more than 99 cards.

Autotest can produce identical multi-file tapes only if the files have identical control-card specifications. The last file to be written must be followed by a card with /EOF/EOJ/ in columns 1-9. When multiple files with identical control-card specifications are processed, each file except the last is followed by a card with /EOF/ in columns 1-5.

If tape labels are to be written, a RDLIN card and a user-information card, if necessary, must precede the first record of each file.

Only sense switch A is used by Autotest when generating a fixed-length tape file. The settings of the other switches are indicated in the TFF card.

#### Output Files

Output files consist of fixed-length data records that can be blocked or unblocked. The blocking factor (number of data records per tape block) is specified in columns 6-9 of the control card. In the event of an incomplete last block, the block is padded with a character indicated in the control card and is written as a full-length block. Terminal record marks can be inserted after each record in a block.

The record size and blocking factor used is limited only by the size of the output-tape record that can be accommodated. The maximum output-tape record size depends on machine size:

- 4,000 characters for an 8K machine.
- 12,000 characters for a 16K machine.

If tape labels are not written, single tape marks are written at the end of each file and, with multi-reel files, at end-of-reel.

If tape labels are written, the program writes a header label followed by a tape mark *before* each file and a trailer label preceded and followed by a tape mark *after* each file. End-of-reel, in the event of multi-reel files, is indicated by an EOR trailer label preceded and followed by a tape mark.

#### Sequence Checking

This program can perform two types of sequence checks on the input records:

1. Interrecord sequence checking is performed by comparing a predetermined field from one record to the next to assure *ascending* sequence of the records within a file. The field to be compared against can be a maximum of ten digits. It must be in the same location in each input record. Also, it must be an all-numeric field (blanks are converted to zeros) contained within one card of the cards comprising each input record. The first sequence number cannot be all zeros. Duplicate numbers are interpreted as a sequence error. Exception records are included in the check.
2. Intrarecord sequence checking is performed by comparing a predetermined field from one card to the next to assure *ascending* sequence of the cards composing a record. The field is normally one or two positions long, allowing as many as 99 cards per record. When this is the case, the sequence can begin again (normally with 01) with the first card of each record. If it is desired to assure ascending sequence of all the cards within the file, rather than within a single record, this field can be increased to a maximum of ten digits. In all cases, the field must be in the same location in each input card and must be an all-numeric field (blanks are converted to zeros). The first sequence number cannot be all zeros.

#### Exception Procedures

Records identified by the presence or absence of a field of up to 72 constant *exception characters* can be bypassed when the card-to-tape operation is performed. A punch in column 16 of the control card specifies that exception procedures are to be followed. A separate card called the *exception card* is supplied with the constant exception characters and the following information:

1. Whether the records bypassed are to be identified by the presence or the absence of the exception characters.



2. The length of the exception field.
3. The location of the field within the input record.

The exception field must be contained within one card, and that card must be in the same location in each record.

#### Field Selection

The program can select fields from the input record and arrange these fields in any order in the output record. A field selected for output must be within the limits of one card of the input record. Up to 16 fields can be selected. They are defined in separate *field-selection* cards in the order of their occurrence in the input record (see *Field Selection Cards*).

Each field definition specifies:

1. The location of the field in the input record.
2. The length of the field (maximum 80 positions).
3. The location in which it is to be inserted in the output record.

Any positions of the output record that are not used are written as blanks.

*Note:* If a group mark is present in the high-order position of a selected field, a wrong-length-record error will occur.

#### Column-Binary Cards

This program can prepare binary tapes from column-binary cards. Input can consist entirely of column-binary cards or column-binary intermixed with BCD (binary-coded-decimal) cards. When processing column-binary cards, each card is treated as a single record. The following operations cannot be performed when processing binary intermixed with BCD records:

1. Sequence checking.
2. Exception procedures.
3. Field selection.
4. Blocking.
5. Group-mark conversion.
6. Record-mark insertion.

When a file contains column-binary cards intermixed with BCD cards, the column-binary cards must have 7 and 9 punches in column 1. The resulting tape records are 84-characters long for BCD cards and 168-characters long for column-binary cards. The last four characters of BCD records and the last eight characters of binary records are written automatically by this program to provide a *look-ahead* feature that enables a program to distinguish between BCD and binary records during tape reading.

Files that contain all column-binary cards do not require the 7 and 9 punches in column 1. The user can specify that either the first 72 or all 80 columns of each

card be used, producing either 144- or 160-character-position tape records. *Look-ahead* characters are not written after the records of all-binary files.

#### Parity

BCD tape records can be written in either even or odd parity. Even parity is written unless odd parity is specified in the control card.

When binary or intermixed binary and BCD records are processed, the binary records are always written in odd parity and the BCD records are always written in even parity. Tape labels are always written in even parity.

#### Group-Mark Conversion

An IBM 7080 group mark (12-5-8) in cards can be converted to an IBM 1400 series group mark (12-7-8) when it is written on tape.

#### Totals Printout

At the end of each file, the program prints the number of cards read and the number of data records (including padded records) written on tape.

*Note:* See *Control Card* for a summary of the control-card information required.

#### Tape Rewind

The user specifies in column 17 of the control card which one of the following options he desires at end-of-job:

1. No tape rewind.
2. Tape rewind.

#### Control Cards

The user must provide three control cards when a tape file consisting of fixed-length records is to be generated. These control cards are:

1. The TFF card, which calls the program used to create the file. This card is punched in the following format:

Columns	Contents
1-3	TFF
4	Output tape unit number (1-6).
5-10	These columns are used to indicate the settings of sense switches B-G respectively. A 1 punched in the corresponding column indicates that the sense switch is ON, and any other punch indicates that the sense switch is OFF.

The TFF card must precede the control and optional cards that will be used to generate the tape file.

2. The program control card which is used to supply the parameters of the file to be generated. This card is punched in the following format:

Columns	Contents
1-5	<i>Input record length.</i> Punch the number of characters in each input record. This field must be all-numeric with high-order zeros punched.
6-9	<i>Blocking factor</i> (number of data records per tape record). The blocking factor must be all-numeric with high-order zeros punched.
10-14	<i>Output record length.</i> These columns need be used only when performing field selection. Punch the number of characters per output data record, not including the terminal record mark if appended by the program.
15	<i>Tape label code.</i> Punch: <i>N</i> — if no labels are to be read or written. <i>1</i> — if IBM standard 120-character labels are to be written on tape. <i>2</i> — if IBM standard 120-character labels on the output tape are to be read and checked on the file-identification field only. <i>3</i> — if IBM standard 120-character labels on the output tape are to be read and bypassed.
16	<i>Exception code.</i> Leave blank if no exception card is used. Punch a <i>1</i> if exception procedures are to be performed.
17	<i>Rewind option.</i> Leave blank if the output tape is not to be rewound at end-of-job. Punch a <i>1</i> if the tape is to be rewound.
18	<i>Parity.</i> Leave blank if the tape is to be written in even parity. Punch a <i>1</i> if odd parity is to be used.
19, 20	<i>Field-selection option.</i> Leave blank if field selection is not to be performed. If field-selection cards are used, punch the number of fields to be selected. Punch the high-order zero if required.
21-25	Not used.
26	<i>Group-mark conversion.</i> Leave blank if group-mark conversion is not desired. Punch a <i>1</i> if IBM 7080 group marks (12-5-8) are to be converted to IBM 1400 series group marks (12-7-8) and then written on tape.
27	<i>Record type.</i> Indicate the record type by punching the correct code as follows: blank — All BCD <i>1</i> — All binary, 80-column records. <i>2</i> — All binary, 72-column records. <i>3</i> — Mixed BCD and 80-column binary records. If this column is punched, columns 1-14 must be blank.
28	<i>Padding character.</i> Punch the character to be used by the program to create padding records

## Columns

## Contents

that will fill out an incomplete block at the end of a file. The following characters, with their card codes indicated in parentheses, cannot be used for padding:

- group mark (12-7-8)
- record mark (0-2-8)
- tape mark (7-8)
- word separator (0-5-8)
- substitute blank (2-8)

29

*Terminal record-mark option.* Punch a *1* if a record mark is to be appended to the end of each record, including padding records, in a tape block.

31-35

*Creation date.* These columns can be used only if a *1* is punched in column 15. Punch the date of the day that the program is being run as follows:

Columns 31-32 — year (i.e., 64)

Columns 33-35 — day of year (001 represents Jan 1, 365 represents Dec 31).

The date punched in these columns is used to check the retention cycle of the output tape and is also the creation date entered in the new header label.

36-40

*Reel serial number.* These columns can be used only if a *1* is punched in column 15. If the retention cycle check of the label on the output tape is bypassed, the contents of these columns are used as the reel serial number in the new header label.

41-51

*Sequence checking.* These columns are used to specify the input record field(s) that are to be used to check the sequence of records in a file and the sequence of cards in a record.

Columns 41-45 — Punch the high-order position of the field within each record that is to be checked for interrecord sequencing.

Columns 46, 47 — Punch the number of characters in the interrecord sequence field.

Columns 48, 49 — Punch the number of the column in each card in which the intrarecord sequence field begins.

Columns 50, 51 — Punch the number of characters in the intrarecord sequence field.

*Note:* Either interrecord or intrarecord sequence checking can be performed without the other.

52-80

Leave blank.

3. The /EOF/ card which must follow intermediate files when a multi-file reel is being created.

4. The /EOF/EOJ/ card which follows the data file and indicates the end of the TFF program. It is punched in the following format:

## Columns

## Contents

1-9

/EOF/EOJ/

### Field-Selection Cards

One or more field-selection card must be supplied if field selection is to be performed. The number of selected fields determines the number of field-selection cards required. Each field definition requires ten positions, which means that the maximum number, or 16 fields, can be defined in two cards.

Selected fields must be defined in the order of their appearance in the input record. Unused columns must be blank.

Each selected-field definition is punched in ten columns as follows:

Positions	Contents
1, 2	Card number. The program computes the number of cards in each record from the number of characters per record.
3, 4	Card column in which the field begins.
5, 6	Number of characters in the field.
7-10	High-order position of the field in the output record.

### Exception Card

An exception card must be supplied if exception procedures are to be performed. The format of the card is as follows:

Columns	Contents
1-72	The information that is compared against to identify an exception record. Punch this information beginning in column 1 and continue as far as necessary.
73	Exception code. The following codes can be used: blank — Records identified by the presence of the exception field are bypassed. I — Records identified by the absence of the exception field are bypassed.
74, 75	Number of columns to be compared.
76-80	High-order position within the record of the field to be compared.

### RDLIN Card

If a new header label is to be written on the tape by the tape file generator, a RDLIN card containing the label information must be supplied by the user.

The format of the RDLIN card is as follows:

Columns	Contents
1-15	Not Used
16-20	RDLIN
21-24	Retention period
25-29	Creation date
30-39	File identification

Columns	Contents
40-44	File serial number
45-48	Reel sequence number
49-52	Leave blank
53	Density
54	Checksum
55	Block sequence
56	Tape checking/interpreting
57	Tape data recording technique
58	Tape data processing technique
59-62	Creating system
63	Record format
64-68	Record length
69-73	Blocking factor/size
74	Checkpoint
75	Not used
76	Indication of whether a user-information card is supplied or not. If label positions 101-120 are to contain anything other than blanks, supply a user-information card and punch a 1 in this column of the RDLIN card.
77-80	Not used.

### User-Information Card

A user-information card is used if information is to be written in positions 101-120 of the header and trailer labels on the output tape. Punch the card as follows:

Columns	Contents
1-20	Information to be written in positions 101-120 of the header and trailer labels.
21-80	Not used.

### Tape-to-Printer Program

The Autotest tape-to-printer program prints the contents of tape records. Field selection ability (fixed-length records only) is provided to permit the user to specify that only certain fields are to be printed and to specify the desired format of the printed output. If tape labels are present, the header label of each file encountered is printed.

For a better understanding of the Autotest tape-to-printer program, the user should read the tape-to-printer section of the Systems Reference Library publication: *Tape Utility Programs with 120-Character Label Capability for IBM 1401 and 1460: Specifications*, Form C24-3156. There are five exceptions to the specifications presented in this publication:

1. The maximum input-tape record size is:  
4,000 characters for an 8K machine.  
12,000 characters for a 16K machine.

2. The program recognizes end-of-job when it processes an ET/P end card instead of by file count or RDLIN card information.
3. A heading line cannot be printed preceding each file when processing multiple files.
4. The tape rewind and unload option at end-of-job is not provided by Autotest.
5. The settings of sense switches B-G are indicated in the T/P call card.

### Input Files

This program can process fixed- or variable-length records, and the tape records can be blocked or unblocked. If blocked, variable-length records are processed, each record within the block (including the last) must be terminated by a record mark. Because the first four characters of blocked, variable-length tape records are normally a *block character count*, this program excludes these positions from the printout. The fifth position of the tape block is considered the first position of the first data record.

The program can process multi-file reels, printing a specified number of files or only those identified by certain header labels. In Autotest, a heading line can not be printed preceding each file when multiple files are processed. If unlabeled files are processed, or if tape labels are bypassed, the exact number of files to be processed is punched in columns 21 and 22 of the control card. When labels are processed, a RDLIN card is supplied for each file to be printed. If a file has a header label for which no RDLIN card is furnished, the program can either halt or bypass this file and continue searching for the file identified by the RDLIN card. The RDLIN cards must be in the same sequence as the tape files. No attempt is made to search for a label on that portion of tape already read.

The program recognizes end-of-job when it processes an ET/P end card.

### Output Format

During processing of labeled files, each header label is printed on a separate page preceding the data file.

Unless excluded by a punch in column 24 of the control card, a page number is printed at the top of each page of the printed output. This number begins with 1 for the first page of each file.

One line of heading information can be printed on each page, on the same line as the page number. A 1-punch in column 23 of the control card indicates the presence of two heading-information cards. The 132-position print line is taken from all 80 columns of the first card and the first 52 columns of the second card.

If no field selection is performed, each record is printed beginning on a new line and continuing for as many lines as necessary.

When performing field selection (fixed-length records only), as many lines are printed as are necessary for the last defined field, up to a maximum of nine lines. A field selected for output must be contained within one 132-character print line. Any portions of the output record that do not contain selected fields are printed as blanks.

Single spacing is always maintained between lines within a record. The spacing between records can be determined either through control-card designation or through first-character forms control.

Column 25 of the control card is used to specify that single, double, or triple spacing be maintained between records. If the first character of each record is to determine spacing, column 25 contains a code indicating that fact.

When operating under first-character forms control, the following codes in the first position of a record cause an immediate space or skip:

<i>Code</i>	<i>Space or Skip Action</i>
Blank	Single space
Zero	Double space
– (B-bit)	Triple space
1-9 or J-R	Skip to channel 1-9 (that is, 1 or J = skip to channel 1; 2 or K = skip to channel 2; etc.).

An ampersand (&) is sometimes used as a first-character control code to cause space suppression. This program does not provide for space suppression. Column 25 of the control card is also used to specify that, in the event of an ampersand forms-control character, the record be either:

1. Bypassed, or
2. Printed after a line containing three asterisks.

*Note:* When fixed-length records with first-character forms control are being processed, the forms-control character is considered to be the first character of the data record and is printed unless excluded through field selection. When variable-length records are processed, the forms-control character is not printed.

### Exception Procedures

Records identified by the presence or absence of a field of up to 72 constant *exception characters* can be bypassed when the tape-to-printer operation is performed. A punch in column 16 of the control card specifies that exception procedures are to be followed. A separate card called the *exception card* (see *Exception Card*) is supplied with the constant exception characters and the following information:

1. Whether the records bypassed are to be identified

by the presence or the absence of the exception characters.

2. The length of the exception field.
3. The location of the field within the input record.

The exception field must be contained within 72 consecutive positions and must be in the same location in each record. Note that when processing with first-character forms control, the forms-control character is considered to be the first character of the data record. When variable-length blocked records are processed, however, the block character count (positions 1-4 of the tape record) is not considered part of the first data record.

Padded records can be bypassed by defining an exception field that consists entirely of the padding character.

#### Field Selection

When fixed-length records are processed, the program can select fields from the input record and arrange these fields in any order and in any position in the print line in the output record. A field selected for output must be within the limits of one print line. Up to 16 fields can be selected. They are defined in separate field-selection cards in the order in which they are to be printed (see *Field-Selection Cards*). The selected fields can be contained in a maximum of nine lines. Each field definition specifies:

1. The location of the field in the input record.
2. The length of the field.
3. The print position in which it is to be printed.

When field selection is performed, the output may consist of a maximum of nine lines per record.

#### Parity

Tape can be read in either even or odd parity. Even parity is assumed unless odd parity is called for in column 18 of the control card.

#### Totals Printout

At the end of each file, the program prints the number of records read and the number of records printed.

#### Tape Rewind

The user specifies in column 17 of the control card which one of the following options he desires at end-of-job:

1. No tape rewind.
2. Tape rewind.

#### Control Cards

The user must provide three control cards when a tape file is to be printed. These cards are:

1. The  $\tau/P$  card which calls the program used to print the tape file. This card is punched in the following format:

Columns	Contents
1-3	$\tau/P$
4	Input tape unit number (1-6)
5-10	These columns are used to indicate the settings of sense switches B-G respectively. A 1 punched in the corresponding column indicates that the sense switch is ON, any other punch indicates that the sense switch is OFF.

The  $\tau/P$  card must precede the program control card, optional cards, and  $\epsilon\tau/P$  card.

2. The program control card which is used to supply the parameters of the file to be printed. This card is punched in the following format:

Columns	Contents
1-5	<i>Input record length.</i> Punch the number of characters in each fixed-length input record. In this case, the field must be all-numeric with high-order zeros punched. If variable-length records are being processed, punch VVVVV.
6-9	<i>Blocking factor</i> (number of data records per tape record). In this case, the field must be all-numeric with high-order zeros punched. If variable-length unblocked records are being processed, punch UUUU. If variable-length blocked records are being processed, punch BBBB.
10-14	Leave blank.
15	<i>Tape label code.</i> Punch: N — If no labels are to be read or written. 1 — If IBM standard 120-character labels are to be read and checked in their entirety. 2 — If IBM standard 120-character labels on the input tape are to be read and checked on the file-identification field only. 3 — If IBM standard 120-character labels on the input tape are to be read and bypassed.
16	<i>Exception code.</i> Leave blank if no exception card is used. Punch a 1 if exception procedures are to be performed.
17	<i>Rewind option.</i> Leave blank if the input tape is not to be rewound at end-of-job. Punch a 1 if the tape is to be rewound.
18	<i>Parity.</i> Leave blank if the tape is to be read in even parity. Punch a 1 if odd parity is to be used.
19, 20	<i>Field-selection option.</i> Leave blank if field selection is not to be performed. If field-selection

<i>Columns</i>	<i>Contents</i>
	cards are used (fixed-length records only), punch the number of fields to be selected. Punch the high-order zero if required.
21, 22	<i>Files control.</i> Punch the number of tape files to be processed. These columns must be punched if no labels are present, or labels are present but bypassed. If tape labels are processed against RDLIN cards (file-identification field only or full check), leave these columns blank.
23-25	<i>Print options.</i> Use these columns as follows: Column 23 — Leave this column blank if a heading line is not to be printed at the top of each page. Punch a <i>I</i> if two cards containing the heading line information are supplied. Column 24 — Punch a <i>I</i> if the normal page numbering routine is to be bypassed. Column 25 — Punch a code indicating the spacing to be maintained between records. The acceptable codes are: <i>Blank or 1</i> — Single-spacing (no additional space) <i>2</i> — Double-spacing (one additional space) <i>3</i> — Triple-spacing (two additional spaces) <i>S</i> — First character control. Records preceded by an ampersand are not printed. <i>*</i> — First character control. Records preceded by an ampersand are printed followed by a line containing only three asterisks.
26-80	Leave blank.

3. The ET/P, which must follow the other required control cards for the program and indicates the end-of-job. It is punched in the following format:

<i>Columns</i>	<i>Contents</i>
1-4	ET/P

#### *Field-Selection Cards*

One or more field-selection cards must be supplied if field selection is to be performed. The number of selected fields determines the number of field-selection cards required. Each field definition requires 11 positions, which means that the maximum number, or 16 fields, can be defined in three cards.

Selected fields must be defined in the order of their appearance in the output record. Unused columns must be blank.

Each selected field definition is punched in 11 columns as follows:

<i>Columns</i>	<i>Contents</i>
1-4	High-order position of the field in the input record.
5-7	Number of characters in the field.
8	Line number.
9-11	High-order print position.

### Exception Card

An exception card must be supplied if exception procedures are to be performed. The format of the exception card is the same as described in the section *Exception Card* under *Tape-File Generation — Fixed-Length Records*.

### RDLIN Card

If input header labels are to be checked, a RDLIN card containing the label information must be supplied by the user. The format of the RDLIN card is the same as described in the section *RDLIN Card* under *Tape-File Generation — Fixed-Length Records*.

### User-Information Card

A user-information card is required if information in positions 101-120 of the input header label contain information to be checked. Punch the card as follows:

Columns	Contents
1-20	Information to be compared against positions 101-120 in the input header label.
21-80	Not used.

### Snapshot Feature (SNP)

The snapshot feature of Autotest provides a means for taking a printout of core storage between specified limits at one or more points in the object program during its execution. The SNP program occupies 810 positions of core storage (14690-15499 [6690-7499]). At the first snapshot, core-storage printout occurs before execution of the instruction specified in the control card. This instruction must be at least a 4-character instruction with operands that are not address-modified. (See *Autotest Specifications*.) After the printout occurs, the program continues.

Additional snapshots can occur at other points in the object program. These are set up by the programmer by using an add-type patch to branch to the SNP program. (See *Additional Snapshots*.)

### SNP Control Card

Only one snapshot control card (for the first snapshot) may be included in a test deck. If more than one is encountered, a message is printed to indicate that only the first snapshot control card is to be processed.

The card must be placed just ahead of either the OBJ card or a DTR control card. If an SNP control card erroneously follows a DTR control card, the snapshot program is not executed during the running of an object program. A message indicating this is printed.

Columns	Contents
1-3	SNP
4-8	Address of Op code of instruction at which the core-storage printout is to occur.
9-13	5-character address of lower-limit (high-order) position of core storage to be printed. This must be the first position after an even hundred.
14-18	5-character address of upper-limit (low-order) position of core storage to be printed (must be an even hundred).
19-80	Leave blank.

*Note:* Addresses must be 5-character numeric with leading zeros punched.

### SNP Execute Card

An SNP execute card (for the first snapshot) is required in the object deck if the programmer wishes to use the snapshot feature. This card must be placed immediately before the disk-trace execute card or the normal execute card in the object program or overlay segment to which the feature applies. When this card is encountered, Autotest replaces the first snapshot instruction with an unconditional branch to the snapshot routine. If the SNP control card was erroneously preceded by a DTR control card, the SNP execute card is ignored and the next card is processed.

The snapshot execute card is punched in the following format for use with Autocoder condensed cards:

Columns	Contents
1-3	006
4-5	08
6	Word separator (0-5-8)
7	/
8-10	H7A[H7/] — 3-character address of snapshot-setup routine.
11-13	080
14	Blank

### Additional Snapshots

If the programmer desires to obtain more than one snapshot in the object program, he can set up, with an add-type autopatch card, a branch to the snapshot program at each additional snapshot location.

On additional snapshots the following rules apply:

1. The snapshot occurs after execution of the specified instruction.
2. Snapshot core-storage limits are the same as those specified by the *original* SNP control card.
3. The snapshot autopatch card must be placed just ahead of the normal execute card in the object program or the overlay segment to which the snapshot applies. *Note:* Autopatching always requires a PCC card.

- Any number of additional snapshots are possible by using snapshot autopatch cards, as long as patching space is available.

#### SNP Print Format

The snapshot program prints lines of 100 characters starting at XX01 or XXX01, the first position given in the control card, and including the character at YY00 or YYY00, the upper position given.

- Every ten positions are followed by a blank. Blocks of 100 positions that are completely blank are bypassed in printing.
- Word-mark indications (l's) are printed beneath their respective characters.
- All unprintable characters are replaced with printable characters identical to replacements in the Autotest core-storage printout (see Figure 1).

#### Print

Position	Output and Explanation
1-10	First ten characters
11	Blank
12-21	Second ten characters
22	Blank
etc.	etc.
100-109	Last ten characters
110-115	Blanks
116-120	Address of 100-character area starting at XX01 or XXX01
121	Dash
122-125	AREA (This word is printed.)
126-132	Blanks

#### Disk-Trace Feature (DTR)

The disk-trace feature of Autotest provides the option for a printout of the data and control words involved in disk read or write instructions.

Two disk-trace programs are provided in 1401/1311 Autotest. The user can specify that the disk-trace output be printed as the trace is performed, or, if a tape unit is available, that the disk-trace output be written on tape and printed at the end of the object run.

The disk-trace program to be used is determined by the contents of column 4 of the DTR control card. If tape is to be used to store the disk-trace output until the end of the object run, the user must punch the number of the available tape unit in column 4. Otherwise, this column is left blank and the disk-trace program that does not use a tape drive is selected.

The length of the data display is determined by the placement of a group-mark with a word-mark in core-storage. The first group-mark with a word-mark en-

countered in the disk read or write area indicates to both disk-trace programs that the end of the data has been reached.

The disk-trace program that does not use a tape drive requires 1,170 positions of core storage. If SNP is used, this disk-trace program is located in positions 13520-14689 [5520-6689] and, if SNP is not used, the program is located in positions 14330-15499 [6330-7499].

Disk read or write instructions located in positions 200-332 cannot be traced, and/or any data read or written by disk instructions and located in positions 200-332 cannot be displayed, if the disk-trace program that does not require a tape drive is used.

The disk-trace program that uses a tape drive requires 810 positions of core storage and is located in positions 13880-14689 [5880-6689] when SNP is used. If SNP is not used, this disk-trace program is located in positions 14690-15499 [6690-7499].

#### DTR Control Card

Only one disk-trace control card may be included in a test deck. This one card can specify a maximum of seven locations at which to trace read- or write-disk instructions. If more than one card is encountered, a message is printed to indicate that only the first disk-trace control card is to be processed.

The disk-trace control card is placed in the test deck after the SNP control card and immediately before the OBJ card.

It is punched in the following format:

Columns	Contents
1-3	DTR
4	Number of the tape unit available (1, 2, 3, 4, 5, or 6). Leave blank if DTR without using a tape drive is desired.
10-14	5-character address of disk read or write instruction.
20-24 30-34 40-44 50-54 60-64 70-74	} Other addresses of disk instructions if desired.

*Note:* Addresses must be 5-character numeric with leading zeros punched. If the disk read or write instructions to be traced are not modified within the object program, the address specified on the DTR control card is the address of the operation code of the disk instruction. However, when a disk read or write instruction to be traced is modified during program execution, or when IOCS is used for disk reading and writing, the following steps must be taken:

- A 4-character NOP (N000) instruction must be inserted immediately ahead of the disk read or write instruction to be traced. If using IOCS, this NOP is automatically generated.
- The 5-character DTR address on the control card is the operation-code position of the NOP instruction. In IOCS,



the NOP precedes the read or write instruction labeled IOCDIN. (Although IOCDIN is the IOCS label for all disk I/O instructions except seek disk, disk trace will only trace read and write instructions. Thus, when the address of the IOCS-generated NOP is given on the DTR control card, all IOCS disk read and write instructions in the object program are traced.)

3. 5-character addresses referring to non-modified, modified (NOP), and IOCS-generated (NOP) disk instructions can be intermixed on the same card.

#### DTR Execute Card

A separate execute card is required in the object deck if the programmer wishes to use the disk-trace feature.

This card is placed immediately before or after the snapshot execute card, or before the normal execute card, in the object program or overlay segment to which the feature applies. When this card is encountered, Autotest replaces the disk read and write instruction to be traced with an unconditional branch to the disk-trace routine.

The disk-trace execute card is punched in the following format for use with Autocoder condensed cards:

Columns	Contents
1-3	006
4-5	08
6	Word separator (0-5-8)
7	/
8-10	H7E[H7V] — 3-character address of disk-trace setup routine
11-13	080
14	Blank

#### DTR Print Format

The disk-trace program prints data in lines of 100 characters, regardless of the mode of operation used. The first line of data in each input/output display is preceded by the control information pertaining to the particular trace. Each succeeding line of data in the display contains only the 100-character positions. The other positions are left blank.

Print Position	Output and Explanation
1	R or W, for read or write operation
2	Blank
3-7	5-character address of read or write instruction
8-9	Displayed only on first line of trace for a particular instruction.
10-19	Sector-control field (before execution of read or write instruction)
20-119	As many as 100 characters of data.

#### Halt Card (HLT)

A halt card is placed in the test deck to indicate to the operator that a disk pack (or tape reel) containing data is to be installed, replaced, or removed. This card is placed anywhere before the OBJ card if a disk pack (or tape reel) is to be installed or replaced. It is placed after the object program if a disk pack (or tape reel) is to be removed.

Punch HLT in columns 1-3 of the card. Punch instructions concerning a disk pack (or tape reel) and drive in the remaining columns. Instructions concerning a precreated tape should include the density setting (high or low) of the tape. The operator must know the correct density setting before installing a tape. When the Autotest program encounters this card in the test deck, the machine stops and a listing of the card is printed. When the operator has performed the tasks indicated, he presses START to continue processing. An object deck can contain any number of halt cards.

#### Object Card (OBJ)

The object card signals the Autotest program that the object programs follow. Punch OBJ in columns 1-3 of the card and place it immediately ahead of the condensed object program. In columns 4-8, punch the starting address of the object-program loader. This is the address punched in the appropriate columns of the Autocoder source program CTL card used when the object program was assembled, or if the columns in the CTL card were not punched, the address is 081. Autotest inserts a branch to the Autotest loader at the address specified in columns 4-8. This is done to handle branches to the loader for overlays and branches to the loader from DA execute cards generated by Autocoder.

#### Autopatch Feature

The autopatch feature of Autotest facilitates changes and corrections to the assembled object programs, making it possible to eliminate program reassemblies. Autopatch provides for deleting any object instruction, adding instructions, replacing instructions, and adding multiple-instruction subroutines that may be branched to from several points in the object program.

It is not necessary for the programmer to compute 3-position alphameric addresses to locate the patch in core, or to provide the necessary patch linkage between the object program and the patch area. Autopatch computes all necessary linkage and inserts patches into an available area of core storage.

When using Autopatch, the programmer must have a program control card (PCC) in his test deck. This card specifies the object program highest core-storage limit,

beyond which patching will originate. Autopatch cards are bypassed and testing continues if no PCC card is present.

*Note:* Autotest does not perform a forms-control operation after printing the patch listing. Therefore, unless the program being tested provides for forms spacing, the printout of the program being tested begins on the line immediately following the patch listing.

### Autopatch Card Format

Patch cards are inserted in the object deck as a group just ahead of the execute card for the particular routine or overlay to which the patch cards apply. Processing of patch cards takes place during the loading of the object program under Autotest control. Patch cards are punched in the following format:

Columns	Contents
1-51	Symbolic source coding:
1-2	Page
3-5	Line
6-15	Label
16-20	Operation
21-51	Operands, Comments (only columns 21-42 print on the patch listing)
52	Blank
53	Punch an <i>A</i> if an add type patch. Punch an <i>R</i> if a replace type patch. <i>Note:</i> A delete-type patch is handled as a replacement patch using an <i>nor</i> .
54	Length of old instruction at patch-point address, not necessary for <i>dc-dcw</i> constants.
55-59	Patch-point 5-character numeric address. For instructions, it refers to the <i>Op-code</i> address. For constants, it refers to low-order position or units position of the address desired for the constant.
60	Number of core positions in new patch instruction (not necessary if <i>dc</i> or <i>dcw</i> ).
61-74	Actual instruction to be patched:
61	Operation code
62-66	A-operand

Columns	Contents
67	A-operand indexing: No — leave blank Yes — Punch 1, 2, or 3, respectively, for index locations 1, 2, or 3.
68-72	B-operand
73	B-operand indexing: No — leave blank Yes — Punch 1, 2, or 3, respectively, for index locations 1, 2, or 3.
74	d-modifier
75	Punch * (11-4-8) — this identifies card as an autopatch card.
76-80	Program identification.

Each time patch cards are processed, a program patch listing (Figure 4) is printed which provides full documentation of all changes being made to an object program. When patch instructions are relocated to the patch area, the address printed under *ADDR* in the program patch listing is indented one position to the right. The left-hand column of this listing gives an indication of how the source program should be corrected. (See *Autopatch Error Messages.*)

Patch cards are inserted in the object program as a group, just ahead of the end card, or, if patches are being made to overlay execute routine, just ahead of the execute card. Processing of patch cards takes place during direct loading of the object program by sensing an asterisk (\*) in column 75. Part of the object program is written out onto disk storage to make room for the main autopatch phase. This phase generates the patch instructions and constants. Autotest requires that the next card after an autopatch grouping be an execute card with either 00608, word separator (0-5-8), and/or 00605, word separator, and *B* punched in columns 1-7 respectively.

If the first execute card after the autopatch cards is intended to clear the card-read area, this area is not cleared. To clear the card-read area, a card with 00605, word separator (0-5-8), and *BE2I [BE2Z]* punched in

PROGRAM PATCH LISTING															
TYPE	PG	LINE	SOURCE	CODING	PATCH	PT	ADDR	OP	AADR	IR	BADR	IR	D	ERROR	NOTES
AP	ADD		MLCWA	05, JOE		00311	05016	L	00005		07422				
AP	ADD	02 031	BE	INTU4	REPL 7	00307	05030	B	01912					S	
AP	ADD		BCE	A, B, C	AT 307	00307	05038	B	06789		07123			C	
AP	ADD	12 345	CW	ONEWM		00307	05042	B	00123						
AP	ADD								N						ERR1
AP	ADD	03 053	DC	LEN5@		00490									
AP	ADD	67 890	DCW	123456789		03801									
AP	ADD														ERR2
AP	ADD		DC	1234		03905									
AP	ADD								N						ERR1

Figure 4. Program Patch Listing Sample

columns 1-10 respectively, must be placed in the deck preceding the execute card. If the next card after autopatch cards is not a proper execute card, Autotest bypasses the rest of the current testing and searches for the next test deck (the next `DC`). There will be a print-out of the offending card and an appropriate message.

Upon successfully completing the testing of any given program, the programmer has the following options:

1. Using the current program patch listing(s) as a guide to file the patch cards in the source program for an updated reassembly. *Note:* For subsequent Autotest use, the program limit required by the `PCC` in columns 10-14 will, in most cases, have to change.
2. Leaving autopatch control cards in with the condensed deck to reload under Autotest control, with autopatching being re-performed.

*Address Types:* Two types of addresses are valid in the patch instruction A- and B-operand fields of the patch card:

1. Actual 5-position numeric—Blanks in any of the five positions are considered invalid; hence leading zeros must be punched.
2. 3-position alphanumeric of the format: `Axxx` where, respectively, A is an A-punch xxx is three characters alphanumeric, and b is a blank.

Example: The A-operand field of a `SEEK-DISK` instruction would be punched in columns 62-66 as `A%F0`.

*Definitions:*

1. Patch-point address (columns 55-59 of patch cards):
  - a. When referring to an instruction patch, the patch-point address is the instruction (Op code) address of the instruction in the object program which is to be deleted or replaced, or after which added patch instructions are to be executed.
  - b. When referring to a *constant patch* (`DC` or `DCW`), the patch-point address designates the core-storage address where the units position or low-order position of the constant will be placed.
2. Number of positions (columns 54 and 60):
  - a. Column 54. Length of old instruction located at the patch-point address. Leave this column blank if a `DC` or `DCW` constant.
  - b. Column 60. Length of new instruction to be patched. Leave this column blank if a `DC` or `DCW` constant.

**Patching Rules**

1. Do not specify as a patch-point instruction any object-program instruction which is being modified in core storage during execution, since this instruc-

tion will be relocated to the patch area if a branch-out linkage must be inserted to effect the patch.

2. Do not specify as a patch-point instruction for an add patch, any unconditional branch, or last card test, because the patch will never be executed.
3. Do not specify patch-point instructions in the middle of a chained instruction series.
4. The maximum size blank constant to be processed by autopatching is 49 positions (#49). Any blank constant of 50 characters or more results in an error message.
5. Any `DC` or `DCW` unsigned patch will be loaded exactly as punched in the card from column 21 to the first blank. In the case of a signed numeric constant, the information from column 22 to the blank is loaded and the sign correctly placed.

**Add Patch**

An add-patch card(s) may be used to insert an additional instruction(s) in an object program at the patch point specified so that it may be executed as part of the assembled program. Autotest sets up the necessary branch-out and branch-back linkages and inserts the additional instruction(s) in the patch area of core storage. The patch-point instruction specified for an add patch must be four positions or more in length.

For example, if the programmer desires to insert additional instruction(s) which are to be executed after the instruction located at the patch-point specified in the object program, he should:

1. Code additional instruction(s) in the Autocoder language.
2. Punch A in column 53 and \* (11-8-4 punch) in column 75.
3. Specify the patch point as the instruction (Op code) address of the instruction after which the additional instruction(s) are to be executed. (For multiple add patches, the same patch-point address for each is specified.)
4. In column 54, the length of the old instruction at the patch point is indicated.
5. In column 60, the number of positions in the new patch-point instruction is specified.
6. The actual machine language Op code of the instruction to be added is punched in column 61.
7. Punch the A-operand in 62-66, the B-operand in 68-72, the d-modifier in 74. If A- or B-operand indexing is required, a 1, 2, or 3 is punched in column 67 or 73 for the appropriate operand.

The Autotest system performs the following:

1. Loads the patch-point instruction from its location in the object program to a location in the patch area.
2. Loads an unconditional branch instruction over the patch-point instruction to effect linkage to the patch area.

3. Loads the additional instruction(s) to the patch area following the relocated patch-point instruction. The 5-position patch address in columns 55-59 in each patch card is compared with the patch-point address of the preceding patch card. If the compare is equal, the patch instructions are located sequentially in the patch area of core storage.
4. Loads an unconditional branch instruction in the patch area following the added instruction(s), to effect linkage back to the next sequential instruction after the patch-point instruction in the object program core-storage area.

#### Replace Patch

Any instruction in object program core storage may be replaced by a replace patch. Three types of replace patches are possible:

1. Replace-patch instruction with the same number of positions as the instruction at the patch point being replaced.
2. Replace-patch instruction with a smaller number of positions than the instruction at the patch point being replaced.
3. Replace-patch instruction with a larger number of positions than the instruction at the patch point being replaced. This type of replace patch requires that the patch-point instruction be four or more positions in length to allow for the insertion of a branch to the patch area.

For example, if the programmer desires to replace instructions at patch points specified in the object program:

1. The replace instruction is coded in the source language of the program.
2. Punch *R* in column 53 and an asterisk (11-8-4) in column 75.
3. The patch point is specified as the instruction (Op code) address of the instruction in the object program to be replaced.
4. In columns 54 and 60, the lengths of the old and new instructions are indicated.
5. In columns 61-74, the actual replacement instruction is punched.

The Autotest system performs the following:

1. Compares the size of the replace-patch instruction with the number of positions in the patch-point instruction which is to be replaced.
2. Loads equal-size replace-patch instructions over the original instruction in object-program core storage.

3. Loads replace-patch instructions with a smaller number of positions than the original instruction to the patch-point area in core storage and fills the remaining positions with sufficient *NOP*'s and word marks.
4. Loads a replace-patch instruction with a larger number of positions into the patch area of core storage, an unconditional branch to the patch area over the original object-program instructions being replaced, and an unconditional branch back to the next sequential instruction in the object program.

#### Delete Patch

A delete patch is a special form of replace patch. Code the autopatch card the same as a replace patch where the replacement instruction is specifically a *NOP* of length one.

#### Branch-Last-Card Modification

Due to additional Autotest control cards which very likely may follow the last object-program data card, all BxxxA instructions must be changed to run with Autotest. To easily convert from a 5-character BxxxA last-card instruction test, it is recommended that autopatch be used. The necessary steps required are:

1. Punch in a pre-selected column of the last data card a specific chosen character. For example, lozenge □ (12-4-8).
2. Determine the core-storage position of the last card read which will contain this chosen character (cccc below).
3. Set up a replace-type patch card as follows to convert to a BCE xxx, ccc, d instruction to simulate last card testing.

Columns	Contents
1-51	Autocoder source coding
52	Blank
53	<i>R</i>
54	5
55-59	<i>ppppp</i> — patch-point address of <u>Bxxx</u> A
60	8
61	<i>B</i>
62-66	<i>iiii</i> — I-address of <u>Bxxx</u> A
67	Blank
68-72	<i>cccc</i> — address of core storage where last card special character will appear.
73	Blank
74	d-modifier special character
75	* (11-4-8 punch)

If the last-card test precedes the read instruction and the special character is punched in an extra card rather than the last data card, or if the special character is destroyed (overlaid) before the BCE test is made, the following steps must be taken.

1. NOP the BLC instruction by using a replace-type patch card punched in the following format.

Columns	Contents
1-51	Autocoder source coding
52	Blank
53	R
54	5
55-59	ppppp — patch-point address of BxxxA
60	1
61	N
62-74	Blank
75	* (11-4-8 punch)

2. Set up an add-type patch to test for the special character after the read instruction, or after the read-error test, if present. Punch the add-type patch card in the following format.

Columns	Contents
1-51	Autocoder source coding
52	Blank
53	A
54	7 (length of read instruction), or 4 (read-error test)
55-59	Patch-point address of the Op-code of the read instruction or read-error test
60	8
61	B
62-66	A-operand of BxxxA
67	Blank
68-72	Address in core storage where last card special character will be stored
73	Blank
74	d-modifier special character
75	* (11-4-8 punch)

#### Final-Halt Modification

To change a final halt in the object program to an unconditional branch to the Autotest linkage routine (for storage print or no storage print), punch a replace patch card in the following format:

Columns	Contents
1-51	Autocoder source coding
52	Blank
53	R
54	n — length of old halt (must be at least 4)
55-59	ppppp — patch-point address of halt.

Columns	Contents
60	4 or 5 — length of new unconditional branch (4 if length of old halt is four, 5 if length is five or more characters).
61	B
62-66	15909 [7909] — Autotest linkage address
67-74	Blank
75	* (11-4-8 punch)

#### DC or DCW Patch

Additional constants, dcw (with word mark in the high-order position) or DC (without word marks) may be added to any specified patch point in core storage not already being used by either the object program or Autotest. Constants may also be patched to replace previous constants. *Note:* Address constants cannot be processed directly by autopatch.

When punching constants, the following rules must be followed:

1. For an alphameric constant, there must be a leading @ (4-8 punch) in column 21 and a terminal @ before or in column 51.
2. For a signed numeric constant, the sign is punched immediately preceding the constant in column 21, and there must be a terminal blank before or in column 51.
3. For an unsigned numeric constant, the constant starts in column 21, and there must be a terminal blank before or in column 51.
4. For a blank constant, there must be a # (8-2-1 punch) in column 21 with the size of the constant punched beginning in column 22. Autopatch cannot process a dc or dcw blank constant equal to or greater than 50 positions. For example, DC #49 is valid, but dcw #50 is invalid for autopatching.

#### Adding Constants to the Original Object Program:

To add a constant to the object program:

1. Punch the constant in source Autocoder format in columns 1-51. See the preceding paragraphs for valid constants.
2. Punch A in column 53.
3. Leave columns 54 and 60 blank.
4. Punch the patch point address in columns 55-59. This is the units position or low-order address desired for the constants.
5. Punch an \* (11-4-8 punch) in column 75.
6. Columns 10-14 of the pcc *must be changed* to reflect the highest position of core storage used by the object program which has been increased by the added constants coded in autopatch.

For an add or replace DC or DCW, Autotest:

1. Checks for DC or DCW in columns 16-18.
2. Analyzes columns 21-51 for Autocoder constant format and sets up the constant.
3. Moves DC constants to the core-storage address specified as the patch point.
4. Loads DCW constants to the core-storage address specified as the patch point.

#### **Subroutine Insertion**

The autopatch feature of Autotest makes it possible to insert one or more subroutine(s), which use the STORE B-ADDRESS REGISTER instruction to establish a linkage back to the main program, in the patch area of core storage and then branch to the subroutine(s) from several points in the object program.

To insert a subroutine without reassembling, code all instructions in the autopatch card format as replace-type patches where the length of the old instruction is specified to be equal to the length of the new instruction.

The first instruction of the subroutine should have a patch-point address equal to or greater than the limit specified in the PCC card that was used before the subroutine was inserted. The patch-point address for each instruction after the first is calculated by adding the instruction length and the patch-point address of

the instruction that will immediately precede it in core storage.

If a subroutine is inserted, a new PCC card must be punched. The information punched in this card is identical to the old PCC card with the exception of the address punched in columns 10-14. To determine the address to be punched in columns 10-14 of the new PCC card, increase the address in columns 10-14 of the old PCC card by at least the number of core-storage positions used by the subroutine.

The branch instruction that serves as an exit from the subroutine must be specified in a special manner to ensure that the branch is executed properly. Specify the length of the old instruction as 5 and the length of the new instruction as 4. The result is a 4-position branch instruction followed by a 1-position NOP with a word mark.

The required branches to the subroutine are inserted by the use of add or replace patches.

#### **80/80 Listing of Output Cards**

At the end of the complete test session, the Autotest program prints a message specifying the procedure to follow in order to obtain an 80/80 listing of all or part of the punched-card input and output. The listing will be completely identified as to the object program and the stacker from which the cards were removed.

## Program Deck Description

The user is supplied with three distinct Autotest punched-card decks. Each of the three decks is used as an independent unit and is punched in the IBM 1401/1311 Autocoder format. The function of each deck is as follows:

- Deck A — Writes Autotest routines onto disk storage and processes the next IDC.
- Deck B — Processes next IDC without building the pack.
- Deck C — Re-establishes Autotest storage printout linkage.

The first two cards in decks A and B are clear-core-storage cards identified by a 1 or 2 punch in column 80, indicating whether it is the first or second card in the deck.

The remaining cards in decks A and B and all the cards in deck C are identified by the following:

Columns	Contents
73-75	Sequence number, beginning with 003 in decks A and B, and 001 in deck C.
76	System modification level.
77-78	System program number: 82 for 16K version 81 for 8K version
79	System version number.
80	A, B, or C to indicate deck in which the card is contained.

Deck A initially writes all Autotest routines and linkages in disk storage. The user must prepare test packets to be placed after deck A in the card read hopper of the IBM 1402 Card Read-Punch. Program operation is automatic after the Autotest routines are written on disk storage. Halts can occur because of object-program halts, required console operator intervention (e.g. to turn on or off sense switches), or error halts. A date card must be punched and inserted into deck A so that the retention cycle of the disk labels can be checked.

Deck B is a smaller version of deck A. Deck B is immediately followed by any number of Autotest test decks and can only be used when all the Autotest routines have been previously written on disk drive zero and not subsequently destroyed. Therefore, an Autotest pack prepared at one time by deck A can be used with deck B to process Autotest test decks (identified by the identification cards). A date card must

also be punched and inserted into deck B so that the disk-label retention cycle can be checked.

Deck C is a restart deck to be used when the object program erroneously destroys the Autotest linkage routine (in high-core locations). When, at an end-of-job or program hang up condition, a programmed or manual branch to the Autotest linkage routine is unsuccessful, deck C is used to load the core-storage printout program into core storage and execute it. Deck C is followed in the feed hopper by the cards still to be processed, and normal Autotest operation continues after the storage print.

## Autotest Setup Instructions

### Storage Considerations

Core-storage requirements are as follows:

1. The Autotest control program is located in core-storage positions 15500 to 15999 [7500 to 7999]. The user must not use this area of core storage when under Autotest control.
2. The snapshot and disk-trace features require additional storage areas in the upper positions of core storage. If the programmer uses one or both of these features in testing his program, he should use a program control card, which specifies the highest core-storage position required by his object program. Autotest can then determine if there is sufficient core storage available to load the particular routine requested (snapshot and/or disk trace). The routine(s) are loaded if the upper positions of core storage are not being used by the object program. Otherwise, an indication is printed that it will not be used.

Snapshot, if called for, will use core-storage positions 14690 to 15500 [6690 to 7500]. Disk trace, if called for without snapshot, utilizes core-storage positions 14690 to 15500 [6690 to 7500]. If disk trace and snapshot are used concurrently in the same test run, disk trace utilizes core-storage positions 13880 to 14690 [5880 to 6690].

Disk-storage limitations are as follows:

1. Disk-storage areas 019000-019979 are restricted from user programming use. If clear disk storage or disk record load attempts to write in these areas, a message will occur and this action (clear or load) will

not occur. The disk area 019980-019999 is available for user pack labeling if desired.

2. When autopatch is used, additional disk storage is required and calculated as follows:
  - a. Take the total number of autopatch cards in any overlay segment.
  - b. Multiply by four.
  - c. Add two.
  - d. The result is the maximum number of sectors required as additional disk storage for autopatch instructions. The highest numbered of these sectors is 018999 and they continue, in descending sequence, as many sectors as required.

### Control-Card Insertion

Each test deck must contain the following in sequence (see Figure 5):

- IDC (identification card)
- OBJ card (object card)
- Object program condensed Autocoder deck
- Object program execute (end) card.

All other cards, including those indicating Autotest optional features, are optional. If used, they must occur according to the following order:

IDC	Additional IDC cards are grouped with the initial IDC card at the beginning of the test deck.
ID1	Immediately follows the IDC card(s).
CCC Console Status Card	Immediately follows IDC and ID1 cards.
PCC	Immediately follows the preceding control cards.
DRL Control Cards— EDRL	Occurs in the test deck anywhere after the IDC card(s) and before any of the SNP, DTR, or OBJ cards.
CDS— Control Cards— ECDS	Same as PCC, except when clear-disk-storage is desired after program execution. Then they are placed after the data cards. If there are no data cards, they are placed after the final execute card.
PRD— Control Card— EPRD	Same as CDS. Print disk can also occur before or after program execution.
TFV— Data Cards— ETFV	Must occur after the PCC card and before any of the SNP, DTR, or OBJ cards.
TFF— Control and Data Cards— /EOF/EOJ/	The TFF card and the control card are required and must occur in that order after the PCC card and before any of the SNP, DTR, or OBJ cards. The other cards, if used with this program, must follow the control card in the following order: exception, field selection, RDLIN, user information, data file, /EOF/ (if a multi-file reel is being created), RDLIN file 2, user-information file 2, data file 2, etc., /EOF/EOJ/ (required).

T/P—  
Control Cards—  
ET/P

The T/P card and the control card are required and must occur in that order after the PCC card and before any of the SNP, DTR, or OBJ cards. The other cards, if used with this program, must follow the control card in the following order: exception, field selection, heading information (2 cards), RDLIN, user information, RDLIN file 2 (if multi-file reel being printed), user-information file 2, etc., ET/P (required).

RDLIN  
(Label Utility)

The RDLIN cards for the disk-label program must be placed in the deck in the following order:

1. Date (if required).
2. RNA and NEW.
3. CHNG, ENTR, DELT, PRNT, PNCH, or PNCX (these cards can be in any order).

HLT

Occurs after the IDC card(s) and anywhere before the OBJ card in the test deck. Also, may occur after the data cards. (If there are no data cards, it occurs after the normal execute card.)

SNP

Can be followed only by a DTR or OBJ card.

DTR

Can be followed only by an OBJ card. SNP and DTR cards must occur as the last card or cards before the OBJ card.

SNP—  
Execute Card

Must be used whenever an SNP control card is used. Occurs before the normal execute card for the program segment to which the snapshot applies. (Only one SNP execute card may be used in a test deck.)

DTR  
Execute Card

Must be used whenever a DTR control card is used. Occurs before the normal execute card for the program segment to which the disk trace applies. (Only one DTR execute card may be used in a test deck.) SNP and DTR execute cards are interchangeable, but must occur as the last card or cards before the normal execute card to which they apply.

Autopatch  
Cards

Occur after object program condensed cards and immediately before any execute cards whether they be:

1. SNP execute
2. DTR execute
3. Program execute (in an overlay)
4. Program end card

### System Preparation

The user must build the Autotest pack with deck A. This deck can be immediately followed by test decks.

If there is to be a special reserved pack for Autotest usage, once the pack is built, deck B can be used to test programs without rebuilding the Autotest pack.

Deck C is a restart deck used to re-establish linkage if the Autotest linkage routine (which loads the core-storage printout program from disk storage into core storage and executes it) is destroyed by the execution of the object program.

### Deck A Operation

1. Punch a disk-label date card with today's date to be inserted between cards 6 and 7 of deck A. Re-



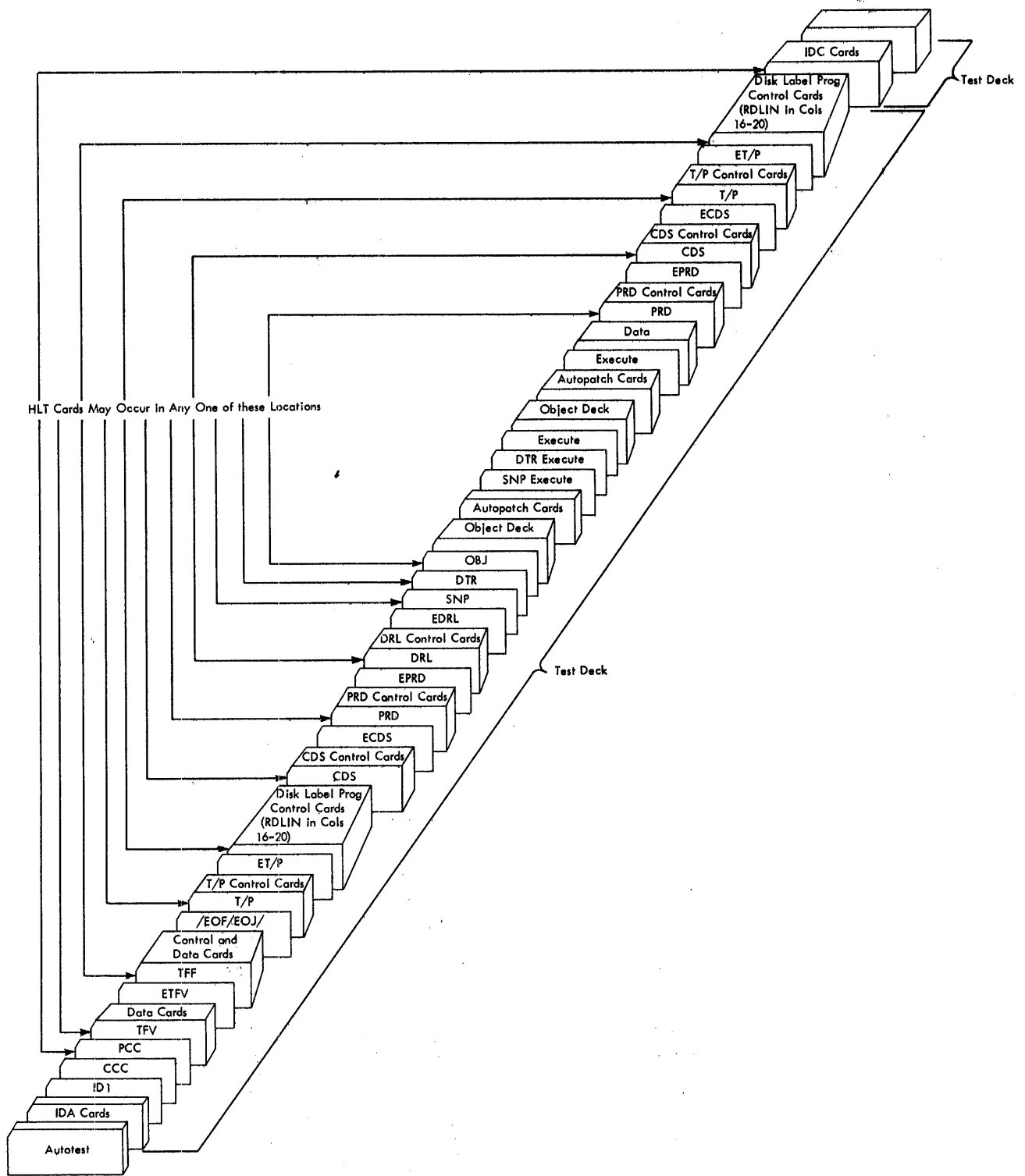


Figure 5. Input Card Sequence Showing All Options

place any previous date card left in deck. This card is punched in the following format:

Columns	Contents
1-3	082
4-5	05
6	word separator (0-5-8)
7-8	year (i.e., 63)
9-11	day of year (001-Jan. 1, 365-Dec. 31)

- Place deck A into the card read hopper followed by the first test deck or decks to be tested.
- Install a pack on drive zero and ready the drive. This pack must have addresses 019000-019979 cleared in load mode referenced to drive zero. The label track is to be addressed as the label track.
- Insert forms on the printer and install an appropriately punched carriage tape.
- Install scratch packs on drives 2, 4, 6, and 8 if available, and ready the drives. These packs should have addressing schemes consistent with the drive on which they are installed.
- Set up scratch tapes on all tape units. Set tapes to high-density, in a ready status, with all file-protect lights off.
- Set sense-switch A on. Set all other sense switches (B, C, D, E, F, G) off.
- Set I/O check-stop switch on.
- Set check-stop switch on.
- Set disk-write switch on.
- Turn write-address key-light off.
- Set address switches to 15909 [7909], press the reset key, the start-reset key, and the program load key.

#### Deck B Operation

- Punch a disk-label date card in the format given in *Deck A Operation*. Insert the date card between cards 6 and 7 of deck B.
- Place deck B into the card read hopper followed by the first test deck or decks to be tested.
- Install and ready Autotest pack previously written by deck A.
- Repeat steps 4 through 12 in *Deck A Operation*.

#### Deck C Operation

- Record all console status conditions.
- Manually print the following areas of storage: 001-100; 101-200; 201-300; 301-400; 15301-15400 [7301-7400]; 15401-15500 [7401-7500].
- Remove the cards from the read hopper and from the normal read stacker.
- Non-process run-out the cards.
- Follow deck C with the cards from step 4 followed by the cards removed from the read hopper in step 3.

- Place the resultant deck in the read hopper.
- Repeat steps 8 through 12 in *Deck A Operation*.

### Autopatch Error Messages

Autopatch cards are checked during processing for the errors noted below. Error messages are printed on the patch listing (see Figure 2) and the error patch is bypassed.

#### Error

Message	Error Made by Programmer
ERR 1	Invalid patch-point address in columns 55-59.
ERR 2	No A or R in column 53 of autopatch card or no Op code in column 61 of an instruction patch card.
ERR 3	Object program patch-point instruction is too small to insert branch linkage to patch area.
ERR 4	Invalid operand columns 62-66 or invalid B-operand columns 68-72.
ERR 5	Source coding error in Autocoder DC or DCW patch card.
ERR 6	Core-storage limits of patch area exceeded. No patch is made.
ERR 7	A 5-character numeric address exceeds 15499 [7499], the last allowable address for core-storage usage with Autotest control.
ERR 8	Length (0, 3, 6, and 9) of the new patch instruction is invalid.
ERR 9	DC or DCW has length indicated as #50 or greater. This is not considered valid in autopatch.

### Halts and Messages During Disk Utility Programs

If a disk error occurs (while clearing disk storage, writing on disk, etc.), the following message is printed:

```
T ERR ADDR X XXXXXX
                drive address
```

T is the type of disk error and can be:

- B = Access busy
- N = Access inoperable
- V = Parity error during read or write operation
- W = Wrong-length record
- X = Unequal address compare

All errors except access inoperable (N) are bypassed and processing continues. Clear-disk-storage and print-disk go on to the next track of the disk area. Disk-record-load and the disk-label program go on to the next card.

A halt occurs for an N ERR. After correcting the disk condition, press START. If the N error cannot be corrected, manually branch to location 15139 [7139] to return control to Autotest.

### Clear Disk Storage

The analysis of each area control card is preceded by the message:

CLEAR DISK

Each area definition is analyzed and printed in the following format:

X XXXXXX-XXXXXX DR-X REF-X CHAR-X TR ERR

This line contains:

1. M or L to indicate move or load mode.
2. The lower and upper limits of the area to be cleared.
3. The disk drive being used.
4. The disk drive to which the addresses on the pack are referenced.
5. The character to be used for clearing.
6. TR if track record format.
7. ERR if there are any of the following errors in the area definition:
  - a. Mode not punched as M or L.
  - b. Lower limit not the zero-sector of a track.
  - c. Lower limit greater than upper limit.
  - d. Drive not 0, 2, 4, 6, 8.
  - e. Reference not 0, 2, 4, 6, 8, \$.

If none of the area definition analyses are flagged with the ERR message, the program goes on to clear the areas defined. If any area is flagged with ERR, the program prints:

\*\*\*OMIT\*\*\*

and reads the next card.

The program checks each area definition prior to processing to see if it occurs within the Autotest area. If so, the program prints:

AUTOTEST AREA IN 019000-019979  
OMIT AREA XXXXXX-XXXXXX

Then, the program goes on to clear the next area.

When the program processes the last area control card, the message:

EOJ CLEAR DISK

is printed and control is turned over to Autotest.

### Print Disk

The analysis of each area control card is preceded by the message:

PRINT DISK

Each area definition is analyzed and printed in the following format:

X XXXXXX-XXXXXX DR-X SC-X WM TR ERR

This line contains:

1. M or L to indicate move or load mode.
2. The lower and upper limits of the area.
3. The disk drive being used.
4. The characters to be substituted for unprintable characters.
5. WM, if word marks are to be printed.
6. TR if track record format.
7. ERR if there are any of the following errors in the area definition:
  - a. Mode not punched as M or L.
  - b. Lower limit not the zero sector of a track.
  - c. Lower limit greater than upper limit.
  - d. Drive is not 0, 2, 4, 6, 8.

If none of the area-definition analyses are flagged with the ERR message, the program goes on to print the area defined.

If any area is flagged with ERR, the program prints:

\*\*\*OMIT\*\*\*

and reads the next card.

When the program processes the last area control card, the message:

EOJ PRINT DISK

is printed and control is turned over to Autotest.

### Disk Record Load

At the beginning of the run, the program prints:

DISK RECORD LOAD

The input card analysis is printed in the following format:

X XXXXXX HO-XXXX LG-XX DR-X SC WM TR ADDR

This line contains:

1. M or L to indicate move or load mode.
2. The address of the sector or track record to be operated upon.
3. HO-XXXX. The high-order position of the data field within the record.
4. The length of the data field (LG-XX).
5. The disk drive being used.
6. SC, if the program is to scan for duplicate addresses and change all sectors on the track with the same address.
7. WM, if a word mark is to be written with the first character in the field.
8. TR, if track-record format.
9. ADDR, if an address is to be changed.
10. The data or address to be inserted.

The program checks each input card for errors and, if any are detected, prints:

ERR OMIT ABOVE CARD

The following conditions are detected as errors:

1. Mode not punched as M or L.
2. The address specified is within the Autotest area.

AUTOTEST PROGRAM IN 019000-019979

is also printed.

3. The field that was to be entered is too long (high-order position of field plus length of field is greater than sector or track record length).
4. The address to be entered is not one of the twenty valid addresses for the track.
5. Drive is not 0, 2, 4, 6, 8.

The program continues to read the next card.

Following an analysis not in error, the program analyzes the setting of the write-address mode-key from the previous disk operation. If the key is set properly, the program continues to process the card. If the write-address mode-key must be changed, the program prints:

TURN WRITE ADDRESS KEY ON/OFF

and halts.

In case of disk error (other than access inoperable), the error is printed and also the message:

OMIT ABOVE CARD

The program continues to read the next card.

When the last input card is processed, the message:

EOJ DISK RECORD LOAD

is printed, and control is turned over to Autotest.

### Disk-Label Program

During the disk-label program processing, the program analyzes the setting of the write-address mode-key from the previous disk operation. If the key is set properly, the program continues to process the next card. If the write-address mode-key must be changed, the program prints the following message and halts.

TURN WRITE ADDRESS KEY ON/OFF

### Set Up Initial Header-Label Track

When a RDLIN card is read with the entry NEW, the program prints:

CREATE LABEL TRACK ON DR-X - SERIAL #XXXXX

If no errors are encountered when the program performs the operation, the program prints:

OK

NOTE: The last track of the pack must contain the address 0X9980 before the program creates the label track.

### Restore Normal Addresses

When a RDLIN card is read with the entry RNA, the program prints:

REMOVE LABEL TRACK ON DR-X

If no errors are encountered when the program performs the operation, the program prints:

OK

### Enter Labels

When a RDLIN card is read with the entry ENTR, the program prints:

ENTER (file identification) ON DR-X  
DELETING LABELS IN XXXXXX-XXXXXX EXP. BY  
YYDDD

Each expired label is deleted and printed following the message:

DELETING . . . . .

This message contains the drive number and the address of the label to be deleted.

After deleting all expired labels, the program prints:

SEARCH COMPLETE

The new label is then written on disk, and printed following the message:

ENTERING . . . . .

This message contains the drive number and the address of the label entered.

### Print Labels

When a RDLIN card is read with the entry PRNT, the program prints:

DISPLAY (file identification)

if a specific header label is to be printed or,

DISPLAY ALL LABELS

if all labels are to be printed.

The printout of each label is preceded by the seven high-order positions of the disk control field used to read the sector.

### Print and Punch Labels

When a RDLIN card is read with either of the entries PNCH or PNCX, the program prints a message indicating the labels to be printed and punched and the format

(standard or expanded `RDLIN`) to be punched. One of the following messages is printed:

```
PUNCH (file identification) ON DR-X 1 CARD/LABEL
PUNCH (file identification) ON DR-X 2 CARD/LABEL
PUNCH ALL LABELS ON DR-X 1 CARD/LABEL
PUNCH ALL LABELS ON DR-X 2 CARD/LABEL
```

At the completion of the run, the last card punched will be the last card in the stacker.

The I/O check-stop switch should be on during this operation. In the event of a punch error, run out the cards in the punch-feed path, and press `START` to re-punch.

#### *Delete Labels*

When a `RDLIN` card is read with the entry `DELT`, the program prints:

```
DELETE (file identification).
```

if a specific header label is to be deleted or,

```
DELETING LABELS IN XXXXXX-XXXXXX EXP BY
YYDDD
```

if all expired labels within specified limits are to be deleted.

Each label found is then printed as read, along with the disk control field used to read the sector.

After deleting all labels specified, the program prints:

```
SEARCH COMPLETE
```

#### *Change Labels*

When a `RDLIN` card is read with the entry `CHNG`, the program prints:

```
CHANGE (file identification)
```

The label is then printed as read, along with the disk-control field used to read the sector, and again as it appears after the change.

After processing the last `RDLIN` card, the program prints:

```
EOJ DISK LABEL PROGRAM
```

and control is transferred to Autotest.

#### *Error Messages During Disk-Label Operations*

If an error, other than the disk operation errors described previously in this section occurs during the execution of the disk-label program, a message is printed. The program does not halt, but instead reads the next card in the read hopper and continues processing accordingly. The messages that can occur and the error that caused the message to print are described in the following paragraphs.

The message:

INVALID CARD

prints if one of the following errors was detected in a `RDLIN` card or a date card.

1. `RDLIN` not punched in columns 16-20.
2. The lower limit is greater than the upper limit.
3. The operation symbol (columns 2-5) is either not valid, or the `RDLIN` card is in the wrong place.
4. The second card of an expanded `RDLIN` entry is missing.
5. The date entered in the date card is either not numeric, or contains a year that is less than 63, or contains a day that is greater than 365.
6. Drive number is not punched in the `RDLIN` card.
7. Drive 0 is indicated in an `RNA` card.

The message:

NOT HDR

should never be printed. This message indicates that during a previous use of the disk pack, a program has written incorrect information in the label track. (The sector read does not have `IHDRb` in the first five positions.) The printing of the message does not indicate that the label track has addresses in a range other than 000180-000199, because an unequal-address-compare would have resulted in this situation.

The message:

NOT EXP

indicates that the program has located an unexpired header label for a file that overlaps the area to be written.

The message:

HDR NOT FOUND

is printed if either a specific header label was requested and none could be found with the same file-identification field, or a header label was not found within the limits specified. When limits are given, the latter situation could occur because the program requires that at least one header label fall within the limits.

The message:

NO ROOM, LAST LABEL READ IS

prints if a header label was to be written in a label track containing 19 unexpired header labels. The message is followed by the information from the last label read.

## **Restarts**

1. If the object program hangs up, the operator should record all console conditions and manually restart at linkage address 15909 [7909] to continue Autotest operation. The storage-print routine follows, if requested.

If this procedure fails to restart the program under Autotest control (if linkage is not re-established) the operator should restart using deck C. (See *Deck C Operation*.)

2. A complete restart procedure is necessary if the Autotest routines on the disk pack have been destroyed. Deck A must be used to recreate the system pack (see *System Preparation*). The next test packet should be processed to prevent recurrence of previous condition.
3. There is also a possibility that user programming may destroy the addressing schemes on one or more packs which will affect their own test run or later test runs. To restore the normal addressing scheme, an IBM customer engineer must be contacted to operate the compare disable switch. Then the Clear Disk program may be run to set up the required addresses on the appropriate pack. Autotest test decks can then be processed with decks A and B dependent on the condition of the system pack.

## **Autotest Halts and/or Messages**

### **Messages Not Accompanied by a Halt**

The messages listed in Figure 6 are printed during an Autotest run to inform the user of program conditions and also to inform the machine operator of the status of an Autotest run. The messages listed are not necessarily in sequence because of varied options of control card ordering or information.

### **Halts and Associated Messages**

All-programmed halts in Autotest are either accompanied by a printed message or identified by a certain defined configuration of the I-address register or of the A-address register. The last two types of identification are limited to some of the error halts.

In the disk-utility programs used with Autotest, all disk-error halts are eliminated, except access-inoperable. An error message is printed and the track in error is bypassed. There are no halts after control-card analyses.

Figure 7 is a listing of the halts and associated messages that can occur during the running of the 1401/1311 Autotest program. The information given for each halt consists of:

1. either the I-address or A-address register setting for halts that do not have an associated message;
2. the message associated with the halt and/or the reason for the halt; and
3. the procedure to be followed by the machine operator.

Message	Explanation
TODAYS DATE XXXXX	This message prints when a date card is loaded with the Autotest deck.
HEADER ROUTINE COMPLETE	Checking of the header label track is complete.
DELETED 0000XXX (Heading Line) (Contents of Label)	This message prints if a label is deleted because it had expired or because of a restart at 333 if it was unexpired. The label that is printed in the message and deleted is located at the address 0000XXX.
IDC (ID1) XXXXXXXXXXX	An image of each IDC or ID1 card read is printed.
PROGRAM CONTROL CARD PCC'XXXXXXXXXXXX	This message prints when a PCC card is read. In addition to a heading line, an image of the PCC card is printed.
NO PROPER LOW-ORDER LIMIT GIVEN	Columns 10-14 of the PCC card are blank or invalid.
SPECIAL RESTRICTED GM REPLACEMENT TO BE USED	The character punched in column 16 of the PCC card is used by the Autotest program during processing as a group-mark replacement character.
WRITE ADDRESS KEY LIGHT ON/OFF FOR OBJECT RUN	The status of the write-address key-light that is required for running the object program (column 15 of PCC card). Autotest causes a halt and prints a message if the setting must be manually changed before executing the object program.
IMPROPER PCC COL 15 ADDRESS MODE KEY ASSUMED IMMATERIAL FOR OBJECT RUN	Column 15 of the PCC is punched, but contains a character other than A, S, 1, or 2. The setting of the write-address key-light is assumed immaterial for running the object program.
STORAGE PRINT-OUT GIVEN AT PROGRAM END	Column 17 of the PCC is blank. Core-storage printout is given after execution of the object program.
NO STORAGE PRINT-OUT	Column 17 of the PCC contains an N.
NO PROGRAM CONTROL CARD, CHECK ASSUMPTIONS MADE	This message prints when a PCC card is not inserted in the test deck. Refer to <u>Program Control Card</u> for the assumptions Autotest makes.
IDC XXXXXXXXXXXXX IDC CARD OUT OF ORDER	This message prints when an IDC card occurs out of order in a test deck. An image of the IDC is printed, the IDC is bypassed, and processing continues with the next card.
XXXXXXXXXX NO RECOGNIZABLE CONTROL CARD, TEST INVALID	A control card, utility call card, utility end card, or an input card is mispunched or out of place in a test deck. Processing continues with the IDC card at the beginning of the next test deck. An image of the error card is printed before the message.
NO IDC CARD, CURRENT TEST INVALID	There is no IDC card with first deck to be tested. Processing continues with next IDC.
NO PCC CARD OR IMPROPER PCC PROGRAM LIMIT NO AUTOPATCHING WILL BE DONE	PCC is absent from the test deck, or address given in columns 10-14 of the PCC is invalid. If any autopatch cards are present, they are not processed.
SNAPSHOT LOCATION WILL BE XXXXX BETWEEN LIMITS XXX01 AND XXX00.	Snapshot indication printed before execution of the object program.
NO ROOM FOR SNP IT IS OMITTED	Snapshot program cannot be executed because the program limit given by PCC columns 10-14 indicates that there is not sufficient core storage available or a DTR card incorrectly preceded the SNP card.
SNP XXXXXXXXXXXXXXX SNP CTL CD ERROR UTILITY IS OMITTED	SNP control card is punched incorrectly. Snapshot is bypassed and processing continues with next card.
EXTRA SNPCARD BYPASSED	More than one SNP control card occurs in the test deck. This card is bypassed.
XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX DISK TRACE WILL OCCUR AT ABOVE LOCATIONS	Indicates as many as seven disk-trace locations given on the DTR control card.
NO ROOM FOR DTR IT IS OMITTED	Disk-trace program cannot be executed; there is not sufficient core storage available.
DTR XXXXXXXXXXXXXXX DTR CTL CD ERROR UTILITY IS OMITTED	DTR control card is punched incorrectly. Disk trace is bypassed and processing continues with next card.
EXTRA DTR CARD BYPASSED	More than one DTR control card occurs in the test deck. This card is bypassed.
NO OBJ CD SEARCH FOR NEXT IDC	OBJ card is missing or out of place in test deck. Program is not executed and processing continues with IDC at the beginning of the next test deck.
AUTOCODER DECK TO BE LOADED	Object program is being loaded.

Figure 6. Messages Not Associated with a Halt (Part 1 of 3)

Message	Explanation
STORAGE PRINT IS COMPLETED AREAS 201-332 AND AREAS 14601 TO 155000 [06601 TO 07500] WERE PRINTED WITHOUT SUBSTITUTION CHARACTERS	This message indicates that the storage print is complete.
PROGRAM PATCH LISTING	Heading of autopatch listing which is printed as autopatch cards are processed.
XXXXXXXXXXXXXXXXXXXX CARD PRINTED WAS OUT OF AUTOPATCH ORDER SEARCH FOR NEXT IDC	Card printed should be an autopatch or an execute card, but is mispunched or out of place in the test deck. Current test deck is bypassed and processing continues with IDC at the beginning of the next test deck.
16 [8] K 1401/1311 AUTOTEST PACK IS BUILT	All deck-A Autotest programs and linkages have been written in disk storage.
BEGIN T/P X	Control has been given to the tape print program. The tape on unit X will be printed.
XXXXXXXXXXXX CORRECT ERROR CARDS AND RETEST PROGRAM	T/P control card error message is preceded by one or more explanatory messages concerning errors. Tape print is bypassed, and processing continues after the ET/P card or when an IDC card is read.
BYPASS T/P MESSAGE XX	The tape-print program has been bypassed, and processing continues after the ET/P card or at the next IDC card. The program had to be bypassed for one of the following reasons that is indicated by XX in the message. 13—Failed to read tape record correctly after 100 attempts, or a wrong-length record was read. 15—Failed to read tape record correctly after 100 attempts with sense switch G on. 30—Tape header label read with file identification other than that in RDLIN card. 31—File-identification field in header label corresponds to that in RDLIN card, but remainder of fields 1-19 and/or user information (positions 101-120) do not correspond. 32—The last card read should have been a RDLIN card but does not have RDLIN in columns 16-20. 33—Tape error occurred while attempting to read a header label. 36—Tape error occurred while attempting to read a trailer label. 38—End of reel.
IN FIELD SELECTION CARD NUMBER OF CHARACTERS TOO LARGE	A field defined in the field-selection card exceeds the length of the maximum print line.
EOJ T/P	T/P program successfully completed.
BEGIN TFV XY	Control has been given to the tape-file generator (variable-length records) program. X indicates the tape drive on which the tape file will be created. Y indicates the parity (B—odd parity; any other character—even parity).
ALPHABETIC INFORMATION IN CC1-4—NOT A TVF OR ETFV—ERROR CARD FOLLOWS XXXXXXXXXXXX	This message indicates that either the ETFV card is missing, or a card is mispunched, or a card is out of order. An image of the card in error is printed. Then, control is returned to Autotest and the program attempts to process the next card.
TAPE RECORD LONGER THAN 13,000 [5000] POSITIONS—FILE ON UNIT X IS INCOMPLETE	The message is self-explanatory. The tape on unit X is tape marked and rewind. Cards are read until an ETFV card or an Autotest control card is read. Then, control is returned to Autotest.
BYPASS TEST—A WRITE TAPE ERROR	An error indication remains after 30 attempts to write a particular record. Cards are read until an IDC card is read. Then, processing continues.
END OF TAPE FILE GENERATOR VARIABLE	The TFV program is complete.
INVALID DTR TAPE SPECIFIED—BYPASS DTR	This message prints if either a tape unit specified was not 1-6, or a tape unit was not specified in the PCC card, or there are no tape drives on line.
NO DA LOADER BRANCH	OBJ card does not have a valid address in columns 4-8. This address is for the loader location. At this location, Autotest inserts a branch to the Autotest loader to handle DA EXEC card branches to the loader and branches to handle overlays.
BYPASS TEST A NON-6 TAPE UNIT CALLED	A TFF, TFV, or T/P call card has specified a tape other than 1, 2, 3, 4, 5, or 6.
BYPASS TEST NO VALID PCC TAPE UNIT	A TFF, TFV, or T/P call card has specified a tape drive not specified in the PCC card.
NO TAPES SPECIFIED	The PCC card is blank from columns 18 to 23.
NO TAPE DIAL CHANGES	On a 6-tape system there is no need to change tapes from the basic 1, 2, 3, 4, 5, 6 tape setting.

Figure 6. Messages Not Associated with a Halt (Part 2 of 3)



Message	Explanation
BYPASS TEST INSUFFICIENT TAPES	The number of tapes specified in the PCC card is greater than the number of tape drives on line.
ZERO TAPE DRIVES SPECIFIED	The operator has indicated no tape drives on line.
BEGIN TFF X	A TFF call card has been read. The file generated is written on tape unit X.
TFF MESSAGE XX	The TFF program and the current test are bypassed to the next test deck because of the following specific TFF message XX condition. 3—Control card error. A complementary message clarifies this situation. 24—Intrarecord sequence error. Either a card is out of sequence or contains a non-numeric value in the sequence field. 25—Interrecord sequence error. A record is either out of sequence or contains a non-numeric value in the sequence field. 27—Incomplete last record. At least one additional card is needed to build the final record in a file of multi-card records. 32—When a RDLIN card was expected, it did not have RDLIN punched in columns 16-20. 33—A tape error occurred while attempting to read a header label. 34—The retention period for the output tape has not expired. 35—A tape error occurred while attempting to write a header label. 36—A tape error occurred while attempting to write a trailer label. 37—30 attempts to write a tape record have failed. 38—End of reel.
IDC FOUND—SKIP TFF	This message prints if either a TFF end card is missing, or an IDC card is out of order, or a data card has IDC in columns 1-3.
EOJ TFF	The TFF program has been completed successfully.
TAPE RD ERROR—BYPASS DTR RECORD	The disk-trace output display at the current tape record is bypassed after 100 unsuccessful read tape attempts.
NO DTR OUTPUT	The DTR program is called for during the object program is called for during the object program run, but there is no output.
END OF DTR OUTPUT	The DTR program output display is concluded.

Figure 6. Messages Not Associated with a Halt (Part 3 of 3)

A-Address Register	I-Address Register	Message and/or Reason	Procedure
		INVALID DATE—CORRECT AND RESTART <u>Reason:</u> the date entered in a load card is invalid. This would result if: 1. the date card is missing, or 2. the date field contains blanks, or 3. the date field contains non-numeric characters, or 4. the year is less than 63, or 5. the day of the year exceeds 365.	Provide a valid date card and restart.
		NOT EXP 0001XX (Heading line) (Contents of a label) <u>Reason:</u> The label that is printed has not expired by today's date. The label is located at 00001XXX.	Manually branch the program to 333 to delete the label and continue processing, or mount a new disk pack and press the start key to process the labels on this pack.
		NOT HDR 00001XX (Heading line) (Contents of header) <u>Reason:</u> This halt can occur only when the pack is being built with deck A or when deck B is being used. The label printed does not have 1HDRb in columns 1-5. The label is located at 0000XXXX.	Manually branch the program to 333 to make the label valid, or press the start key to reprocess the labels in the label track.
		T DSK ERR 0 0001XX (Heading line) (Contents of label) <u>Reason:</u> This halt can occur only when the pack is being built with deck A or when deck B is being used. A disk error of type T has occurred while attempting to read or write the indicated label in the indicated disk location. T is the type of disk error and can be: N—Access inoperable V—Parity or write disk check X—Unequal address compare B—Access busy W—Wrong-length record	Press the start key to reprocess the label track.
b999		A message is not associated with this halt. <u>Reason:</u> A disk error has occurred during deck-A processing when an attempt was made to write the Autotest routines on drive zero.	<ol style="list-style-type: none"> <li>1. Turn the write-address key-light off if it is on.</li> <li>2. Correct the drive zero access if it is inoperable.</li> <li>3. Turn the disk-write switch on if it is off.</li> <li>4. Press the start key to retry the operation an additional nine times.</li> <li>5. If several attempts are unsuccessful and a 1311 malfunction has not occurred, retry the operation with another deck A.</li> </ol>
		ARE THERE 6 TAPE DRIVES ON LINE? IF NO, PRESS START. IF YES, PRESS START RESET THEN START <u>Reason:</u> Autotest must know the number of tapes on line.	Follow the procedure indicated by the message.
		IF ALL CONDITIONS MET, PRESS START <u>Reason:</u> This message is preceded by a series of messages regarding tapes. If all conditions in the messages are met, press the start key to continue.	Perform all required operations and press the start key.
		DIAL TAPE UNITS TO XXXXXX <u>Reason:</u> Manual intervention is needed.	Set the address-select dials on the tape units as specified in the message. Press the start key to continue processing.
		TURN ON SENSE SWITCHES—XXXXXX <u>Reason:</u> Manual intervention is required.	Turn on the sense switches indicated in the message and turn all others off. Press the start key to continue processing.
		HLT XXXXXXXXXXXX <u>Reason:</u> Manual intervention is required.	Perform the operation indicated in the message.
		TURN WRITE ADDRESS KEY LIGHT ON/OFF <u>Reason:</u> Manual intervention is required.	Perform the operation indicated in the message. Press the start key to continue processing.

Figure 7. Halts and Messages (Part 1 of 2)

A-Address Register	I-Address Register	Message and/or Reason	Procedure
	15,805 7,805	A message is not associated with this halt. <u>Reason:</u> A disk-write error has occurred.	<ol style="list-style-type: none"> <li>1. Turn the write-address key-light off if it is on.</li> <li>2. If a halt occurs, press the start key to try the write-disk operation nine more times.</li> <li>3. If a halt occurs, it may be due to a 1311 malfunction. Correct the condition and restart the program.</li> </ol>
	3,203 3,203 3,209 3,209 3,215 3,215 3,221 3,221 15,272 7,272 15,343 7,343 15,973 7,973	A message is not associated with this halt. <u>Reason:</u> A disk-read error under Autotest control.	<ol style="list-style-type: none"> <li>1. Press the start key to try the operation nine more times.</li> <li>2. If the halt reoccurs, the error is due to either a 1311 malfunction, or part of the Autotest pack has been destroyed. Restart the program with deck A and the test deck.</li> </ol>
b8bb		A message is not associated with this halt. <u>Reason:</u> A write-tape error that cannot be corrected has occurred during DTR processing.	<p>Do one of the following:</p> <ol style="list-style-type: none"> <li>1. Replace the tape on which the error occurred (the last tape selected) and retest the current program using Autotest deck B. Testing must start with the IDC card previously read.</li> <li>2. Manually branch the program to 15909 7909 . The object program run is bypassed and the contents of core storage are printed or not depending on the user's specification.</li> <li>3. Push the start key to retry the write tape operation 30 more times.</li> </ol>
	15,765 7,765	A message is not associated with this halt. <u>Reason:</u> The write-address key-light must be turned off to perform the write-disk operations that precede the execution of the autopatch routine.	Turn the write-address key-light off. Press the start key to resume processing.
		LAST CARD—PRESS START CONTINUES TESTING—START RESET—START FOR 80 X 80 LIST <u>Reason:</u> The last card of a test deck has been read. The user is given the options described in the message.	<p>To continue testing:</p> <ol style="list-style-type: none"> <li>1. Place more test decks in the read hopper.</li> </ol> <p>To obtain an 80/80 list:</p> <ol style="list-style-type: none"> <li>1. Place the cards from all stackers (including cards previously removed from the stackers) into the read hopper. (The punch or read mechanisms do not have to be cleared.)</li> <li>2. Press the start-reset key and then the start key to continue processing.</li> </ol>
		END OF AUTOTEST 80 X 80 LIST <u>Reason:</u> Autotest end of job.	Either terminate the run or place more cards in the read hopper and press the start key to continue the 80/80 list.

Figure 7. Halts and Messages (Part 2 of 2)

## Index

Additional Snapshots .....	19	Halt Card (HLT) .....	21
Add Patch .....	23	Halts and Messages .....	30, 34
Autopatch Card Format .....	22	Identification Card (IDC) .....	7
Autopatch Error Messages .....	30	Machine Requirements .....	5
Autopatch Feature .....	21	Object Card (OBJ) .....	21
Autotest Halts and Messages .....	34	Operating Procedures .....	27
Autotest Optional Features .....	8	Patching Rules .....	23
Autotest Setup Instructions .....	27	PDR Control Cards .....	11
Autotest Specifications .....	6	PRD End Card .....	11
Autotest Standard Features .....	6	PRD Print Format .....	11
Branch-Last-Card Modification .....	24	Print and Punch Labels .....	33
CDS Addresses .....	9	Print Disk .....	31
CDS Control Cards .....	9	Print Disk (PRD) .....	11
CDS End Card .....	10	Print Labels .....	32
Change a Label (CHNG) .....	14	Print Labels (PRNT) .....	14
Change Labels .....	33	Program Control Card (PCC) .....	7
Clear Disk Storage .....	9, 31	Program Deck Description .....	27
Clear Storage and Load Routine .....	6	Program Features .....	10
Control Cards .....	15, 18	Punch and Print Labels (PNCH) .....	14
Control-Card Insertion .....	28	RDLIN Card .....	12, 17, 19
Console-Status Card .....	7	Replace Patch .....	24
Core Storage Printout .....	6	Restarts .....	34
Date Card .....	13	Restore Normal Addresses .....	13, 32
DC or DCW .....	25	Segment Mark .....	6
Deck A Operation .....	28	Set Up Initial Header-Label Track .....	32
Deck B Operation .....	30	Set Up Initial Header-Label Track (NEW) .....	13
Deck C Operation .....	30	Snapshot Feature (SNP) .....	6
Delete Labels .....	33	Snapshot Routine .....	6
Delete Labels (DELT) .....	14	SNP Control Card .....	19
Delete Patch .....	24	SNP Execute Card .....	19
Disk-Label Program .....	11, 32	SNP Print Format .....	20
Disk Record Load .....	31	Specifications .....	5
Disk Record Load (DRL) .....	10	Stacker-Identification Cards .....	7
Disk-Trace Feature (DTR) .....	20	Storage Considerations .....	27
DRL Input Cards .....	10	Subroutine Insertion .....	25
DRL End Card .....	11	System Preparation .....	28
DTR Control Card .....	20	Tape-File Generation — Fixed-Length Records .....	14
DTR Execute Card .....	21	Tape-File Generation — Variable-Length Records .....	14
DTR Print Format .....	21	Tape-to-Printer Program .....	17
Enter a New Label (ENTR) .....	13	Tape Utility Programs .....	8
EOF Card .....	16	User Information Card .....	17, 19
Error Messages During Disk-Label Operations .....	33		
Exception Card .....	17, 19		
Field-Selection Cards .....	17, 18		
Final-Halt Modification .....	25		

International Business Machines Corporation  
Data Processing Division  
112 East Post Road, White Plains, N.Y. 10601

## READER'S COMMENT FORM

Autotest for IBM 1401/1460 with IBM 1311, Specifications and Operating Procedures, C24-3195-2

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

- |  | Yes                      | No                       |
|--|--------------------------|--------------------------|
| ● Does this publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Did you find the material:             |                          |                          |
| Easy to read and understand?             | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use?            | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete?                                | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated?                        | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level?        | <input type="checkbox"/> | <input type="checkbox"/> |
- What is your occupation? \_\_\_\_\_
  - How do you use this publication?

As an introduction to the subject?	<input type="checkbox"/>	As an instructor in a class?	<input type="checkbox"/>
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in a class?	<input type="checkbox"/>
For information about operating procedures?	<input type="checkbox"/>	As a reference manual?	<input type="checkbox"/>

Other \_\_\_\_\_

- Please give specific page and line references with your comments when appropriate. If you wish a reply, be sure to include your name and address.

### COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

fold

fold

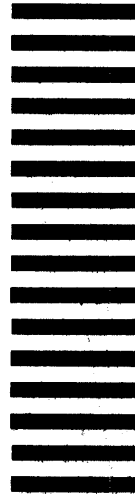
FIRST CLASS  
PERMIT NO. 387  
ROCHESTER, MINN.

**BUSINESS REPLY MAIL**  
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation  
Systems Development Division  
Development Laboratory  
Rochester, Minnesota 55901

Attention: Programming Publications, Dept. 425



fold

fold

IBM401-1311/1460-1311

Printed in U. S. A.

C24-3195-2



International Business Machines Corporation  
Data Processing Division  
112 East Post Road, White Plains, N.Y. 10601