DOSSIER

SOFTWARE REFERENCE MANUAL

National Computer Systems, Inc.

REVISION 9

RELEASE 3.2

October, 1980

Part No. 202-145-355

DOSSIER

SOFTWARE REFERENCE MANUAL

National Computer Systems, Inc.

REVISION 9

RELEASE 3.2

October, 1980

Part No. 202-145-355

REVISION 9: DOSSIER SOFTWARE REFERENCE MANUAL

October, 1980

Revision 9 of this document incorporates technical and editorial corrections. None of the changes reflect software modifications or additions.

Specific changes in the text are indicated by a heavy line in the outside margin. Lines by the page number indicate pagination changes only. Small lines, squares actually, between text lines indicate deletions.

The changes in Chapter Five resulted from a revision of the Self-description Inventory form.

Use the form at the back of this manual to report corrections or suggested changes. Direct correspondence to:

Sentry Publications
National Computer Systems, Inc.
4401 West 76th Street
Minneapolis, Minnesota 55435

Refer to part number 202 145 355 when describing this manual.

P R E F A C E

The Sentry 70 optical mark reader was developed to handle documents which are both meaningful to the user of a form and readable by a machine. The OMR reads pencil marks on documents which are designed in any number of ways to collect and communicate information from people directly. One document provides the means to explain a problem, record the response and provide the response to the computer system. No data transfer steps, such as punching cards, are required and consequently no related data errors are created. Applications for the Sentry are limited only by need and imagination. Use of the NCS form can be equally at home in a kindergarten classroom or hospital records department. And the only special tool required is a #2 pencil.

TABLE OF CONTENTS

Part Number - 202-145-355
June, 1980

INTRODUCTION

PURPOSE

This is a reference manual for the DOSSIER Command Language. The DOSSIER Command Language is a tool for writing programs to process scannable documents. The user-designed program is part of a larger software set which controls the optical mark reader (OMR) system.

THE OPTICAL MARK READER

An NCS Sentry 70 OMR System consists of from 3 to 8 components. The basic system includes a scanner, teleprinter and control cabinet. The control cabinet contains a processor, magnetic tape unit, and cassette tape unit.

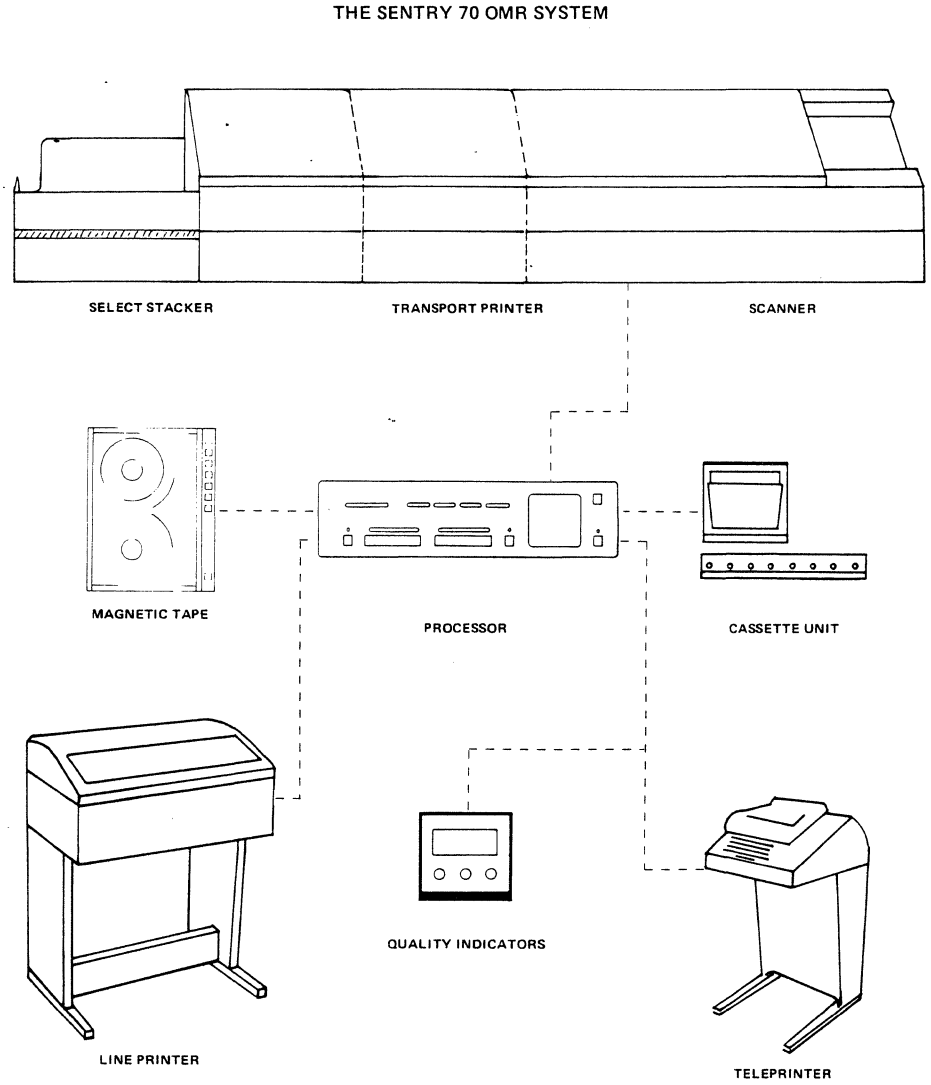


Figure 1-1. Sentry 70 System Components

The scanner reads the documents. The processor instructs the scanner. The teletype allows communication between the operator and the processor. The tape receives and stores the scanner output record. With the basic system this output record must be read and processed by another system.

Refer to The Reference Manual for the NCS Optical Mark Reader System for a complete description of the physical components of the system. An understanding of the hardware components will be of assistance to the user when writing a DOSSIER program.

THE PROCESSOR

The processor requires three software components to process a document:

1. The user-designed program in the DOSSIER Command Language.
2. The DOSSIER Interpreter.
3. SYMON, the system monitor.

The DOSSIER Interpreter and SYMON are considered part of the system and are provided on cassette tapes. The DOSSIER Command Language is the user's tool to prepare programs to process documents.

Figure 1-1 represents the minimum components of the basic OMR system.

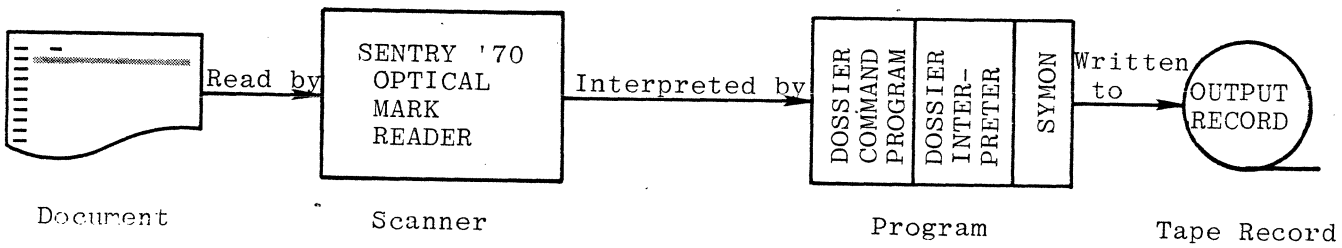


Figure 1-2. The OMR System Schematic

THE DOSSIER COMMAND LANGUAGE

A DOSSIER Command Language program defines how data, read from a document, is to be resolved, edited, and evaluated. In addition, the program defines the format of the output tape record.

The data is written on a 7 or 9 track tape according to the user's equipment configuration.

The Sentry 70 OMR can read marks placed nearly anywhere on a document. DOSSIER can resolve data in any response configuration and from both sides of the document simultaneously.

The DOSSIER Command Language provides the user with eight command groups. These are:

- Execution Linkage
- Identification and Set-up
- Resolution
- Validation
- Data Manipulation
- Arithmetic
- Sequence and Control
- Output

Each command has an associated list of parameters that interface with the DOSSIER Interpreter. At run time, the DOSSIER Interpreter and the user written DOSSIER command program are loaded into storage and executed as one program. Keeping the Interpreter separate from the command program reduces assembly time, removes the need for reserved words, and reduces the amount of storage required for DOSSIER statements.

READING TECHNIQUES

The 7008,7010,7015 and 7020 scanners read responses from a document by detecting the amount of light passing through the page. Light sources above the document and series of photocells under the document are utilized to read responses and to convert each possible response into a value between 0 and 15. When no document is in the read station, no light is blocked and the resulting output is 0. A clean document with no marks would result in a "paper level" output of approximately three from each photocell. Any marks on the document would produce output readings from 4 through 15, depending on the density of the mark.

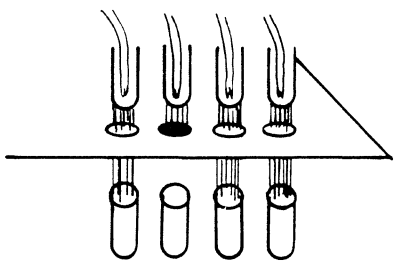


Figure 1-3. Document Under Read Head

Documents are printed on special Trans-optic® bond paper using transparent inks. The high quality paper guarantees uniform "paper level" readings across thousands of documents over extended periods. A series of timing marks printed along one edge of the document indicate when the scanner is to sample a row of responses. If a document has only one mark printed along the timing track, then only one row of responses will be read into the system. However, an 8 1/2" X 12" document could have 69 marks in the timing track, and would therefore read 69 rows of responses into the system. The photocells in the read head are also spaced six per inch permitting a maximum of 47 responses across the width of the 8 1/2" page.

The results is that an 8 1/2" x 12" document contains more than 3000 possible response positions. This permits complete freedom in forms design and virtually any layout to satisfy the user's specific needs.

SYSTEM CAPABILITY

A Sentry 7008 or 7010 scanner permits the user to scan 3000 documents per hour. Document sizes may vary from 3 2/3" x 7 1/2" up to 8 1/2" x 12". A 7015 scanner handles documents up to 11" x 17 7/8" and 13" x 17" and scans at a rate of 6000 per hour. A 7018 scanner handles up to 8 1/2" x 12" forms at a rate of 6000 sheets an hour and a 7020 scanner handles up to 8 1/2 " x 12" documents at a rate of up to 7000 per hour.

When a document is scanned, an image of the document is transferred into memory. A user-written DOSSIER command program resolves the document image, validates the quality of the responses, scores the resolved results, and generates an output tape record containing the resolved and validated data.

Although an 8 1/2" x 12" document contains more than 3000 possible response positions, the user-written program must only define which response positions are to be interrogated. The response positions are interpreted as elements in an item or a row of items, arrays, and grids. The responses are resolved into either the selected response, a multiple, or an omission. After resolution, the data may be edited, tested, and scored against standards established in the program.

OUTPUT TAPE RECORD

An output tape record is generated for each document processed. The record tape may be written on 7-track tapes at either 556 or 800 bpi, or 9-track tapes at 800 bpi or 1600 bpi.

Each tape record consists of fixed front end information plus as many additional characters as the user desires. The fixed information in each record provides positive correlation between the tape record and specific document. Also contained in the first 24 characters are error flags to indicate when a document contains questionable data and an invalid response count to provide a quick indication of the quality of the data on the document.

The Sentry 70 series of products provides the most effective means available for accurately resolving marked responses on a document.

THE NAME

The name DOSSIER is an acronym for DOcument Scanning and Scoring Interpretive Executive Routine.

DOCUMENT DEFINITION

PURPOSE

In order to write a DOSSIER program, the user must be able to recognize basic features of a document. This section describes a document, defines terms and explains how the terms are used to specify locations on a document. It describes how an x,y grid system is used and how information is grouped into items, arrays and grids. The section closes with an explanation of orientation and root response position.

DOCUMENT FEATURES

Every document consists of a number of essential features defined by the terms below. Figure 2-1 displays the features.

Timing Track	The series of short black rectangles running in a straight line along one edge of each document. Also referred to as the y-axis.
Timing Mark	A single short black rectangle in the timing track.
Leading Edge	The edge of the sheet which passes first under the read head. Parallel to the x-axis.
Guide Edge	The guide edge is placed nearest the front when loaded into the OMR hopper.
Trailing Edge	The edge to pass last under the read head.
Front Side	The side facing up when the answer sheet passes under the read head of the OMR. When the sheet is properly oriented with the timing track along the left and the leading edge at the top, the side facing you is the front.
Skunk Marks	Black marks which occupy a few response positions usually at the leading edge of the document. These marks uniquely identify the sheet as to its type.
Response Position	(also bubble or response circle). Any position on the sheet which is capable of being read by a single photocell. Identified by one x,y coordinate.
Bias Bar	A bar of ink, the color in which the document was printed. The bar has no other marks. It runs the width of the document perpendicular to the timing track. The Bias Bar is used to ensure that each photocell screens out marks below the ink level. The scanner reads the bias bar first and uses the "bias level" as a starting point for determining mark intensities on that document.

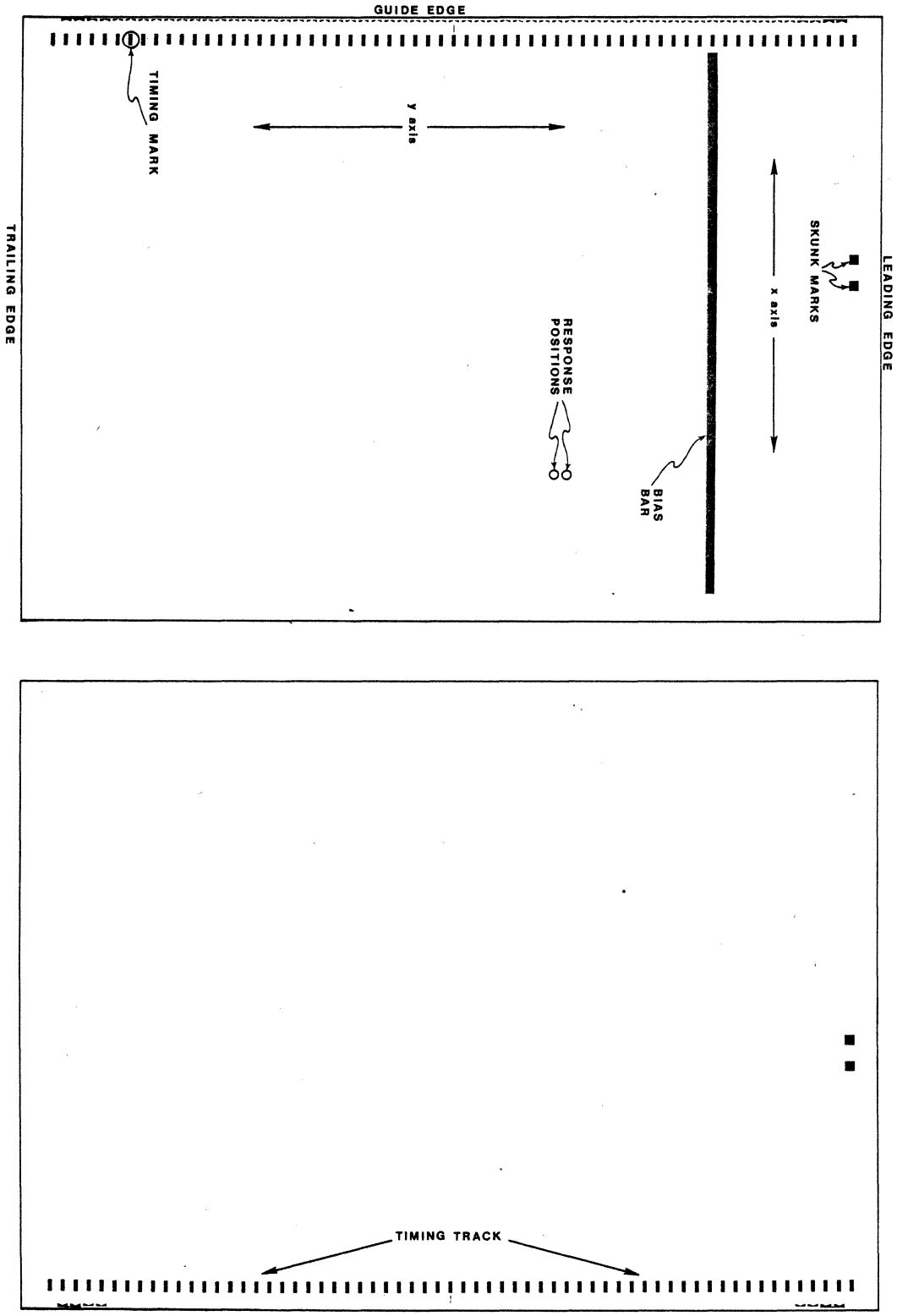


Figure 2-1. Document Front and Back

X,Y AXIS POSITION

Figure 2-1 shows both sides of a document from which the responses have been removed. The front side of a document has the timing marks located on the left side and the skunk marks at the top. On the back side, the skunk marks also appear at the top but the timing marks appear on the right.

The x-axis is positioned by drawing a horizontal line through the top timing mark parallel to the leading edge. The y axis is positioned by drawing a vertical line through the timing track. At the intersection of x and y axis, x has a value of zero and y a value of one. This has been established so that the upper left response position will have a value of one for both the x and y coordinates. Coordinate values are expressed as an x,y number pair. Therefore the intersection of the x and y axis has the coordinates (0,1).

RESPONSE POSITIONS

All response positions are expressed as a positive number and increase as they move away from the x,y axis. Unit spacing across the x axis is always uniform with one unit space equal to the distance between two adjacent photocells; six units per inch.

Unit spacing along the y axis is determined by placement of the timing marks. Therefore, the distance between two adjacent printed timing marks always equals one unit. Spacing between timing marks may not always be uniform but is always considered one unit. Figure 2-2 illustrates how the y coordinate value is determined.

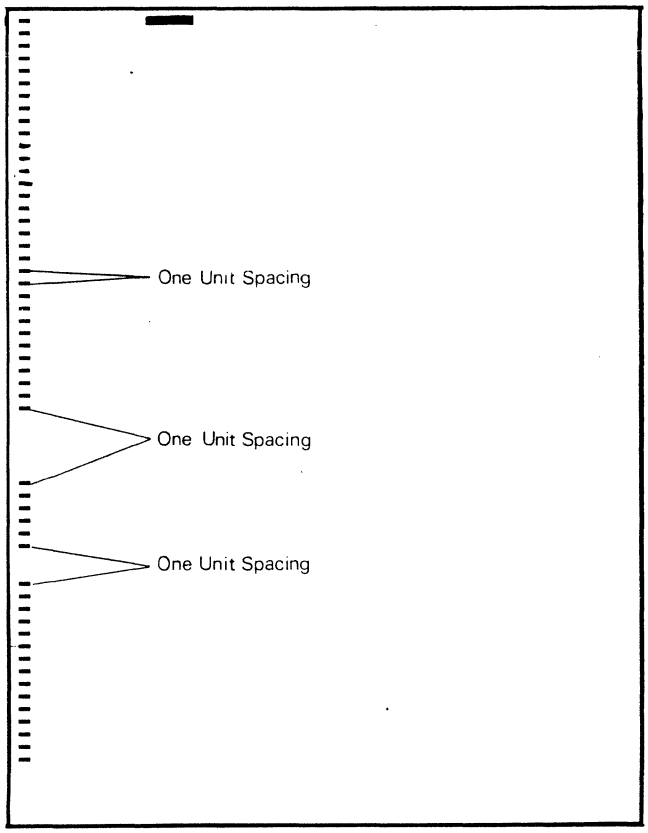


Figure 2-2. Unit Spacing

DOCUMENT FIELDS

Figure 2-3 illustrates various methods used to structure information fields on a document. Shown on the illustration are fields with uniform and nonuniform response placement, fields forming arrays, item columns and alphabetic or numeric grids. DOSSIER selectively analyzes information fields by using the x,y positions to locate each field.

Responses to one given question are either uniform or nonuniform. Uniform/nonuniform refers to the spacing between each response position on both the x and y axes. Uniformity requires identical spacing, any other configuration is nonuniform. Figure 2-4 displays some examples.

A. ORGANIZATION

1. ITEM

An item consists of all responses that can be selected for a given question. Items 2 through 26 in Figure 2-3 are items having uniform spacing. Each item has uniform response placement because the x,y spacing between each response is identical. Item 1 in Figure 2-3 has nonuniform spacing because the x,y spacing between response position 25 through 29, and 30 through 34 is different than the response positions between other responses in the same item.

2. ITEM COLUMN

An item column is formed when a set of consecutively numbered items form one column. The responses appearing in the column must be uniformly spaced. Items 2 through 8, 9 through 15, 16 through 22, 23 through 25, and 26 each form an item column.

3. ARRAY

An array is formed when a set of item columns is uniformly spaced. The number of items in each column must be equal. Items 2 through 22 (see Figure 2-3) constitute an array. All items in the array must be numbered consecutively.

4. GRID

A grid is a form of an item column. Grids allow the system to extract alphabetic, numeric, alphanumeric, or binary information from the grid field. An item in an alphabetic grid has △,A,B.....Z responses (△ represents a blank). An item in numeric grid has 0 through 9 responses. An item in an alphanumeric grid consists of alphabetic responses immediately followed by numeric responses (△,A-Z,0-9). An item in binary grid has 1,2,4,8,16,...,32,768 response position values. A binary grid may not contain more than 16 response positions per item. A decimal grid is a specialized binary grid composed of items of up to four responses. The response positions have the values 1, 2, 4, 8.

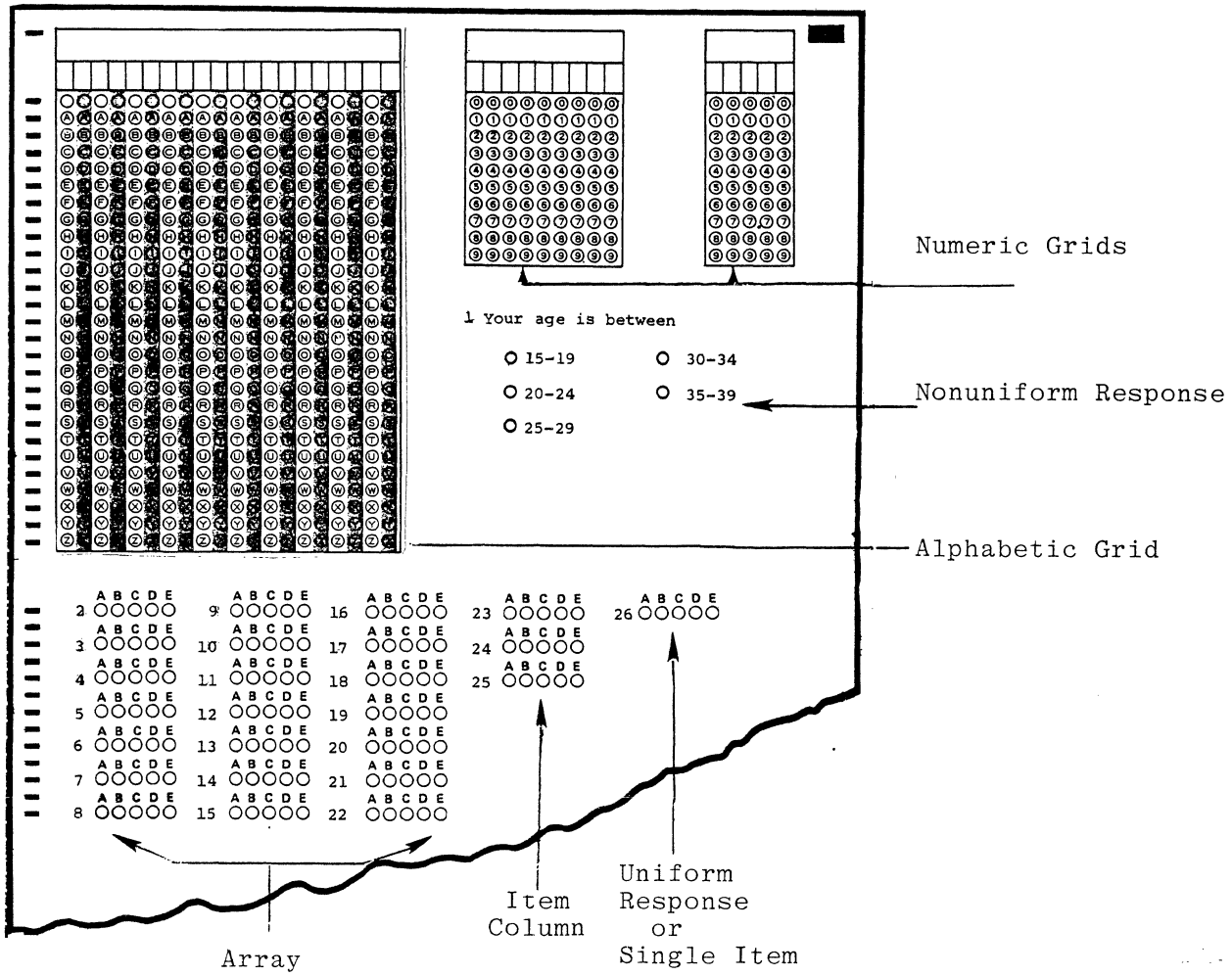


Figure 2-3 Document Fields

Uniform Responses

1. A B C D E

2. A B C D E

3. A-C D-F G-I J-L

Nonuniform Responses

7. The animal is a

tiger camel horse dog cow

8. Beige Aquamarine Navy

Figure 2-4
Uniform/Nonuniform Responses

B. ORIENTATION

Arrays, item columns, or grids, must be either horizontally or vertically oriented on a document. The orientation depends on the orientation of one response in any item in the configuration. The following procedure defines orientation.

Orientation Procedure

STEP	PROCEDURE
1	Orient the document so the leading edge appears at the top of the document.
2	Select any item in either an array or item column.
3	Imagine a straight line through the selected item's response.
4	If the line is vertical, the orientation is vertical; if the line is horizontal, the orientation is horizontal.

1. ITEM COLUMN AND ARRAY EXAMPLES

Figure 2-5 illustrates a section of a properly oriented document. Items 1 through 10 form a horizontal array. Items 11 through 14 form a horizontal item column. Item 40 and 41 and items 42 through 49 form two vertical arrays.

The horizontal array consists of two columns. The first column contains items 1 through 5. The second column contains items 6 through 10. The vertical array containing 40 and 41 consists of two columns with only one item in each column.

Although items 1 through 14 were configured into one array and one item column, the user must not conclude items having similiar symmetry must be configured as illustrated in the example. Items 1 through 5, 6 through 10, and 11 through 14 could each be defined as an item column. However, configuring items into arrays conserves processor storage and decreases processing time.

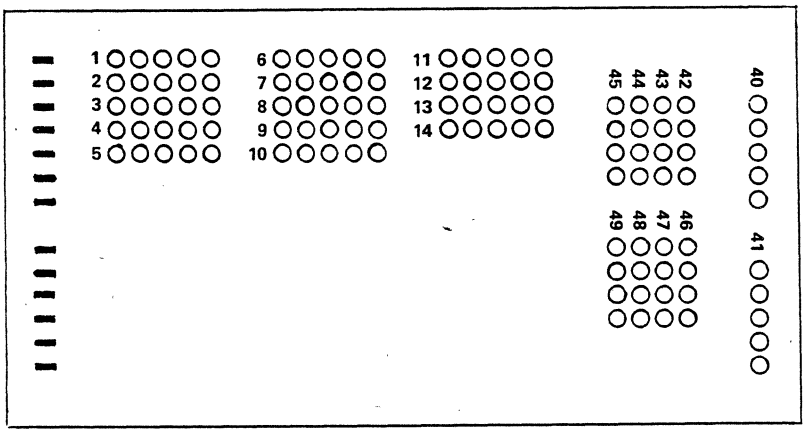


Figure 2-5. Item and Array Examples

2. GRID EXAMPLES

Figure 2-6 shows four types of grids. Both orientations are demonstrated.

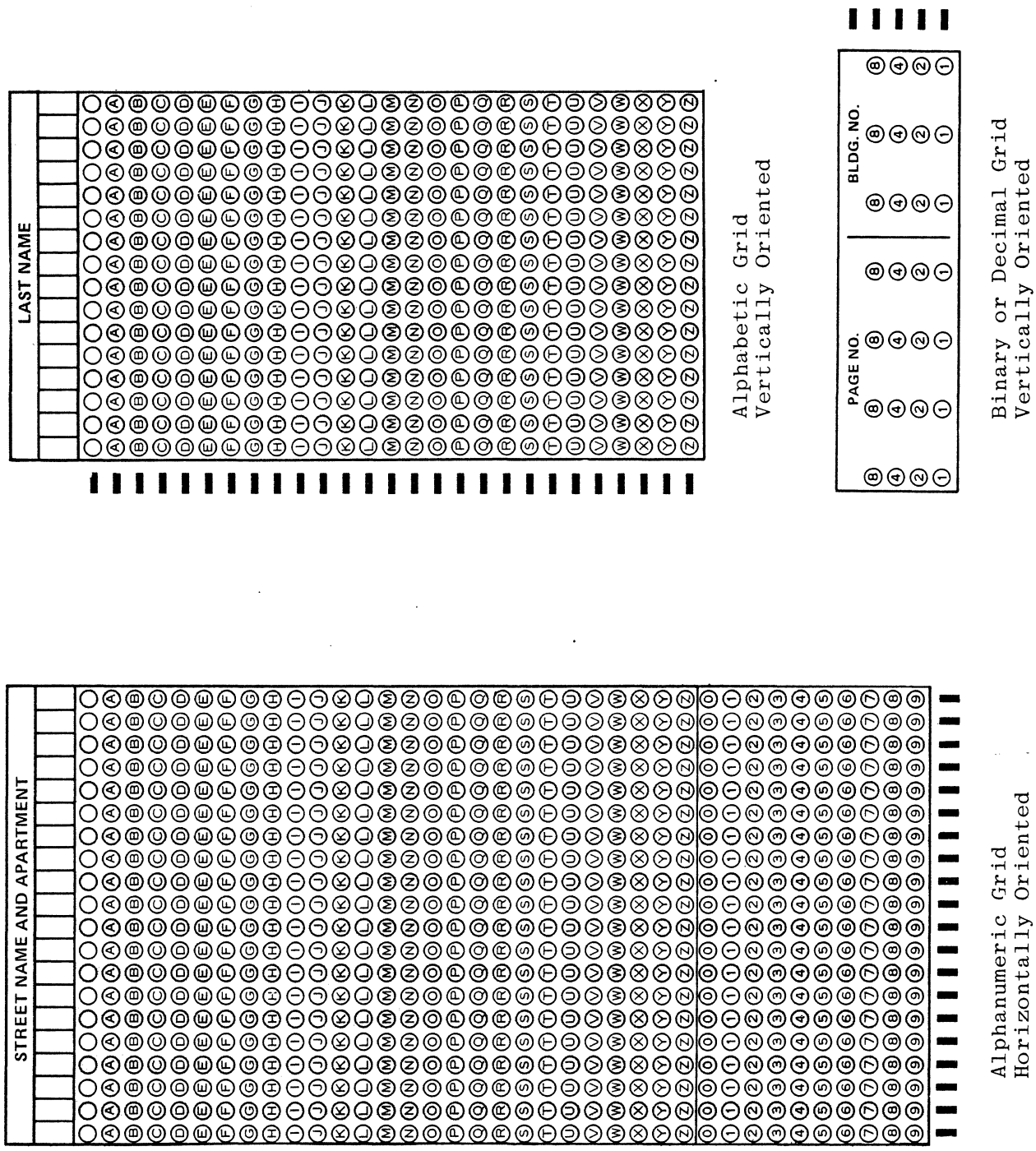


Figure 2-6. Alphabetic, Alphanumeric, Binary and Decimal Grids

C. ROOT RESPONSE POSITION

A root response position references the x,y number pair for the first response position of an item, item column, array, or grid. Figure 2-7 illustrates the root response position for each field on the document.

Binary Grid

The root response position on a binary grid is the least significant response position in the first item to be resolved.

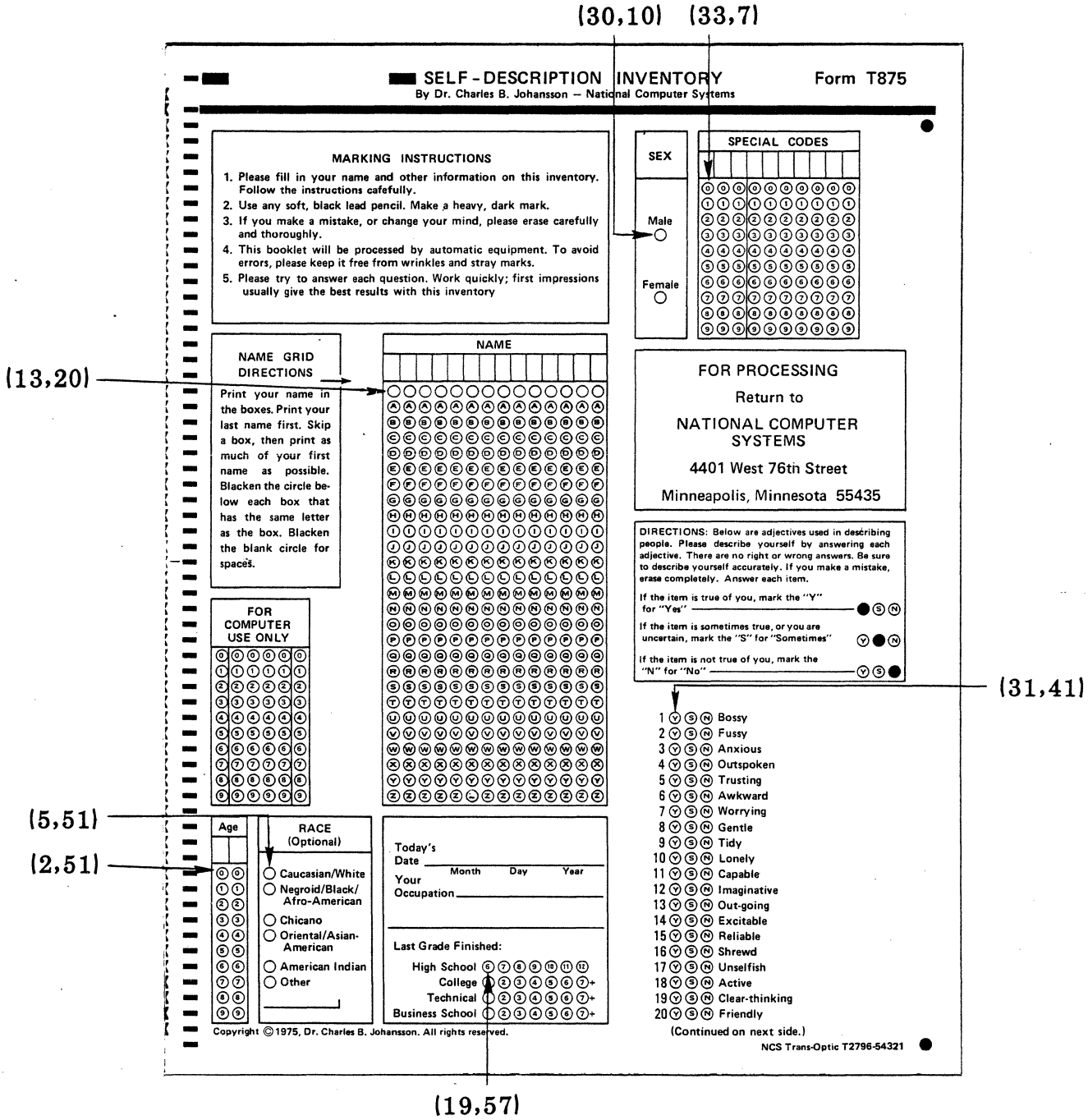
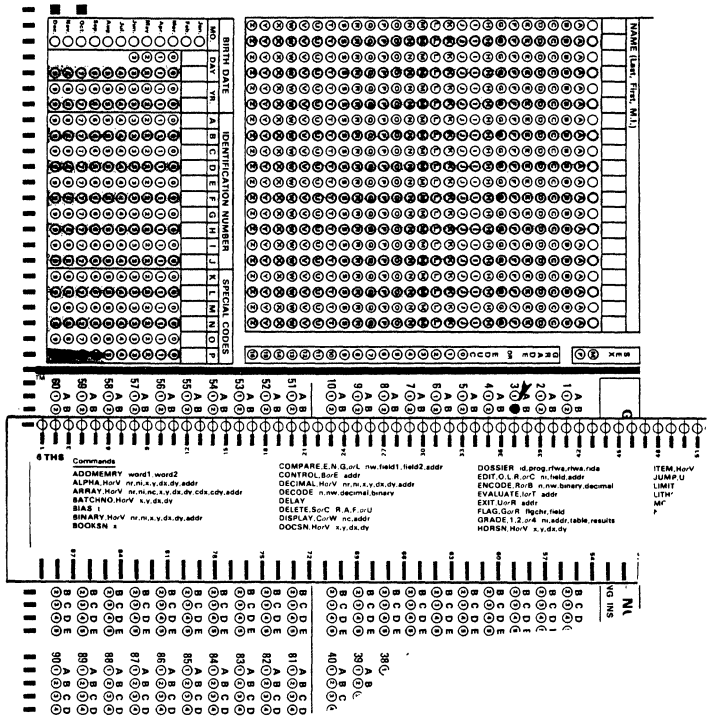
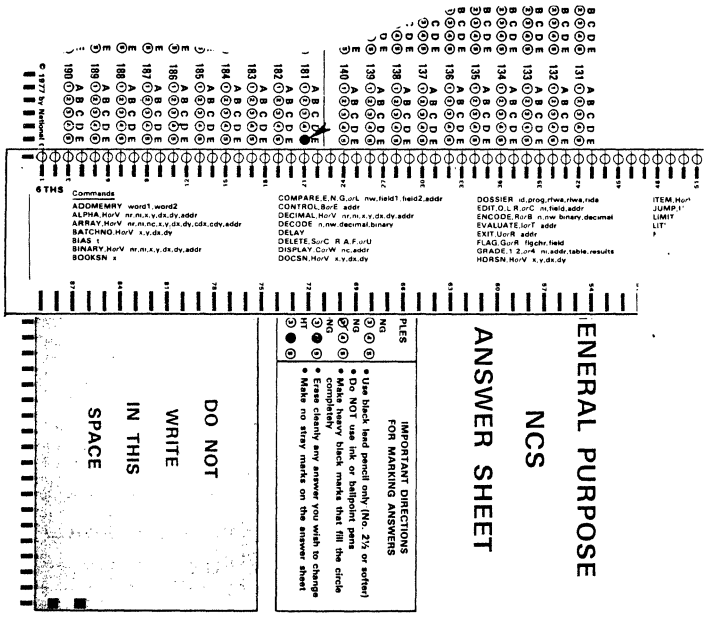


Figure 2-7 Root Response Positions



Front of Form



Back of Form

Figure 2-8 Forms Ruler Determining x Coordinate

The x coordinate for each root response position is located by using the six units per inch scale on the forms ruler. Position the ruler so the zero unit is centered in the middle of the timing mark associated with the root response position. Locate the x coordinate by counting the number of unit spaces to the root response position (see Figure 2-8).

DOCUMENT PROCESSING

PURPOSE

A document is processed when all its response positions have been read and the data resolved, validated, manipulated and written onto tape. This section explains in detail the reading, resolution, validation, and manipulation functions of DOSSIER. In addition, the storage format and output structure of the data are described.

OVERVIEW OF DOCUMENT PROCESSING

The Sentry 70 system hardware reads every x position associated with a timing mark and records in memory a number (0 through F) proportional to the intensity of the mark. This process is referred to as resolution scanning.

The DOSSIER command set may then be used to:

- Resolve selected document fields in memory.
- Total the correct and incorrect number of answers.
- Validate data placement in a field.
- Evaluate the quality of the read.
- Manipulate data into a useful format for continued processing.

Command paths available to the user are displayed in Figure 3-1.

RESOLUTION SCANNING

Resolution scanning performs the following functions:

- Location - selects the document field in memory referenced by the resolution command.
- Mark Discrimination - selects the response value having the darkest mark.
- Storage - generates a sequential list of response values for items appearing in the field.

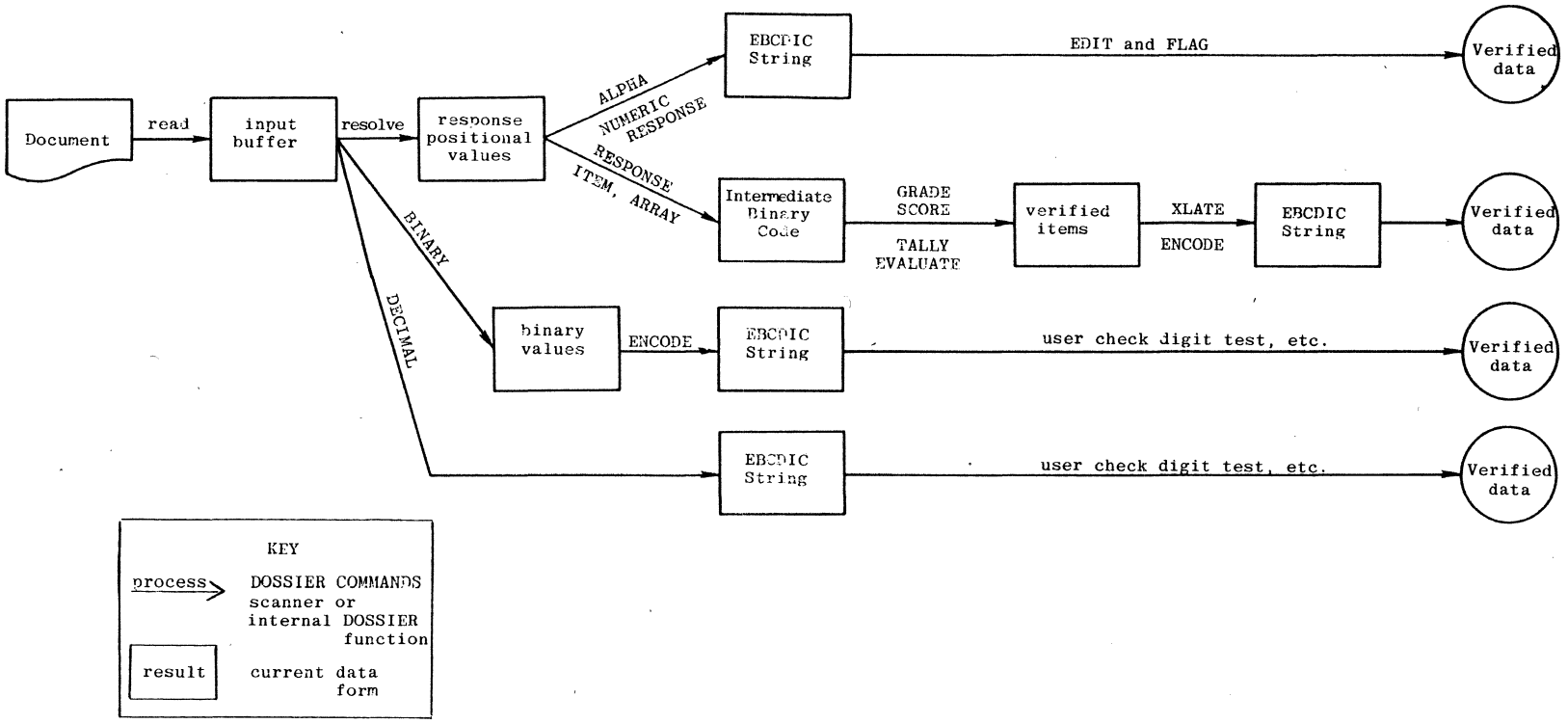


Figure 3-1. Data Resolution, Validation and Manipulation Paths.

LOCATION

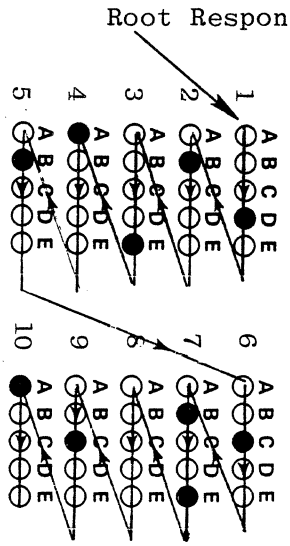
All resolution scanning starts from the referenced field's root response position. The root response positions for all items being scanned are assigned a binary value of one. The remaining response positions for each item are numbered consecutively. The root response position of an alphabetic or alphanumeric grid is assigned a "blank" EBCDIC code and the remaining responses for each item in the grid are assigned A through Z EBCDIC codes if the grid is alphabetic or A through Z; 0 through 9 if the grid is alphanumeric. Each item in a numeric grid has response positions assigned EBCDIC codes of 0 through 9.

Binary grids must not contain more than 16 response positions per item. The first response position is assigned a value equal to 2^0 and the last response position a value of 2^{15}. Binary grids are the only fields that can contain multiple marks in any item.

Array Resolution Scanning

Although arrays have many items, one array resolution command can scan all array responses. Scanning begins by addressing all intensity values associated with the root response item. If a mark is found in one of the response positions for an item, a two character hexadecimal number indicating the positional value for the selected response is transferred to the output record buffer area and inserted in the lower eight bits of the first word of the buffer. If no marks are found, a hexadecimal 00 is transferred to the lower eight bits of buffer storage. If more than one mark is found, a hexadecimal FF is transferred to the lower eight bits of buffer storage. After the root response item has been resolved and the selected response transferred to the lower eight bits of a buffer word, the next item in the same column is addressed and resolved. The resolved results are transferred to the lower eight bits of the next buffer word. When all items associated with a column are resolved and the selected responses are transferred to storage, the successive items in the remaining columns are resolved into selected responses and stored contiguously.

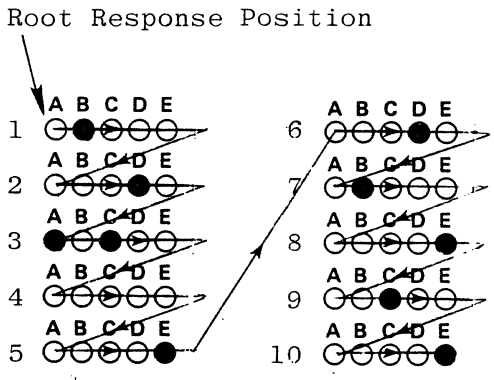
Figure 3-2 shows the response for a two column, five row vertically oriented array. The arrow running through the responses visually demonstrates how DOSSIER scans the array. The block beside the array expresses the buffer contents except for the upper eight bits of each buffer word. The buffer's high order bits will be discussed later in this section. Figure 3-3 illustrates how a horizontally oriented array is scanned.



MEMORY

Location	Contents
WORDN	XX04
WORDN+1	XX02
+2	XX05
+3	XX01
+4	XX02
+5	XX03
+6	XXFF
+7	XX00
+8	XX03
+9	XX01

Figure 3-2
Order of Vertical Array Resolution



MEMORY

Location	Contents
WORDN	XX02
WORDN+1	XX04
+2	XXFF
+3	XX00
+4	XX05
+5	XX04
+6	XX02
+7	XX05
+8	XX03
+9	XX05

Figure 3-3
Order of Horizontal Array Resolution

Nonuniform Response Resolution Scanning

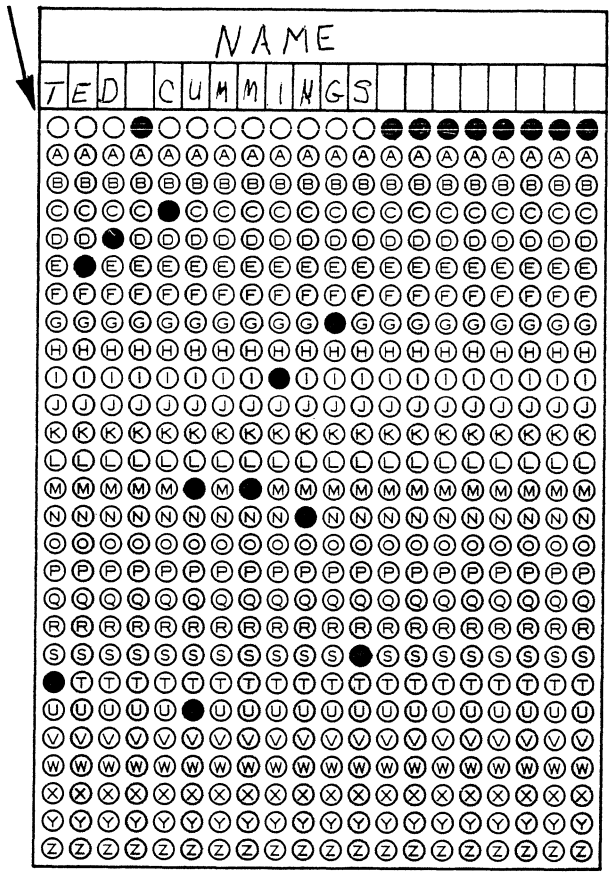
Nonuniform Response scanning is similar to array scanning; however, one resolution command will be required to resolve each set of uniform responses belonging to an item. The first response position resolved is the first position designated by the first resolution command.

Grid Resolution Scanning

Grid resolution scanning first resolves the item having the root response and then resolves the remaining responses of the grid's items. If the mark appears in an item, DOSSIER transfers the marked position's EBCDIC value to the buffer word's lower eight bits. Items containing multiple responses are assigned a 5C (the EBCDIC code for an *) in the lower eight bits of the user defined output record buffer area. Items left blank have a 60 (the EBCDIC code for an -) assigned to the lower eight bits of the output record buffer area.

Figure 3-4 shows the responses selected by Ted Cummings when completing the alphabetic name grid. The block to the right expresses the grid buffer contents. The upper eight bits are again missing and are discussed in a subsequent paragraph.

Root Response Position



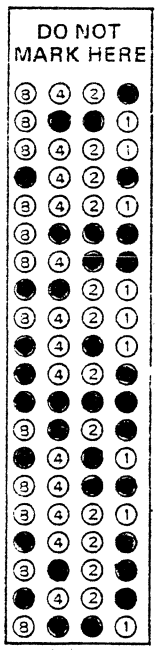
MEMORY

Location	Contents EBCDIC	CHARACTER
WORDN	XXE3	T
WORDN+1	XXC5	E
.	XXC4	D
.	XX40	
.	XXC3	C
	XX5C	*
	XX60	-
	XXD4	M
	XXC9	I
	XXD5	N
	XXC7	G
	XXE2	S
	XX40	
	XX40	
	XX40	
	XX40	
	XX40	
	XX40	
	XX40	
	XX60	

Figure 3-4 Alphabetic Grid Resolution

Binary grids may resolve response positions into one binary value or one EBCDIC number per item. Binary grids that produce binary data must not contain more than 16 response positions per item. Binary grids producing EBCDIC numbers must not contain more than four response positions per item. Binary grid resolution producing EBCDIC data inserts a 5C (the EBCDIC for an *) in the buffer for any item containing a value greater than hexidecimal nine. As before, all resolution scanning starts at the root response position.

Figure 3-5 illustrates a completed binary grid. The decimal buffer contents indicate the resolved results when resolution produces EBCDIC numbers and the binary buffer contents indicate the resolved results when resolution processing produces binary numbers. All 16 bits of each buffer address are illustrated as four hexadecimal numbers.



MEMORY	
DECIMAL	BINARY
XXF1	0001
XXF6	0006
XXF0	0000
XXF9	0009
XXF0	0000
XXF7	0007
XXF3	0003
XX5C	000C
XXF0	0000
XX5C	000A
XXF9	0009
XX5C	000F
XXF5	0005
XX5C	000A
XXF3	0003
XXF0	0000
XXF9	0009
XXF5	0005
XXF9	0009
XXF6	0006

Figure 3-5 Binary Grid Resolution

MARK DISCRIMINATION

Each time the scanner detects a timing mark, one hexadecimal digit is written in memory for each photocell in the read head. The hexadecimal number represents the intensity of darkness detected by a photocell and range from 0 for complete translucense to F for complete opacity. See Figure 3-6.

A response is classified as "light" if the intensity of the mark is at least four levels above the paper level. The response qualifies as a dark response if the intensity level is greater than hexadecimal B. Items with all responses below the "light" level are classified as omissions.

Paper level for a given sheet is determined by the non reading bias bar on a document. When the bias bar is read by the BIAS command, the system averages the light intensities and uses the average as paper level. When the BIAS command is omitted, a paper level of hexadecimal two is assumed and read quality is sacrificed.

The Sentry 70 system software is designed to reliably select a response position in an item that contains more than one marked response position. If a reliable selection cannot be made, the item is flagged as a multiple response. For reliable response selection to be made, one response position in an item must pass one of the following tests:

1) Light Mark Delta - the response position having the darkest mark must be two levels darker than the response position having the next darkest mark if all reported intensities are in the light range or light and dark range.

2) Dark Mark Delta - the response position having the darkest mark must be three levels darker than the response position having the next darkest mark if all reported intensities are in the dark range.

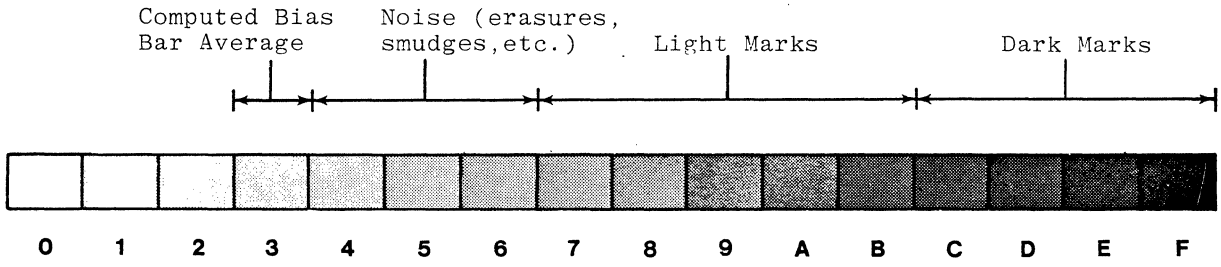


Figure 3-6. Mark Discrimination Levels

NOTE: Occasionally it is useful to raise the light mark threshold when resolving items having only one response position. Raising the light mark threshold reduces the possibility of reading a response position that was poorly erased. The light mark threshold, dark mark threshold, light mark delta, and dark mark delta are located in memory locations A4 through A7 and may be changed by the applications program. Appendix E defines the contents of memory location A4 through A7.

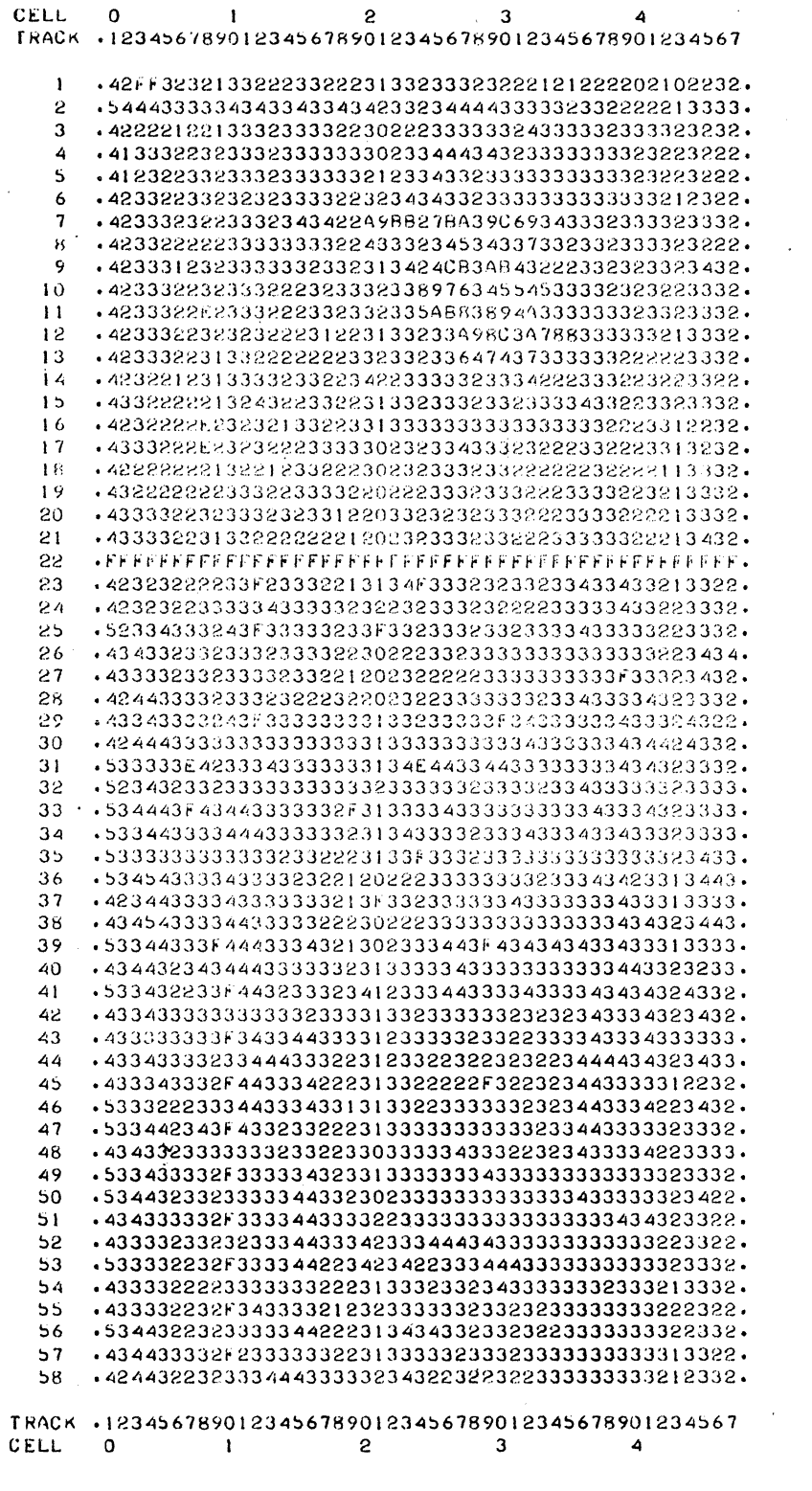


Figure 3-7. Stored Light Intensity Characters

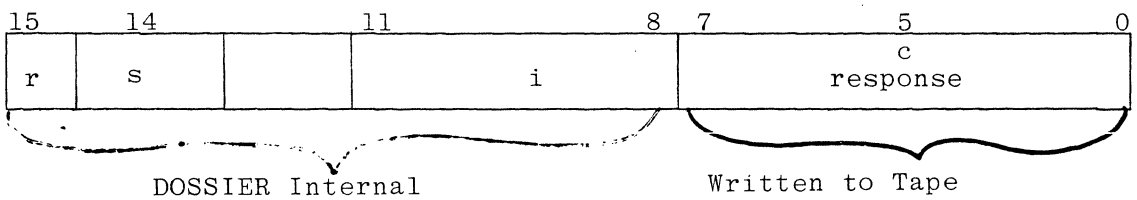
STORAGE

All resolved responses or stored data require one word of storage. When data is transferred from storage to high performance tape, only the information in the lower eight bits is transferred. Therefore, each character written to tape must be stored in one memory word.

The storage area containing the information to be written to tape comprises the output record buffer. The output record buffer is declared by the "rfwa" and "rlwa" parameters of a DOSSIER command. The transfer from the output record buffer to tape occurs when the WRITE command is executed.

Buffer Contents for Resolved Items

Figure 3-8 depicts the contents of a memory word for one resolved item. All resolved items except items in binary grids create words having identical formats. Words created by the resolution of binary grids can use bits 8-15 for resolved data.



r - Bit 15 = one when the resolved response is light.

s - Bit 14 = one when the resolved response was selected from an item having multiple marks.

i - Bits 8-11 = hexadecimal intensity level for the selected response.

c - Bit 5 = one if the response is scored as a correct response by the SCORE command. Not affected by GRADE command.

Figure 3-8 Resolved Item in Memory Word.

Buffer Contents for Other Data

Figure 3-9 depicts the contents of a memory word for other than resolved data. This data can be produced by the system when the processor executes a command that generates data or manipulates resolved data. An example of data generation commands are SCORE, EVALUATE, ENCODE, FLAG, and TRANSLATE.

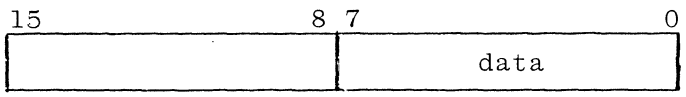


Figure 3-9. Data in Memory Storage

TEST SCORING

DOSSIER SCORE and GRADE commands allow correct answer processing. The number of correct answers is determined by comparing the resolved response to the answers in a right response table. Each time a correct answer is found and the SCORE command is coded, an internal right count accumulator is incremented by one and each word containing a correct response has bit 5 set. GRADE command processing is similiar; however, bit five is not set when a word contains a correct response.

The number of incorrect answers is also accumulated by the GRADE and SCORE commands. The incorrect count only reflects the number of wrong responses and does not reflect the number of omitted and multiple items.

The contents of the right count accumulator are transferred to the output record buffer storage area as a decimal value. The contents of the wrong count accumulator are stored in the DOSSIER Interpretive Routine in binary format. The address of the wrong count accumulator may be found in Appendix E. The contents of the right and wrong count accumulator are set to zero at the initiation of the next SCORE or GRADE command.

The SCORE, GRADE and TALLY commands accumulate the number of multiple responses, omissions, imbedded omissions, light marks and dark marks. Each of the five cumulative totals is stored internally until an EVALUATE command accesses the cumulative total and formats them into a specified storage area. The numbers accumulated can reflect the total counts for several sections of a document.

The imbedded omission count for a resolved response area reflects the total number of unanswered items before the last completed item. The total omission count for a resolved response area reflects the total number of unanswered items. Figure 3-10 shows ten items which contain one imbedded omission and three total omissions.

	A	B	C	D	E		A	B	C	D	E
1	o	●	o	o	o	6	o	o	●	o	o
2	o	o	o	●	o	7	o	o	●	o	o
3	o	o	o	o	●	8	●	o	o	o	o
4	o	o	o	o	o	9	o	o	o	o	o
5	●	o	o	o	o	10	o	o	o	o	o

Figure 3-10. One Imbedded Omission, Three Total Omissions

Right Response Table Construction

Right response tables are used in correct answer processing. They can be constructed to contain one correct answer per word, or packed two or four correct answers per word. The tables are always constructed sequentially with the first answer in the first word of the table. If the table contains two or four answers per word, the first answer must be located in the most significant part of the first word. Figure 3-11 shows how answers appear in consecutive words in memory when a table is constructed to hold one answer per word, two answers per word, or four answers per word.

	15----------8	7----------0
Word 1	X X	Answer #1
Word 2	X X	Answer #2
Word 3	X X	Answer #3
Word 4	X X	Etc.

	15----------8	7----------0
Word 1	Answer #1	Answer #2
Word 2	Answer #3	Answer #4
Word 3	Answer #5	Etc.

	15-------12	11-------8	7--------4	3---------0
Word 1	Answer #1	Answer #2	Answer #3	Answer #4
Word 2	Answer #5	Answer #6	Answer #7	Answer #8
Word 3	Answer #9	Etc.		

Figure 3-11
Right Response Tables Packed One, Two, or Four Per Word.

Right Response Flags

The user may insert into the right response table a hexadecimal FO, Fl, or F2 in place of any correct response. When the system encounters one of these hexadecimal numbers during the grading or scoring of an item, the system either skips marking the item, marks the item as being correct, or marks the item as being incorrect. Figure 3-12 defines the operation performed by each hexadecimal flag.

HEXADECIMAL NUMBERS	OPERATION
FO	Ignore item.
Fl	Accept item as correct.
F2	Accept item as incorrect.

Figure 3-12. Answer Flags

Because right response flags require eight bits of storage, they can only be packed two correct answers per word.

VALIDATION

Validation is checking stored data fields for the presence of omitted or multiply marked responses. In addition, it can mean analyzing for questionably high levels of light marks. The purpose is to enable significant data editing by the scanner before processing by a mainframe.

DOSSIER offers a range of data validation commands, plus the ability to access user prepared assembly language subroutines to augment the validation capabilities. Validation includes completeness and justification checks, mark quality analysis and flags to mark validation failures.

COMPLETENESS AND JUSTIFICATION CHECKS

Individual fields can be tested by EDIT for the presence of omits and multiples. The test can be made on the entire field (complete) or only until an omit is read in a specified direction (left or right justification). One omit within the data or one multiple will cause a failure.

There are a variety of options for handling an edit failure. The action generally depends on the severity of the error.

1. If the error should be corrected before the data goes to the mainframe, the record should not be written to tape. In this case, the error should stop the scanner to allow the operator to correct and reprocess the sheet or the sheet should be selected to the alternate hopper (an equipment option) for later correction.

2. If the error is not detrimental to later processing, but requires notification, a flag can be written in the record and the sheet stacked with the regular stack.

3. Flags in the record can be used in other ways also. Records can be flagged and written while the sheets are sent to the alternate hopper. Those records could be rejected on the mainframe while the record list, with the flags, could be used as an audit trail to correct the selected documents. Flags can be printed directly on the documents if the scanner includes the optional transport printer. In this case sheets can be sent to either hopper.

4. Messages for the operator or to create an audit trail can be displayed on the terminal printer.

FLAGGING THE RECORD

In most cases a record failing any edit checks is flagged. Flagging requires that a FLAG command be executed if the edit check fails. FLAG does three things:

1. Places a user-specified EBCDIC character in the edit flag area of the output record.

2. Places a > in character position 24 of the output record.

3. Places a ? in the record position specified by the program.

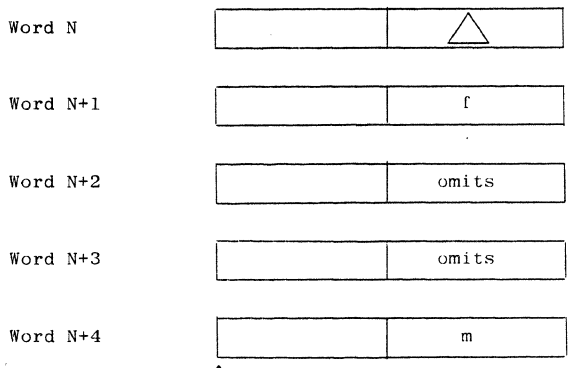
If the record is not to be written to tape FLAG can stop the system and display the error on the terminal printer, enabling the operator to correct it.

The EBCDIC character used as a flag is chosen by the programmer. A data sheet marked with the flags for each edited grid should be available for the operator to enable easy identification of fields in error. Flags can indicate the field, such as N for name, B for birthdate or they can be assigned in alphabetic order in the sequence edited--A for the first field, B for the second and so forth.

MARK QUALITY ANALYSIS - EVALUATE

The TALLY, GRADE, and SCORE commands accumulate the total number of omissions, imbedded omissions, multiple responses, and light and dark responses in a field and store the accumulated counts within DOSSIER. These counts are retrieved by the EVALUATE command, converted to usable EBCDIC data, and placed in five storage locations. These storage locations contain a blank position, a single-digit mark intensity factor, either a two-digit imbedded omissions count or total omissions count, and a single-digit multiple response count. At the completion of the EVALUATE command, the omission and the multiple counts are added to the invalid counter and all five counters accumulated by the SCORE, GRADE, and TALLY commands are reset to zero.

Figure 3-13 shows the format of the assembled characters as they would appear in the output record. Figure 3-14 lists the numeric values, called mark intensity factors, used to express the accumulated percentage of light marks.



Where △ = a blank position.

f = mark intensity factor.

omits = number of imbedded omissions or total omissions. 00-99.

m = number of multiple responses. 0-9.

NOTE: Each character position requires one word of storage and produces EBCDIC coded data.

Figure 3-13. Data produced by EVALUATE and Stored in Memory.

CHARACTER	% OF LIGHT MARKS
-	No light marks
0	0 - 9%
1	10 - 19%
2	20 - 29%
3	30 - 39%
4	40 - 49%
5	50 - 59%
6	60 - 69%
7	70 - 79%
8	80 - 89%
9	90 - 99%
$	100% light marks
#	Section totally omitted

Figure 3-14. Mark Intensity Factors by Percentages.

Applying The Data Produced By Evaluate

Any document's field that has a high % of light marks should be remarked and reprocessed. The average mark intensity factor for a batch of documents can be easily determined by visually inspecting the mark intensity factor on the output record listing. Documents having a high mark intensity factor and high imbedded omission and/or omission count could indicate that many light pencil marks were not scored or tallied. These documents should be remarked and then reprocessed. A high multiple count could indicate the student has placed many stray marks on a document. These documents should be removed, inspected for stray marks, cleaned up, and reprocessed if stray marks exist.

INFORMATION TRANSFER

The lower eight bits of each word stored in the output buffer area is transferred to a seven or nine-channel high-performance tape when the DOSSIER command program executes a WRITE command and the scanner has sensed the end of the document. Every record placed on tape starts with a 24-character fixed field followed by an optional variable length grid edit flag field and fields created by the resolution and validation commands. It is the programmer's responsibility to declare a buffer storage area that contains one word for each character written to tape. Storage is defined by using the BSS pseudo operation code and the DOSSIER command. A description of the BSS pseudo operation code is located in Chapter 4.

Systems having nine-channel tape units and reading and writing EBCDIC code may write binary as well as EBCDIC data. Systems having 7 channel or 9 channel ASCII code units must first translate all data in the buffer to EBCDIC before issuing the WRITE command. The WRITE command automatically translates all data to the appropriate character code as the data is written to tape. After information has transferred to tape, DOSSIER clears the output buffer to blanks and then processes the next document.

TWENTY-FOUR CHARACTER FIXED FORMAT FIELD

Figure 3-16 illustrates the format of the 24-character field. The contents and source of each of the fields is illustrated in Figure 3-17. Some fields are inserted as the command program executes and others are inserted by the WRITE command.

The WRITE command signifies the end of current document processing and the start of processing for the next document.

NOTE: Fields inserted in the output at WRITE time cannot be retrieved by addressing the output buffer area. However, the contents of many fields can be obtained by addressing direct storage locations listed in Appendix E.

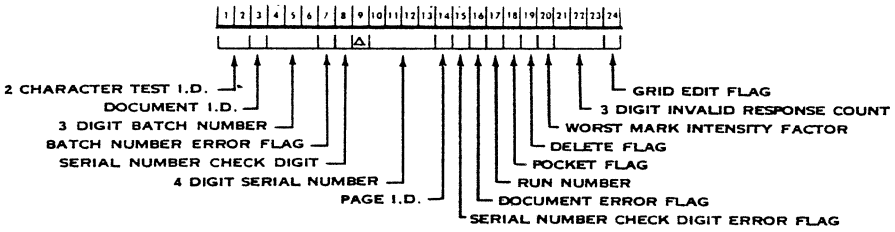


Figure 3-16
24-Character Fixed Format

CONTENTS	1st CHAR POS	CHAR FLD LGTH	SOURCE	REMARKS
Test ID	1	2	DOSSIER command id parameter	
Document ID	3	1	SKUNK command id parameter	If no SKUNK command is coded, this field is blank
Batch Number	4	3	Entered by operator at run time	A mandatory operator entry
Batch Number Error Flag	7	1	> symbol is generated when the resolved batch number grid contains an omission or multiple response	Contains a blank when no errors are encountered
Serial Number Check Digit	8	1	1) Computed internally 2) Read from a header document 3) Read from document when the run number is > 1	
Serial Number	10	4	1) Entered by operator at run time 2) Read from header document 3) Read from document when the run number is > 1	A mandatory operator entry that can be modified by a header document. The serial number is incremented by one after the output buffer is transferred to tape or the data is deleted from tape by the execution of the DELETE command
Page ID	14	1	PAGEID command id parameter	If no PAGEID command is coded, this field is blank

FIGURE 3-17 TAPE RECORD FIELD CONTENTS CONTINUED

CONTENTS	1st CHAR POS	CHAR FLD LGTH	SOURCE	REMARKS
Serial Number Check Digit Error Flag Symbol	15	1	< symbol generated by the system when the serial number check digit read from a document or header document does not compute	This field is blank when no errors are encountered
Document Error Flag Symbols S,L,T,C,U,N, or B	16	1	Symbols generated by the system S-Short document; missing timing marks L-Long document; extra timing marks T-Timing Error C-Content error; data in one or more fields is unacceptable U-Undefined document; the program did not recognize a skunk mark for the scanned document N-Serial number error; serial number grid improperly coded B-Batch number error; batch number grid improperly coded	This field is blank when no errors are encountered
Run Number	17	1	Entered by operator at run time	A mandatory operator entry
Pocket Flag Symbol	18	1	P symbol is generated when a document is diverted to the select stacker	

FIGURE 3-17. TAPE RECORD FIELD CONTENTS CONTINUED

CONTENTS	1st CHAR POS	CHAR FLD LGTH	SOURCE	REMARKS
Delete Flag Symbol	19	1	D symbol is generated when the buffer contents are omitted from the tape	
Worst Mark Intensity Factor	20	1	Indicates the worst mark intensity value found by an EVALUATE command	Fields analyzed by EVALUATE are compared for the one with the highest percentage of light marks.
Three-Digit Invalid Response Count	21	3	The accumulative sum of all omission and multiple counts computed by each EVALUATE command	The internal invalid response counter is reset after output record buffer is written on tape.
Three-Digit Timing Mark Count			Length Error	Shows the actual timing mark count on a LENGTH ERROR
Grid Error Flag Symbol	24	1	> symbol generated by the system any time a field fails an edit test	If no FLAG commands are executed, this position contains a blank.

FIGURE 3-17 TAPE RECORD FIELD CONTENTS

GRID EDIT FLAG FIELD FORMAT

Figure 3-18 depicts the format of the grid edit flag field. This field is omitted if the FLAG command is not used by the command program. However, if any editing is being done, the user must define sufficient output record storage for the edit flags. One storage location must be assigned to each possible edit flag. If the user wants a blank in the output record following the last edit flag, the field size is simply defined as having one additional storage location, and a trailing blank will result in the output record.

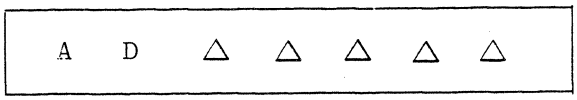
CHARACTER POSITION 25----------n

Where n is any integer value greater than 25.

Figure 3-18. Grid Edit Flag Field

As previously mentioned, grids failing edit tests result in an identifying character being placed in the next available flag field storage location. Figure 3-19 illustrates the flag field for a document where the first and fourth grids failed the edit test. Since the grid edit flag field is seven words long, seven fields on the document could have been edited.

25----------31



Where △ indicates a word containing a blank character position.

Figure 3-19. Flags in a Grid Edit Flag Field

SPACING INFORMATION IN THE OUTPUT RECORD

All data placed in an output record buffer field is stored in the lower eight bits of each word. If the buffer storage field is longer than the inserted data, the lower eight bits of each unused word are blank filled. Users may use blanks to separate data fields in the output record by declaring each field to be longer than necessary.

FIELDS CONTAINING DATA PRODUCED BY SCORE, EVALUATE AND RESOLUTION COMMANDS

Fields created by these commands always appear after the edit grid flag field. Normally, an identifying grid, such as a document's name grid, is the first field in the output record after the grid edit flag field. This facilitates a manual editing process.

TRANSPORT PRINTER

Systems having a transport printer may use instructions to assemble and write data in any one of three zones on a document. All zones are on one line and are located between the timing track and the edge of the document. Zone one is located on the top third of a document, zone two on the middle third of a document and zone three in the bottom third of a document. See Figure 3-20.

The scanning applications program accumulates data in a scribe buffer and the SCRIBE command prints the buffer contents on the document. The scribe buffer may contain up to 32 characters of data and all data must be packed two characters per word.

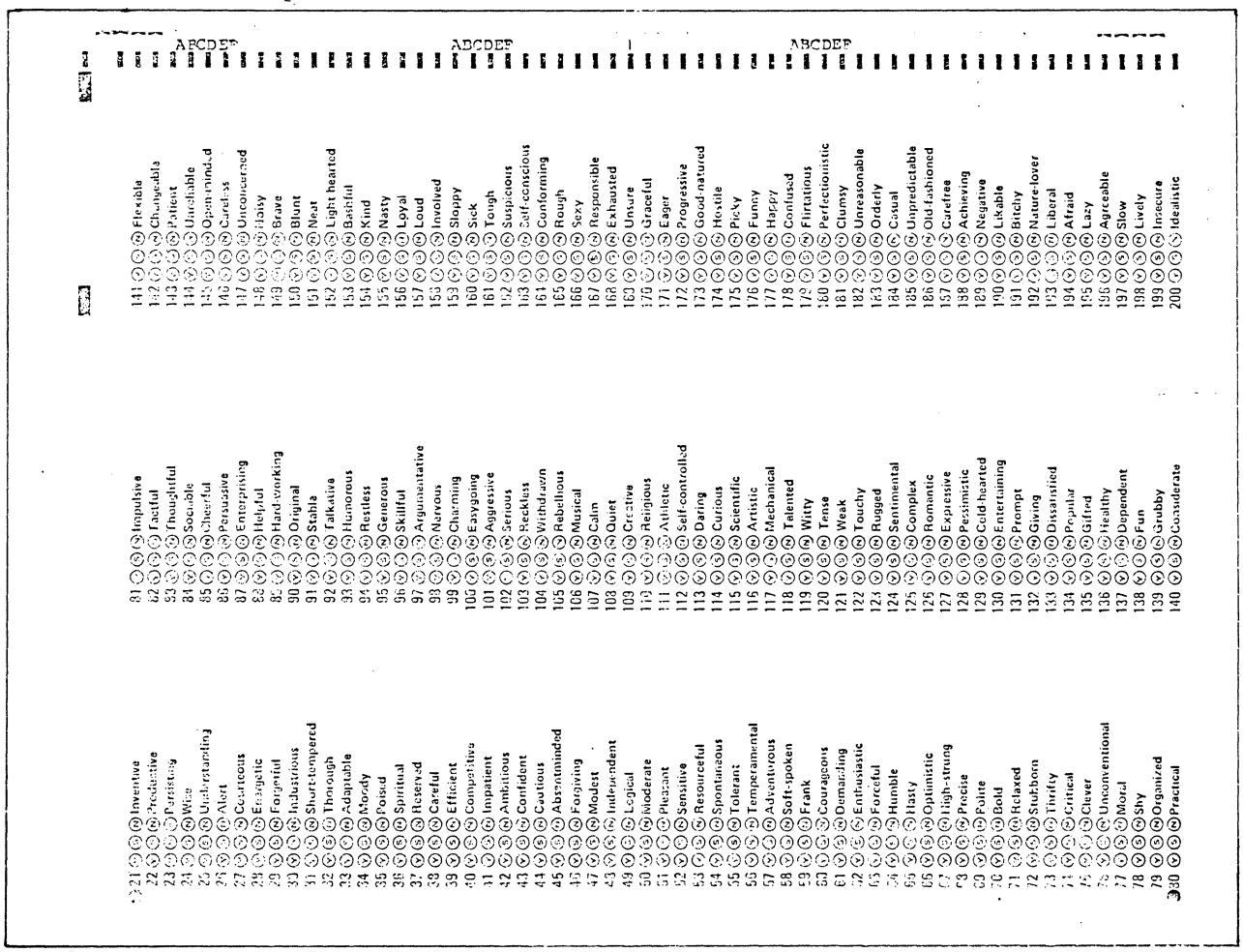


Figure 3-20. Scribed Document.

7018 users only: Because of the rapid scan rate on 7018 systems, the transport printer has time to print only two zones on each sheet. Each zone is 28 characters long instead of 32 characters as on other systems.

DOSSIER COMMANDS

This chapter describes all DOSSIER commands and groups them into eight divisions.

- Execution Linkage
- Identification and Set-Up
- Resolution
- Validation
- Data Manipulation
- Arithmetic
- Sequence Control
- Output

DOSSIER CODING FORM

The following rules indicate how a DOSSIER command should be written.

COLUMNS

Symbolic labels start in column one. The DOSSIER command begins in column ten. Mandatory parameters start in column 24. Comments usually begin in column 45 but may begin after the parameters if a space is left between the parameters and the comments.

SYMBOLIC LABELS

Labels start in column one, must start with an alphabetic character, may contain up to eight characters and may not contain a plus (+), minus (-), equal (=), comma (,), ampersand (&), dollar sign ($), asterisk (*) or blank ( ).

COMMANDS

Commands begin in column ten and must be expressed in upper case letters.

PARAMETERS

Mandatory parameters start in colum 24, are separated by commas and are terminated by a single space.

LITERALS

A literal is an expression for an actual data value. There are three forms for literals in DOSSIER. The three are interchangable as parameters in a DOSSIER command but their stored forms, which depend on the input format, should be a consideration when determining which form to use.

Literals can be expressed as EBCDIC display characters, decimal numbers or hexadecimal numbers. See the chart for specific explanations.

Literal Expression	How to Express It	Stored Form	Literal (Example)	Stored Form
EBCDIC Display Character	= C where C is any EBCDIC character listed in Appendix A. NOTE: an EBCDIC character is a single digit or character. For example =6 is a display character but =16 is not.	EBCDIC code (see Appendix A)	= M = 7 = * = = -	00D4 00F7 005C 0040 0060
Decimal Number	Standard decimal number	Binary form of decimal number - displayed in hexadecimal	1 10 26 100	0001 000A 001A 0064
Hexadecimal Number	Hexadecimal number preceded by zero (0)	Binary form of hexadecimal number - displayed as entered	01 010 087 0100	0001 0010 0087 0100

CHART 4-1. Literals

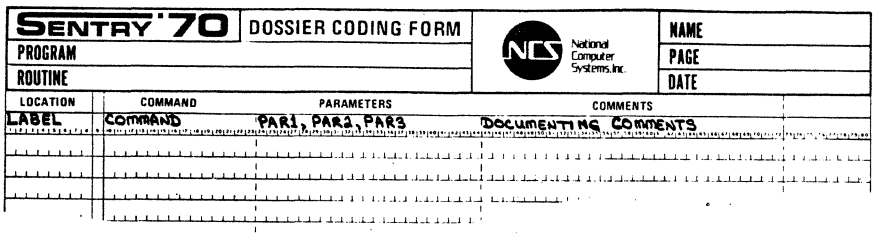


Figure 4-1. Coding Format for DOSSIER Commands

ILLUSTRATION FORMAT

The format for each command appears first as a model with all variants, then as an example. The following points indicate how the illustrated notation should be interpreted.

1. If a command has several variants, all variants are included in a }
2. Optional modifiers and accompanying parameters are enclosed in [ ]
3. Required parameters are listed in order, in lower case letters.

```
FLAG,G  }
FLAG,R  }       flgchr,field

FLAG,R          D,DATE

                ***

PAGEID[,S]      id

PAGEID          1
```

Figure 4-2. Example Illustrations

DOSSIER

EXECUTION LINKAGE COMMAND

The DOSSIER command provides linkage parameters between the external coding and the internal interpretive system.

DOSSIER

A required command which must follow immediately after the IDNT command. This command provides linkage parameters between the command program and the interpreter.

```
DOSSIER          id,prog,rfwa,rlwa,rida

DOSSIER          SD,BEGIN,RECB,RECE,LSTNME
```

COMMAND	DESCRIPTION
DOSSIER	Provides linkage between the command program and the interpreter.

PARAMETER	DESCRIPTION
id	Specifies two alphanumeric identification characters that are stored in the output record's test identifier field and are listed on the teleprinter during program initialization.
prog	Specifies the address of the first DOSSIER command that processes the document.
rfwa	Specifies the address of the first word in the output record storage area.
rlwa	Specifies the address of the last word plus one in the output record storage area.
rida	The address of 16 consecutive characters in the output record buffer to be saved when the buffer is written to tape. The buffer is saved when the record is deleted by the DELETE command but not when a red Document Quality Indicator is encountered in error stop mode.[1] The contents of this field are displayed on the terminal printer on an error-stop to indicate to the operator which document was the last one processed. For this reason an easily recognized field, such as name or ID should be used.

[1]A description of error stop mode and document quality indicators found in Appendix C.

Storage Requirements - None. Parameters supplied by this command are stored in the interpreter storage area. The parameters defined by this command program allow the command program to be assembled independently of the interpreter.

NOTE: The alphanumeric characters for the id parameter are EBCDIC characters. See Appendix A.

LIMIT

IDENTIFICATION AND SET-UP COMMANDS

Commands in this section identify the documents being scanned and impose controls to ensure effective processing. Commands listed in this section are:

- LIMIT
- SKUNK
- BIAS
- PAGEID
- POCKET
- DELETE

LIMIT

A mandatory command that specifies the maximum number of photocells and timing marks on a document. If more than one type of document is processed by a command program, one LIMIT command must exist for each document type. If the program specifies more than one LIMIT command, the parameters from the last LIMIT command executed declare the number of photocells and timing marks that appear on the document being scanned. This command performs these functions:

During Assembly

- Is used to generate bounds errors on the assembly listing if an x or y parameter exceeds the declared limit.
- Is used to calculate the size of the document image.

During Execution

- Declares the number of timing marks on each document in the batch.
- Declares the number of photocells to be averaged from the Bias bar to calculate the document paper level.

LIMIT	x,y
LIMIT	43,61

PARAMETER	DESCRIPTION
x	Specifies the number of photocells the document covers. Maximum for a 7010 or 7020 system is 47, whereas a 7015 system could have 60.
y	Specifies the number of timing marks on a document. Maximum for a 7010 or 7020 is 69, whereas a 7015 system could have 105.

Storage Requirements - One word.

CAUTION

During assembly, the parameters of the last LIMIT read in the assembly path are applied to any following resolution. Therefore a resolution subroutine relating to a previously defined LIMIT placed after a different LIMIT will cause a "C" error during assembly if the resolution exceeds the bounds of the most recently assembled LIMIT. The problem is schematically shown below.

EXAMPLE:

LOCATION	COMMAND	PARAMETERS	COMMENTS
A	LIMIT	47,61	
	.		
	.		ROUTINE A
	.		
	JUMP,U	SPECRES	
B	LIMIT	42,55	
	.		
	.		ROUTINE B
	.		
	WRITE	0	
C	LIMIT	40,52	
	.		
	.		ROUTINE C
	.		
	WRITE	0	
SPECRES	ITEM,H	4,5,41,50,1,1,FIELDA	This will cause a "C" (exceeding LIMIT size) assembly error. The original x location is beyond the last LIMIT read, (LIMIT 40,52). Also after resolving three of the items, the y location would exceed 52.
	.		
	.		
	.		

If the LIMIT relating to the subroutine were restated at the subroutine entry point, that LIMIT would apply to the section of code and the assembly would not result in an error.

```
SPECRES     LIMIT      47,61
            ITEM,H     4,5,41,50,1,1,FIELDA
              .
              .
              .
```

SKUNK

SKUNK

Identifies a document by its skunk marks. Checks specified response positions on a document for the dark marks which comprise the skunk marks. Referred to as the skunk test. If the skunk marks are recognizable (all specified response positions have dark marks) then processing is transferred to the address specified by the command. If they are not recognizable, processing continues with the next sequential command, unless SKUNK,U was executed. In that case, control is passed directly to the WRITE command and no further instructions are executed for that document.

SKUNK[,U]	nr,x_1,y_1,·····x_n,y_n,id,addr
SKUNK,U	2,3,1,4,1,D,START

COMMAND	DESCRIPTION
SKUNK	Specifies that program continuation starts at the location designated by the addr parameter if the system recognizes the skunk marks. Otherwise, processing continues with the next sequential instruction.
SKUNK,U	Performs the same function as the SKUNK command when the skunk marks are recognized. When the skunk marks are not recognized, a WRITE command is automatically executed. In Error-stop Mode, the scanner will stop. In Non-stop mode, a U is inserted in position 16 of the output record and all characters after position 24 are blank filled.

PARAMETER	DESCRIPTION
nr	Specifies the number of response positions covered by skunk marks.
x_1,y_1 . . x_{nr},y_{nr}	A number pair specifying the coordinate values for a response position covered by the skunk marks. One number pair is required for each position covered.
id	Specifies one EBCDIC character that is inserted in the third character position of the output record if all defined x,y response positions contain dark marks.
addr	Identifies the address of the next command to be processed if all defined x,y response positions contain a dark mark.

Storage Requirements - Three words plus one additional word for each response position specified by the x,y parameters.

Application

The SKUNK,U variant must be coded as the last SKUNK command in a chain of one or more SKUNK commands.

CAUTION

DOSSIER checks only the specified response positions of the SKUNK command. SKUNK marks in all identified response positions plus any other skunk marks will pass a skunk test. Therefore, when programming a multiple forms program, define the document with the greatest number of skunk marks first continuing to define the documents with decreasing number of marks.

EXAMPLE:

LOCATION	COMMAND	PARAMETERS	COMMENTS
	.	.	
	.	.	
	.	.	
	SKUNK	6,4,1,5,1,6,1,7,1,8,1,9,1,A,DOC.A	If document has skunk marks in these 6 response positions go to DOC.A
	SKUNK	4,4,1,5,1,6,1,8,1,B,DOC.B	If it has these 4 skunk marks go to DOC.B
	SKUNK	2,6,1,9,1,C,DOC.C	If it has these 2 skunk marks go to DOC.C
	SKUNK,U	1,5,1,D,DOC.D	If it has this skunk mark go to DOC.D. If not, the document is unrecognizable.
	.		
DOC.A	.		Process document A
	.		
	.		
DOC.B	.		Process document B
	.		
	.		
DOC.C	.		Process document C
	.		
	.		
DOC.D	.		Process document D
	.		

BIAS

The BIAS command reads the Bias (color) bar, and increases the reliability of the data to be scanned. A paper level for each document is found and the mark discrimination is based on this paper level. BIAS also locates photocells producing false intensity levels.

The BIAS level is determined by averaging the paper (plus ink) intensities of the photocells declared by the LIMIT command. Since this occurs for each document, the average paper opacity and ink intensities are adjusted to allow for shifts in paper and ink levels. (Note: The BIAS command cannot compensate for a single sheet of paper that has a wide range of opacity readings caused by paper non-uniformity.)

BIAS	t
BIAS	3

PARAMETER	DESCRIPTION
t	Specifies the timing mark that contains a Bias bar. This timing mark must not contain response positions or skunk marks.

Storage Requirements - One word.

If BIAS is not used the paper level defaults to 2 which places the light mark threshold at 6.

BIAS must follow LIMIT since limit specifies how many photocells are to be included in the average. It is possible to have one LIMIT command just for the BIAS bar and execute a second LIMIT after the BIAS for the remainder of the sheet. This would be useful if the Bias bar did not extend the full width of the sheet.

Cell Error

BIAS reports any photocell reading out of range on five consecutive sheets. A photocell is out of range if it is zero or more than four above the paper level average. The error halts the system and produces a CELL ERR n (where n is the photocell number) message on the terminal printer. The purpose of the check is to locate calibration failures. The failure can have other causes. For causes and recovery see the DOSSIER section of the operating manual for your system.

NOTE: A response of Y (or CR) to the CONTINUE? Message of CELL ERR n will suppress subsequent reports of a cell error for cell n, which could lead to serious data errors.

Adjusting the Light Mark Threshold

The light mark threshold (paper level +4) is stored in location A4. This value may be changed by the user before resolution of a particularly troublesome area of a form.

For example, if you are consistantly missing light marks in a grid, you could lower the light mark threshold as follows:

```
SUBMEMRY        ØA4,ONE                     Lower Threshold
BINARY,H        8,3,12,16,1,1,BADGRID       Resolve
ADDMEMRY        ØA4,ONE                     Reset Threshold
```

If you were consistently reading smudges, you could raise the light mark threshold in a similar manner.

PAGEID

PAGEID

An optional command that stores the EBCDIC character from the command in the 14th character position of the output record. This command can also control when the serial number will increment. When this command is omitted, the serial number increments after each record is written to tape.

PAGEID[,S]	id
PAGEID	1

COMMAND	DESCRIPTION
PAGEID	Sets the Page ID and allows the serial number to increment each time an output record is written to tape.
PAGEID,S	Sets the Page ID and holds the serial number constant when an output record is written to tape.

PARAMETER	DESCRIPTION
id	Specifies the EBCDIC character to be inserted in the 14th character of the output record.

Storage Requirements - One word.

NOTE: Use of PAGEID does not override a previously executed PAGEID,S command.

Application

This command is used to store the page number of a multipage booklet in the output record and keep the serial number constant for each page of the booklet. The commands below demonstrate how SKUNK commands and PAGEID commands could be processed to assign the same serial number to each sheet in the same booklet. For this routine to function the batch must be structured so that all sheets are in ascending order and all sheets are located with the correct booklet. Each booklet contains four sheets.

EXAMPLE:

```
LOCATION        COMMAND         PARAMETERS

                SKUNK           2,x,y,x,y,1,SHEET1
                SKUNK           2,x,y,x,y,2,SHEET2
                SKUNK           2,x,y,x,y,3,SHEET3
                SKUNK,U         2,x,y,x,y,4,SHEET4
SHEET1          PAGEID,S        1
                  .
                  .
SHEET2          PAGEID,S        2
                  .
                  .
SHEET3          PAGEID,S        3
                  .
                  .
SHEET4          PAGEID          4
                  .
                  .
                  .
                  .
```

POCKET

POCKET

Controls the disposition of documents to the select or normal stacker. The command may be located any place in the execution path, but does not execute until a WRITE command executes. A POCKET,C cancels any POCKET,S with the same parameter which appears before it in the execution path. A document pocketed on a red condition does not stop a scanner running in Error Stop Mode.

This command will execute with no effect on a scanner without a select stacker. Note that this will result in no action on a red error condition.

POCKET,S POCKET,C	R A F U
POCKET,S	R

COMMAND	DESCRIPTION
POCKET,S	Allows a document meeting the specified parameter to be placed in the select stacker.
POCKET,C	Cancels any POCKET,S command with the same parameter appearing in the execution path before the POCKET,C.

PARAMETER	DESCRIPTION
R	Causes POCKET execution when the sixteenth character position in the output buffer is non blank, i.e., FLAG,R, length error, unidentified document, etc. When this condition exists, the Document Quality Indicator signals a red light.[1]
A	Causes POCKET execution when the scanned document fails a FLAG,G test. When these conditions exist, the Document Quality Indicator signals an amber light.[1]
F	Causes POCKET execution when any red or amber conditions exist.
U	Causes unconditional POCKET execution.

Storage Requirement - One word.

Application

(See next page.)

1. For a description of Error Stop Mode and Document Quality Indicator, see Appendix C.

The examples below illustrates POCKET commands in a program.

LOCATION	COMMAND	PARAMETERS
	POCKET,S	A
	.	
	.	
	POCKET,S	R
	.	
	.	
	POCKET,C	F
	.	
	.	
	POCKET,S	U

The first command causes the scanned document to appear in the select stacker when the Document Quality Indicator signals an amber condition.

The second command causes the scanned document to appear in the select stacker when the Document Quality Indicator signals a red condition. Since the first command is still selected, one command, POCKET,S, having an F parameter could have been coded in place of the first two commands.

The third command causes the scanned document to appear in the normal stacker because it cancels the pocketing condition established by command 1 and 2.

The fourth command causes the scanned document to appear in the select stacker and does not depend on the indicator signaled by the Document Quality Indicator.

CAUTION

POCKET,S R or F (if used) should appear in the execution path prior to the SKUNK, U command. Otherwise the POCKET command will not be executed if the skunk mark is not recognized. In that case, an invalid document would be stacked with valid documents in Non-stop Mode.

DELETE

DELETE

Controls the transfer of output buffer data to tape. This command can inhibit the transfer of buffer data or enable the transfer when the execution sequence has previously inhibited transfer. The command may be located any place in the execution path but does not execute until the WRITE command executes. A DELETE,C command cancels a DELETE,S command with the same parameter appearing before DELETE,C in the execution path. Data not transferred from the buffer to tape is lost unless it is moved out of the buffer storage area before the WRITE command executes.

DELETE,S DELETE,C	{ R A F U }
DELETE,S	R

COMMAND	DESCRIPTION
DELETE,S	Allows the system to omit writing an output record according to the specified parameter.
DELETE,C	Cancels any previously executed DELETE,S command having the same parameter and appearing in the execution path.

PARAMETER	DESCRIPTION
R	Causes DELETE execution when the sixteenth character position in the output buffer is non blank, i.e., FLAG,R, length error, unidentified document, etc. When these conditions exist, the Document Quality Indicator signals a red light.[1]
A	Causes DELETE execution when the scanned document fails a FLAG,G test. When these conditions exist, the Document Quality Indicator will signal an amber light.[1]
F	Causes DELETE execution when any conditions specified by the R or A parameter exist.
U	Causes unconditional DELETE execution.

Storage Requirement - One word.

Application

(See next page.)

1. See Document Quality Indicators, Appendix C.

The example below illustrates DELETE commands in a program.

LOCATION	COMMAND	PARAMETER
	DELETE,S	F
	.	
	.	
	DELETE,C	A
	.	
	.	
	.	
	DELETE,S	U

The first command causes the output record to be deleted when the Document Quality Indicator signals a red or amber light.

The second command cancels the A parameter automatically set by the F parameter in the first command. The execution of the second command causes the output record to be deleted when the Document Quality Indicator signals a red light.

The third command causes the output record to be deleted unconditionally.

CAUTION

DELETE,S R or F (if used) should appear in the execution path prior to the SKUNK,U command. Otherwise, the DELETE command will not be executed if the skunk mark is not recognized. In that case, an invalid document will be recorded on tape in Non-stop Mode.

RESOLUTION COMMANDS

The commands in this section are used to resolve items, arrays, and grids. The listed commands are:

- ITEM
- ARRAY
- RESPONSE
- ALPHA
- NUMERIC
- BINARY
- DECIMAL
- LITHOID
- BOOKSN
- HDRSN
- DOCSN
- BATCHNO

Storage

The resolved contents from these commands are stored in binary or EBCDIC display form. If storage is in binary, the contents should be changed to EBCDIC before it is written to tape. Figure 4-3 classifies the resolution commands according to their storage formats.

BINARY	EBCDIC
ITEM	
ARRAY	
RESPONSE	RESPONSE
	NUMERIC
	ALPHA
	DECIMAL
BINARY	
	LITHOID
	BATCHNO
	HDRSN
	DOCSN
	BOOKSN
omit 00 multiple FF	omit 60 (-) multiple 5C (*)

Figure 4-3
Storage Formats

ITEM

ITEM

Resolves items in an item column into a singular binary result per item. The column resolved must have uniform spacing between responses and items. This command may also be used to resolve a single item having uniform responses.

ITEM,H ITEM,V	nr,ni,x,y,dx,dy,addr
ITEM,H	12,1,14,29,-1,0,BDATE

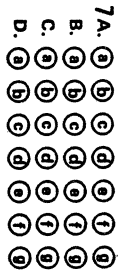
COMMAND	DESCRIPTION
ITEM,H	Specifies that items in the column are horizontally oriented.
ITEM,V	Specifies that items in the column are vertically oriented.

PARAMETER	DESCRIPTION
nr	Specifies the number of responses for any item in the column. This number must not exceed 31 if the resolved responses are to be scored with the SCORE command. Otherwise, the number must not exceed 127.
ni	Specifies the number of items in the column.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The address of the first resolved item. The remainder of the resolved items are stored in contiguous storage locations.

Storage Requirements - Five words:

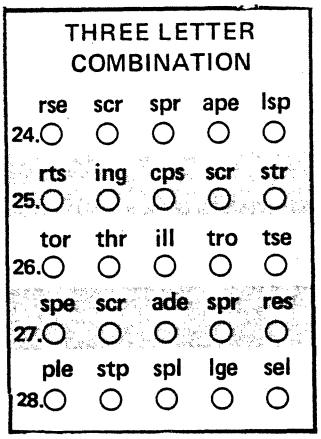
EXAMPLE:

The following are coded as positioned on the document.



EXAMPLE 1:

LOCATION	COMMAND	PARAMETERS	COMMENTS
	.		
	.		
	.		
	ITEM,V	7,4,29,34,-1,1,ASSESS1	Resolution progressed toward the timing track with one space between responses.
	.		
	.		
	.		
	.		
ASSESS1	BSS	4	4 spaces reserved in output record
	.		
	.		
	.		
	.		



EXAMPLE 2:

	.		
	.		
	.		
	ITEM,H	5,5,32,18,2,2,TLC	A horizontally oriented item column with 2 spaces between responses.
	.		
	.		
	.		
	.		
TLC	BSS	5	5 spaces reserved in output record.

ARRAY

ARRAY

Resolves items appearing in an array into a singular binary result per item. The array resolved must have uniform spacing between responses, items, and columns.

ARRAY,H ARRAY,V	nr,ni,nc,x,y,dx,dy,cdx,cdy,addr
ARRAY,H	5,16,5,44,28,-1,2,-9,0,ANS

COMMAND	DESCRIPTION
ARRAY,H	Specifies that items in the array are horizontally oriented.
ARRAY,V	Specifies that items in the array are vertically oriented.

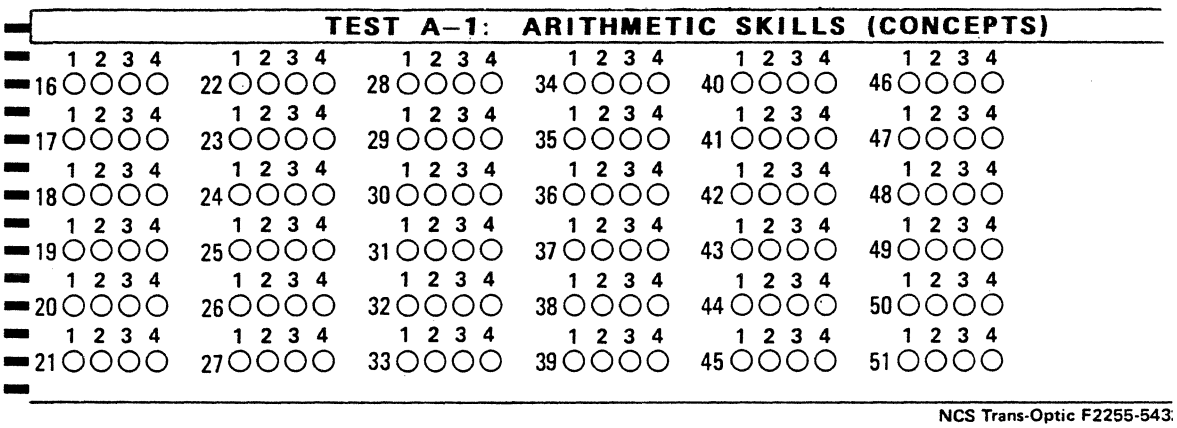
PARAMETER	DESCRIPTION
nr	Specifies the number of responses in each item in the array. The number must not exceed 31 if the resolved responses are to be scored with the SCORE command. Otherwise, the number must not exceed 127.
ni	Specifies the number of items in each column.
nc	Specifies the number of columns in the array.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
cdx	The x coordinate difference between the root response position and the first response position in the next adjacent column. The number selected must be preceded by a minus sign if column to column resolution progresses toward the timing track.

PARAMETER	DESCRIPTION
cdy	The y coordinate difference between the root response position and the first response position in the next adjacent column. The number selected must be preceded by a minus sign if column to column resolution progresses toward the leading edge
addr	The address of the first resolved item. The remainder of the resolved items are placed in consecutive storage locations.

Storage Requirements - Six words.

EXAMPLE:

The array is coded as it appeared on the document.



LOCATION	COMMAND	PARAMETERS	COMMENTS
	.		
	.		
	.		
	ARRAY,H	4,6,6,2,52,1,2,6,0,TESTA1	A horizontal array with 4 responses per item, 1 space between responses in each item and 2 spaces between items. There are 6 spaces between the first response in each column.
	.		
	.		
	.		
	.		
	.		
	.		
	.		
	.		
	.		
TESTA1	BSS	36	36 spaces reserved in output record.

RESPONSE

Resolves a uniform string of response positions. The response positions may comprise one total item or part of a nonuniform item. RESPONSE is used for items which cannot be classified as item columns, arrays or any of the grid types.

If the uniform string of positions comprises one total item, RESPONSE,S is used.

QUARTER:
- ○ FIRST
- ○ SECOND
- ○ THIRD
- ○ FOURTH

Use RESPONSE,S

If the uniform string of positions is part of a nonuniform item, RESPONSE is used, unless it is the final string of positions for the item, then RESPONSE,S is used.

26.

○ A	○ D	○ F			
○ B	○ E	○ G	Use	RESPONSE	(for A-C)
○ C		○ H		RESPONSE	(for D-E)
		○ I		RESPONSE,S	(for F-I)

The resolved result can be stored in binary or EBCDIC format, depending on the "iv" parameter specified.

RESPONSE
RESPONSE,S
} nr,iv,x,y,dx,dy⊏,addr]

RESPONSE 7,1,9,35,-1,0
RESPONSE,S 7,8,9,39,-1,0,GRADE

COMMAND	DESCRIPTION
RESPONSE	This command is used to resolve all uniform response sets except the last uniform response.
RESPONSE,S	Resolves the last uniform response in an item having nonuniform responses, or resolves one item having uniform responses. The S parameter causes the result to be stored in the output tape buffer area.

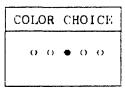
PARAMETER	DESCRIPTION
nr	Specifies the number of responses in the uniform response set. This number must not exceed 31 if the resolved response is to be scored with the SCORE command. Otherwise, the number must not exceed 127.
iv	Specifies the initial value for the string of responses being resolved. The initial value is assigned to the first response in the string. Sequential responses will be assigned consecutive values. This parameter may be expressed in any literal form (reference page 4 - 1). If a decimal or hexadecimal value is to be scored, it must not exceed 31. Otherwise it may not exceed 255. Decimal and hexadecimal initial values generate a "00" for omitted responses and "FF" for multiples. EBCDIC initial values yield a minus (-) for omits and an asterisk (*) for multiples. Zero cannot be used as an initial value with decimal or hexadecimal literals because the binary 00 is used for an omit.
x,y,	An x,y coordinate pair identifying the grid values of the first response in the uniform response set being resolved.
dx	The difference between the x coordinate values for two adjacent responses. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent responses. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr.	This parameter is only used when coding a RESPONSE,S command. The address specifies the storage location of the resolved result.

Storage Requirements - RESPONSE Four words
RESPONSE,S Five words

INITIAL VALUE (iv)

The chart below displays the stored result of resolving the same question with a variety of initial values. The question is resolved with:

```
RESPONSE,S   5,iv,16,10,1,0,CHOICE
```



INITIAL VALUE (iv)	RESPONSE POSITION 3 in STORAGE (in hexadecimal)	OMIT	MULTIPLE
1	03	00	FF
=1	F3	-	*
=A	C3	-	*
=V	E7	-	*
5	07	00	FF
16	12	00	FF
=16	Invalid EBCDIC Character		
=6	F8	-	*

Maintaining Single "iv" Storage Format for Items

Care should be taken when assigning initial values in a RESPONSE string. Since only one item is being resolved, each possible response choice should be stored in the same format, binary or EBCDIC. Therefore each RESPONSE should be assigned the same type of initial value. If a combination of EBCDIC and binary initial values is used, the omits and multiples will appear in the format used for the RESPONSE,S. This is clarified in the next two examples.

EXAMPLE 1: To Resolve Item in Figure 4-4.

```
RESPONSE          4,=1,...
RESPONSE          4,=5,...
RESPONSE,S        1,9,...,MEALS
```

A response of 1-8 will be stored in EBCDIC and the response 9 will be stored in binary. Omits and multiples will appear in binary. It will be difficult to edit or translate this field.

EXAMPLE 2: To Resolve Item in Figure 4-4.

```
RESPONSE          4,=1,...
RESPONSE          4,=5,...
RESPONSE,S        1,=9,...,MEALS [valid]
```

All the responses will be stored in EBCDIC.

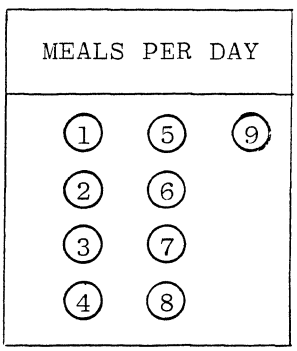


Figure 4-4.

Sequential Values and EBCDIC Display Characters

The "iv" is incremented to position of the marked response. For example if the "iv" is 1 and the response choice marked was five, the value stored would be 5. If the "iv" is 86 and the response choice marked is eight, the value stored would be 94. This is very direct for decimal or hexadecimal numbers.

The concept is less obvious with EBCDIC display characters. The characters do not follow a complete numerical sequence. For example, if an "iv" is =A and the response choice marked was L, the value stored would be 00CC rather than 00D3, which is the EBCDIC code for L. Therefore to capture an alphabetic string the item has to be broken into strings correlating to EBCDIC display characters. The breaks come between I and J, (where the value jumps from C9 to D1), and between R and S, (where the value changes from D9 to E2).

For example consider an item having 26 responses labeled A through Z (such as an alphabetic grid without the initial blank). The code would be:

```
RESPONSE      9,=A,x,y,dx,dy
RESPONSE      9,=J,x,y,dx,dy
RESPONSE,S    9,=S,x,y,dx,dy,ALPHGRID
```

CAUTION

If an item having nonuniform response sets is being resolved, all response sets belonging to the item must be resolved before any other item, array, or grid is resolved.

ALPHA

Resolves an alphabetic or alphanumeric grid into a single EBCDIC character per item. Grid response values must start with a space and proceed in alphabetical order, then zero through nine for the alphanumeric grid.

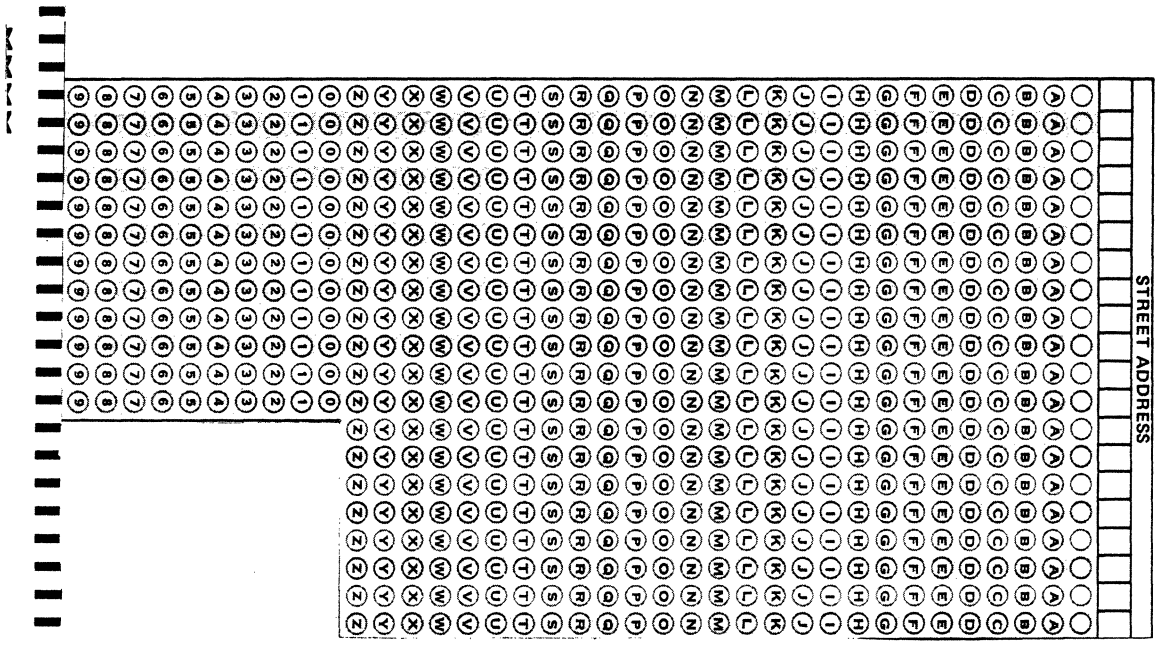
ALPHA,H ALPHA,V	nr,ni,x,y,dx,dy,addr
ALPHA,H	27,6,44,45,-1,2,FSTNAM

COMMAND	DESCRIPTION
ALPHA,H	Specifies that items in the grid are horizontally oriented.
ALPHA,V	Specifies that items in the grid are vertically oriented.

PARAMETER	DESCRIPTION
nr	Specifies the number of responses to be scanned for each item in the grid. This number should be the actual number of responses for the grid, up to 27 for an alphabetic and 37 for an alphanumeric.
ni	Specifies the number of items in the grid.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The address of the first resolved item. The remainder of the resolved items are placed in consecutive storage locations.

Storage Requirements - Five words.

EXAMPLE:



LOCATION	COMMAND	PARAMETERS	COMMENTS
	.		
	.		
	.		
	ALPHA,H	37,12,37,4,-1,1,ADD#	Alphanumeric grid, horizontal, with 37 response positions per item. Scanning progress toward the timing track.
	ALPHA,H	27,8,37,16,-1,1,STREET	Alphabetic grid with 27 response positions per item.
	.		
	.		
	.		
ADD#	BSS	12	12 and 8 spaces reserved in output record.
STREET	BSS	8	

NUMERIC

Resolves a numeric grid into a single EBCDIC character per item.

NUMERIC,H NUMERIC,V	nr,ni,x,y,dx,dy,addr
NUMERIC,H	2,1,15,37,-2,0,SEX

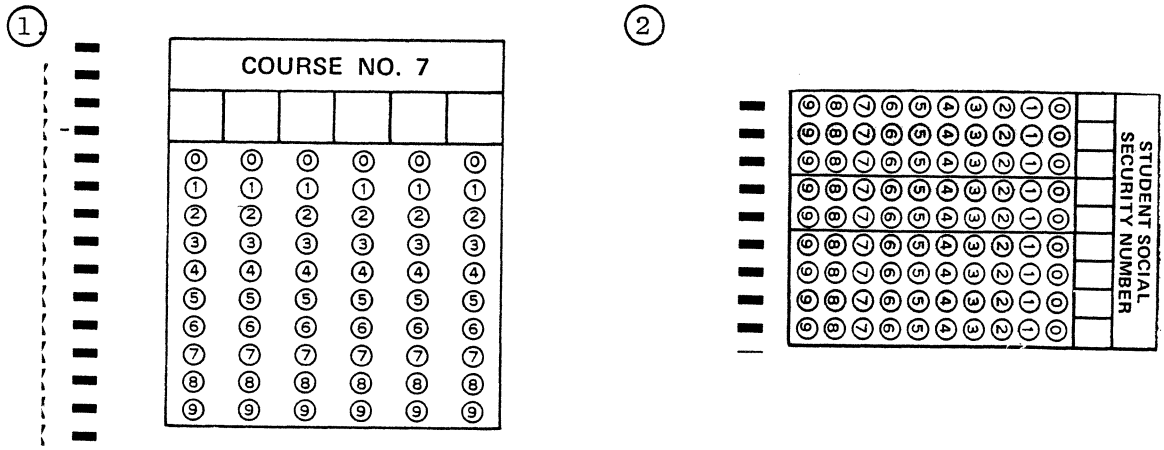
COMMAND	DESCRIPTION
NUMERIC,H	Specifies that items in the grid are horizontally oriented.
NUMERIC,V	Specifies that items in the grid are vertically oriented.

PARAMETER	DESCRIPTION
nr	Specifies the number of responses to be scanned for each item in the grid. The number selected must not exceed ten.
ni	Specifies the number of items in the grid.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate values for adjacent response of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate values for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The address of the first resolved item. The remainder of the resolved items are in consecutive storage locations.

Storage Requirements - Five words.

EXAMPLE:

These numeric grids are coded as they were oriented on the documents.



EXAMPLE 1:

LOCATION	COMMAND	PARAMETER	COMMENTS
	.		
	.		
	.		
	NUMERIC,V	10,6,4,34,2,1,COUR7	
	.		
	.		
	.		
COUR7	BSS	6	6 spaces reserved in the output record.
	.		
	.		
	.		

* * *

EXAMPLE 2:

	.		
	.		
	.		
	NUMERIC,H	10,9,11,17,-1,1,SS	Resolution scanned progresses toward the timing track.
	.		
	.		
	.		
SS	BSS	9	9 spaces reserved in the output record.
	.		
	.		
	.		

BINARY

BINARY

Resolves a binary grid into a singular binary number per item

BINARY,V BINARY,H	nr,ni,x,y,dx,dy,addr
BINARY,V	4,9,45,4,-1,-2,SOCNB

COMMAND	DESCRIPTION
BINARY,H	Specifies items in the grid are horizontally oriented.
BINARY,V	Specifies items in the grid are vertically oriented.

PARAMETER	DESCRIPTION
nr	Specifies the number of responses to be scanned for each item in the grid. This number must not exceed 16.
ni	Specifies the number of items in the grid.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate values for adjacent vertical items or two adjacent responses of a horizontal item. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate values for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The address of the first resolved item. The remainder of the resolved items are placed in consecutive storage locations.

Storage Requirements - Five words.

CAUTION

BINARY and DECIMAL commands read multiple responses as a single value with no mark discrimination by intensity. All marks above the light mark threshold are read as binary ones. Therefore, particular care must be taken when marking binary grids. Marks must be above the light mark threshold to be read. This is a case when it is recommended that a check digit be used with each binary and decimal grid.

EXAMPLE:

512	256	128	64	32	16	8	4	2	1		
o	o	o	o	o	o	o	o	o	o	FIRST	SERIAL #
o	o	o	o	o	o	o	o	o	o	SECOND	
o	o	o	o	o	o	o	o	o	o	THIRD	

LOCATION	COMMAND	PARAMETER	COMMENTS
	.		
	.		
	.		
	BINARY,H	10,3,41,3,-3,2,SERIAL	Resolve the three items into three words of storage at SERIAL. Change 3 binary words to three 4- digit decimal numbers to be stored in the output record at SERNUM.
	ENCODE,B	4,3,SERIAL,SERNUM	
	.		
	.		
	.		
SERIAL	BSS	3	Reserve 3 words in working storage.
	.		
	.		
	.		
SERNUM	BSS	12	Reserve 12 words in output record.
	.		
	.		
	.		

NOTE: Root response position is the least significant response bit in the first item to be resolved.

DECIMAL

DECIMAL

Resolves a binary grid into a singular EBCDIC decimal number per item. If an item contains a binary number greater than $9_{(16)}$, an asterisk is placed in the storage location.

DECIMAL,H DECIMAL,V	nr,ni,x,y,dx,dy,addr
DECIMAL,H	4,1,37,4,-1,0,CKDGT

COMMAND	DESCRIPTION
DECIMAL,H	Specifies items in the grid are horizontally oriented.
DECIMAL,V	Specifies items in the grid are vertically oriented.

PARAMETER	DESCRIPTION
nr	Specifies the number of responses for any item in the grid. This number must not exceed four.
ni	Specifies the number of items in the grid.
x,y	An x,y coordinate pair identifying the location of the root response position.
dx	The difference between the x coordinate value for adjacent responses of a horizontal item or two adjacent vertical items. The dx values must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The address of the first resolved item. The remainder of the resolved items are placed in consecutive storage locations.

Storage Requirements - Five words.

CAUTION

The DECIMAL and BINARY commands read multiple responses as a single binary value with no discrimination by intensity. All marks above the light mark threshold are read as binary ones. Therefore, particular care must be taken when marking binary grids. Marks must be above the light mark threshold to be read and below it to be ignored. This is a case when it is recommended that a check digit be used with each decimal and binary grid.

LOCATION	COMMAND	PARAMETERS	COMMENTS
	.		
	.		
	.		
	DECIMAL,H	4,6,41,11,1,3,PUPIL	Horizontally oriented grid, root response at 41,11, a difference of 1 space between responses and 3 spaces between items.
	.		
	.		
	.		
	.		
	.		
	.		
PUPIL	BSS	6	6 spaces reserved in the output record.
	.		
	.		
	.		
	.		

This grid also could be resolved with BINARY,H. In that case the results would have to be encoded to obtain the same decimal code provided by the DECIMAL resolution.

NOTE: Root response position is the least significant response bit in the first item to be resolved.

LITHOID

LITHOID

Resolves a one item, 18 response position binary grid that is imprinted on a document. When resolution completes, the binary number is expressed as a six digit EBCDIC decimal integer. If the number is less than six digits, the field contains leading zeros.

LITHOID,H LITHOID,V	x,y,dx,dy,addr
LITHOID,H	41,6,1,0,IDENT

PARAMETER	DESCRIPTION
x,y	An x,y coordinate pair identifying the location of the root response position. The root response position is usually surrounded by a square box. In most cases it is the first digit after the printed number.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be proceeded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate value for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be proceeded by a minus sign if resolution scanning progresses toward the leading edge.
addr	The storage location of the most significant decimal integer.

Storage Requirements - Four words.

Application

The lithoid is used to identify and match the output from two separately processed sheets which were originally one document (sliced booklet). Figure 4-5 displays a document with the lithoid printed on both halves. The sheet is perforated for easy separation. The lithoid is unique for each document and is applied when the document is printed.

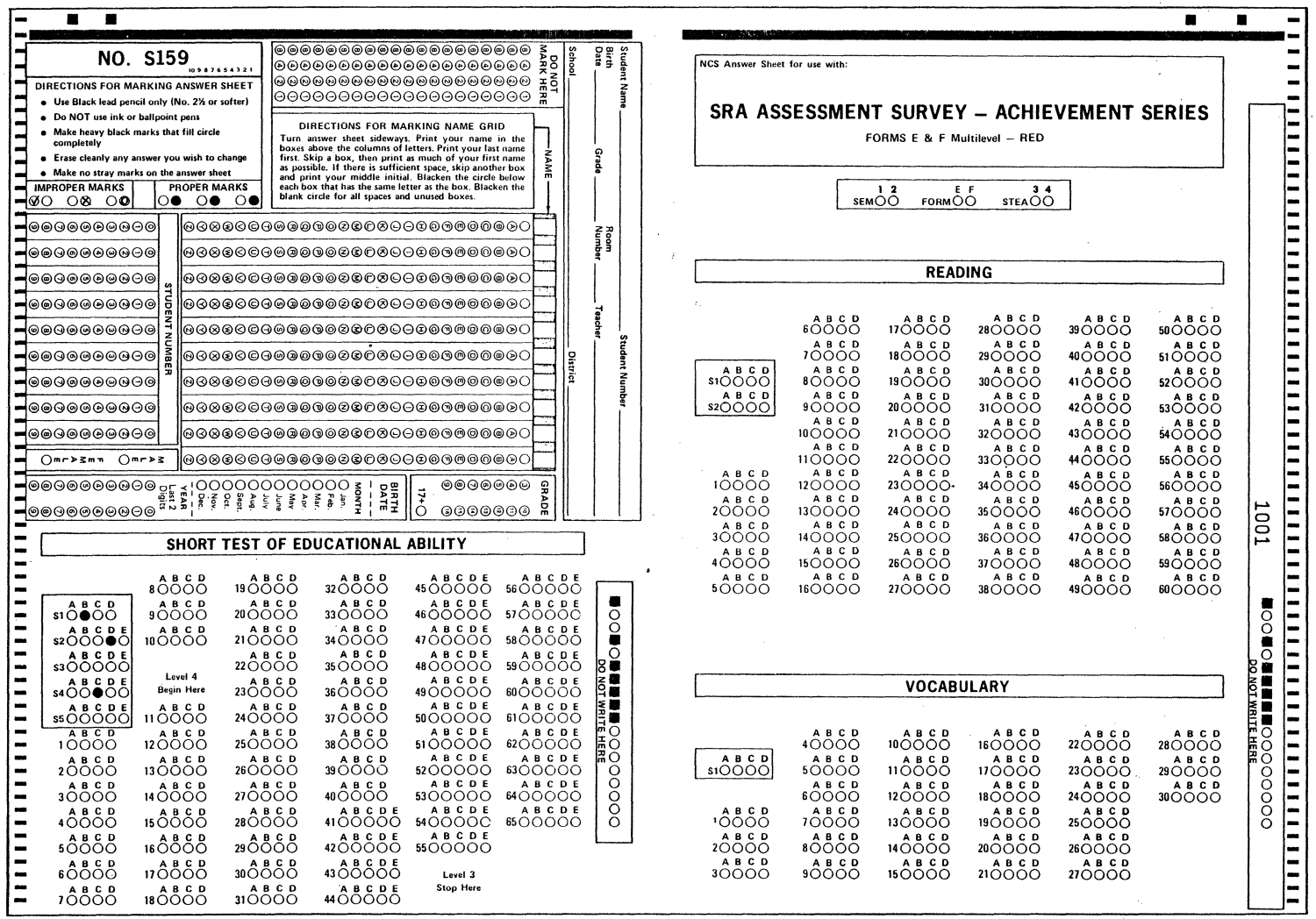


Figure 4-5. Document with Lithoid

HDRSN

Resolves a serial number grid from a header document. The most significant digit is a check digit and the remaining four digits are a user-supplied serial number. The resolved check digit and serial number are transferred to the output record if the check digit can be correctly computed from the serial number, and the serial number grid is free of omissions and multiple responses. Otherwise, the output record contains a serial number that is one greater than the previous output record's serial number and the calculated check digit. Two error flags are also written into the output record buffer; < flag in character position 15 and a C in character position 16.

The check digit is calculated by multiplying the least significant serial number digit by two, the next digit by three, the following digit by four, and the most significant digit by five. The calculation is completed by summing the four products and using the sum's least significant digit for the check digit.

EXAMPLE:

```
HDRSN  =  6359
9 x 2  =    18
5 x 3  =    15
3 x 4  =    12
6 x 5  =    30
            75
```

Check Digit = 5

```
HDRSN,H          x,y,dx,dy
HDRSN,V

HDRSN,V          42,3,1,2
```

COMMAND	DESCRIPTION
HDRSN,H	Specifies the header grid items are horizontally oriented.
HDRSN,V	Specifies the header grid items are vertically oriented.

PARAMETER	DESCRIPTION
x,y	An x,y coordinate pair identifying the grid's root response position.

PARAMETER	DESCRIPTION
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate values for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.

Storage Requirements - Three words.

Application

A header document precedes a batch of documents and normally contains information that is common to all documents in that batch. If the document contains a serial number grid that is resolved by this command, the output record created by this document will contain the number specified in the grid. The remaining output records created from successively scanned documents will contain serial numbers incremented from the serial number in the header grid.

DOCSN

DOCSN

Resolves the five digit serial number grid of a document. The most significant digit is a check digit and the remaining four digits are a user supplied serial number. The resolved check digit and serial number are transferred to the output record if the operator enters a run number greater than one. The check digit must be correct and the serial number grid free of omissions and multiple responses. If the serial number read from the document fails any of the requirements, the number inserted into the output record is one greater than the serial number from the previous document and the check digit is calculated from the incremented serial number. Serial number grids that have a check digit error, an omission, or a multiple response also have a < flag in character position 15 and a N in character position 16 of the output record.

The check digit is calculated by multiplying the least significant serial number digit by two, the next digit by three, the following digit by four and the last digit by five. The calculation is completed by summing the four products and using the least significant digit for a check digit.

EXAMPLES:

```
DOCSN = 2156
6 x 2 =   12
5 x 3 =   15
1 x 4 =    4
2 x 5 =   10
          41
```

Check Digit = 1

```
DOCSN,H          x,y,dx,dy
DOCSN,V

DOCSN,V          42,3,1,2
```

COMMAND	DESCRIPTION
DOCSN,H	Specifies that the serial number grid items are horizontally oriented.
DOCSN,V	Specifies that the serial number grid items are vertically oriented.

PARAMETER	DESCRIPTION
x,y	An x,y coordinate pair identifying the grid's root response position.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate values for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.

NOTE: This command is ignored if the run number is 1.

Storage Requirements - Three words.

Application

A serial number is coded on a document when a document that has failed an edit test is corrected. The number placed on the document is taken from the output record listing. When the document is reprocessed, the serial number coded into the grid is forced into the output record. The reprocessed data is then merged into one correct output record file.

BOOKSN

BOOKSN

Resolves a serial number and batch number binary grid on a 7010 folded document when the run number is greater than one. The operations performed by this command are ignored when the run number is one. Resolution of a valid binary serial and batch number causes the system to insert the decimal data in the batch and serial number fields of the output record. The most significant digit of the batch number is obtained from the most significant digit of the batch number entered at program initialization. The two least significant batch number digits resolved by this command can not exceed 19. Data resolved by this command is invalid if the bit pattern is not odd parity, or a bit combination containing a decimal number contains a hexidecimal character greater than nine. Invalid resolved data causes the system to:

- Insert a N in character position 16 of the output record.
- Insert a < flag in character position 15 of the output record.
- Store in the serial number field the next consecutive serial number.
- Store in the batch number field the same batch number as the last batch number on tape.

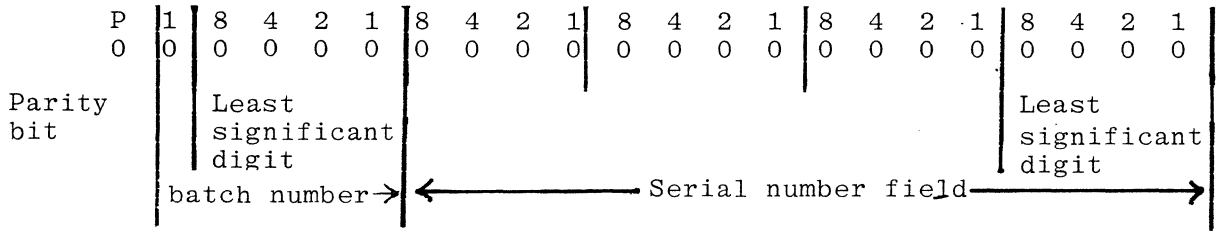
```
BOOKSN          x

BOOKSN          47
```

PARAMETER	DESCRIPTION
x	Identifies the photocell that reads the serial number grid. (Assumes y = 5, dy = 2, dx = 0)

Storage Requirements - One Word

Application

Each digit of the batch and serial number for the document is coded in binary when a document field fails an edit test. The number placed on the document is taken from the output record listing. When the field failing the test is corrected and the document is reprocessed, the operator enters a run number of 2 and the scanned serial and batch number is inserted into the batch and serial number field of the output record. The figure below defines which bits in the grid are associated with the decimal numbers in the batch and serial number fields.



Note: If the batch number is not coded on the sheet the original batch number is left untouched.

BATCHNO

BATCHNO

Resolves a header document batch number grid, inserts the resolved results in the batch number field of the output record, and sets the serial number to 0001. If the resolved number grid contains an omission or multiple response, the batch number field in the output record contains the batch number assigned to the previous document, and the serial number field is one greater than the serial number field for the previous document. Also, a > is inserted in character position seven, and a B is inserted in character position 16 of the output record.

BATCHNO,H BATCHNO,V	x,y,dx,dy
BATCHNO,V	30,27,-1,1

COMMAND	DESCRIPTION
BATCHNO,H	Specifies that the batch number grid is horizontally oriented.
BATCHNO,V	Specifies that the batch number grid is vertically oriented.

PARAMETER	DESCRIPTION
x,y	An x,y coordinate pair identifying the root response position of the grid.
dx	The difference between the x coordinate values for two adjacent responses of a horizontal item or two adjacent vertical items. The dx value must be preceded by a minus sign if resolution scanning progresses toward the timing track.
dy	The difference between the y coordinate values for two adjacent horizontal items or two adjacent responses of a vertical item. The dy value must be preceded by a minus sign if resolution scanning progresses toward the leading edge.

Storage Requirements - Three words.

VALIDATION COMMANDS

Commands discussed in this section include;

- TALLY
- GRADE
- SCORE
- EVALUATE
- EDIT
- FLAG

TALLY

Counts the number of total omissions, imbedded omissions, multiple responses, and light and dark responses in any resolved section of a document. Each of the five counts are stored internally. The counts incremented and their locations in DOSSIER are:

00B0	OMITCNT	Imbedded Omits
00B1	TOTLCNT	Total Omits
00B2	MULTCNT	Multiples
00B5	DARKCNT	Dark Marks
00B4	LITECNT	Light Marks

The counts will be accumulated by successive TALLY, GRADE, and/or SCORE commands. EVALUATE is used to retrieve the counts. They will be cleared by the next EVALUATE or WRITE command executed. TALLY (and GRADE and SCORE) accept either binary or EBCDIC omits and multiples. (00B3 is explained under EVALUATE.)

```
TALLY          ni,addr

TALLY          100,QUEST
```

PARAMETER	DESCRIPTION
ni	Specifies the number of contiguous resolved responses in the output record buffer area to be tallied. Must not exceed 255.
addr	Specifies the address of the first item in the output record buffer to be tallied.

Storage Requirements - Two words.

GRADE

Counts the number of correct and incorrect answers in a resolved section of a document. The correct and incorrect counts are accumulated by comparing each resolved response to a correct response in the right response table. If a resolved response matches the correct response in the right response table, the correct score accumulator is incremented. The incorrect count does not include the number of multiples or omissions. The correct score accumulator is transferred to three consecutive words of the output buffer area in EBCDIC format. The scores are stored in binary format at location $AF_{(16)}$ WRONGCNT and $AE_{(16)}$ RIGHTCNT. The counts are cleared prior to being filled by the current command.

GRADE also performs the TALLY function. The five counts will be added to any values stored by previous TALLY, GRADE, or SCORE commands which have not been cleared by an EVALUATE or WRITE.

GRADE terminates when the correct number of items specified by this command are graded or a right response of zero is found in the right response table.

```
GRADE,1
GRADE,2     ni,addr,table,results
GRADE,4

GRADE,4     80,ANS,ADVJR,RESULTS
```

COMMAND	DESCRIPTION
GRADE,1	Specifies that only one correct answer is stored in the lower eight bits of each word in the right response table.
GRADE,2	Specifies that two correct answers are stored in each word of the right response table. The first correct answer is stored in the most significant eight bits of the first word and the remaining responses are stored consecutively in eight-bit blocks.
GRADE,4	Specifies that four correct answers are stored in each word of the right response table. The first correct answer is stored in the most significant four bits of the first word and the remaining responses are stored consecutively in four-bit blocks. This command may not be used if the correct response has a value that exceeds a hexadecimal E or a hexadecimal F0, F1, or F2.

PARAMETER	DESCRIPTION
ni	Specifies the number of resolved responses to be scored. Must not exceed 255.
addr	Specifies the address of the first item in the output record buffer to be scored.
table	Specifies the starting address of the right response table.
results	Specifies the starting address of three character output record buffer containing the accumulated score.

Storage Requirements - Four words.

Reading Keys

A key, or right response table, may be read from a header and resolved into a storage area outside the record area. The header can be a different form or a specially identified test form. A typical key identifier is all one character in an identifying field such as id number or name.

Special Scoring Cases

Individual items can be bypassed, scored always correct, or always incorrect by storing a special character in the right response table Refer to the table below for specific characters. Note that 00 signals the end of the right response table.

The total omit count (TOTLCNT) does not terminate at the end of the key (00) but continues through the number of responses specified by the "ni" parameter.

EBCDIC CHARACTER	CONTENTS	RESULTS
0	$F0_{(16)}$	Ignore Response
1	$F1_{(16)}$	Accept as Correct Response
2	$F2_{(16)}$	Accept as Incorrect Response
(none)	00	End of Key

Figure 4-6. Special Cases for the Right Response Table.

Storing Keys in Binary or EBCDIC

Right response tables may be built for scoring either binary or EBCDIC data. The one exception is for data resolved by the NUMERIC command. Because the EBCDIC digits 0-2 are the same as the special key characters, (F0,F1,F2), a score key for results of a numeric grid would produce a score different from the one intended.

Application

A brief four response example is coded in each of the grade variants to clarify the coding relationship between the command and the right response table. The resulting content of memory is displayed following the validly coded examples.

EXAMPLE 1:

```
LOCATION        COMMAND         PARAMETERS
                .
                .
                .
                GRADE,1         4,SECTION1,KEY1,SCORE1
                .
                .
                .
KEY1            HEX             02,04,F0,08
                .
                .
                .
```

CONTENTS OF MEMORY	LOCATION		CONTENTS
	KEY1		0 0 0 2
		+1	0 0 0 4
		+2	0 0 F 0
		+3	0 0 0 8

* * *

EXAMPLE 2:

```
                .
                .
                .
                GRADE,2         4,SECTION2,KEY2,SCORE2
                .
                .
                .
KEY2            HEX             0204,F008
                .
                .
                .
```

CONTENTS OF MEMORY	LOCATION		CONTENTS
	KEY2		0 2 | 0 4
		+1	F 0 | 0 8

* * *

EXAMPLE 3:

```
                .
                .
                .
                GRADE,4         4,SECTION3,KEY3,SCORE3
                .
                .
                .
KEY3            HEX             Can't be packed 4 to a word if F0 is used.
```

NOTE: GRADE,1 and SCORE,1 mask off the upper eight bits of each item response read. They then compare the response to the sixteen bits of the word in the answer key. Therefore responses read from a key sheet must have the upper eight bits masked off before being stored in the key area of the program.

When moving the responses to the key use MOVE,M as illustrated.

```
MOVE,M          nw,source,destin,00FF
```

The lower eight bits of each word will remain unchanged as they are moved while the upper eight bits are cleared to zeros.

The alternative is to use the 2 or 4 variants of GRADE or SCORE and use PACK,2 or PACK,4. Both of these commands discard the upper portion of each word during the move and pack operation.

SCORE

SCORE

This command is identical to GRADE except, in addition, bit 5 of the resolved response is set to 1 if the resolved response matches the correct response in the right response table.

SCORE,1 SCORE,2 SCORE,4	ni,addr,table,results
SCORE,4	80,ANS,ADVJR,RESULTS

COMMAND	DESCRIPTION
SCORE,1	Specifies one correct answer is stored in the lower eight bits of each word in the right response table.
SCORE,2	Specifies that two correct answers are stored in each word of the right response table. The first correct answer is stored in the most significant eight bits of the first word and the remaining responses are stored consecutively in eight-bit blocks.
SCORE,4	Specifies four correct answers are stored in each word of the right response table. The first correct answer is stored in the most significant four bits of the first word and the remaining responses are stored consecutively in four-bit blocks. This command may not be used if the correct response is a hexadecimal F0, F1, or F2, or has a response positional value that exceeds a hexadecimal E.

PARAMETER	DESCRIPTION
ni	Specifies the number of resolved responses to be scored. Must not exceed 255.
addr	Specifies the address of the first item in the output record buffer to be scored.
table	Specifies the starting address of the right response table.
results	Specifies the starting address of a three word output record buffer containing the accumulated number of correct responses.

Storage Requirements - Four words.

Application

SCORE should be used only when it is necessary to display in the output record exactly which responses were correct in addition to the number of correct responses. Coding SCORE is deceptively complicated because of the effect of setting bit five. An example of setting bit five for a right response is shown below:

Bits	7	6	5	4	3	2	1	0
	0	0	0	0	0	1	0	0

Right Response
Before Scoring

Bits	7	6	5	4	3	2	1	0
	0	0	1	0	0	1	0	0

Right Response
After Scoring

In the example, executing the ARRAY command provides a set of responses labeled ANSWERS. ANSWERS is compared to the right response table TABLE by SCORE. The scored results remain in binary format at ANSWERS. Bit 5 is set for those answers which are correct. This effectively adds 32 to their binary value. To be readable in the output record, ANSWERS is translated to EBCDIC by XLATE according to the characters defined in XTABLE.

XTABLE in this example is 39 characters and includes in order:

- The omit character
- The multiple character
- The five response choices
- 27 blanks to fill the table up to position 33.
- Five characters to display the correct answer for a response choice. The result is that incorrect answers will be displayed as the original response values of 1-5 and correct answers as A-E.

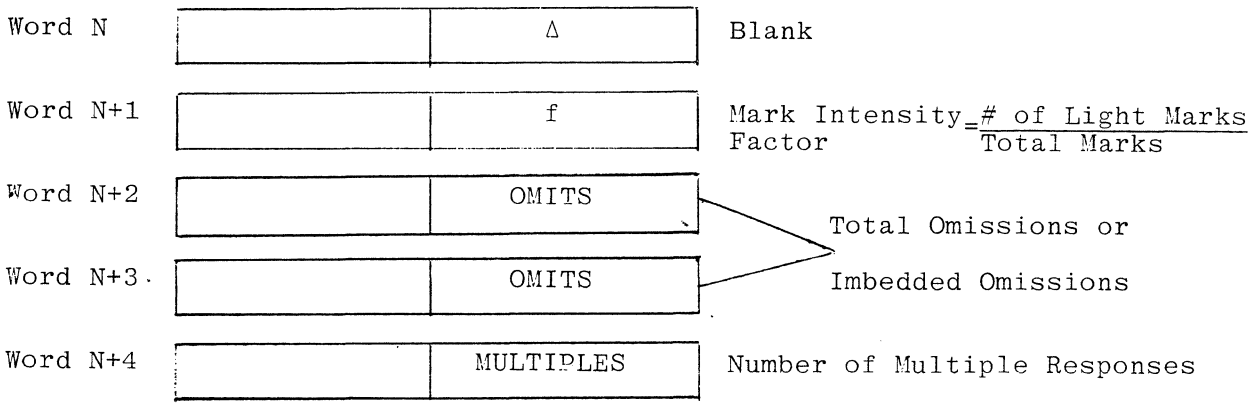
EXAMPLE:

```
LOCATION      COMMAND       PARAMETERS                         COMMENTS
              .
              .
              .
              ARRAY         ........ ANSWERS
              .
              .
              .
              SCORE,4       12,ANSWERS,TABLE,RESULTS
              .
              .
              .
              XLATE         12,ANSWERS,XTABLE
              .
              .
              .
TABLE         HEX           3141,5113,2121
XTABLE        BCD           39,-*12345                         ABCDE
                                      27 blanks
                               wrong responses                 correct
                               multiple                        responses
                               omit
```

EVALUATE

Retreives the accumulated counts stored by previous TALLY, SCORE, and GRADE commands and transfers to the output buffer area the percentage of light marks, the number of imbedded or total omissions, and the number of multiple responses in one or more sections of the document. The imbedded or total omissions and the multiple counts are summed and added to the Total Invalid Response Counter (00B3). The internal counters set by the TALLY, SCORE, and GRADE commands are then reset to zero. The binary representation for the percent of light marks may be obtained by addressing location $AC_{(16)}$.

The results of the EVALUATE command are stored in EBCDIC in five words of the output buffer



The first character is blank. The second character represents the percentage of light marks which is the mark intensity factor. The chart below defines the percentage range for each defined character for "f".

BINARY IN LOCATION $AC_{(16)}$	CHARACTER IN OUTPUT RECORD	% OF LIGHT MARKS
0	#	Section Totally omitted
1	-	No light marks
2	0	0 - 9%
3	1	10-19%
4	2	20-29%
5	3	30-39%
6	4	40-49%
7	5	50-59%
8	6	60-69%
9	7	70-79%
10	8	80-89%
11	9	90-99%
12	$	100% Light Marks

The third and fourth characters express the number of omissions or imbedded omissions. Omission counts exceeding 99 will be expressed as 99.

The fifth character expresses the number of multiple responses. Multiple counts exceeding 9 will be expressed as 9.

EVALUATE,I EVALUATE,T	addr
EVALUATE,I	RESULTSE

COMMAND	DESCRIPTION
EVALUATE,I	Specifies that the imbedded omission count is to be inserted in the output record buffer area, and that the imbedded omission count and the multiple count are added to the total invalid count.
EVALUATE,T	Specifies that the total omission count is to be inserted into the output record buffer area. The total omission count and multiple count are added to the total invalid count.

PARAMETER	DESCRIPTION
addr	Specifies the address of the first character of the EVALUATE field in the output record buffer.

NOTE 1: The mark intensity factor (SECTFACT stored in location $AC_{(16)}$) is calculated for the stored counts for each EVALUATE. After it is calculated, it is compared to the document mark intensity factor (DOCUFACT stored in location AD). If the current SECTFACT is higher than the current DOCUFACT, the value in DOCUFACT is replaced by the value in SECTFACT. DOCUFACT represents the lightest section evaluated on the sheet scanned. It is stored in character position twenty of the output record.

Storage Requirements - Two words.

EDIT

Verifies whether data in the field has been totally completed, left justified, right justified, or totally omitted. A field fails a completeness test if a word in the field contains an omission or a multiple. A field fails a justification test if information in the grid contains an imbedded omission or a multiple response. A field also fails a justification test if a left justified field does not contain data in the field's left most word and a right justified field does not contain data in the field's right most word. Finally, a field fails the omission test if any item in the grid has been completed. Fields that pass verification tests cause the program to branch to a user specified address.

EDIT checks for both binary or EBCDIC omits or multiples (00,FF) or (60,5C).

```
EDIT,C  }
EDIT,L  }
EDIT,R  }        ni,field,addr
EDIT,O  }

EDIT,C           8,BDATE,CONT.2
```

COMMAND	DESCRIPTION
EDIT,C	Verifies a field has been totally completed.
EDIT,L	Verifies a field has been left justified.
EDIT,R	Verifies a field has been right justified.
EDIT,O	Verifies a field has been totally omitted.

PARAMETER	DESCRIPTION
ni	Specifies the number of items in the field to be edited. This number should never exceed 255.
field	Specifies the first word address of the field containing the data needing verification.
addr	Specifies the address of the next DOSSIER command executed if the specified field is properly verified.

Storage Requirements - Three words.

Application

The following example verifies if the field stored at NUMERICA is right justified. If it is right justified, execution continues at location ED4. If not, the output record contains:

- C in character position 16
- > in character position 24
- A in the edit flag field (begins in position 25)
- ? in character position NUMERICA-1

The scanner feed mechanism stops in Forced Run Mode if Data Switch 1 is not set and output record transfer is inhibited.

EXAMPLE:

LOCATION	COMMAND	PARAMETERS	COMMENTS
.			
.			
.			
ED3	EDIT,R	2,NUMERICA,ED4	Edit grid for right justification
	FLAG,R	A,NUMERICA	Flag Grid Error
ED4	....	.....	
.			
.			
.			
.			

For another example of EDIT, see the application section of the FLAG command

FLAG

Identifies fields in storage and gives the user the option of stopping the scanner feed mechanism and inhibiting the transfer of data to tape. Fields are identified by:

- placing a question mark in the storage location immediately before the specified field.
- inserting a > in character position 24 of the output record.
- inserting a specified character in the first blank position of the grid edit flag field (beginning in position 25 of the output record).
- inserting a C in character position 16 of the output record.

The scanner halts automatically when WRITE is executed if a C is inserted in character position 16 of the output record and Data Switch 1 on the processor is off (normal Error Stop Mode). Halting the scanner feed mechanism also inhibits the transfer of data to tape. Only a C is inserted in character position 16 of the output record if Data Switch 1 is on and the contents of the buffer are written to tape.

FLAG,G FLAG,R	flgchr,field
FLAG,R	B,BDATE

COMMAND	DESCRIPTION
FLAG,G	Specifies that only a field is to be flagged. Flagging a field does not insert a C into character position 16 of the output record nor does it stop the feed mechanism.
FLAG,R	Specifies that both the field and record are to be flagged. This command causes a field to be identified, stops the scanner feed mechanism, and inhibits the transfer of data to tape if Data Switch 1 is off.

PARAMETER	DESCRIPTION
flgchr	Specifies the character that is placed in the edit flag field. Any valid EBCDIC character appearing in Appendix A may be used.
field	Specifies the location of the field that is to have a question mark inserted in the preceding location.

Storage Requirements - Two words.

Application

(See next page.)

The EDIT and FLAG commands are used to validate data in a field. Assume a symbolic location NAME contains the name of an applicant and the symbolic location SSN contains the applicant's social security number. The NAME field is to be checked for left justification and flagged if it is not, and the SSN field is to be checked for completeness and the record flagged if it is not; code the following:

```
LOCATION        COMMAND         PARAMETERS

                EDIT,L          19,NAME,CKSSN
                FLAG,G          N,NAME
CKSSN           EDIT,C          9,SSN,CØNT
                FLAG,R          S,SSN
                .               .
                .               .
                .               .
                .               .
CONT            .               .
.               .               .
.               .               .
.               .               .
NAMEFLAG        BSS             1
NAME            BSS             19
SSNFLAG         BSS             1
SSN             BSS             9
.               .               .
.               .               .
.               .               .
```

If the NAME field was not left justified and SSN field was not complete, the output record field would contain:

- C in character position 16
- > in character position 24
- N in character position 25
- S in character position 26
- ? in character position NAME-1
- ? in character position SSN-1

The scanner feed mechanism stops if Data Switch 1 is off, and output record transfer is inhibited.

If only the NAME field was not left justified, the output record field would contain:

- > in character position 24
- N in character position 25
- ? in character position NAME-1

The feed mechanism is not affected by this operation and an output record is written.

If only the SSN field was not complete, the output record field would contain:

- C in character position 16
- > in character position 24
- S in character position 25
- ? in character position SSN-1

The scanner feed mechanism stops if Data Switch 1 is off and output record transfer is inhibited.

DATA MANIPULATION COMMANDS

Commands in this section permit the user to move and store data, to convert EBCDIC data to binary and binary data to EBCDIC, and to pack and unpack stored information. The commands described are:

- STORE
- MOVE
- ENCODE
- DECODE
- REPLACE
- PACK
- XPAND
- XLATE

STORE

Stores immediate data in one or more words. All data stored is right justified. Data requiring less than 16 bits of storage is filled with leading zeros.

STORE[,B]	[nw,]value,destin
STORE	=B,SEX

COMMAND	DESCRIPTION
STORE	Stores data in one word.
STORE,B	Stores data in a block of words.

PARAMETER	DESCRIPTION
nw	Specifies the number of words in the block if the STORE,B variant is used. This parameter must be omitted when only STORE is coded. Must not exceed 255.
value	Specifies the immediate operand. Use a valid literal.
destin	Specifies the address of the first stored character. If the specified value parameter is to be stored in a block of memory words, the character will be stored in nw contiguous locations starting at destin.

Storage Requirements - Three words.

MOVE

MOVE

Moves a contiguous block of words from one storage location to another. During the move process, a source information field may be logically ANDed word for word with a one word mask and the results inserted in the destination field. The source is left unchanged by the move operation.

MOVE[,M]	nw,source,destin[,mask]
MOVE	10,RAW,WORK

COMMAND	DESCRIPTION
MOVE	Moves a contiguous block of words from one storage location to another.
MOVE,M	Masks a block of contiguous words and places the masked results in storage.

PARAMETER	DESCRIPTION
nw	Specifies the number of words to be moved and/or masked. Must not exceed 255.
source	Specifies the address of the first word to be moved and/or masked.
destin	Specifies the destination address of the first word that has been moved and/or masked.
mask	Specifies the mask. The mask may be expressed as a decimal number or four hexadecimal characters. When expressed as four hexadecimal characters, a zero must be placed before the most significant character.

Storage Requirements - MOVE - Three words.
MOVE,M - Four words.

Application

Assume a user wishes to identify which items in a test were SCORED correctly but not necessarily which response positions were selected by the student taking the test. Also, assume the scored results are stored at location EXAM. To mask out the selected responses and store the logical results in CORRES, code the following:

```
MOVE,M          80,EXAM,CORRES,00020
```

Preceding Zero — Hexadecimal Mask Value

The same results as the results produced by the MOVE,M command above can be produced by the MOVE,M command below. The only difference between the two commands is the MOVE,M command below has a decimal mask.

```
MOVE,M          80,EXAM,CORRES,32
```

Decimal Mask Value

ENCODE

ENCODE

Translates binary data to EBCDIC decimal numbers.

Converts either the lower eight bits of binary response value to an EBCDIC decimal number (ENCODE,R) or the lower fifteen bits of a binary word to an EBCDIC decimal number (ENCODE,B).

Each binary value encoded to an EBCDIC decimal number is inserted in the lower eight bits of memory word. If the decimal field is longer than the decimal number, the decimal number is inserted right-justified in the field and unused field positions are filled with EBCDIC decimal zeros. If the receiving field is too short to contain the decimal number, the most significant decimal digits that do not fit are truncated. The upper eight bits of all words in the decimal field are filled with zeros.

ENCODE,R ENCODE,B	n,nw,binary,decimal
ENCODE,R	2,1,GRADE,DECGRADE

COMMAND	DESCRIPTION
ENCODE,R	Specifies only the lower eight bits of a word are to be encoded. If the lower eight bits are all zeros (binary code for an omission) the result is an EBCDIC minus. If the eight bits are all ones (binary code for a multiple) it encodes to an EBCDIC asterisk.
ENCODE,B	Specifies that the lower fifteen bits of the word are to be encoded. Bit 15 is stripped off in the process. Do not use for binary response values which are the result of ITEM, ARRAY, and RESPONSE commands because of the undetermined contents of the upper eight bits.

PARAMETER	DESCRIPTION
n	Specifies the number of decimal digits in the result. This number may be in the range of one to five when ENCODE,B is used, or one to three when ENCODE,R is used.
nw	The number of words containing binary data to be converted. This integer must not exceed 255.
binary	The address of the first binary word to be converted. The contents of the addressed word must not exceed $7FFF_{(16)}$.
decimal	The address of the most significant digit in the decimal field.

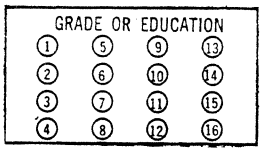
Storage Requirements - Three words.

Application

ENCODE can convert a binary value stored in a word to the equivalent EBCDIC decimal value. ENCODE,R is useful when converting data fields on a document from a resolved binary value to an EBCDIC decimal character. Consider the following example.

Below is shown a grade or education field that must be resolved into a binary value. Assume this field must be displayed and edited using one of the grid edit commands. The following coding will convert the binary field to EBCDIC decimal data format, or an asterisk or minus EBCDIC character if the field contains a multiple response or an omission.

```
ENCODE,R      2,1,BGRADE,DGRADE
```



At command completion, the BGRADE field remains unchanged, and if the student selected response position nine, DGRADE contains:

```
DGRADE        40F0,40F9
```

If BGRADE indicates a multiple response, DGRADE contains:

```
DGRADE        405C,405C
```

When BGRADE indicates an omission, DGRADE contains:

```
DGRADE        4060,4060
```

The following charts show how the system encodes a binary number when ENCODE,B command is coded.

```
ENCODE,B      4,5,BINDAT,DECDAT
```

	15			0
BINDAT	0	8	0	0
+1	1	7	F	6
+2	3	0	0	0
+3	0	1	3	1
+4	0	0	5	9

	14							7-------0	
DECDAT	0 0 0 0 0 0 0 0							F2	1st binary value
+1	0 0 0 0 0 0 0 0							F0	
+2	0 0 0 0 0 0 0 0							F4	
+3	0 0 0 0 0 0 0 0							F8	

$0800_{(16)} = 2048_{10}$

	14							7-------0	
DECDAT +4	0 0 0 0 0 0 0 0							F6	2nd binary value
+5	0 0 0 0 0 0 0 0							F1	
+6	0 0 0 0 0 0 0 0							F3	
+7	0 0 0 0 0 0 0 0							F4	

$17F6_{(16)} = 6134_{10}$

	14							7-------0	
DECDAT + 8	0 0 0 0 0 0 0 0							F2	3rd binary value
+ 9	0 0 0 0 0 0 0 0							F2	
+10	0 0 0 0 0 0 0 0							F8	
+11	0 0 0 0 0 0 0 0							F8	

$3000_{(16)} = (1)2288_{10}$ ← discarded

	14							7-------0	
DECDAT +12	0 0 0 0 0 0 0 0							F0	4th binary value
+13	0 0 0 0 0 0 0 0							F3	
+14	0 0 0 0 0 0 0 0							F0	
+15	0 0 0 0 0 0 0 0							F5	

$0131_{(16)} = 0305_{10}$

	14							7-------0	
DECDAT +16	0 0 0 0 0 0 0 0							F0	5th binary value
+17	0 0 0 0 0 0 0 0							F0	
+18	0 0 0 0 0 0 0 0							F8	
+19	0 0 0 0 0 0 0 0							F9	

$0059_{(16)} = 0089_{10}$

CAUTION

Failure to use the ENCODE,R command to encode binary numbers produced by the resolution process could cause the system to produce anomalous results.

DECODE

DECODE

Converts an EBCDIC decimal field into a binary value and stores the value in a memory word. The decimal number converted must not exceed 65535. Decimal numbers smaller than five digits may be extended to five digits by adding leading zeros. Decimal integer values larger than 32767 when decoded will set bit 15 and the remaining 14 bits will represent a negative binary value, if later resolution looks at all 16 bits.

```
DECODE          n,nw,decimal,binary

DECODE          3,1,GRID,BASE2
```

PARAMETER	DESCRIPTION
n	Specifies the number of decimal digits in each field that is to be converted to a binary value. Must be between one and five, inclusive.
nw	The number of binary words that exist after each decimal field is converted to a binary value.
decimal	The beginning address of the most significant digit in the decimal field to be converted.
binary	The storage address of the first converted binary value.

Storage Requirements - Three words.

Application

DOSSIER has no base 10 arithmetic instructions. Therefore, EBCDIC coded decimal data must be converted to the equivalent binary value before arithmetic operations can be performed.

Assume that a three item numeric grid was resolved and the character format must be resolved to binary format before an arithmetic assembly language subroutine is performed. Assume the grid field's resolved contents contain no omissions or multiple responses and are stored in a field labeled GRID. Also assume the binary output is to be stored in BASE2. To convert the decimal number to a binary number, code the following:

```
DECODE          3,1,GRID,BASE2
```

If the GRID field contained 40F3,C0F7, and 40F4; after execution of the DECODE command the GRID field would remain unchanged and the BASE2 one word field will contain 0176. The contents of all memory fields are expressed hexadecimally.

DECODE may also be used to segment a decimal field and produce one binary value for each segment. Assume the decimal field DEC contains C0F9, 40F8, 40F7, 40F6, 40F5, C0F4, C0F3, 40F2, and 40F1. To segment the decimal field into three fields each having three digits and producing three binary values, code the following:

```
DECODE          3,3,DEC,BINARY
```

At the completion of the command, the DEC field will remain unchanged and the BINARY field will contain 03DB, 028E, and 0141. As before, all memory fields are expressed hexadecimally.

CAUTION

Numbers greater than 32767, when decoded, set bit 15. In some cases (e.g. SUBMEMRY, ADDMEMRY) bit 15 will be read as a sign, causing the remaining fifteen bits to be read as a negative number. ENCODE removes bit 15.

REPLACE

REPLACE

Allows the selective replacement of the lower eight bits of a field. At command completion the upper eight bits of all words in the field are left unchanged.

```
REPLACE,L
REPLACE,T >        nw,from,to,addr
REPLACE,A

REPLACE,T          ll,=-,=  ,LSTNME
```

COMMAND	DESCRIPTION
REPLACE,L	Replaces leading field contents.
REPLACE,T	Replaces trailing field contents.
REPLACE,A	Replaces any character in the field.

PARAMETER	DESCRIPTION
nw	Specifies the length of the replacement field, this must not exceed 255.
from	Specifies the character presently in the field to be replaced. Use a valid literal.
to	Specifies the replacement character. Use a valid literal.
addr	First word address of the replacement field.

Storage Requirements - Three words

Application

(See next page.)

This command is used to replace unwanted characters that might exist in a field. In the examples below, only the lower eight bits of each word appear.

Example 1: Assume after resolution of an alphabetic grid the field NAME contains:

```
JOHN DOE-------
```

After coding:

```
REPLACE,T 15,=-,040,NAME
```

the field contains:

```
JOHN DOE
```

Example 2: Assume after resolution of a numeric grid the field TOTAL contains:

```
---165
```

After coding:

```
REPLACE,L  6,=-,=0,TØTAL
```

the field contains:

```
000165
```

Example 3: Assume you want spaces to represent multiples and omissions in your output record. Just before the WRITE command in your program, you would code:

```
REPLACE,A  250,=-,= ,RECB+24
REPLACE,A  250,=*,= ,RECB+24
```

PACK

PACK

Stores either the lower order eight bits of two consecutive words into one word or the lower order four bits of four consecutive words into one word.

PACK,2 PACK,4	nw,source,destin
PACK,2	8,BUFFR,TABL2

COMMAND	DESCRIPTION
PACK,2	Specifies that the low order eight bits of two consecutive words are to be stored in one word. Data is packed into the destination field left-justified. Unfilled positions in the last word will be filled with binary zeros.
PACK,4	Specifies that the low order four bits of four consecutive words are to be stored in one word. Data is packed into the destination field left-justified. Unfilled positions in the last word will be filled with binary zeros.

PARAMETER	DESCRIPTION
nw	Specifies the number of input words to be packed. This integer must not exceed 255.
source	Specifies the address of the first word to be packed.
destin	Specifies the address of the first packed word. The source and destination cannot be at the same location. If they are, the first word will be lost.

Storage Requirements - Three words.

Application

PACK will allow the packing of resolved data. A header sheet could contain score tables for the succeeding documents. After resolving the header's score table, considerable memory could be saved by packing the right response table.

EXAMPLE 1:

```
PACK,2      8,BUFFR,TABL
```

source data

	7----------0
BUFFR	00 | 05
+1	00 | 21
+2	00 | 1B
+3	00 | 0E
+4	00 | 03
+5	00 | 11
+6	00 | 12
+7	00 | 09

packed data

	15---------8	7----------0
TABL	05	21
+1	1B	0E
+2	03	11
+3	12	09

EXAMPLE 2:

```
PACK,4      7,RECD,TABL4
```

source data

	3------0
RECD	000 | 4
+1	000 | 1
+2	000 | 7
+3	000 | 3
+4	000 | B
+5	000 | 2
+6	000 | E

packed data

	15---12	11--8	7---4	3-----0
TABL4	4	1	7	3
+1	B	2	E	0

'0' Fill

XPAND

Unpacks one word of data into either the lower eight bits of two words or the lower four bits of four words. Bits not filled by the operation are set to zero.

XPAND,2 XPAND,4	nw,source,destin
XPAND,2	7,TABL2,HEXDAT

COMMAND	DESCRIPTION
XPAND,2	Specifies that the system place the first eight bits in the source field in the lower eight bits of the first word in the destination field. The remaining groups of eight bits in the source field will be placed consecutively in the lower eight bits of words in the destination field.
XPAND,4	Specifies that the system place the first four bits in the source field in the lower four bits of the first word in the destination field. The remaining groups of four bits in the sources field will be placed consecutively in the lower four bits of words in the destination field.

PARAMETER	DESCRIPTION
nw	Specifies the number of words in the result. Must not exceed 255.
source	Specifies the beginning address of the packed field.
destin	Specifies the beginning address of the unpacked field.

Storage Requirements - Three words.

Application

(See next page)

EXAMPLE 1 :

```
XPAND,2      7,TABL2,HEXDAT
```

source data

	11--------8	7----------0
TABL2	21	3A
+1	05	17
+2	4D	2F
+3	0A	11

unpacked data

		7----------0
HEXDAT	0 0	21
+1	0 0	3A
+2	0 0	05
+3	0 0	17
+4	0 0	4D
+5	0 0	2F
+6	0 0	0A

EXAMPLE 2 : (Using the Source Data of Example 1)

```
XPAND,4      7,TABL2,BINDAT
```

		3-------0
BINDAT	0 0 0	2
+1	0 0 0	1
+2	0 0 0	3
+3	0 0 0	A
+4	0 0 0	0
+5	0 0 0	5
+6	0 0 0	1

XLATE

XLATE

Allows the user to translate any previously resolved binary data to any EBCDIC coded characters. Translation is accomplished by defining in a table the desired EBCDIC character for each binary positional value. Each word in the table defines two EBCDIC characters that are to replace the binary values. The first word defines the EBCDIC characters replacing the binary omission (00) and multiple response (FF). The second word defines the EBCDIC characters replacing binary values one and two, the third word defines the EBCDIC characters replacing binary values three and four, etc. The translation process overlays the information it is translating, and leaves the upper 8 bits in the translated word unchanged.

```
XLATE          nw,resp,table

XLATE          80,ARRAY80,XLATE
```

PARAMETER	DESCRIPTION
nw	Specifies the number of words to be translated, This number must not exceed 255.
resp	Identifies the address of the first word to be translated.
table	Identifies the first word address of the translate table.

Storage Requirements - Three words.

Application

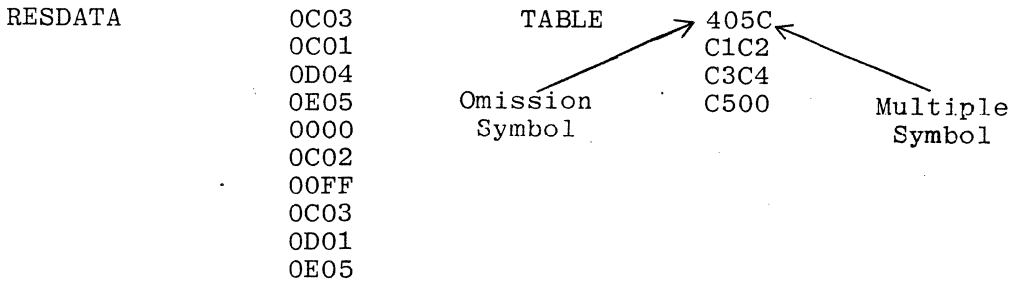
The translate command allows the user to change resolved binary positional values to any EBCDIC character.

EXAMPLE:

```
LOCATION    COMMAND     PARAMETERS              COMMENTS
            XLATE       10,RESDATA,TABLE
            .           .
            .           .
            .           .
RESDATA     BSS         10                      In output record
            .           .
            .           .
            .           .
TABLE       BCD         7, *ABCDE               Working storage
            .           .
            .           .
            .           .
```

This XLATE will translate ten consecutive words starting at RESDATA according to TABLE. Omits will be translated to blanks (40), multiples to "*" (5C), binary 01 to "A" (C1), binary 02 to "B" (C2), etc.

Contents of memory before command execution:



Contents of memory after command execution:

RESDATA	0CC3	TABLE	D6D4
	0CC1		C1C2
	0DC4		C3C4
	0EC5		C500
	00D6		
	0CC2		
	00D4		
	0CC3		
	0DC1		
	0EC5		

CAUTION

Translate tables that do not define sufficient EBCDIC values to translate resolved binary data cause indeterminate results when the binary positional value being translated exceeds the length of the translate table.

ADDMEMRY

ARITHMETIC COMMANDS

Two arithmetic commands either add or subtract the contents of two binary words. Commands referenced in this section are:

- ADDMEMRY
- SUBMEMRY

ADDMEMRY

Adds the contents of word one to word two and stores the results in word one.

```
ADDMEMRY        word1,word2

ADDMEMRY        TOTAL2,TOTAL1
```

PARAMETER	DESCRIPTION
word1	Specifies the address of the original addend and of the final accumulated result.
word2	Specifies the address of the augend.

Storage Requirements - Three Words.

SUBMEMRY

SUBMEMRY

Subtracts the contents of word 2 from word 1 and stores the results in word 1.

```
SUBMEMRY        word1,word2

SUBMEMRY        GROSS,NET
```

PARAMETER	DESCRIPTION
word1	Specifies the address of the subtracted result.
word2	Specifies an address of a word whose contents are to be subtracted from word 1.

Storage Requirements - Three words.

SEQUENCE AND CONTROL COMMANDS

Commands in this section can alter the sequential execution of commands. Commands listed in this section are:

- JUMP
- EXIT
- REENTER
- CONTROL
- TEST
- COMPARE
- DISPLAY
- DELAY

JUMP

Causes program execution to branch to an addressed DOSSIER routine and if specified, store the JUMP command address +1 in the memory location immediately before the branching address.

JUMP,U JUMP,R	addr
JUMP,U	GRIDD

COMMAND	DESCRIPTION
JUMP,U	Specifies an unconditional branching operation to an addressed DOSSIER command.
JUMP,R	Specifies an unconditional branching operation to an addressed DOSSIER routine, and stores the return address in the memory location immediately before the branching address.

ADDRESS FIELD	DESCRIPTION
addr	The address of the first operation to be executed after the execution of the JUMP instruction.

Storage Requirements - Two words.

Application

JUMP,U transfers control to any command in a DOSSIER program. JUMP,R transfers control to a DOSSIER subroutine and provides for a return to the program.

Assume that many different documents having different skunk marks are to be scanned and all documents have name grids that have the same orientation, root response position, response placement, and number of items. After each document is identified, the same subroutine could be used to resolve the name grid for all documents. The following code could be used to enter the subroutine.

LOCATION	COMMAND	PARAMETER	COMMENTS
	JUMP,R	NAMGRD	Instruction 1
	.		Instruction 5
	.		Instruction 6

To exit the subroutine, the following would be coded:

LOCATION	COMMAND	PARAMETER	COMMENTS
EXIT	JUMP,U	**	Instruction 4
NAMGRD	ALPHA,H	27,----------	Instruction 2
	JUMP,U	EXIT	Instruction 3

When the system executes the JUMP,R command (instruction one), the address of instruction five is inserted in the address field of the instruction four and processing continues at location NAMGRID (instruction two). After instruction three is executed, processing branches to EXIT (instruction four) where another unconditional branch returns the user to instruction five (the next instruction in the main program).

EXIT

Causes program execution to branch to an addressed assembly language routine, and if specified, stores the location of the DOSSIER command after the EXIT command in the memory location immediately before the branching address.

```
EXIT,U  }
EXIT,R  }        addr

EXIT,R           SUB1
```

COMMAND	DESCRIPTION
EXIT,U	Specifies an unconditional branching operation to an addressed assembly language routine.
EXIT,R	Specifies an unconditional branching operation to an addressed assembly language routine and stores the return address in the memory location immediately before the branching location.

ADDRESS FIELD	DESCRIPTION
addr	The address of the first assembly language operation to be executed after the execution of the EXIT instruction.

Storage Requirements - Two words.

Application

EXIT,U command transfers control to an assembly language program. EXIT,R transfers control to an assembly language subroutine and stores the address of the next DOSSIER statement to be executed after the subroutine completes. This instruction is extremely useful when data manipulation operations not provided in DOSSIER must be executed. Code the following to enter an assembly language subroutine.

LOCATION	COMMAND	ADDRESS FIELD
	EXIT,R	ASSLSUB
	.	
	.	

To exit an assembly language subroutine and return to DOSSIER processing, the REENTER command must be executed.

REENTER

REENTER

Causes the system to exit an assembly language processing subroutine and start processing a DOSSIER routine at the address specified. If the user selects byte mode addressing in the assembly language routine, it is his responsibility to return to word mode addressing prior to executing a REENTER command. Byte mode is not compatible with the DOSSIER system monitor SYMON.

```
REENTER          addr

REENTER          RET1
```

ADDRESS FIELD	DESCRIPTION
addr	The address of the first DOSSIER command executed after the execution of the REENTER command. This address must be filled with ** when the address is supplied by the EXIT,R command.

Storage Requirements - Three words.

Application

The REENTER command is used to enter a DOSSIER processing routine from an assembly language routine. To exit an assembly language program that was entered by an EXIT,R instruction, code the following:

```
LOCATION        COMMAND     ADDRESS FIELD        COMMENTS

LEAVE           REENTER     **
ASSLSUB         LDA         HDR
                STA         STRT
                 .
                 .
                JMP         LEAVE
```

At the completion of the assembly language program starting at location ASSLSUB, control is transferred to location LEAVE. Location LEAVE transfers control to the command following the EXIT,R DOSSIER command that called the ASSLSUB subroutine.

CONTROL

CONTROL

Provides an automatic exit to an assembly language routine that either executes before the command program starts or when the command program terminates. When branching occurs, the address of the next executable command is stored at the branching address and execution of the assembly language routine starts at the address after the branching address. A CONTROL,B command executes after the run number is entered or after terminating the tape positioning operator run option with a carriage return. When entering a addressed CONTROL,B routine, the A register will contain one if entered by toggling run or zero if entered by terminating the tape positioning option by pressing the carriage return. The CONTROL,E executes when the operator presses the SENSE switch to close out the tape.

```
CONTROL,B }
CONTROL,E }      addr

CONTROL,B        HDRLBL
```

COMMAND	DESCRIPTION
CONTROL,B	Specifies the addressed routine is to be executed at the beginning of the command program.
CONTROL,E	Specifies the addressed routine is to be executed at command program termination.

PARAMETER	DESCRIPTION
Addr	Specifies the first address of the assembly language routine.

Storage Requirement - Two Words

Application

This command allows the user to write an assembly language routine that will write header and trailer labels on tape.

Reentry

Reentry to DOSSIER from the routine referenced by a CONTROL command entails a JMP to an indirect address. A basic model for the routine exit is:

```
LOCATION   COMMAND   PARAMETERS      COMMENTS
CONTB                **              CONTROL,B ENTRY
           LAP       0               Initialize
           STA       KOUNT              Counter
             .
             .
             .
           JMP       $CONTB          Return
```

TEST

There are four variants of the test instruction. The first compares the character to the memory word and branching occurs if the contents are equal. The second variant tests the character to the memory word and branching occurs if the contents are not equal. Variant three tests the character to the memory word and branching occurs if the character is greater than the memory word. The last variant tests the character to the memory word and branching occurs if the character is less than the memory word. The character under test is only tested against the lower eight bits of a memory word. The memory word tested must be positive.

```
TEST,E  )
TEST,N  }        char,field,addr
TEST,G  }
TEST,L  )

TEST,E           =D,DECZERO,OMIT
```

COMMAND	DESCRIPTION
TEST,E	Compares a character to the contents in the lower eight bits of a memory word. Branching occurs if the character is identical to the character stored in the contents of the lower eight bits of the memory word.
TEST,N	Compares a character to the contents in the lower eight bits of a memory word. Branching occurs if the character is NOT identical to the character stored in the lower eight bits of the memory word.
TEST,G	Compares a character to the contents in the lower eight bits of memory word. Branching occurs if the binary value of the character is greater than the binary value in the lower eight bits of memory word. (Branch if char>field.)
TEST,L	Compares a character to the contents in the lower eight bits of memory word. Branching occurs if the binary value of the character is less than the binary value in the lower eight bits of the memory word. (Branch if char<field.)

PARAMETER	DESCRIPTION
char	Specifies an immediate test operand. Use a valid literal.
field	Specifies the location of a one word field whose lower eight bits are being compared against the char field.
addr	Specifies the location of the next DOSSIER command executed if the character passes the comparison test.

Storage Requirements - Three words.

COMPARE

COMPARE

Compares the lower eight bits of each word in Field 1 to the lower eight bits of each word in Field 2 for either equality or inequality and transfers control to a user specified address when the tested condition exists.

```
COMPARE,E
COMPARE,N        nw,field1,field2,addr
COMPARE,G
COMPARE,L

COMPARE,E        7,WORK,SPACES,OVFLOW
```

COMMAND	DESCRIPTION
COMPARE,E	Compares the data in field 1 to the data in field 2 for equality.
COMPARE,N	Compares the data in field 1 to the data in field 2 for inequality.
COMPARE,G	Compares the data in field 1 to the data in field 2 to see if field 1 is greater than field 2.
COMPARE,L	Compares the data in field 1 to the data in field 2 to see if field 1 is less than field 2.

PARAMETER	DESCRIPTION
nw	Specifies the number of words to compare in each field. This must not exceed 255.
field1	Specifies the beginning address of the compare field.
field2	Specifies the beginning address of the field that field 1 is being compared against.
addr	Specifies the location of the next DOSSIER command executed if the condition specified by the operation code exists.

Storage Requirements - Four words.

DISPLAY

Sets a flag to print a message on the teleprinter when the WRITE command is executed.

The DISPLAY,C command causes the scanner to pause and a display message to be printed on the teleprinter. Scanning automatically resumes when teleprinting is complete. (The DISPLAY,C command can not be used on 7018 systems.)

The DISPLAY,W command causes the system to halt and print a program display message plus the normal 32 character/16 character identification fields. The system waits for an operator response before continuing. Since the system halts, extra sheets may pass through the scanner without being processed. These sheets must be returned to the input hopper before continuing.

DISPLAY commands may be anywhere in the execution path; however, DOSSIER only executes the last DISPLAY command in the path. The display message field must be in EBCDIC format and must be packed two characters per word.

DOSSIER error messages override user DISPLAY messages.

```
DISPLAY,C }
DISPLAY,W }        nc,addr

DISPLAY,C          5,ERCODE
```

COMMAND	DESCRIPTION
DISPLAY,C	Causes the document feed mechanism to stop momentarily and a message to be listed on the teleprinter. After the message is typed, processing continues. (Caution: Do not use this command with 7018 systems.)
DISPLAY,W	Causes the document feed mechanism to stop and a message to be listed on the teleprinter. Processing stops and waits for an operator acknowledgement before continuing.

PARAMETER	DESCRIPTION
nc	Specifies the number of characters in the displayed message. The message may be from 1 to 32 characters long.
addr	Specifies the address of the field containing the message. The message must be packed two characters per word.

Storage Requirements - Two words.

Application

This command may be used to list part of the output record on the teleprinter or inform the operator of a program detected exception.

Example 1: The DISPLAY,W will be performed when WRITE is executed. The scanner will stop, the teleprinter will print "IS KEY CORRECT?" Plus the normal 32 character/16 character identification fields and the system will wait for an operator response.

```
LOCATION    COMMAND        PARAMETERS              COMMENTS
            DISPLAY,W      16,KEYMSG               Address of message
            .
            .
            WRITE          0
            .
            .
            .
            .
KEYMSG      BCD            16,IS KEY CORRECT?Δ     Display message
```

Example 2: DISPLAY,C causes the scanner to pause at WRITE while the message is printed on the teleprinter. The message will be variable depending on the edit flags set earlier in the program execution. The DISPLAY command requires that the message be packed two characters per words. The PACK,2 command fills that requirement.

```
            .
            .
            .
            PACK,2         10,EDITFLGS,FLAGMSG
            DISPLAY,C      10,FLAGMSG
            .
            .
            .
            WRITE          0
            .
            .
            .
FLAGMSG     BSS            5                       10 characters re-
                                                   served for message
                                                   (not in output
                                                   record area).
```

CAUTION

7015 and 7018 users should not use the DISPLAY,C option. If a 7015 system encounters a DISPLAY,C command, it is converted to DISPLAY,W before execution. If a 7018 system encounters a DISPLAY,C command, it will cause numerous errors.

DELAY

DELAY

Delays processing until the complete document has passed under the read head. Document validity is unknown until the document has passed completely under the read head. If a length error or a timing error is found, control is given directly to WRITE command. This command allows for verifying document validity before setting internal flags. DELAY should be used prior to changing any internal switches or counters.

Caution must be used in the placement of this command within your program. If the DELAY command is placed too early in the program, there may be insufficient time to complete the rest of the program.

```
DELAY

DELAY
```

Storage Requirements - One word.

Application

Test booklet sequencing is an example of when this command is useful. When processing booklets it is desirable to maintain sequence control of the pages within the booklet. This is done with an internal counter corresponding to the pages of the booklet. (ADDMEMRY is used to increment the internal counter.)

EXAMPLE:

```
LOCATION        COMMAND         PARAMETERS              COMMENTS

                DOSSIER         .......
                .
                .
                .
                SKUNK           ....PG1
                SKUNK,U         ....PG2
                .
                .
                .
PG1             TEST,E          0,COUNTER,PG1.1         Right Page?
                FLAG,R          S,...                   No. Flag Sequence Error
                JUMP,U          TAG20                   And Reject Record
                                                        Resolution,Scoring,etc.

PG1.1           .....
                .
                .
                .
                DELAY                                   Wait until all system
                                                        errors are processed.
                ADDMEMRY        COUNTER,ONE             Increment internal
                                                        counter.
                WRITE           0                       Not illustrated is the
                                                        saving of record prior
                                                        to the WRITE on any
                                                        page before the last.
                .
                .
                .
PG2             TEST,E          1,COUNTER,PG2.1         Right Page?
                FLAG,R          S,...                   No.Flag Sequence Error
                JUMP,U          TAG20                   And Reject Record
PG2.1           .....                                   Resolution, Scoring,
                                                        etc.
                .
                .
                .
                DELAY                                   Wait until all system
                                                        errors are processed.
                ADDMEMRY        COUNTER,ONE             Increment internal
                                                        counter.
                TEST,N          2,COUNTER,TAG20         Both pages processed?
                STORE           0,COUNTER               Yes Reset Counter.
TAG20           WRITE           0                       Write
```

If a length or timing error is found, the ADDMEMRY, TEST and STORE commands will be bypassed. An error will be displayed on the teleprinter and the error can be corrected (if in error stop mode). The document can be re-processed in the correct sequence as the system is still set to process the corrected document, not the one following it.

WRITE

OUTPUT BUFFER TRANSFER COMMAND

The WRITE command causes the data in the output buffer to be written to tape. The SCRIBE command writes data in the scribe area in the document.

WRITE

Transfers to high performance tape the data assembled in the output record buffer area. Before transfer occurs, the following fields are inserted into the twenty-four (24) character fixed format field.

- Batch Number
- Batch Number Error Flag
- Serial Number Check Digit
- Serial Number
- Page I.D.
- Serial Number Check Digit Error Flag
- Document Error Flag
- Run Number
- Pocket Flag
- Delete Flag
- Worst Mark Intensity Factor
- Invalid Response Count

Systems equipped with 7 or 9 channel ASCII tape units must first translate binary data residing in the buffer to EBCDIC data before the WRITE command is issued. Failure to translate binary data will cause indeterminate data to reside on tape. Control transfers to the first command referenced by the DOSSIER command's "prog" parameter after execution of the WRITE command.

```
WRITE        u

WRITE        O
```

ADDRESS FIELD	DESCRIPTION
u	An integer value between zero and three specifying the address of the tape unit. Must be identical for each WRITE command in the program.

Storage Requirements - One word.

Application

If a user wishes to use the transport printer to print prefix data on the document, fields inserted by the WRITE command must be obtained by accessing storage locations listed in Appendix E. One field that cannot be accessed by addressing storage locations in Appendix E is the serial number check digit field. The following assembly language routine may be used to calculate a check digit.

```
ASSEMBLER: CLASS-II/15N R-3.1  PROGRAM: OTISCKDT                    DATE: 1/30/75                PAGE: 0005

ERRORS LOCN  INST     LABEL     COMMAND       PARAMETERS          COMMENTS                          SEQ NUMB

                                ********************************                                    0094
                                *                              *                                    0095
                                *  CALCULATE SERIAL CHECK DIGIT *                                    0096
                                *                              *                                    0097
                                ********************************                                    0098
                                                                                                    0099
                                PROGRAMMER A. CLARK                                                 0100
                                DATE FEBRUARY 12, 1975                                              0101
                                                                                                    0102
                                FUNCTIONS WITH ANY DOSSIER RELEASE                                  0103
                                THAT IS EQUAL TO OR GREATER THAN                                    0104
                                3.0                                                                 0105
                                                                                                    0106
                                CALLING COMMAND                                                     0107
                                      COMMAND       PARAMETER                                       0108
                                      JUMP,R        SERCKDGT                                        0109
                                                    OUT                                             0110
                                                                                                    0111
                                WHERE                                                               0112
                                SERCKDGT  NAME OF THE ROUTINE THAT                                  0113
                                          COMPUTES THE CHECK DIGIT                                  0114
                                OUT       THE STORAGE LOCATION                                      0115
                                          WHERE THE RESULTS OF                                      0116
                                          THIS ROUTINE ARE STORED.                                  0117
                                          THE CHECK DIGIT IS                                        0118
                                          STORED IN EBCDIC FORMAT                                   0119
                                          IN THE LOWER EIGHT BITS                                   0120
                                          OF THE MEMORY WORD.                                       0121
 21FA  0000     REM                                                                                 0122
                                GENERAL COMMENTS                                                    0123
                                SYMBOLIC ADDRESS USED IN THIS                                       0124
                                ROUTINE MUST NOT APPEAR IN THE                                      0125
                                COMMAND PROGRAM.                                                    0126
                                                                                                    0127
                                THIS ROUTINE SHOULD BE PLACED                                       0128
                                BEFORE THE END STATEMENT. THE                                       0129
                                REMARKS MAY BE OMITTED.                                             0130
                                                                                                    0131
      0089     SERSTR    EQU       089                                                             0132
 21F8 5600               JUMP,U    **                                                              0133
 21FC FFFF
 21FD 8304     SERCKDGT  MOVE,M    4,SERSTR,SERBIN,0F                                              0134
 21FE 0089
 21FF 2222
 2200 000F
 2201 5200               EXIT,U    CALCKDGT                                                        0135
 2202 2203
 2203 B707     CALCKDGT  LDA       $SERCKDGT-1                                                     0136
 2204 9A1A               STA       ADPTR+2                                                         0137
 2205 DE09               IMS       SERCKDGT-1        RTN PTR SET                                   0138
 2206 B218               LDA       SERBIN            MSD OF SERIAL NUMBER                          0139
 2207 1051               ALA       2                 TIMES 4                                       0140
 2208 8A19               ADD       SERBIN            PLUS 1                                        0141
 2209 9A18               STA       SERBIN            ACCUMULATED SUM                               0142
 220A B218               LDA       SERBIN+1          MSD-1 OF SERIAL NUMBER                        0143
 220B 1051               ALA       2                 TIMES 4                                       0144
 220C 8A15               ADD       SERBIN            ADD TO ACCUMULATED SUM                        0145
 220D 9A14               STA       SERBIN                                                          0146
 220E B215               LDA       SERBIN+2          MSD-2 OF SERIAL NUMBER                        0147
 220F 1050               ALA       1                 TIMES 2                                       0148
 2210 8A13               ADD       SERBIN+2          PLUS 1                                        0149
 2211 8A10               ADD       SERBIN            ACCUMULATED SUM                               0150
 2212 9A0F               STA       SERBIN                                                          0151
 2213 B211               LDA       SERBIN+3          LSD OF SERIAL NUMBER                          0152
 2214 1050               ALA       1                 TIMES 2                                       0153
 2215 8A0C               ADD       SERBIN            ACCUMULATED SUM                               0154
 2216 9A08               STA       SERBIN                                                          0155
 2217 F900               REENTER   *+1                                                             0156
 2218 5000
 2219 221A
 221A 9601               ENCODE,B  4,1,SERBIN,SERBIN ENCODE ACCUMULATED SUM                        0157
 221B 2222
 221C 2222
 221D 8201     ADPTR     MOVE      1,SERBIN+3,**     PLACE LSD OF SUM IN USER AREA                 0158
 221E 2225
 221F FFFF
 2220 5000               JUMP,U    SERCKDGT-2        ROUTINE DONE                                  0159
 2221 21F8
 2222          SERBIN    BSS                                                                       0160
```

SCRIBE

Sets a flag to transfer data from the scribe buffer to the transport printer when the WRITE command is executed. Data in the SCRIBE buffer must be packed two characters per word. DOSSIER only executes the last SCRIBE command in the execution path.

This command can only be used when the system includes a transport printer. Otherwise internal status checks will not allow processing to be continued.

SCRIBE,1 SCRIBE,2 SCRIBE,3 SCRIBE,R	nc,addr
SCRIBE,R	32,SCANBUF
7018 users only: SCRIBE,R	28,SCANBUF (The transport printer prints 28 characters maximum on 7018 systems.)

COMMAND	DESCRIPTION
SCRIBE,1	Causes scribe buffer data to be printed on the top third of the document in the area.
SCRIBE,2	Causes scribe buffer data to be printed on the middle third of the document in the scribe area.
SCRIBE,3	Causes scribe buffer data to be printed on the bottom third of the document in the scribe area.
SCRIBE,R	Causes scribe buffer data to be printed on one of three zones in the scribing area (or one of the two zones for 7018 systems). The zone selected is determined by the run number. The run number is entered by the operator at program initialization. A run number of one causes printing on the top third of the document, a run number of two on the middle third of the document and a run number of three or greater on the bottom third of the document. (7018 users enter a run number of either one or two. Caution: Do not enter a run number higher than two on 7018 systems).

PARAMETER	DESCRIPTION
nc	Specifies the number of characters in the scribe buffer. This value must not exceed 32. (For 7018 systems it must not exceed 28.)
addr	Specifies the address of the first word in the SCRIBE buffer. Data in the SCRIBE buffer must be packed two characters per word.

Storage Requirements - Two words

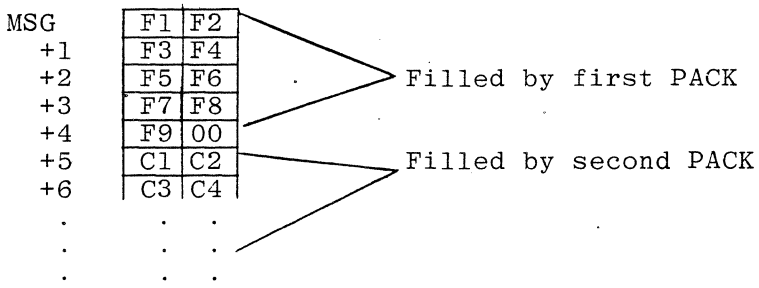
Application

An incorrect method to scribe a nine-digit ID, a blank, and a 12-character last name onto the middle third of each document would be:

```
        PACK,2     9,ID,MSG
        PACK,2     12,LASTNAME,MSG+5
        SCRIBE,2   22,MSG
        .  .
        .
        .
MSG     BSS        11
```

The ID and last name would appear without the intervening blank. This occurs as a result of the first PACK command, where the first four and one half words of MSG would be filled in with the nine-digits.

The lower half word of MSG+4 would be filled with binary zeros since the ID had an odd number of digits.

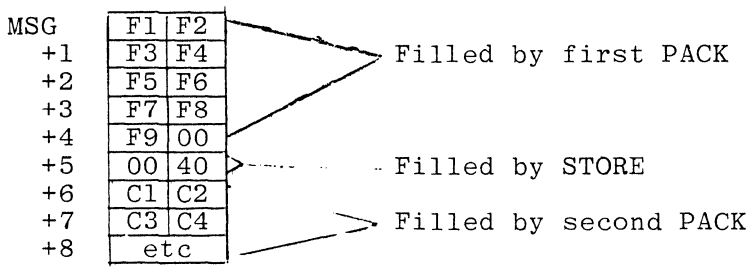


When printed, the 00 is ignored since it is a null character. Therefore, the two fields will not be separated by a blank.

A correct method to scribe the ID + name is as follows. If there is no blank after the ID, one may be placed explicitly as follows:

```
        PACK,2     9,ID,MSG
        STORE      = ,MSG+5
        PACK,2     12,LASTNAME,MSG+6
        SCRIBE,2   24,MSG
        .
        .
        .
MSG     BSS        12
```

The MSG area, after execution of the above code segment, would appear as:



A SCRIBE from this MSG area will produce a 9-digit ID, a blank and a 12-character last name.

CAUTION

On a red error condition, causing the system to halt in Error Stop Mode, the document will not be scribed. In this event if a POCKET,S R,F or U has been executed in the program, the halt is overridden and the document will be scribed and pocketed.

ASSEMBLER PSEUDO OPERATIONS

A pseudo operation is a specialized instruction to the assembler which is not assembled itself but directs a specific segment of the assembly. This section explains the most useful assembler pseudo operations. A complete list may be found in the CLASS Reference Manual.

PROGRAM LINKAGE

Two pseudo-ops are required to begin and end the program.

IDNT

A one to eight character name that identifies the name of the program being assembled. Must be the first command of each program.

```
IDNT        name
```

PARAMETER	DESCRIPTION
name	One to eight character program name that must start with an alphabetic character and must not contain a plus (+), minus (-), comma (,), equals (=), colon (:), ampersand (&) or space ( ).

END

The last source statement in the program signifying to the assembler the end of source statements.

```
END
```

STORAGE ALLOCATION

Data storage is reserved with BSS.

BSS

Reserves storage for a block of data.

```
BSS         n
```

PARAMETER	DESCRIPTION
n	Specifies the number of words of storage that will be required to contain the block.

DATA DEFINITION

Data is specified and stored with an appropriate pseudo-op from this group.

BCD

Inserts into contiguous memory words the EBCDIC character format for each character appearing in the parameter field. Each word contains two EBCDIC characters that are left-justified into the field. If the message contains an odd number of characters, the lower eight bits of the last word containing the message are zero filled.

```
BCD          n,message
```

PARAMETER	DESCRIPTION
n	Specifies the number of characters in the message.
message	The message to be stored. Any valid character appearing in the EBCDIC character set (see Appendix A) may be selected.

BCDR

Inserts into contiguous memory words the EBCDIC character code for each character appearing in the parameter field. Each word containing a character of text stores an EBCDIC character in the lower eight bits of memory word. The upper eight bits contain zeros. Identical to TEXT.

```
BCDR         n,message
```

PARAMETER	DESCRIPTION
n	Specifies the number of characters in the message.
message	The message to be stored. Any valid character appearing in the EBCDIC character set (see Appendix A) may be selected.

(blank)

Causes the expression(s) in the PARAMETER field to be evaluated and stored in consecutive memory locations.

label (blank) $exp_1, exp_2, exp_3, \ldots exp_n$

PARAMETER	DESCRIPTION
$exp_1, exp_2, exp_3, \ldots$	Numeric expressions or literal constants. A numeric expression must be resolvable into a 16-bit constant. The expression may consist of a signed (+ or -) decimal integer, octal number, hexadecimal number, symbolic address or combinations of these.

DEC

Assembles decimal data constants into consecutive storage locations.

DEC a,b,c

PARAMETER	DESCRIPTION
a,b,c,...	Decimal constants 0 - 65535 (2^{16}-1). A character other than the digits 1 - 9 will be flagged (F) and a zero will be substituted. Signs (+ or -) are invalid characters.

HEX

Stores hexadecimal constants in contiguous memory locations. Each group must be separated by a comma. The last group terminates with a blank. Every group fills one memory location. No group may contain more than four hexadecimal characters. Groups containing less than four hexadecimal characters store data right justified with leading zeros in the memory field.

HEX aaaa,bbbb,cccc,---

PARAMETER	DESCRIPTION
aaaa	Groups of hexadecimal characters
bbbb	
cccc	

OCT

This pseudo-op causes the octal numbers in the PARAMETER field to be converted to hexadecimal and stored into consecutive words. Consecutive numbers must be separated by commas. The first blank encountered in the PARAMETER field terminates the field.

OCT $num_1, num_2, num_3, ...num_n$

Only the numeric character 0 through 7 are valid in an octal numnber. A number containing a character other than 0 through 7 will be flagged by CLASS with an 'F' error code and a value of 0 will be substituted for the illegal number. Signed numbers (+ or -) are not allowed under the OCT pseudo-op. Octal to hexadecimal conversion is performed $2^{16}-1$. Numbers larger than this will be converted incorrectly.

If a symbol is present in the LOCATION field, it is assigned to the first word stored by this pseudo-op. The Location Counter will be incremented by 1 for each word stored.

TBL2

Causes the expressions in the parameter field to be assembled, two eight-bit values per word, into consecutive storage locations. If the number of expressions is odd, the last word will contain zero in the lower eight bits.

TBL2 $exp_1,exp_2,exp_3,...exp_n$

PARAMETER	DESCRIPTION
$exp_1,exp_2,exp_3,...$	Numeric expressions or literal constants within the range 0 - 255.

TBL4

Causes the expressions in the parameter field to be assembled, four four-bit values per word, into consecutive storage locations. If the number of expressions is not a multiple of four, the last word will contain 0 in the lower order unused bits.

TBL4 $exp_1,exp_2,exp_3,...exp_n$

PARAMETER	DESCRIPTION
$exp_1,exp_2,exp_3,...$	Numeric expressions or literal constants within the range 0 - 15.

TEXT

Inserts into contiguous memory words the EBCDIC code for each character appearing in the parameter field. Each word contains an EBCDIC character in the lower eight bits. Identical to BCDR.

```
                TEXT            n,message
```

PARAMETER	DESCRIPTION
n	Specifies the number of characters in the message.
messge	The message to be stored. Any valid character appearing in the EBCDIC character set (see Appendix A) may be selected.

ASSEMBLER CONTROL

EQU equates a name to the value of an expression.

EQU

Assigns a value to the tag in the location field.

```
     tag            EQU          exp
```

PARAMETER	DESCRIPTION
exp	Numeric expressions which must be resolvable into a 16-bit constant. The expression may consist of a signed (+ or -) decimal integer, octal number, hexadecimal number, symbolic address or combinations of these. If the name field contains an *, the tag in the location field is assigned the current address of the program counter.

CON AND ORG

The CON and ORG pseudo-ops both perform identical functions and may be used interchangeably. These pseudo-ops set the assembler's Location Counter to a specified value (address).

```
     tag            CON or ORG     exp
```

PARAMETER	DESCRIPTION
exp	Hexadecimal expressions which must be resovable into a 16-bit constant. This expression must be a positive value. See Appendix J for more information.

OUTPUT LISTING CONTROL

The format of the output listing of the assembled program is controlled by this set of pseudo-ops.

EJCT

Causes one or more of the following to occur on the output listing device:

a) Teleprinter
   3 lines are skipped in the listing.

b) Line Printer
   A page eject is performed to the top of the next page and the column headings and page numbers are printed at the top of the new page before normal listing resumes.

```
EJCT
```

EJCT is not printed on the output listing.

NLST

Suppresses listing of the assembled program. May be cancelled by LIST. A NLST encountered while a previous NLST is in effect will be ignored.

```
NLST
```

NLST is not printed on the assembly output listing.

LIST

Causes assembly time program listing to resume following an NLST. If an NLST is not encountered prior to a LIST, LIST is ignored.

```
LIST
```

LIST is not printed on the assembly output listing.

LSYM

Causes the program symbol table to be sorted into alphabetical order and printed at the end of the output listing.

```
LSYM
```

LSYM does not appear on the assembly output listing.

REM

Allows a remark to be displayed on the program listing. Any characters after REM will be printed.

```
REM
```

REM does not appear on the output listing.

REPT

Allows a specified number of input records to be repeated a specified number of times. The input records to be repeated must be immediately after REPT in the execution path.

```
REPT          num1,num2
```

PARAMETER	DESCRIPTION
num1	Specifies the number of times the input record is to be repeated. Must be a decimal number.
num2	Specifies the number of input records to be repeated. Must be a decimal number.

DOSSIER COMMAND RECAP

DOSSIER COMMAND RECAP (CONT.)

DOSSIER COMMAND RECAP (CONT.)

CODING

This chapter demonstrates how to code a DOSSIER program. The self-description inventory was selected because the document is well designed, offers a variety of fields for resolution and requires many of the commands described in the preceding chapter.

DOCUMENT SPECIFICATIONS

Programs are coded to meet the specifications given to the programmer by the test administrator or system analyst. This information must include validation and scoring procedures, and define the content and format of the output record.

VALIDATION PROCEDURES

1) Test the Name grid for left justification. If not left justified, assign the grid an edit flag of N and flag the record.

2) Test the Sex field for completeness. If not complete, assign the field an edit flag of S and flag the field.

3) Test the Special Codes grid for completeness. If not complete, assign the grid an edit flag of C and flag the record.

4) Test the Age grid for completeness. If not complete, assign the grid an edit flag of A and flag the grid.

5) Test the read quality of the 200 items. If they contain 30% or more light marks (mark intensity factor 3 or above) along with omits and multiples, assign an edit flag of Q and flag the field.

6) Test if the skew marks are read. If not read, assign an edit flag of Z and flag the item responses.

SCORING PROCEDURES

All 200 adjectives are to be graded as Y,S or N, with totals obtained for each letter.

POCKET AND SCRIBE PROCEDURES

Sheets are sent to the alternate hopper if a FLAG,G is executed (failure in SEX, AGE or read quality). The record is written to tape. Sheets with records written to tape, which includes selected sheets, are scribed with the contents of SPECIAL CODES and the three totals obtained for Y,S and N.

OUTPUT RECORD FORMAT

Information placement occurs in the output record in the following sequence:

CHARACTER POSITION	DESCRIPTION
25-30	Alphabetic edit flag for the Name grid.
"	Alphabetic edit flag for the Sex field
"	Alphabetic edit flag for the Special Codes grid.
"	Alphabetic edit flag for the Age grid.
"	Alphabetic edit flag for the read quality.
"	Alphabetic edit flag for the skew mark read.
31	Blank
32	Blank
33	Percentage of light marks.
34-35	Total omissions count.
36	Total multiples count.
37	Blank
38-40	Total of items answered Y.
41	Blank
42-44	Total of items answered S.
45	Blank
46-48	Total of items answered N.
49	Blank
50-63	Resolved contents of Name grid.
64	Blank
65	Resolved contents of Sex field
66	Blank
67-76	Resolved contents of Special Codes grid.
77	Blank
78-79	Resolved contents of Age grid.
80	Blank
81-82	Resolved contents of High School response of Last Grade Finished field.
83	Resolved contents of College response of Last Grade Finished field.
84	Resolved contents of Technical response of Last Grade Finished field.
85	Resolved contents of Business School response of Last Grade Finished field.
86	Blank
87-286	Resolved responses of adjective items.

SELF-DESCRIPTION INVENTORY

Form T377

By Dr. Charles B. Johansson — National Computer Systems

MARKING INSTRUCTIONS

1. Please fill in your name and other information on this inventory. Follow the instructions carefully.
2. Use any soft, black lead pencil. Make a heavy, dark mark.
3. If you make a mistake or change your mind, please erase carefully and thoroughly.
4. This inventory will be processed by automatic equipment. To avoid errors, please keep it free from wrinkles and stray marks.
5. Please try to answer each question. Work quickly; First impressions usually give the best results with this inventory.

SPECIAL CODES

NAME GRID DIRECTIONS

Print your name in the boxes. Print your last name first. Skip a box, then print as much of your first name as possible. Blacken the circle below each box that has the same letter as the box. Blacken the blank circles for blank boxes.

NAME

ADDRESS

CITY

STATE

ZIP

SEX

Male

Female

Age

LAST GRADE FINISHED:

High School: 9 10 11 12

College: 1 2 3 4+

Technical: 1 2 3 4+

Business School: 1 2 3 4+

FOR COMPUTER USE ONLY

FOR PROCESSING

Return to

NATIONAL COMPUTER SYSTEMS

4401 West 76th Street

Minneapolis, Minnesota 55435

DIRECTIONS: Below are adjectives used in describing people. Please describe yourself by answering each adjective. There are no right or wrong answers. Be sure to describe yourself accurately. If you make a mistake erase completely. Answer each item.

If the item is always, or usually, true of you, mark the "Y" for "Yes" — (Y) S N

If the item is sometimes true, or you are uncertain, mark the "S" for "Sometimes" — Y (S) N

If the item is not true, or rarely true of you, mark the "N" for "No" — Y S (N)

1 Y S N Bossy
2 Y S N Fussy
3 Y S N Anxious
4 Y S N Outspoken
5 Y S N Trusting
6 Y S N Awkward
7 Y S N Worrying
8 Y S N Gentle
9 Y S N Tidy
10 Y S N Lonely
11 Y S N Capable
12 Y S N Imaginative
13 Y S N Out-going
14 Y S N Excitable
15 Y S N Reliable
16 Y S N Shrewd
17 Y S N Unselfish
18 Y S N Active
19 Y S N Clear-thinking
20 Y S N Friendly

(Continued on next side.)

NCS Trans-Optic F4289:109876

21 Y S N Inventive
22 Y S N Productive
23 Y S N Persisting
24 Y S N Wise
25 Y S N Understanding
26 Y S N Alert
27 Y S N Courteous
28 Y S N Energetic
29 Y S N Forgetful
30 Y S N Industrious
31 Y S N Short-tempered
32 Y S N Thorough
33 Y S N Adaptable
34 Y S N Moody
35 Y S N Poised
36 Y S N Spiritual
37 Y S N Reserved
38 Y S N Careful
39 Y S N Efficient
40 Y S N Competitive
41 Y S N Impatient
42 Y S N Ambitious
43 Y S N Confident
44 Y S N Cautious
45 Y S N Absentminded
46 Y S N Forgiving
47 Y S N Modest
48 Y S N Independent
49 Y S N Logical
50 Y S N Moderate
51 Y S N Pleasant
52 Y S N Sensitive
53 Y S N Resourceful
54 Y S N Spontaneous
55 Y S N Tolerant
56 Y S N Temperamental
57 Y S N Adventurous
58 Y S N Soft-spoken
59 Y S N Frank
60 Y S N Courageous
61 Y S N Demanding
62 Y S N Enthusiastic
63 Y S N Forceful
64 Y S N Humble
65 Y S N Hasty
66 Y S N Optimistic
67 Y S N High-strung
68 Y S N Precise
69 Y S N Polite
70 Y S N Bold
71 Y S N Relaxed
72 Y S N Stubborn
73 Y S N Thrifty
74 Y S N Critical
75 Y S N Clever
76 Y S N Unconventional
77 Y S N Moral
78 Y S N Shy
79 Y S N Organized
80 Y S N Practical
81 Y S N Impulsive
82 Y S N Tactful
83 Y S N Thoughtful
84 Y S N Sociable
85 Y S N Cheerful
86 Y S N Persuasive
87 Y S N Enterprising
88 Y S N Helpful
89 Y S N Hard-working
90 Y S N Original
91 Y S N Stable
92 Y S N Talkative
93 Y S N Humorous
94 Y S N Restless
95 Y S N Generous
96 Y S N Skillful
97 Y S N Argumentative
98 Y S N Nervous
99 Y S N Charming
100 Y S N Easygoing
101 Y S N Aggressive
102 Y S N Serious
103 Y S N Reckless
104 Y S N Withdrawn
105 Y S N Rebellious
106 Y S N Musical
107 Y S N Calm
108 Y S N Quiet
109 Y S N Creative
110 Y S N Religious
111 Y S N Athletic
112 Y S N Self-controlled
113 Y S N Daring
114 Y S N Curious
115 Y S N Scientific
116 Y S N Artistic
117 Y S N Mechanical
118 Y S N Talented
119 Y S N Witty
120 Y S N Tense
121 Y S N Weak
122 Y S N Touchy
123 Y S N Rugged
124 Y S N Sentimental
125 Y S N Complex
126 Y S N Romantic
127 Y S N Expressive
128 Y S N Pessimistic
129 Y S N Cold-hearted
130 Y S N Entertaining
131 Y S N Prompt
132 Y S N Giving
133 Y S N Dissatisfied
134 Y S N Popular
135 Y S N Gifted
136 Y S N Healthy
137 Y S N Dependent
138 Y S N Fun
139 Y S N Grubby
140 Y S N Considerate
141 Y S N Flexible
142 Y S N Changeable
143 Y S N Patient
144 Y S N Unreliable
145 Y S N Open-minded
146 Y S N Careless
147 Y S N Unconcerned
148 Y S N Noisy
149 Y S N Brave
150 Y S N Blunt
151 Y S N Neat
152 Y S N Light-hearted
153 Y S N Bashful
154 Y S N Kind
155 Y S N Nasty
156 Y S N Loyal
157 Y S N Loud
158 Y S N Involved
159 Y S N Sloppy
160 Y S N Sick
161 Y S N Tough
162 Y S N Suspicious
163 Y S N Self-conscious
164 Y S N Conforming
165 Y S N Rough
166 Y S N Sexy
167 Y S N Responsible
168 Y S N Exhausted
169 Y S N Unsure
170 Y S N Graceful
171 Y S N Eager
172 Y S N Progressive
173 Y S N Good-natured
174 Y S N Hostile
175 Y S N Picky
176 Y S N Funny
177 Y S N Happy
178 Y S N Confused
179 Y S N Flirtatious
180 Y S N Perfectionistic
181 Y S N Clumsy
182 Y S N Unreasonable
183 Y S N Orderly
184 Y S N Casual
185 Y S N Unpredictable
186 Y S N Old-fashioned
187 Y S N Carefree
188 Y S N Achieving
189 Y S N Negative
190 Y S N Likable
191 Y S N Bitchy
192 Y S N Nature-lover
193 Y S N Liberal
194 Y S N Afraid
195 Y S N Lazy
196 Y S N Agreeable
197 Y S N Slow
198 Y S N Lively
199 Y S N Insecure
200 Y S N Idealistic

DOSSIER PROGRAMMING SEQUENCE

DOSSIER commands can be coded in any order that completes the resolution and validation task.

Figure 5-1 illustrates the basic programming sequence for any DOSSIER TASK. Documents are identified by the SKUNK command and their x,y limits set by the LIMIT command. The BIAS command dynamically adjusts the system to paper level each time a sheet is scanned. DOSSIER commands resolve, score, validate, and manipulate the data on the identified document. The WRITE command writes the output record to tape, and DOSSIER automatically returns control to the start of the program. When the last document in the batch is validated, the operator terminates the resolution process and closes out the job.

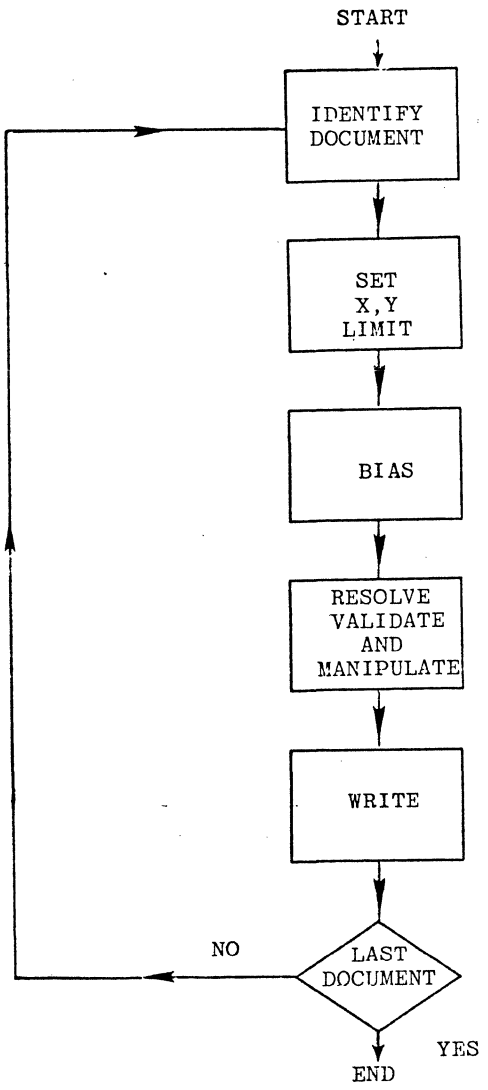


Figure 5-1. DOSSIER Programming Sequence

CODING THE SELF-DESCRIPTION INVENTORY

The last pages in this chapter contain a listing of source statements that resolve and validate the self-description inventory. The source statements shown in this listing fulfill the specifications described earlier in the chapter. The paragraphs that follow explain the purpose of various sets of instructions appearing in the listing.

Figure 5-2 illustrates the coding of the IDNT and END job control statements. The IDNT statement must precede the DOSSIER command and the END statement must follow the last statement in the program.

The IDNT parameter assigns a name to the program. The name must start with an alphabetic character, must not exceed eight characters in length, and must not contain a plus, minus, equals, comma, or space character.

LOCATION	COMMAND	PARAMETER
	IDNT	SELFDISC
	Source Statements	
	END	

Figure 5-2. IDNT and END Usage

EXECUTIVE LINKAGE AND IDENTIFICATION

The DOSSIER,POCKET,SKUNK and LIMITS commands are illustrated in Figure 5-3.

DOSSIER identifies the program as SP, locates the next processing command at BEGIN, places the record between RECB and RECE and specifies the name field as the identifier of the last document processed.

POCKET specifies that FLAG,G failures will be selected. In this case the location of POCKET before SKUNK is not critical, but when the POCKET is for red failures, it must be executed prior to SKUNK to enable documents failing the SKUNK test (a red failure) to go to the secondary hopper.

Documents that fail the SKUNK,U test causes the immediate execution of WRITE. Documents that pass the SKUNK test cause program execution to branch to START.

The LIMIT command limits the maximum value x and y can have in the root response position of a resolution command. It also declares the number of timing marks on a document and specifies the memory requirements for the scanner buffer. Coding which specifies an x or y value that exceeds the values in the LIMIT command causes assembly errors. Documents having an incorrect number of timing marks are flagged as short or long (character position 16 of the output record).

```
        DOSSIER     SD,BEGIN,RECB,RECE,NAME
BEGIN   POCKET,S    A
        SKUNK,U     2,2,1,14,1,I,START
START   LIMIT       47,62
```

Figure 5-3. Document Identification

Figure 5-4 illustrates the BIAS command. BIAS reads the ink and paper of the BIAS bar located at the second timing mark and establishes the paper level for the document.

```
BIAS            2
```

Figure 5-4. BIAS Bar Read.

DOCSN is illustrated in figure 5-5. DOCSN resolves the For Computer Use Only serial number grid. The response value of the middle four columns are checked against the check digit of the first column. This command is skipped when the run number is one.

```
DOCSN,V         21,51,1,1               SERIAL NUMBER RESCORE GRID
```

Figure 5-5. Serial Number Identification

The identification command illustrated by figure 5-6 actually functions as a document read quality indicator. SKUNK assess whether or not the skew marks (the two black circles on either corner opposite the timing track) were read. If they were not read, the document was out of alignment and was read incorrectly. Such a document would be flagged with a Z.

```
SKUNK           2,47,3,47,62,I,ED.7     CHECK SKEW MARKS
FLAG,G          Z,RESP                  SKEW MARKS DID NOT READ
```

Figure 5-6. Skew Mark Check

RESOLUTION

Figure 5-7 illustrates the commands which resolve the fields on the front of the document. The statements are organized according to completion of scan. For example, Special Codes is resolved before Name because it passes under the read head first.

High School (in the Last Grade Finished field) is resolved separately from the other school choices because it must be expanded to two digits which requires a separate data manipulation. Front page resolution continues with Sex, Age, and the remaining school choices. It is concluded with the item column of adjectives.

The Sex field is resolved as a nonuniform item having two symmetrical response sets. Although the item could have been resolved with a single RESPONSE command, the use of two commands allows the record to display an F or M without using an XLATE command. The =F or =M "iv" assigns F or M to the SEX field of the output record.

Special Codes and Age are resolved with NUMERIC commands and Name with ALPHA,V. High School is resolved as a single item by RESPONSE,S and the "iv" of 6 will store a binary number from 6 through 12 in HS. HS is converted to EBCDIC by ENCODE. The remaining items in Last Grade Finished are resolved as one horizontal item column by ITEM,H. The results are stored in CO. The first 20 adjectives form an item column resolved by ITEM,H.

```
NUMERIC,V       10,10,33,7,1,1,SPCDE
ALPHA,V         27,14,13,20,1,1,NAME
RESPONSE        1,=M,9,40,0,0
RESPONSE,S      1,=F,9,44,0,0,SEX
NUMERIC,V       10,2,8,51,1,1,AGE
RESPONSE,S      7,6,14,54,0,1,HS      HIGH SCHOOL OF LGF
ITEM,V          7,3,15,54,1,1,CO      REMAINDER OF LGF
ITEM,H          3,20,31,41,1,1,RESP
```

Figure 5-7. Resolution Commands - Front.

Figure 5-8 illustrates the two commands needed to resolve the back of the document. The columns are not uniformly spaced and cannot be resolved with one ARRAY. The ARRAY,H command resolves items 21 through 140 and the responses are stored in RESP+20. ITEM,H resolves items 141 through 200 with storage in RESP+140.

```
ARRAY,H     3,60,2,45,3,-1,1,-16,0,RESP+20
ITEM,H      3,60,12,3,-1,1,RESP+140
```

Figure 5-8. Resolution Commands - Back

VALIDATION

Figure 5-9 illustrates the biographical data validation commands used in the SELFDISC program. EDIT,L tests the Name field for left justification and transfers control to the instruction at location ED.1 if the field is left justified. If it is not, FLAG,R will set an N flag in the record. EDIT,C tests the Sex field for completeness, transferring control to ED.2 if complete and to FLAG,G, which flags the field, if not. The remaining commands cause similar editing.

```
            EDIT,L      14,NAME,ED.1
            FLAG,R      N,NAME
ED.1        EDIT,C      1,SEX,ED.2
            FLAG,G      S,SEX
ED.2        EDIT,C      10,SPCDE,ED.3
            FLAG,R      C,SPCDE
ED.3        EDIT,C      2,AGE,ED.4
            FLAG,G      A,AGE
```

Figure 5-9. Biographical Data Validation

Figure 5-10 illustrates the scoring validation commands. GRADE,4 compares the 200 item response values in RESP to the right response table YTAB and stores the result in YSCRE. EVALUATE,T retrieves the count of total omissions and multiples and the % of light marks from the data stored internally by GRADE,4 and stores the data in INVCNT. EVALUATE is executed at this point, before the following two GRADE commands, because each of those commands would add a complete set of data, creating a result 3 times the actual number.

Two TEST commands are executed here for the same reason. The TEST,G tests the mark intensity factor stored in DOCUFACT in binary form (see page 4-53) and transfers to ED.6 if the mark intensity factor is less than 5. If not, then the % of light marks is 30% or greater and TEST,E tests the contents of storage location OB3, the invalid response count, (see Appendix E, page E-3). If the contents is 0, control is transferred to ED.6. If not 0, then both conditions of flag Q have been met and FLAG,G flags the field.

The two TEST commands displayed two alternatives for referencing the field to be compared. TEST,G referenced a location which used EQU to equate that location to the actual storage location OAD. TEST,E directly referenced the storage location OB3. Both techniques are equally valid.

The two remaining GRADE,4 commands count the S and N responses.

```
ED.4    GRADE,4         200,RESP,YTAB,YSCRE   ITEMS ANSWERED Y
        EVALUATE,T      INVCNT                OUTPUT OMIT & MULT COUNTS
        TEST,G          5,DOCUFACT,ED.5       TEST FOR 30 PER CENT LIGHT
        TEST,E          0,0B3,ED.5            TEST FOR ANY INVALIDS
        FLAG,G          Q,RESP                FLAG READ QUALITY
ED.5    GRADE,4         200,RESP,STAB,SSCRE   ITEMS ANSWERED S
        GRADE,4         200,RESP,NTAB,NSCRE   ITEMS ANSWERED N
```

Figure 5-10. Scoring Validation Commands

DATA MANIPULATION

Figure 5-11 illustrates the data manipulation commands for SELFDISC. XLATE translates the 200 adjective response values of RESP according to the translation table RESPTAB. Omits are translated to a blank, multiples to *, response positions of 1 to Y, of 2 to S, and of 3 to N. The translations replace the originals in storage. The second XLATE translates the school choices other than High School into values of 1 through 6 in the same manner.

ENCODE,R translates the binary number 6 through 12 stored as the response to High School into a 2 digit EBCDIC decimal number. Omits will be translated to - and multiples to *.

REPLACE,A replaces an EBCDIC - (omit) with a blank for any of the 33 characters of the Name, Sex, Spcde, Age, and HS fields. Note that this was executed after the HS field was changed to EBCDIC. Coding time was saved by organizing the output record with these fields continguous which allowed them to be changed with one REPLACE command.

```
XLATE       200,RESP,RESPTAB     TRANSLATE RESPONSES
ENCODE,R    2,1,HS,HS            OUTPUT 2 DIGIT HIGH SCHOOL
XLATE       3,CO,SCHTAB          TRANSLATE LAST GRADE FINISHED
REPLACE,A   35.=-,= ,NAME        CONVERT GRID OMITS TO SPACES
```

Figure 5-11 Data Manipulation Commands.

DATA TRANSFER

Figure 5-12 illustrates the SCRIBE and WRITE commands.

Messages to be scribed must be in packed format. The output of Spcde and the Y,S and N counts are packed into the scribe message.

A program will not run without the WRITE command which transfers data from the output record buffer to tape.

```
ED,7    PACK,2      10,SPCDE,MSG
        PACK,2      12,YSCRE,MSG+5
        SCRIBE,R    22,MSG
        WRITE       0
```

Figure 5-12 SCRIBE and WRITE Commands.

OUTPUT RECORD

Figure 5-13 illustrates how to allocate storage for the output record buffer and the constants required in the program. EJCT will not appear in the assembly listing. Instead it will cause a "top of page" function at that point in the listing. REPT saves keying time as the record following it will be repeated 10 times to provide a 200 character right response table. A description of the operation codes used in this section appears in Appendix C.

```
         EJCT
RECB     BSS     24                  STANDARD DOSSIER FRONT END
FLAGS    BSS     7                   EDIT FLAGS
INVCNT   BSS     6                   TOTAL OMMISSIONS + MULTIPLES
YSCRE    BSS     4                   ITEMS ANSWERED Y
SSCRE    BSS     4                   ITEMS ANSWERED S
NSCRE    BSS     4                   ITEMS ANSWERED N
NAME     BSS     15
SEX      BSS     2
SPCDE    BSS     11
AGE      BSS     3
HS       BSS     2                   LAST GRADE HIGH SCHOOL
CO       BSS     1                   LAST YEAR COLLEGE
TECH     BSS     1                   LAST YEAR TECHNICAL SCHOOL
BUSS     BSS     2                   LAST YEAR BUSINESS SCHOOL
RESP     BSS     200                 ITEM RESPONSES
RECE     BSS     1
         EJCT
RESPTAB  BCD     5, *YSN             RESPONSE TRANSLATION TABLE
SCHTAB   BCD     9, *1234567         LAST GRADE TRANSLATION TABLE
MSG      BSS     11
YTAB     REPT    10,1
         HEX     1111,1111,1111,1111,1111
STAB     REPT    10,1
         HEX     2222,2222,2222,2222,2222
NTAB     REPT    10,1
         HEX     3333,3333,3333,3333,3333
DOCUFACT EQU     0AD
         END
```

Figure 5-13. Declaration of Storage

Figure 5-14 illustrates how to interpret an assembly listing.

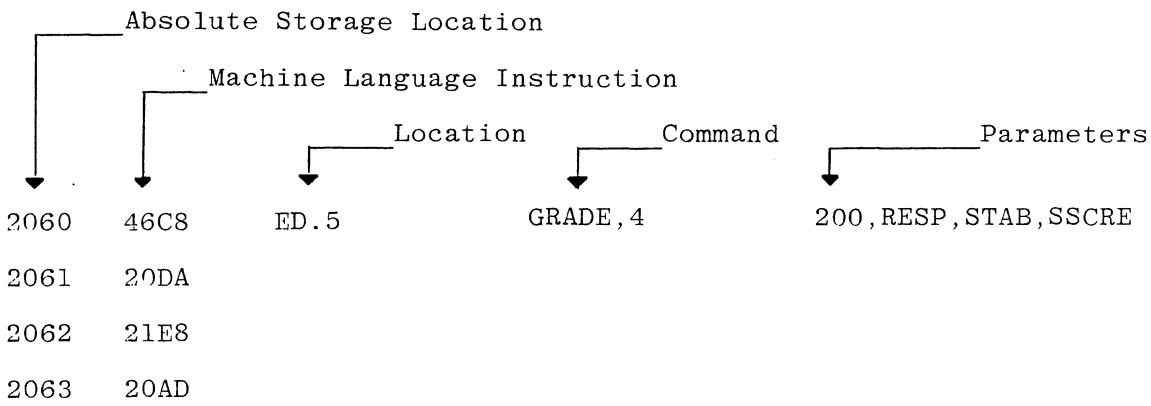
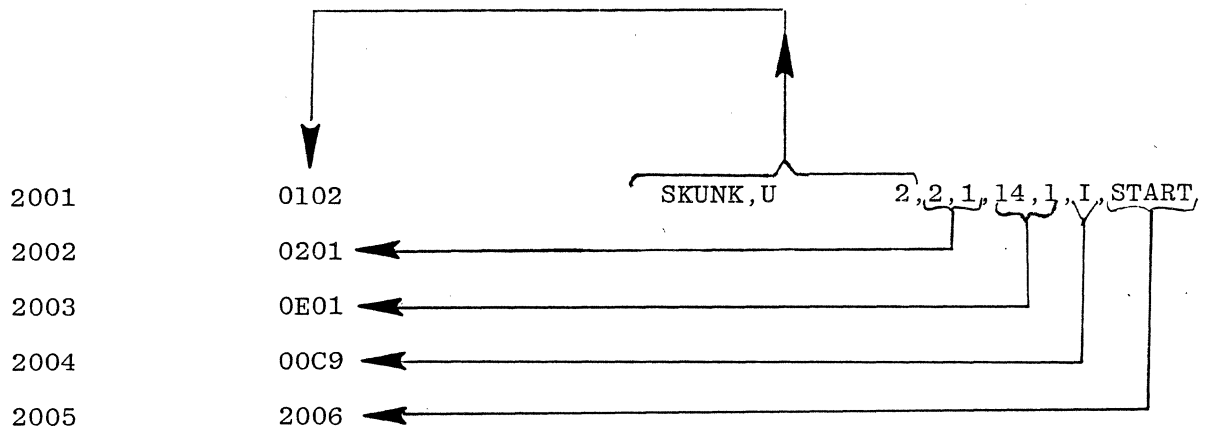


Figure 5-14. Sample Assembly Listing

Figure 5-15 illustrates the tabular format for the SKUNK command and shows how the assembler expands the command. The assembler expands other DOSSIER commands in a similar manner. The user should refer to Appendix B to identify the parameter locations for the DOSSIER command in question.



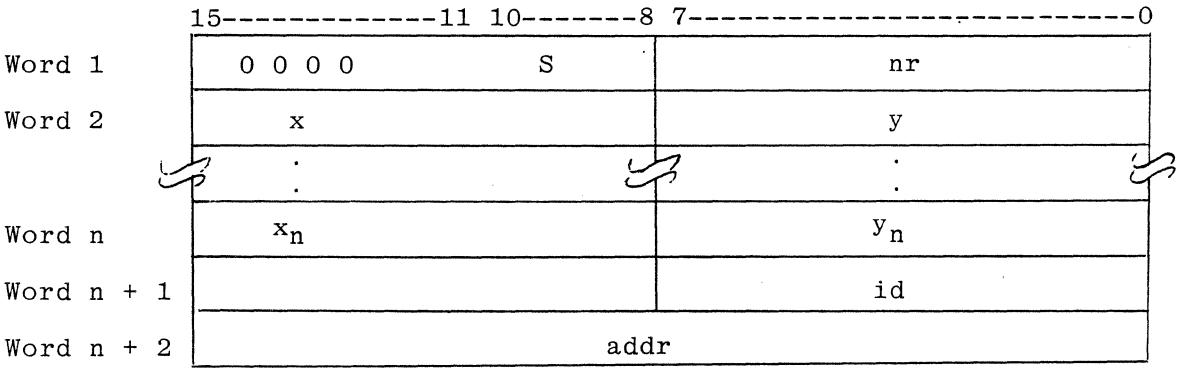


Figure 5-15. Expansion of the SKUNK Command.

```
ERRORS LOCN  INST      LABEL   COMMAND      PARAMETERS          COMMENTS                                          SEQ NUMB
                               IDNT         SELFDISC                                                              0001
                                                                                                                  0002
                               THIS PROGRAM WILL RESOLVE THE SELF DESCRIPTION                                     0003
                               INVENTORY AND WRITE A FORMATTED RECORD TO TAPE                                     0004
                                                                                                                  0005
                               EDIT FLAGS:                                                                        0006
                                         N - NAME GRID IS NOT LEFT JUSTIFIED                                      0007
                                         S - SEX IS NOT FILLED IN                                                 0008
                                         C - SPECIAL CODES GRID IS NOT COMPLETE                                   0009
                                         A - AGE IS NOT COMPLETE                                                  0010
                                         Q - ITEMS CONTAIN 30 PER CENT LIGHT                                      0011
                               MARKS AND THERE ARE MULTS OR OMITS                                                 0012
                                         Z - SKEW MARKS DID NOT READ                                              0013
                                                                                                                  0014
                               OMITTED ITEMS ARE TRANSLATED TO SPACES                                             0015
                               MULTIPLE MARKS ARE TRANSLATED TO ASTERISKS                                         0016
                               SEX IS TRANSLATED TO:                                                              0017
                                                   M - MALE                                                       0018
                                                   F - FEMALE                                                     0019
                               ITEMS ARE TRANSLATED TO Y S N                                                      0020
                               AN INVALID ITEM COUNT OF TOTAL OMMISSIONS AND                                      0021
                               MULTIPLES IS OUTPUT                                                                0022
                               A COUNT OF ITEMS ANSWERED Y, S, AND N IS COMPUTED                                  0023
                                                                                                                  0024
                               DOCUMENTS WITH FLAGGED ONLY EDIT ERRORS ARE SELECTED.                              0025
                               DOCUMENTS WRITTEN TO TAPE ARE SCRIBED WITH THE                                     0026
                               CONTENTS OF SPECIAL CODES AND THE Y, N AND S COUNTS.                               0027
                                                                                                                  0028
                                                                                                                  0029
                                                                                                                  0030
0002  00E2                     DOSSIER      9D,BEGIN,RECB,RECE,NAME                                               0031
0003  00C4
0004  2000
0005  2084
0006  21A2
0007  2085
      2000
2000  C001     BEGIN           POCKET,S     A                                                                     0032
2001  0102                     SKUNK,U      2,2,1,14,1,I,START                                                    0033
2002  0201
2003  0C01
2004  00C9
2005  2006
      002F     START           LIMIT        47,62                                                                 0034
      003C
2006  083E
2007  0902                     BIAS         2                                                                     0035
2008  002F
2009  2A04                     NUMERIC,V    10,10,33,7,1,1,SPCDE                                                  0036
200A  000A
```

```
ERRORS LOCN  INST      LABEL     COMMAND       PARAMETERS                  COMMENTS                    SEQ NUMB
       2008  2107
       200C  0101
       200D  20C6
       200E  2818                ALPHA,V       27,14,13,20,1,1,NAME                                    0037
       200F  000E
       2010  0D14
       2011  0101
       2012  2085
       2013  1801                RESPONSE      1,=M,9,40,0,0                                           0038
       2014  4004
       2015  0928
       2016  0000
       2017  1901                RESPONSE,S    1,=F,9,44,0,0,SEX                                       0039
       2018  40C6
       2019  092C
       201A  0000
       201B  20C4
       201C  2A0A                NUMERIC,V     10,2,8,51,1,1,AGE                                       0040
       201D  0002
       201E  0833
       201F  0101
       2020  20D1
       2021  1907                RESPONSE,S    7,6,14,54,0,1,HS      HIGH SCHOOL OF LGF                0041
       2022  0006
       2023  0E36
       2024  0001
       2025  2004
       2026  2007                ITEM,V        7,3,15,54,1,1,CO      REMAINDER OF LGF                  0042
       2027  0003
       2028  0F36
       2029  0101
       202A  2006
       202B  1400                DOCSN,V       21,51,1,1             SERIAL NUMBER RESCORE GRID        0043
       202C  1533
       202D  0101
       202E  2103                ITEM,H        3,20,31,41,1,1,RESP                                     0044
       202F  0014
       2030  1F29
       2031  0101
       2032  20DA
       2033  2303                ARRAY,H       3,60,2,45,3,-1,1,-16,0,RESP+20                          0045
       2034  023C
       2035  2003
       2036  FF01
       2037  F000
       2038  20EC
       2039  2103                ITEM,H        3,60,12,3,-1,1,RESP+140                                 0046
       203A  003C
       203B  0C03
       203C  FF01
```

```
ERRORS LOCN  INST     LABEL   COMMAND       PARAMETERS                COMMENTS                        SEQ NUMB
       2030  2160
       203E  710C             EDIT,L        14,NAME,ED.1                                              0047
       203F  2085
       2040  2043
       2041  7905             FLAG,R        N,NAME                                                    0048
       2042  2035
       2043  7301     ED.1    EDIT,C        1,SEX,ED.2                                                0049
       2044  20C4
       2045  2048
       2046  78E2             FLAG,G        3,SEX                                                     0050
       2047  20C4
       2048  730A     ED.2    EDIT,C        10,SPCDE,ED.3                                             0051
       2049  20C6
       204A  2040
       2048  79C3             FLAG,R        C,SPCDE                                                   0052
       204C  20C6
       204D  7302     ED.3    EDIT,C        2,AGE,ED.4                                                0053
       204E  2001
       204F  2052
       2050  78C1             FLAG,G        A,AGE                                                     0054
       2051  2001
       2052  46C8     ED.4    GRADE,4       200,RESP,YTAB,YSCRE       ITEMS ANSWERED Y                0055
       2053  20DA
       2054  2186
       2055  20A9
       2056  4900             EVALUATE,T    INVCNT                    OUTPUT OMIT & MULT COUNTS       0056
       2057  20A3
       2058  8A05             TEST,G        5,DOCUFACT,ED.5           TEST FOR 30 PER CENT LIGHT      0057
       2059  00AD
       205A  2060
       2058  9800             TEST,E        0,083,ED.5                TEST FOR ANY INVALIDS           0058
       205C  0085
       205D  2060
       205E  78D8             FLAG,G        Q,RESP                    FLAG READ QUALITY               0059
       205F  20DA
       2060  46C8     ED.5    GRADE,4       200,RESP,STAB,SSCRE       ITEMS ANSWERED S                0060
       2061  20DA
       2062  21E0
       2063  20AD
       2064  46C8             GRADE,4       200,RESP,NTAB,NSCRE       ITEMS ANSWERED N                0061
       2065  20DA
       2066  221A
       2067  2081
       2068  30C8             XLATE         200,RESP,RESPTAB          TRANSLATE RESPONSES             0062
       2069  20DA
       206A  21A3
       2068  9101             ENCODE,R      2,1,HS,HS                 OUTPUT 2 DIGIT HIGH SCHOOL      0063
       206C  2004
       206D  2004
       206E  3005             XLATE         5,CD,SCHTAB               TRANSLATE LAST GRADE FINISHED   0064
```

```
ERRORS LOCN  INST      LABEL    COMMAND       PARAMETERS            COMMENTS                        SEQ NUMB

       206F  20D6
       2070  21A6
       2071  5823               REPLACE,A     35,=-,= ,NAME         CONVERT GRID OMITS TO SPACES    0065
       2072  6040
       2073  2085
       2074  0002               SKUNK         2,47,3,47,62,I,ED.7   CHECK SKEW MARKS                0066
       2075  2F03
       2076  2F3E
       2077  00C9
       2078  2078
       2079  7BE9               FLAG,G        Z,RESP                SKEW MARKS DID NOT READ         0067
       207A  20DA
       207B  A10A       ED.7    PACK,2        10,SPCDE,MSG                                          0068
       207C  20C6
       207D  21AB
       207E  A10C               PACK,2        12,YSCRE,MSG+5                                        0069
       207F  20A9
       2080  2180
       2081  AC16               SCRIBE,R      22,MSG                                                0070
       2082  21AB
       2083  5810               WRITE         0                                                     0071
```

ERRORS	LOCN	INST	LABEL	COMMAND	PARAMETERS	COMMENTS	SEQ NUMB
	2084		RECB	BSS	24	STANDARD DOSSIER FRONT END	0073
	209C		FLAGS	BSS	7	EDIT FLAGS	0074
	20A3		INVCNT	BSS	6	TOTAL OMMISSIONS + MULTIPLES	0075
	20A9		YSCRE	BSS	4	ITEMS ANSWERED Y	0076
	20AD		SSCRE	BSS	4	ITEMS ANSWERED S	0077
	20B1		NSCRE	BSS	4	ITEMS ANSWERED N	0078
	20B5		NAME	BSS	15		0079
	20C4		SEX	BSS	2		0080
	20C6		SPCDE	BSS	11		0081
	20D1		AGE	BSS	3		0082
	20D4		HS	BSS	2	LAST GRADE HIGH SCHOOL	0083
	20D6		CO	BSS	1	LAST YEAR COLLEGE	0084
	20D7		TECH	BSS	1	LAST YEAR TECHNICAL SCHOOL	0085
	20D8		BUSS	BSS	2	LAST YEAR BUSINESS SCHOOL	0086
	20DA		RESP	BSS	200	ITEM RESPONSES	0087
	21A2		RECE	BSS	1		0088

```
ERRORS LOCN  INST      LABEL     COMMAND   PARAMETERS            COMMENTS                        SEQ NUMB
       21A3  405C      RESPTAB   BCD       5, *YSN               RESPONSE TRANSLATION TABLE      0090
       21A4  CBE2
       21A5  D500
       21A6  405C      SCHTAB    BCD       9, *1234567           LAST GRADE TRANSLATION TABLE    0091
       21A7  F1F2
       21A8  F3F4
       21A9  F5F6
       21AA  F700
       21AB            MSG       BSS       11                                                    0092
             000A      YTAB      REPT      10,1                                                  0093
       21B6  1111                HEX       1111,1111,1111,1111,1111                            0094
       21B7  1111
       21B8  1111
       21B9  1111
       21BA  1111
       21BB  1111                HEX       1111,1111,1111,1111,1111                            0094
       21BC  1111
       21BD  1111
       21BE  1111
       21BF  1111
       21C0  1111                HEX       1111,1111,1111,1111,1111                            0094
       21C1  1111
       21C2  1111
       21C3  1111
       21C4  1111
       21C5  1111                HEX       1111,1111,1111,1111,1111                            0094
       21C6  1111
       21C7  1111
       21C8  1111
       21C9  1111
       21CA  1111                HEX       1111,1111,1111,1111,1111                            0094
       21CB  1111
       21CC  1111
       21CD  1111
       21CE  1111
       21CF  1111                HEX       1111,1111,1111,1111,1111                            0094
       21D0  1111
       21D1  1111
       21D2  1111
       21D3  1111
       21D4  1111                HEX       1111,1111,1111,1111,1111                            0094
       21D5  1111
       21D6  1111
       21D7  1111
       21D8  1111
       21D9  1111                HEX       1111,1111,1111,1111,1111                            0094
       21DA  1111
       21DB  1111
       21DC  1111
       21DD  1111
```

ERRORS	LOCN	INST	LABEL	COMMAND	PARAMETERS	COMMENTS	SEQ NUMB
	21DE	1111		HEX	1111,1111,1111,1111,1111		0094
	21DF	1111					
	21E0	1111					
	21E1	1111					
	21E2	1111					
	21E3	1111		HEX	1111,1111,1111,1111,1111		0094
	21E4	1111					
	21E5	1111					
	21E6	1111					
	21E7	1111					
		000A	STAB	REPT	10,1		0095
	21E8	2222		HEX	2222,2222,2222,2222,2222		0096
	21E9	2222					
	21EA	2222					
	21EB	2222					
	21EC	2222					
	21ED	2222		HEX	2222,2222,2222,2222,2222		0096
	21EE	2222					
	21EF	2222					
	21F0	2222					
	21F1	2222					
	21F2	2222		HEX	2222,2222,2222,2222,2222		0096
	21F3	2222					
	21F4	2222					
	21F5	2222					
	21F6	2222					
	21F7	2222		HEX	2222,2222,2222,2222,2222		0096
	21F8	2222					
	21F9	2222					
	21FA	2222					
	21FB	2222					
	21FC	2222		HEX	2222,2222,2222,2222,2222		0096
	21FD	2222					
	21FE	2222					
	21FF	2222					
	2200	2222					
	2201	2222		HEX	2222,2222,2222,2222,2222		0096
	2202	2222					
	2203	2222					
	2204	2222					
	2205	2222					
	2206	2222		HEX	2222,2222,2222,2222,2222		0096
	2207	2222					
	2208	2222					
	2209	2222					
	220A	2222					
	220B	2222		HEX	2222,2222,2222,2222,2222		0096
	220C	2222					
	220D	2222					
	220E	2222					

ERRORS	LOCN	INST	LABEL	COMMAND	PARAMETERS	COMMENTS	SEQ NUMB
	220F	2222					
	2210	2222		HEX	2222,2222,2222,2222,2222		0096
	2211	2222					
	2212	2222					
	2213	2222					
	2214	2222					
	2215	2222		HEX	2222,2222,2222,2222,2222		0096
	2216	2222					
	2217	2222					
	2218	2222					
	2219	2222					
		000A	NTAB	REPT	10,1		0097
	221A	3333		HEX	3333,3333,3333,3333,3333		0098
	221B	3333					
	221C	3333					
	221D	3333					
	221E	3333					
	221F	3333		HEX	3333,3333,3333,3333,3333		0098
	2220	3333					
	2221	3333					
	2222	3333					
	2223	3333					
	2224	3333		HEX	3333,3333,3333,3333,3333		0098
	2225	3333					
	2226	3333					
	2227	3333					
	2228	3333					
	2229	3333		HEX	3333,3333,3333,3333,3333		0098
	222A	3333					
	222B	3333					
	222C	3333					
	222D	3333					
	222E	3333		HEX	3333,3333,3333,3333,3333		0098
	222F	3333					
	2230	3333					
	2231	3333					
	2232	3333					
	2233	3333		HEX	3333,3333,3333,3333,3333		0098
	2234	3333					
	2235	3333					
	2236	3333					
	2237	3333					
	2238	3333		HEX	3333,3333,3333,3333,3333		0098
	2239	3333					
	223A	3333					
	223B	3333					
	223C	3333					
	223D	3333		HEX	3333,3333,3333,3333,3333		0098
	223E	3333					
	223F	3333					

```
ERRORS LOCN  INST      LABEL     COMMAND   PARAMETERS              COMMENTS                SEQ NUMB.

       2240  3333
       2241  3333
       2242  3333                HEX       3333,3333,3333,3333,3333                      0098
       2243  3333
       2244  3333
       2245  3333
       2246  3333
       2247  3333                HEX       3333,3333,3333,3333,3333                      0098
       2248  3333
       2249  3333
       224A  3333
       2248  3333
             00A9      DOCUFACT  EQU       0A9                                           0099
                               END                                                       0100
       0008  3C50
       0009  3FFF
       000A  0000
       000B  0010
0000 ERRORS
```

APPENDIX A
INTERNAL EBCDIC CHARACTER SET

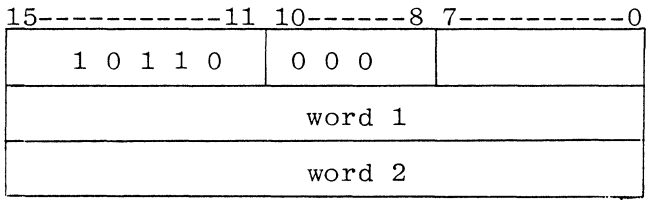
Bit Positions 0, 1, 2, 3	Second Hexadecimal Digit	Quad 1				Quad 2				Quad 3				Quad 4				
		00				01				10				11				Bit Positions 6, 7
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 4, 5
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
0000	0					SP	&	-									0	
0001	1							/						A	J		1	
0010	2													B	K	S	2	
0011	3													C	L	T	3	
0100	4													D	M	U	4	
0101	5													E	N	V	5	
0110	6													F	O	W	6	
0111	7													G	P	X	7	
1000	8													H	Q	Y	8	
1001	9													I	R	Z	9	
1010	A					¢	!		:									
1011	B					.	$	,	#									
1100	C					<	*	%	@									
1101	D					(	)	_	'									
1110	E					+	;	>	=									
1111	F					|	¬	?	"									

NOTE: SP = a space.

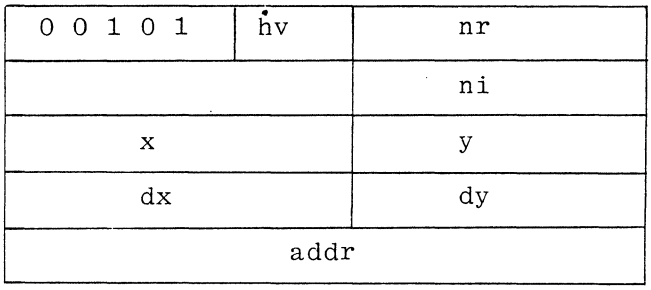
APPENDIX B
COMMAND EXPANSIONS

The following lists the binary expansion for each DOSSIER command instruction. For convenience, each command is listed alphabetically and an application is included at the end of the list.

ADDMEMRY ADDMEMRY word1,word2



ALPHA ALPHA,V or H nr,ni,x,y,dx,dy,addr



The hv field contains:

000 when ALPHA,V is coded.

001 when ALPHA,H is coded.

where

nr = number of responses
ni = number of items
nw = number or words
nc = number of columns

ARRAY

ARRAY,V or H nr,ni,nc,x,y,dx,dy,cdx,cdy,addr

	15--------11	10----8	7----------0
Word 1	0 0 1 0 0	hv	nr
Word 2	nc		ni
Word 3	x		y
Word 4	dx		dy
Word 5	cdx		cdy
Word 6	addr		

The hv field contains:

010 when ARRAY,V is coded.
011 when ARRAY,H is coded.

BATCHNO

BATCHNO,V or H x,y,dx,dy

	15--------11	10----8	7----------0
Word 1	0 0 0 1 0	hv	
Word 2	x		y
Word 3	dx		dy

The hv field contains:

000 when the BATCHNO,V is coded.
001 when the BATCHNO,H is coded.

BIAS

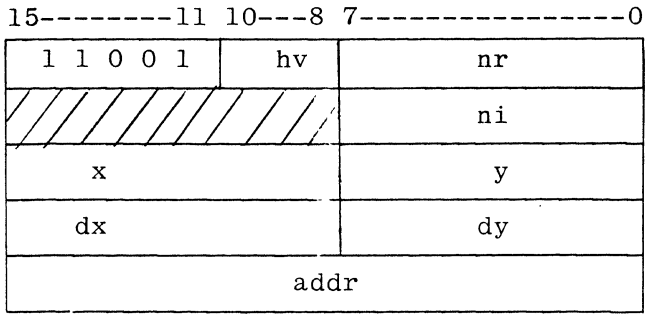
BIAS t

	15--------11	10----8	7----------0
	0 0 0 0 1	0 0 1	t
	//////////		n

The n field is filled by the x parameter in the LIMIT command.

BINARY

BINARY,H or V nr,ni,x,y,dx,dy,addr

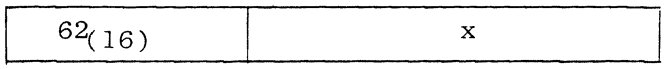


The hv field contains:

000 when BINARY,V is coded.
001 when BINARY,H is coded.

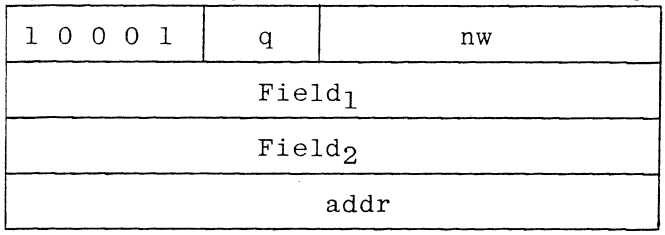
BOOKSN

BOOKSN x



COMPARE

COMPARE,EorNorGorL nw,fieldl,field2,addr



The q field contains:

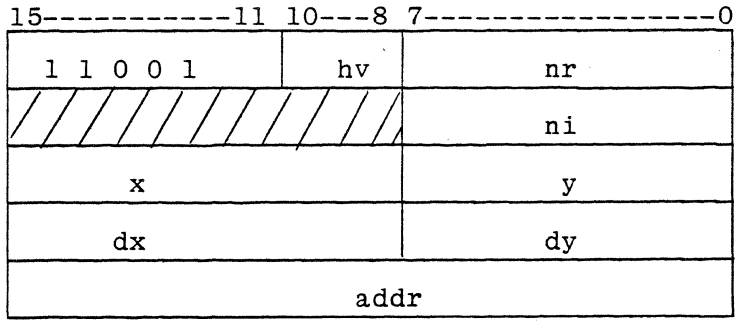
100 when COMPARE,E is coded.
101 when COMPARE,N is coded.
110 when COMPARE,G is coded.
111 when COMPARE,L is coded.

CONTROL

See END. The addresses for the CONTROL routine are stored in locations 0026 and 0027. They appear with the END statement.

DECIMAL

DECIMAL,V or H nr,ni,x,y,dx,dy,addr

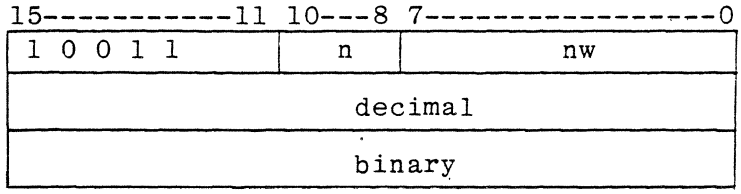


Where the hv field contains:

010 when DECIMAL,V is coded.
011 when DECIMAL,H is coded.

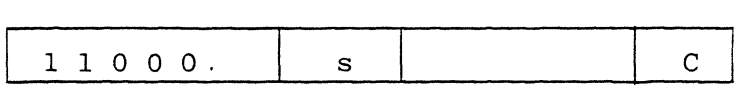
DECODE

DECODE nw,decimal,binary



DELETE

DELETE,S or C R or A or F or U



The s field contains:

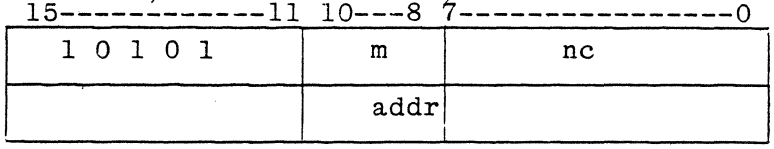
010 when DELETE,S is coded.
011 when DELETE,C is coded.

The C field contains:

001 when the A parameter is coded.
010 when the R parameter is coded.
011 when the F parameter is coded.
100 when the U parameter is coded.

DISPLAY.

DISPLAY,C or W nc,addr



The m field contains:
010 when the DISPLAY,C is coded.
011 when the DISPLAY,W is coded.

Here nc means number of characters.

DOCSN

DOCSN,H or V x,v,dx,dy

	15---11	10---8	7---0
Word 1	1 0 0 1 0	n	ni
Word 2	x		y
Word 3	dx		dy

The hv field contains:
100 when the DOCSN,V is coded.
101 when the DOCSN,H is coded.

DOSSIER

DOSSIER id,prog,rfwa,rlwa,rida

	15---8	7---0
LOCN 02		1st char id
LOCN 03		2nd char id
LOCN 04	prog	
LOCN 05	rfwa	
LOCN 06	rlwa	
LOCN 07	rida	

END

END

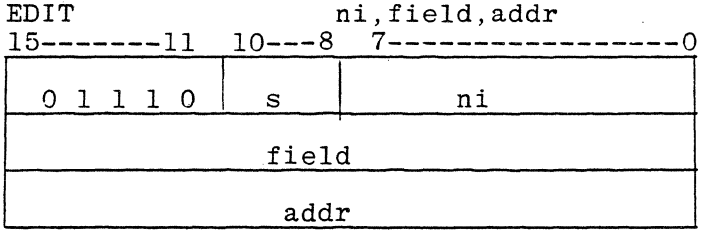
	15---8	7---4	3---0
LOCN 08	Document Image Buffer Start Addr.		
LOCN 09	Document Image Buffer End Addr.		
LOCN 0A		(unused)	
LOCN 0B		0001	tu
*LOCN 26	CONTROL,B Routine Start Address		
*LOCN 27	CONTROL,E Routine Start Address		

* Appears only if CONTROL is coded.

The tu field contains:
0000 when WRITE unit is 0.
0001 when WRITE unit is 1.
0010 when WRITE unit is 2.
0011 when WRITE unit is 3.

EDIT

EDIT ni,field,addr

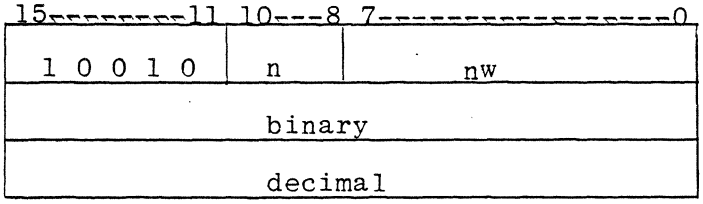


The s field contains:

000 when EDIT,0 is coded.
001 when EDIT,L is coded.
010 when EDIT,R is coded.
011 when EDIT,C is coded.

ENCODE

ENCODE,R or B n,nw,binary,decimal

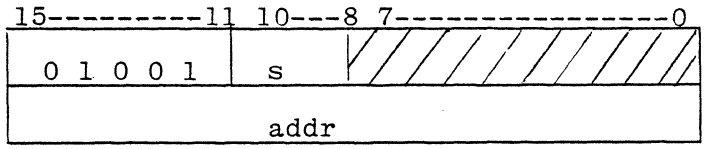


The n field contains:

000 when the ENCODE,R is coded and n is 1.
001 when the ENCODE,R is coded and n is 2.
010 when the ENCODE,R is coded and n is 3.
011 when the ENCODE,B is coded and n is 1.
100 when the ENCODE,B is coded and n is 2.
101 when the ENCODE,B is coded and n is 3.
110 when the ENCODE,B is coded and n is 4.
111 when the ENCODE,B is coded and n is 5.

EVALUATE

EVALUATE,I or T addr

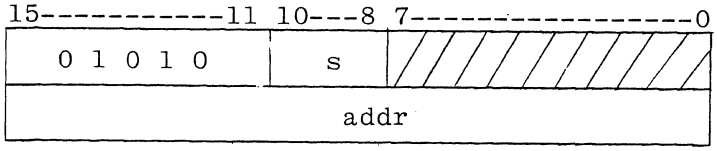


The s field contains:

000 when EVALUATE,I is coded.
001 when EVALUATE,T is coded.

EXIT

EXIT,U or R addr

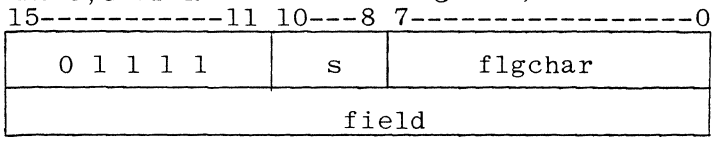


The s field contains:

010 when EXIT,U is coded.
011 when EXIT,R is coded.

FLAG

FLAG,G or R flgchar,field

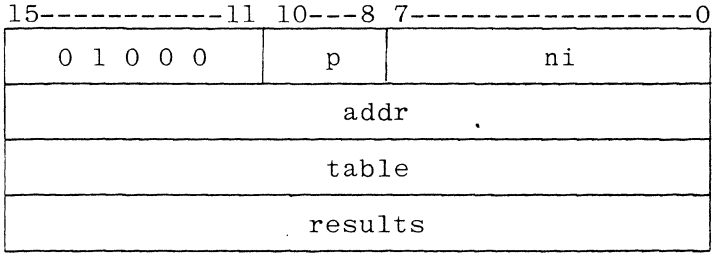


Where the s field contains:

000 when FLAG,G is coded.
001 when FLAG,R is coded.

GRADE

GRADE,1 or 2 or 4 ni,addr,table,results

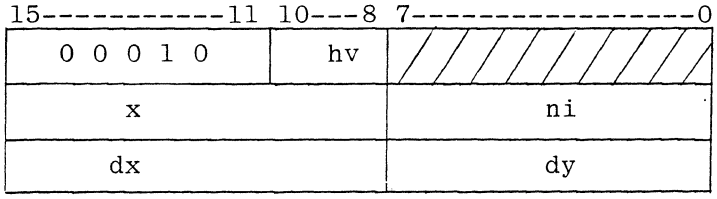


The p field contains:

100 when GRADE,1 is coded.
101 when GRADE,2 is coded.
110 when GRADE,4 is coded.

HDRSN

HDRSN,V or H x,y,dx,dy

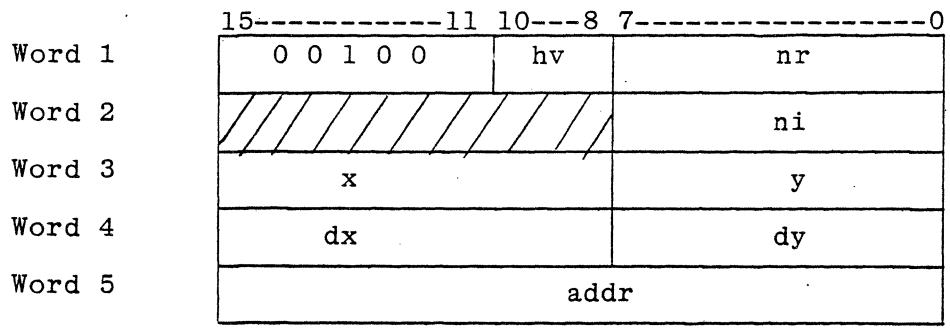


The hv field contains:

010 when HDRSN,V is coded.
011 when HDRSN,H is coded.

ITEM

ITEM,V or H nr,ni,x,y,dx,dy,addr

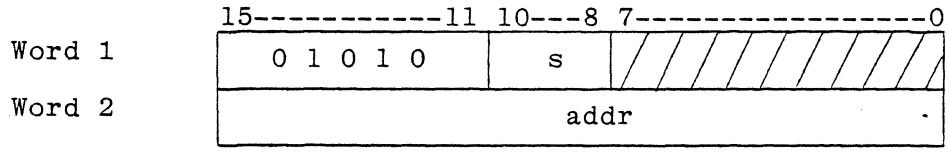


The hv field contains:

000 when ITEM,V is coded.
001 when ITEM,H is coded.

JUMP

JUMP,U or R addr

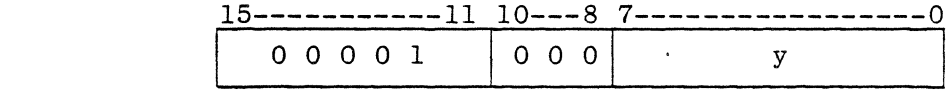


The s field contains:

000 when JUMP,U is coded.
001 when JUMP,R is coded.

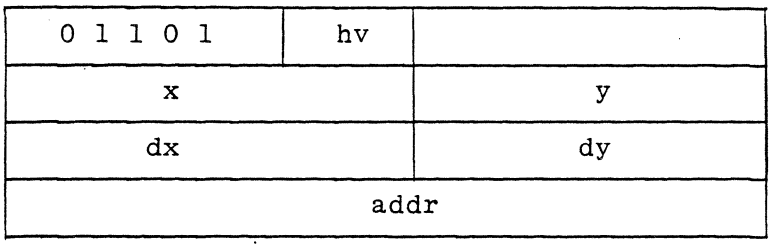
LIMIT

LIMIT x,y



LITHOID

LITHOID,V or H x,y,dx,dy,addr

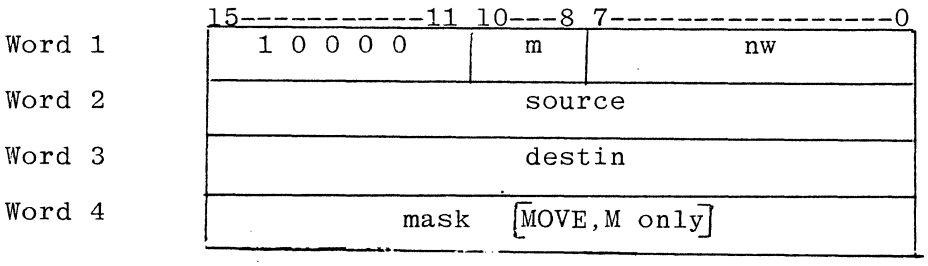


The hv field contains:

010 when LITHOID,V is coded
011 when LITHOID,H is coded

MOVE

MOVE [,M] [nw,] source,destin,mask

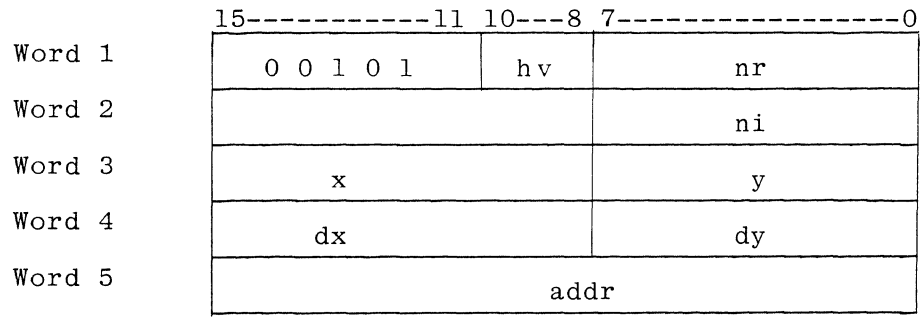


The m field contains:

010 when MOVE is coded.
011 when MOVE,M is coded.

NUMERIC

NUMERIC,H or V nr,ni,x,y,dx,dy,addr

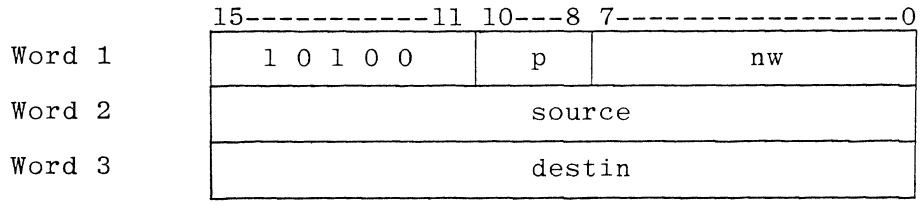


The hv field contains:

010 when NUMERIC,V is coded.
011 when NUMERIC,H is coded.

PACK

PACK,2 or 4 nw,source,destin

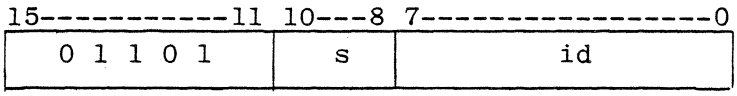


The p field contains:

001 when PACK,2 is coded.
010 when PACK,4 is coded.

PAGEID

PAGEID,S id

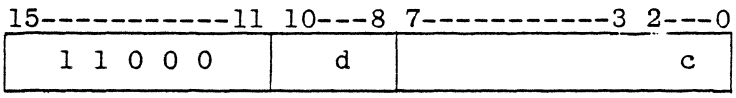


The s field contains:

000 when PAGID is coded.
001 when PAGID,S is coded.

POCKET

POCKET,S or C R or A or F or U



The d field contains:

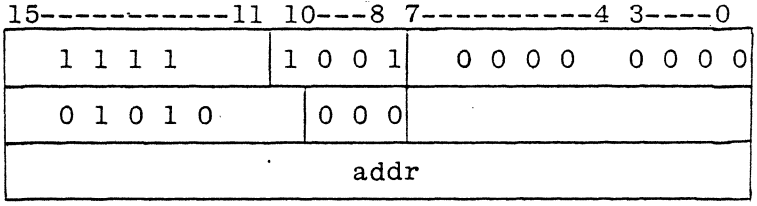
000 when POCKET,S is coded.
001 when POCKET,C is coded.

The c field contains:

001 when A parameter is coded.
010 when R parameter is coded.
011 when F parameter is coded.
100 when U parameter is coded.

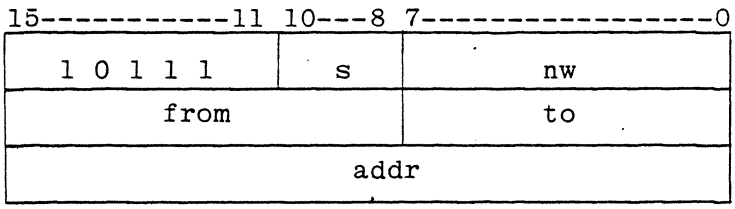
REENTER

REENTER addr



REPLACE

REPLACE,L or T or A nw,from,to,addr

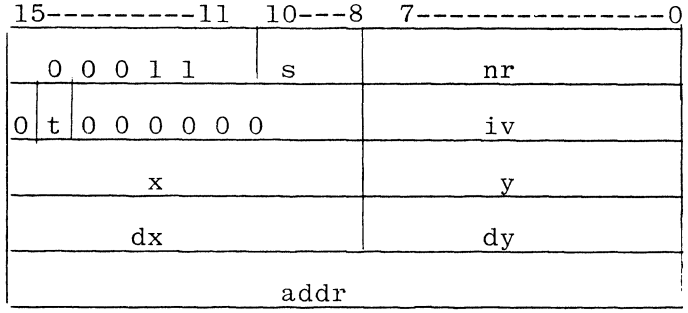


The s field contains:

001 when REPLACE,L is coded.
010 when REPLACE,T is coded.
011 when REPLACE,A is coded.

RESPONSE

RESPONSE,[S] nr,iv,x,y,dx,dy,[addr]



The s field contains:

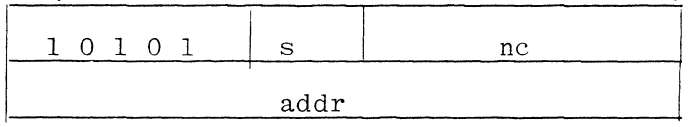
000 when RESPONSE is coded.
001 when RESPONSE,S is coded.
Word 5 is only present when RESPONSE,S is coded.

The t field contains:

0 when the iv parameter is an integer or hexadecimal value.
1 when the iv parameter is =c; where c equals any valid EBCDIC character appearing in Appendix A.

SCRIBE

SCRIBE,1or2or3orR nc,addr

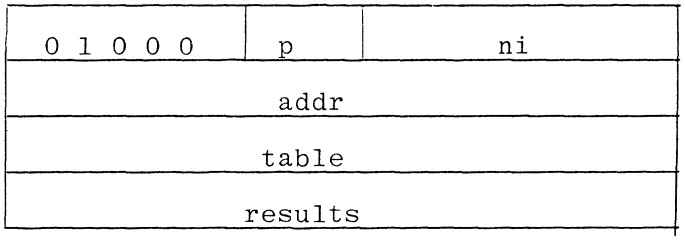


The s field contains:

101 when SCRIBE,1 is coded.
110 when SCRIBE,2 is coded.
111 when SCRIBE,3 is coded.
100 when SCRIBE,R is coded.
nc means number of characters.

SCORE

SCORE,1 or 2 or 4 ni,addr,table,results

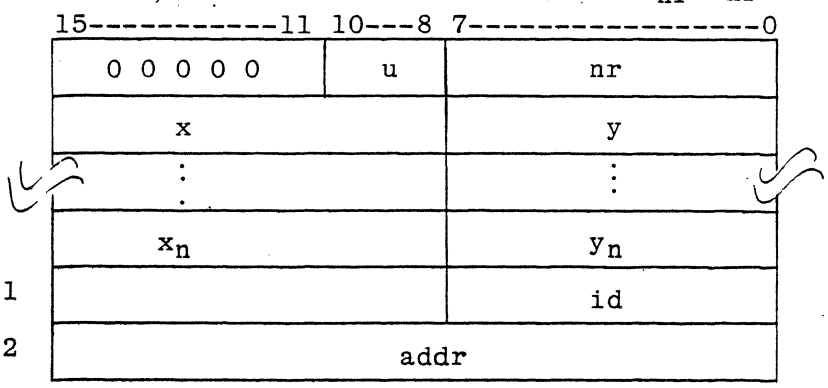


The p field contains:

000 when SCORE,1 is coded.
001 when SCORE,2 is coded.
010 when SCORE,4 is coded.

SKUNK

SKUNK,U nr,x_1,y_1,...,x_{nr},y_{nr},id,addr

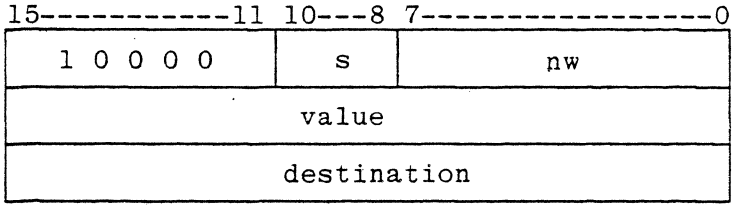


The u field contains:

000 when SKUNK is coded.
001 when SKUNK,U is coded.

STORE

STORE,[B] (nw),value,destin

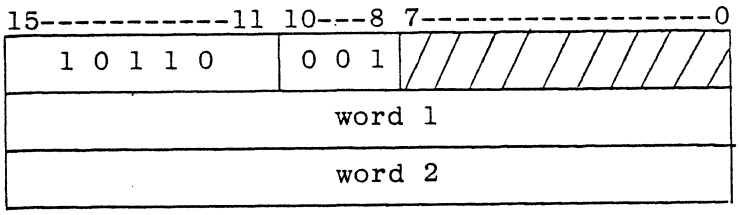


The s field contains:

0 when STORE is coded.
1 when STORE,B is coded

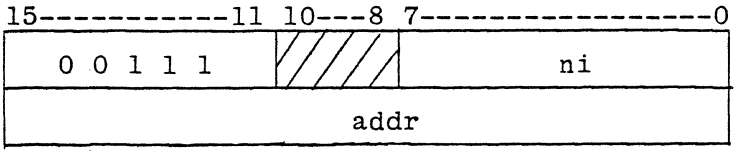
SUBMEMRY

SUBMEMRY word1,word2



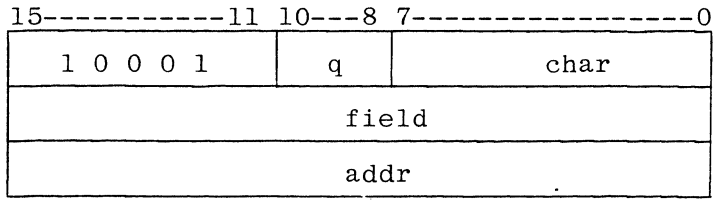
TALLY

TALLY ni,addr



TEST

TEST,EorNorGorL char,field,addr

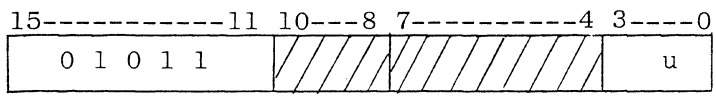


The q field contains:

000 when TEST,E is coded.
001 when TEST,N is coded.
010 when TEST,G is coded.
011 when TEST,L is coded.

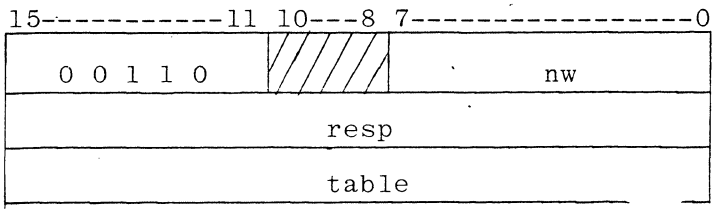
WRITE

WRITE u



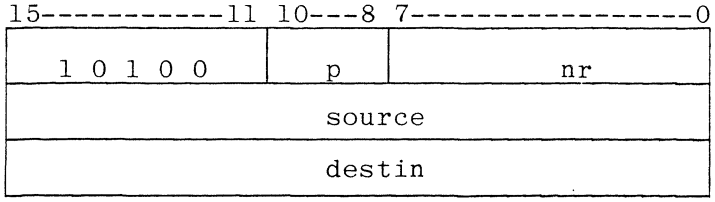
XLATE

XLATE nw,rep,table



XPAND

XPAND,2 or 4 nr,source,destin

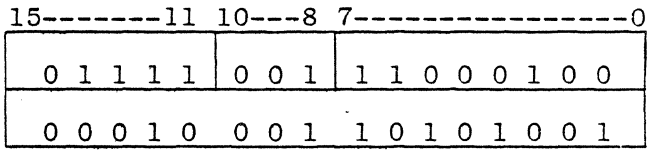


The p field contains:

101 when XPAND,2 is coded.
110 when XPAND,4 is coded.

Application: FLAG,R

LOCN	INST	LABEL	COMMAND	PARAMETERS
1134	79C4		FLAG,R	D,DC
1135	11A9			



APPENDIX C

DOCUMENT QUALITY INDICATORS

ERROR STOP AND NON-STOP MODE

Normal DOSSIER scanning is done in error stop mode. Each time a document causes a red document quality condition the system halts and prints:

```
FRE-c
..32 characters  ..16 characters
CONTINUE?
```

where c is the character identifying the error. The 32 characters, the first 32 of the record, identify the document causing the error. The 16 characters, from the field specified in the DOSSIER command, identify the last document processed.

If data switch 1 is set prior to running, the system will be in non-stop mode and it will not halt on red error conditions. All records will be transferred to tape with any errors marked in the record as shown in figure C-1, unless a DELETE command is executed.

DOCUMENT QUALITY INDICATORS

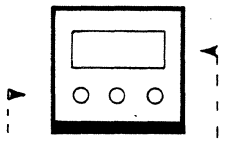
DOSSIER includes a visual and tape record flag system for indicating an error on a document being scanned. Two external indicators, (the digital display and the indicator lamps) and one or more characters placed in the record flag the error. The three indicator lamps signal the document quality as:

red - an error sufficiently serious to stop scanning (e.g. unidentified skunk mark, wrong number of timing marks, FLAG,R executed).

amber - a field failed an edit test but scanning continues.

green - good document.

A four digit code appears in the digital display to signal which error occurred. A good document displays the serial number. The indicators and flags are specified in the chart on the next page.



INDICATOR LAMPS	DIGITAL DISPLAY	CHARACTER	RECORD POSITION	DESCRIPTION
red	9999	U	16	The document just scanned failed all skunk tests.
red	9090	T	16	The document just scanned encountered a timing error.
red	8888	C > f	16 24 25+n	A grid just failed an edit test and the output record for the document was flagged. n = number of previous flags set for this document. f = flag character.
red	⌐XX	L	16	The document scanned was too long. The display indicates the number of extra timing marks that were read.
red	⌐XX	S	16	The document scanned was too short. The display indicates the number of missing timing marks.
amber	2222	> f	24 25+n	A grid just failed an edit test and the grid was flagged.
red	1111	N	16	The resolution of a document or book serial number grid produced invalid data.
red	0000	B	16	The resolution of a batch number grid produced invalid data.
green	XXXX			The document scanned contains none of the above indications. The digital display identifies the serial number of the document just scanned.

Figure C-1. Document Quality Indicators

APPENDIX D
AVAILABLE STORAGE

DOSSIER, SYMON and the document image buffer occupy a specific number of words of storage. The number of words is determined by the DOSSIER variant and the size of the largest document to be resolved by the program. The table gives the number of words of storage used by each DOSSIER variant.

To calculate the number of words remaining for storing the application program, add the number of words required for the document image to the number used by DOSSIER & SYMON. Substract that sum from the available memory.

EXAMPLE:

Available Memory		Variant		Document Image Size		Words Remaining
8K System		10SDT0		Document with 60 timing marks		
8192 Words	-	(4192 Words	+	720 Words	) =	3280 Words

DOSSIER VARIANT	WORDS USED BY DOSSIER & SYMON	DOCUMENT IMAGE BUFFER SIZE
10SDTO 10SDPTO 10SDT1 10SDT2 10SDT3 10SDT4 10SDT9 10SDTA 10SDTB	4192 4256 4320 4320 4320 4320 4320 4384 4384	12 X # of timing marks
15TO 15PTO 15T1 15T2 15PT3 15T4 15T9 15TB	4096 4160 4224 4224 4288 4224 4224 4288	15 X # of timing marks
20STO	4128	12 X # of timing marks

Table of Storage Requirements

APPENDIX E
DIRECT STORAGE AVAILABLE TO THE USER

The following listing identifies storage areas that are available to the user. The quantities in these areas may be changed or used by the programmer to gather additional data or change the mark discrimination techniques. In addition, the program has 16 direct storage locations which can be used when writing assembly language subroutines.

DATE

The date, as typed in at program load time, is stored at location 0100_{16}.

EXAMPLE: Date 02/18/76

0100 00F0	0
0101 00F2	2
0102 0061	slash
0103 00F1	1
0104 00F8	8
0105 0061	slash
0106 00F7	7
0107 00F6	6
0108 000D	carriage return

```
ERRORS LOCN  INST      LABEL     COMMAND      PARAMETERS          COMMENTS                                  SEQ NUMB
                                               ********************************************              0590
                                               DOSSIER DIRECT STORAGE AVAILABLE TO THE USER.             0591
                                                THE FOLLOWING DOSSIER PARAMETERS MAY BE                  0592
                                                ACCESSED BY A DOSSIER COMMAND PROGRAM. THE               0593
                                                PARAMETER LOCATION ASSIGNMENTS MAY NOT BE                0594
                                                CHANGED UNLESS THE DOSSIER INTERNAL REFERENCE            0595
                                                SPEC IS ALSO UPDATED TO REFLECT THE CHANGE.              0596
                                               ********************************************              0597
                                                                                                        0598
             00A0                 CON          0A0                 ASSIGN DOSSIER DIRECT STORE          0599
                                                                                                        0600
                                               ********************************************              0601
                                               CENTRAL HUB STRUCTURE PARAMETERS                         0602
                                               ********************************************              0603
                                                                                                        0604
   00A0      0000      POINTER                                      POINTER TO CURRENT COMMAND           0605
   00A1      0000      OPRNCODE                                     COMMAND OP-CODE   (BITS 11-15)     0606
   00A2      0000      MODIFIER                                     COMMAND MODIFIER  (BITS 08-10)     0607
   00A3      0000      PRIMARY                                      1ST PARAMETER     (BITS 00-07)     0608
                                                                                                        0609
                                               ********************************************              0610
                                               BASIC MARK RESOLUTION PARAMETERS                         0611
                                                THESE VALUES HAVE BEEN PRE-SET TO                       0612
                                                THEIR CURRENT VALUES TO READ CLOSELY                    0613
                                                CONTROLLED FORMS. OLD FORMS WITH NON-                   0614
                                                UNIFORM INK MAY REQUIRE ADJUSTMENTS TO                  0615
                                                THESE PARAMETER VALUES.                                 0616
                                               ********************************************              0617
                                                                                                        0618
   00A4      0006      LOWLIMIT     LOW.MRK                         LIGHT MARK THRESHOLD                 0619
   00A5      0002      LOWDELTA     LOW.DLT                         LIGHT MARK DELTA                     0620
   00A6      000C      TOPLIMIT     TOP.MRK                         DARK MARK THRESHOLD                  0621
   00A7      0003      TOPDELTA     TOP.DLT                         DARK MARK DELTA                      0622
                                                                                                        0623
                                               ********************************************              0624
                                               BASIC PHOTOCELL RELIABILTIY PARAMETERS                   0625
                                                THESE CONTROLS ARE USED WITH THE 'BIAS'                 0626
                                                COMMAND TO CHECK THE READING QUALITY                    0627
                                                OF EACH PHOTOCELL ON THE INK BAR.                       0628
                                               ********************************************              0629
                                                                                                        0630
   00A8      FFFC      BIASLIM      -BIAS.CNT+1-NCS.SPCL            CONSECUTIVE SHEET COUNT              0631
   00A9      0000      BIASERR                                      BAD PHOTOCELL NUMBER                 0632
   00AA      0009      BIASDELT     BIAS.1ST                        BIAS DELTA * 2 (ROUND/TRUNCATE)      0633
   00AB      0000      BIASAVGR                                     SHEET BIAS BAR AVERAGE               0634
```

```
ERRORS LOCN  INST    LABEL     COMMAND    PARAMETERS          COMMENTS                                SEQ NUMB
                                          ********************************************            0636
                                          DOCUMENT QUALITY COUNTERS & CONTROLS                      0637
                                          ********************************************            0638
                                                                                                    0639
       00AC  0000    SECTFACT                               SECTIONAL LIGHTNESS FACTOR              0640
       00AD  0000    DOCUFACT                               DOCUMENT LIGHTNESS FACTOR               0641
       00AE  0000    RIGHTCNT                               CORRECT ANSWER COUNT                    0642
       00AF  0000    WRONGCNT                               INCORRECT ANSWER COUNT                  0643
       00B0  0000    OMITCNT                                SECTION IMBEDDED OMITS                  0644
       00B1  0000    TOTLCNT                                SECTION TOTAL OMITS                     0645
       00B2  0000    MULTCNT                                SECTION MULTIPLES                       0646
       00B3  0000    INVLCNT                                TOTAL INVALIDS (OMITS & MULTS)          0647
       00B4  0000    LITECNT                                SECTION LIGHT COUNT                     0648
       00B5  0000    DARKCNT                                SECTION DARK COUNT                      0649
                                                                                                    0650
                                          ********************************************            0651
                                          BATCH/SERIAL NUMBER HOLDING STORAGE.                      0652
                                           BATCH/SERIAL NUMBER MOVED INTO OUTPUT                    0653
                                           RECORD WHEN 'WRITE' COMMAND PROCESSED.                   0654
                                          ********************************************            0655
                                                                                                    0656
            0003    BSN.SIZ   EQU        3                   3 DIGIT BATCH #                     0657
            0004    DSN.SIZ   EQU        4                   4 DIGIT SERIAL #                    0658
                                                                                                    0659
       00B6  00F0    BSN.BUF   TEXT       3,001               3 DIGIT BATCH #                     0660
       00B7  00F0
       00B8  00F1
       00B9  00F0    DSN.BUF   TEXT       4,0001              4 DIGIT SERIAL #                    0661
       00BA  00F0
       00BB  00F0
       00BC  00F1
       00BD  00F0    RSN.BUF   TEXT       1,0                 1 DIGIT RUN #                       0662
```

```
ERRORS LOCN  INST      LABEL     COMMAND      PARAMETERS          COMMENTS                              SEQ NUMB

                                            ********************************************               0664
                                            SYSTEM COUNTERS                                            0665
                                            ********************************************               0666
                                                                                                       0667
             00BE      CNTRORG   EQU        *                                                          0668
       00BE  0000      READTOTL             0                   SCANNED SHEETS COUNTER                 0669
       00BF  0000      SHEETS               0                   PROCESSED SHEETS COUNTER               0670
       00C0  0000      FORCED               0                   FORCED RUN STOPS COUNTER               0671
       00C1  0000      RECORDS              0                   WRITTEN RECORDS COUNTER                0672
       00C2  0000      POCKETS              0                   POCKETED SHEETS COUNTER                0673
       00C3  0000      DELETES              0                   DELETED RECORDS COUNTER                0674
       00C4  0000      PARITY               0                   PARITY ERROR COUNTER                   0675
       00C5  0000      BIASFAIL             0                   BIAS CHECK FAILURES                    0676
       00C6  0000      BIASTOT              0                   BIAS BAR AVERAGE TOTAL                 0677
       00C7  0000      ERRORS               0                   FLAGGED SHEETS COUNTER                 0678
       00C8  0000                           0                   DOCUMENT CONTENT ERRORS                0679
       00C9  0000                           0                   DOCUMENT SEQUENCE ERRORS               0680
       00CA  0000                           0                   SERIAL # ERRORS                        0681
       00CB  0000                           0                   BATCH # ERRORS                         0682
       00CC  0000                           0                   SHORT DOCUMENTS                        0683
       00CD  0000                           0                   LONG DOCUMENTS                         0684
       00CE  0000                           0                   SKUNK ERRORS                           0685
       00CF  0000                           0                   TIMING ERRORS                          0686
             00D0      CNTREND   EQU        *                                                          0687
```

```
ASSEMBLER: CLASS-II/15N R-3.2  PROGRAM: DOSSIER                          DATE: 03/19/76                   PAGE: 0021

ERRORS LOCN  INST      LABEL    COMMAND      PARAMETERS           COMMENTS                                SEQ NUMB

                                          *********************************************                   0689
                                          USER DIRECT STORAGE MEMORY ASSIGNMENTS.                         0690
                                           THIS DIRECT STORAGE MEMORY BLOCK IS RESERVED                   0691
                                           FOR USE BY A DOSSIER COMMAND PROGRAM. THIS                     0692
                                           AREA ASSIGNMENT MAY BOT BE CHANGED UNLESS THE                  0693
                                           DOSSIER INTERNAL REFERENCE SPEC IS ALSO                        0694
                                           UPDATED TO REFLECT THE CHANGE.                                 0695
                                          *********************************************                   0696
                                                                                                          0697
             00D0               CON       0D0                  SET ASIDE USER DIRECT STORE                0698
      00D0             USERFWA  BSS       16                                                              0699
```

APPENDIX F
TELEPRINTER MESSAGES

The list below describes messages that can appear on the teleprinter during CLASS or DOSSIER execution. These are explained in the operations manual for the system under ASSEMBLY or DOSSIER.

MESSAGES	DEFINITION	ERROR CONDITION	ASSEMBLY = A DOSSIER = D
CELL ERRn	Cell Error - Cell n	Yes	D
DOC OVRUN	Document Over Run	Yes	D
END INP	End Input	Yes	A
FRE C	Forced Run Error - Content	Yes	D
FRE L	Forced Run Error - Long	Yes	D
FRE S	Forced Run Error - Short	Yes	D
FRE T	Forced Run Error - Timing	Yes	D
FRE U	Forced Run Error - Undefined	Yes	D
LNG ERR n	Length Error	Yes	A
OBJ ERR	Object Error	Yes	A
OPR EOF	Operator End of File	No	D
OPR EOJ	Operator End of Job	No	D
OPR EOT	Operator End of Tape	No	D
OPR ERR	Operator Error	Yes	D
OPR SRCH	Operator Search	No	D
PAR ERR	Parity Error	Yes	A
PARAM ERR	Parameter Error	Yes	A
PROG WAIT	DISPLAY,W Command Execution	No/Yes	D
RDY INP	Ready Input	Yes	A
RDY L-PRT	Ready Line Printer	Yes	D
RDY MTU	Ready Magnetic Tape Unit	Yes	D
RDY OBJ	Ready Oject File	Yes	A
RDY T-PRT	Ready Transport Printer	Yes	D
REOT	Read Double File Marker	No	D
RLP	Read Load Point	No	D
RPE	Read Parity Error	Yes	D
RTM	Read Tape Mark	No	D
URWPE	Irrecoverable Parity Error	Yes	D
WEOT	End of Tape	Yes	D
WLD n	Write Timing Error	Yes	D
WPE n	Write Parity Error	Yes	D
*	Program Initialized	No	D

APPENDIX G
DEBUGGING COMMAND PROGRAMS

SYMON II CONSOLE DEBUG

DEBUG is an on-line conversational utility program which aids the user in debugging his program with the aid of the teleprinter. DEBUG is present in SYMON II only (8k or greater system).

The user can enter DEBUG as follows:

TO SET (P) REGISTER

ALPHA LSI

PROCEDURE	EXTERNAL EFFECTS
a) Set STOP	STOP lamp on
b) Press RESET, Clear SREG	
c) Enter 011E	(On Hexadecimal keyboard)
d) Press WRITE	WRITE lamp on
e) Press P	011E displayed on data lamps
f) Clear STOP	STOP lamp off
g) Press RUN	

ALPHA 16

PROCEDURE	EXTERNAL EFFECTS
a) Press RESET	
b) Enter 011E	(with data switches)
c) Press P	
d) Press ENTRY	011E displayed on data lamps
e) Press Run	

On entry DEBUG will type a question mark (?).

At this point the operator has two options: Type I or J.

1) I for Inspect/Change Memory

   The I function allows the user to display individual memory locations and change their values if desired. The user types the address of the first word to be inspected followed by a period.

DEBUG will then respond with the address and the contents of that address. If the value is to be changed, a new value is entered. If the next consecutive location is be inspected, a space is typed. If the previous consecutive location is to be inspected, a comma is typed. When all corrections have been made, a period is typed to terminate inspection. The absence of a period, a space, or a comma will cause the correction not to be entered.

2) J for Jump

The J function allows the user to unconditionally transfer control. The user types the address of where he wants control to transfer followed by a space, comma, or period delimiter. When DEBUG is complete, return to control can occur only at the beginning of the program (at address 550 for DOSSIER).

When an illegal command is entered, DEBUG will ignore the command, type a left arrow (←), new line and a question mark (?). It will then wait for an I or J.

All addresses and data parameters are entered as hexadecimal values. Leading zeros need not be entered. If an error has been made in entering a parameter, it may be reentered by typing a RUBOUT (DELETE) followed by the correct value or by continuing to type in hexadecimal values with the leading zeros included. Only the last four characters will be accepted. (See EXAMPLE. Sixth line).

EXAMPLE:

In the following example all DEBUG typeouts are underlined. Operator entered characters are not underlined.

?I13.	Inspects Location 13.
0013 0800Δ	Space typed to examine next location.
0014 9214,	Comma typed to examine previous location.
0013 0800 F612.	New data typed to change address 0013 from 0800 to F612. Period returns to Command mode.
?I2→3.	Inspects Location 3.
0003 0000 F6 F714.	Address 0003 changed to 0000 to F714. (Error F6 is ignored, only F714 is used). Period returns to Command mode.
?P←	Illegal command character (not I or J).
?J550	Return to start of program.

DOSSIER "STEP" MODE DEBUG

A programmer may use an option in DOSSIER to perform a one command step through his program. The programmer can only enter the option when the teleprinter responds with an asterisk. The option is entered by clearing all Data Switches, and entering zero in the A register as follows:

ALPHA LSI

PROCEDURE	EXTERNAL EFFECTS
a) Set STOP	STOP lamp on
b) Press RESET, Clear SREG	
c) Enter 0000	(On Hexadecimal keyboard)
d) Press WRITE	WRITE lamp on
e) Press A	0000 displayed on data lamps
f) Clear STOP	STOP lamp off
g) Press RUN	

ALPHA 16

PROCEDURE	EXTERNAL EFFECTS
a) Press RESET	
b) Enter 0000	(with data switches)
c) Press A	
d) Press ENTRY	0000 displayed on data lamps
e) Press RUN	

The command program is then initialized by pressing RUN, but only one document must be passed under the read head. The system will halt prior to executing each DOSSIER command. When the system halts, the

- "X" register will contain the first word address of the next command to be executed.
- "A" register will contain the first word of the next command to be executed.

By pressing RUN, the system will execute the next command and halt.

If the stepping operation encounters an assembly language routine, the system will execute all machine instructions and then continue stepping DOSSIER commands when the REENTER command executes. When the system halts, the programmer may use the SYMON debug routine located at address $011E_{(16)}$ to inspect changes in memory caused by the previously executed command.

This procedure requires that a LIMIT command precede all resolution commands. Otherwise the operation will result in a length error. Another document must not pass under the read head until all DOSSIER commands for that document, including the WRITE command, are executed.

APPENDIX H
SETTING DATA SWITCHES ON ALPHA CONSOLES

CONSOLE DATA LAMPS

Both CAI ALPHA consoles are designed with a series of 16 data indicator lamps across the top. These data lamps are sequentially marked with numerals 0-15 (see Figure H-1). The 16 lamps are 4 sets of 4 digit binary numerals. For general operation, only lamps 0-3 are used. Lamps 0-3 indicate the data switches with the lamp numeral corresponding directly to the data switch number.

ALPHA 16

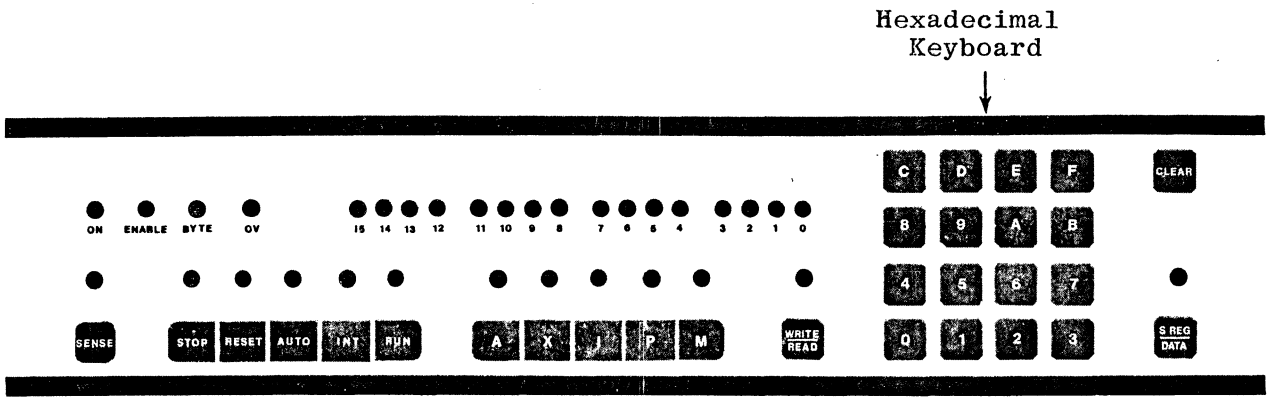
On the ALPHA 16, a data switch is set by depressing the appropriately numbered switch. Multiple switches may be set.

ALPHA LSI

To set an ALPHA LSI data switch requires setting the S REG/DATA switch and entering the appropriate number from the hexadecimal entry keyboard. The data lamps respond with the binary representation of the hexadecimal number entered. For quick reference, data switches are set from the hexadecimal keyboard as displayed by table H-1. Multiple switches may be entered.

TO SET DATA SWITCH	PRESS HEXADECIMAL KEYBOARD DIGIT
0	1
1	2
2	4
3	8

TABLE H-1.



ALPHA LSI Console

APPENDIX I

SPECIAL MAINFRAME COMPATIBILITY REQUIREMENTS

DIGITAL EQUIPMENT CORPORATION DEC-10 COMPATIBLE TAPES

OUTPUT RECORDS

A special variant of DOSSIER reads and writes DEC 10 compatible 9 track tapes. Output records for this variant must be multiples of 5 characters. In addition, the last five characters must be unused by the command program. DOSSIER (the DEC 10 Variant) fills the last five characters of each output record with a carriage return, a line feed, and 3 null characters.

Example:

If your output record requires 87 characters, you must add 8 additional characters using

```
FILLER          BSS          8
```

to make it a multiple of five and leave room for the CR, LF and 3 nulls.

INPUT RECORDS

Input source records to the assembler for this variant must be 85 characters long ending with CR, LF and 3 null characters.

HONEYWELL 6000 COMPATIBLE TAPES

OUTPUT RECORDS

A special variant of DOSSIER reads and writes HONEYWELL 6000 compatible 9 track tapes. Output records for this variant must be multiples of 12 characters.

Example:

If your output record requires 62 characters, you must add

```
FILLER          BSS          10
```

to make it a multiple of 12.

INPUT RECORDS

Input source records to the assembler for this variant must be 84 characters long.

APPENDIX J

THE USE OF PSEUDO-OPS FOR PROGRAMS WITH MULTIPLE SHEETS

PSEUDO-OP DESCRIPTION

The CON and ORG pseudo-ops both perform identical functions and may be used interchangeably.

```
CON          expression
ORG          expression
```

These pseudo-ops set the assembler's Location Counter to a specified value (address).

The Location Counter within CLASS is preset to a value beyond the end of the System Monitor (SYMON) at the beginning of both the first and second pass of the assembly. It is then incremented by '1' for each word of binary object code generated by the assembler. It may be reset to any desired value during program assembly by using a CON or ORG pseudo-op.

USAGE

A practical application for these pseudo-ops is a DOSSIER program for multiple sheets with each sheet having a unique output record. CON or ORG will allow reuse of the addresses in your output record description by assigning new labels and new sizes.

This example shows how coding was done to define the longest output record, which determines the length of all output records.

SENTRY 70 CODING FORM

National Computer Systems, Inc

PROGRAM SAMPLE USE OF CON PSEUDO-OP

ROUTINE DEFINING OUTPUT RECORDS FOR MULTIPLE FORMS

NAME

PAGE 1 of 1

DATE

```
LOCATION  OPERATION  ADDRESS FIELD      COMMENTS                          IDENT
 RECB     BSS        24
NAME      BSS        20
ID        BSS        10                OUTPUT RECORD FOR
ID        BSS        10                  LARGEST SHEET
ID        BSS        10                IN THE PROGRAM
ID        BSS        10
RECE      EQU        *

          CON        RECB+24
ZIP       BSS        7                 OUTPUT DESCRIPTION
HOUSENUM  BSS        10                  FOR ANOTHER SHEET
HOUSENUM  BSS        10
HOUSENUM  BSS        10

          CON        RECB+24
GRADE     BSS        2
SCHOOL    BSS        24                THIRD SHEET

          CON        RECE              WHEN ALL RECORDS ARE DEFINED

     WORKING STORAGE
          END
```

Ø = ALPHA O O = NUMERIC O I = ALPHA I 1 = NUMERIC 1 Z = ALPHA Z 2 = NUMERIC 2 202 145 413

Sample Use of the Con Pseudo-Op to
define Outpur Records for Multiple Forms

In the example above, the CON pseudo-op was used to enable redefinition of the existing output record for a second sheet. ONLY positions to be used need to be specified. The unused portion will be space filled for this document. CON is used again to specify a third sheet's output record. This may be done as many times as necessary.

When all the records are described, it is important that CON be used to reset the assembly address pointer to the end of the longest output record area, to allow any working storage locations to assemble in their proper location in the program.

COMMENT SHEET
DOSSIER SOFTWARE REFERENCE MANUAL
Part No. 202-145-355-7

FROM NAME:

BUSINESS
ADDRESS:

COMMENTS: (DESCRIBE ERRORS, SUGGESTED ADDITION OR DELETION AND INCLUDE PAGE NUMBER, ETC.)

CUT ALONG LINE

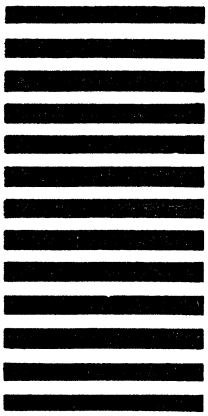
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

FOLD

FIRST CLASS
PERMIT NO. 1840
MINNEAPOLIS, MINN.

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



National Computer Systems, Inc.

c/o Publications

4401 West 76th Street
Minneapolis, Minnesota 55435

FOLD

CUT ALONG LINE

SENTRY 70 | DOSSIER CODING FORM

National Computer Systems, Inc.

NAME

PROGRAM

PAGE

ROUTINE

DATE

LOCATION		COMMAND	PARAMETERS	COMMENTS
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80	

A = ALPHA 0 | 0 = NUMERIC 0 | T = ALPHA T | 1 = NUMERIC 1 | 2 = ALPHA 2 | 2 = NUMERIC 2