File Number 1401/1460-30
Form C24-1462-3

IBM Systems Reference Library

Input/Output Control System (on Tape) Specifications and Operating Procedures IBM 1401 and 1460

Program Number *1401 - IO - 065*

This publication describes the programming required to use IOCS to control the input/output of data from card reader, card punch, printer, and tape files. The IOCS descriptive entries (DIOCS and DTF) and macro instructions are explained in detail. The types of data records and tape labels handled by IOCS are defined. Also, two sections are included that should be especially useful to experienced programmers: (1) Summaries — briefly lists storage-area considerations, macro instructions, and processing-overlap considerations; (2) Program Operation — describes IOCS library routines, labels, halts, and error indications.

IOCS is a supplement to the 1401 *Autocoder* program, and the reader should be familiar with the program described in *Autocoder (on Tape) Specifications for* IBM *1401 and 1460,* Form C24-1434.

Fourth Edition

This is a reprint of C24-1462-2 incorporating changes released in the following Technical Newsletter:

Form Number	Pages Affected	Date
N24-0286	7, 8, 9, 10, 11, 12, 12.1, 25, 26, 28, 29, 33, and 34	December 23, 1964

Significant changes or additions to the specifications contained in this publication will be reported in subsequent revisions or Technical Newsletters.

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

Address comments concerning this manual to: IBM Corporation, Programming Publications, Dept. 425, Rochester, Minn. 55901.

Contents

The IBM 1401 and 1460 Input/Output Control System eliminates the need for detailed programming of standardized input and output routines. IOCS requires descriptive entries and macro instructions in addition to those used by the *Autocoder* program. With these, the user has access to routines for reading and writing, blocking and deblocking, file labeling, and error checking.

IOCS library routines are selected and tailored automatically by the *Autocoder* processor to satisfy the particular requirements of each job. *Autocoder* generates the minimum number of instructions needed according to the detailed information the user supplies in the descriptive entries.

Although primarily concerned with magnetic-tape files, IOCS also applies to unit-record files in the IBM 1402 Card Read-Punch and to continuous forms prepared by the IBM 1403 and 1404 Printers.

The IBM 1401 system used to assemble programs with IOCS must have at least:

- 4,000 positions of core storage
- Four IBM 729 II, 729 IV, 729 V, or 7330 Magnetic Tape Units
- IBM 1403, Model 2, or 1404 Printer
- IBM 1402 Card Read-Punch
- Advanced-Programming feature
- High-Low-Equal Compare feature
- Sense switches (Not required for assembly from a source program card deck, but necessary for all other *Autocoder* operations.)

The IBM 1460 system used to assemble programs with IOCS must have at least:

- 8,000 positions of core storage
- Four IBM 729 II, 729 IV, 729 V, 729 VI, or 7330 Magnetic Tape Units
- IBM 1403 Printer, Model 2
- IBM 1402 Card Read-Punch
- Indexing-and-Store-Address-Register feature
- Sense switches (Not required for assembly from a source-program card deck, but necessary for all other *Autocoder* operations.)

The resultant object program can be used in any IBM 1401 system equipped with the advanced programming and high-low-equal compare features or in any IBM 1460 system equipped with the indexing-and-store-address-register feature. Sense switches are also required if tape scanning will be performed (see *DIOCS Readerror).*

General Description

The IOCS descriptive entries and macro instructions are punched in *Autocoder* cards and assembled with the source program.

The descriptive entry cards are inserted immediately behind the *Autocoder* JOB and CTL cards, and ahead of the user's source program. They describe the input/output files used in the program and consist of two types of entries:

DIOCS —"Descriptive IOCS" that describes generally the machine configuration and the files used in the program (Figure 1).

DTF —"Define The File" that describes, in detail, an individual file (Figure 2). A DTF entry must be included for each file processed by IOCS — unit-record files, tape files, and printer.

Label	Operation	OPERAND
	DIOCS	
DIOCSORG		700
IODEVICES		READER,,PUNCH,,TAPE
TAPEUSE		INPUT
FEATURES		RELEASE
LABELDEF		STANDARD,,TM,,CHECK
COUNTS		RECORD,,HASH
ALTDRIVE		YES
EXITS		6,,7
RWDOPTION		UNLOAD
READERROR		CLEAN,,TAPE,,4

Figure 1. Sample DIOCS Entry

Label	Operation	OPERAND
	DTF	FILEA
FILETYPE		TAPE,,INPUT
MODEPAR		LOAD
CHANDRIVE		2
ALTTAPE		5
RECFORM		BLOCKED
SIZEREC		120
BLOCKSIZE		600
IOAREAS		INPUTA
WORKAREA		WORKA
EOFADDR		ENDJOB
EXTADDR		HEAD2
TYPELABEL		STANDARD
CHECKLABEL		ALL
HEADER		MO PAYROLL,,62074,,030
SERIALNUM		54003
WLRADDR		RECERR

Figure 2. Sample DTF Entry

Each DIOCS and DTF entry consists of a set of cards: a header card followed by several detail entry cards. The number of detail cards is governed by the factors that must be specified for a particular job. These entries are explained in detail under *Descriptive Entries.*

The macro instructions are entered in the source program whenever an input or output record is to be read or written. They provide linkage to the IOCS library routines that read, write, block, deblock, and check records without further programming on the part of the user. Four basic operations are:

GET —moves a record from a file (tape or unit record) to an input or work area where it can be processed.

PUT —moves a record from an output or work area to a file (tape, unit-record, or printer).

OPEN —activates any file for processing, and checks or writes IBM standard header labels in tape files.

CLOSE —deactivates any file after processing is completed, and writes IBM standard trailer labels in tape files. Standard input trailers are automatically checked immediately before the CLOSE.

Because IBM standard header and trailer labels are processed automatically by IOCS, it is to the user's advantage to include them whenever possible. However, tape records with no header or trailer labels or with nonstandard labels can be processed. In this case, the user must write the program instructions to process his labels.

Other macro instructions used by IOCS are: RELSE (release), FEORL (Forced End-of-Reel), DCLOS (Dump-Close), and RDLIN (Read Label Information).

Processing Overlap Special Feature

When this special feature is installed in an IBM 1401 or 1460 Data Processing System, records can be read, written, and punched in the *overlap mode* by IOCS. This feature provides a reduction in over-all job time by efficient use of the high-speed processing and input/output capabilities of the 1401 and 1460. It is described in detail in *Special Features for IBM 1401 and 1460,* Form A24-3071.

Whenever this feature is used in conjunction with IOCS, all input and output operations must be handled by IOCS. The special programming and timing considerations of overlap are completely absorbed by IOCS. Therefore, throughout his program, the user considers the records as being read or written in the *non-overlap mode.* To provide for this, he specifies an *overlap* operation in the DIOCS entry (FEATURES). He must also plan for this feature in the specifications of certain DTF entries and in the allocation of input/output and work areas in storage for certain types of records. To gain the greatest time saving advantage of this feature, two input/output areas should be allotted when tape records are processed. Card records require a work area, and they cannot utilize the read release or punch release special feature. Cards in an input card file cannot be selected. All cards in the card reader stack in the normal pocket. For printer operations, an additional macro instruction (SPACE/SKIP) is provided.

Throughout this bulletin, overlap is indicated whenever it affects a specification.

Record Types and Storage Areas

The IBM tape IOCS processes records that are blocked or unblocked and fixed-length or variable-length. Although all the records in a given file must be the same type, IOCS can process several different-type files in the same operation. The four different combinations of blocking and length are illustrated in the *Schematic of Record Types and Input Areas* (Figure 3). For each combination, the schematic illustrates an input area, several records read in, and the *Autocoder* DA statement for the area. The labels containing the letters DTF refer to the corresponding DTF entry specifications.

Whenever IOCS controls the input/output of any records, a record-mark code may be included *only* to indicate the end of a record. It must not be used for *any* other purpose. Also, because IOCS requires storage positions 90, 91, 95, and 96 when clearing the index registers, these positions must not be used by the programmer.

Unblocked Records

Unblocked records (Figure 3A, B) are read in, or written, one record at a time. This record type includes tape records, card input and output files, and printer operations.

Input/Output Areas

Each input/output storage area allotted for these records should be equivalent in length to the size of a single record. In the case of variable-length records, the area must provide for the largest record.

For tape records, the input/output areas must be defined by DA statements and a group-mark with word-mark must follow the area. The label (symbolic address) of the DA statement must be included in the DTF entries (IOAREAS). Because unblocked records (with one input area specified) are normally processed in the input area and reference must be made to individual fields, word marks must be provided to define these fields, as in any 1401 or 1460 operation. The fields can be labeled and defined with high-order word marks by the DA statement. Or, if the tape contains word marks, they can be inserted by reading the tape record in the load mode.

When IOCS is to check the length of fixed-length input tape records (as specified by the DTF WLRADDR entry), one extra position must be allotted in the input area, for IOCS use. For example, if 100-character records are to be processed, the DA statement for the input area must specify 101 positions. After a record is read in from tape, the extra position is located between the data record and the group-mark with word-mark position. When a correct-length record has been read, this position contains a group mark that was generated when the tape interrecord gap was sensed. If a wrong-length record was read, this position may contain any character other than a group mark.

When an unblocked variable-length record is read in, IOCS automatically inserts a record mark immediately following the record in the input area. This replaces the group mark.

Unblocked tape records processed in the *overlap mode* may be processed in a work area, or in the input/output area if no work area is specified. When two input/output areas are specified (DTF IOAREAS), however, the records can be processed in these areas only if indexing is also DTF-specified.

Because card-read, card-punch, and printer input/output areas are fixed, they need not be defined or included in the DTF entries.

Work Areas

Generally, unblocked records (with one input/output area specified) do not require record work areas, as blocked records do, because fields can be readily defined and used directly in the input/output area. If a work area is desired, however, it must be defined by a DA statement. In this case the individual fields are labeled and defined by high-order word marks in this work area, rather than in the input/output areas. The label of the work area DA statement may be included in the DTF entry (WORKAREA) or referred to in the GET or PUT macro instructions.

The work area must be followed by a record mark or a group-mark with word-mark, if it is used for a fixed-length unblocked record that does not contain a record mark as its last character. Whenever IOCS is to move an unblocked *variable-length* record to or from a work area, that record *must* contain a record mark as its last character.

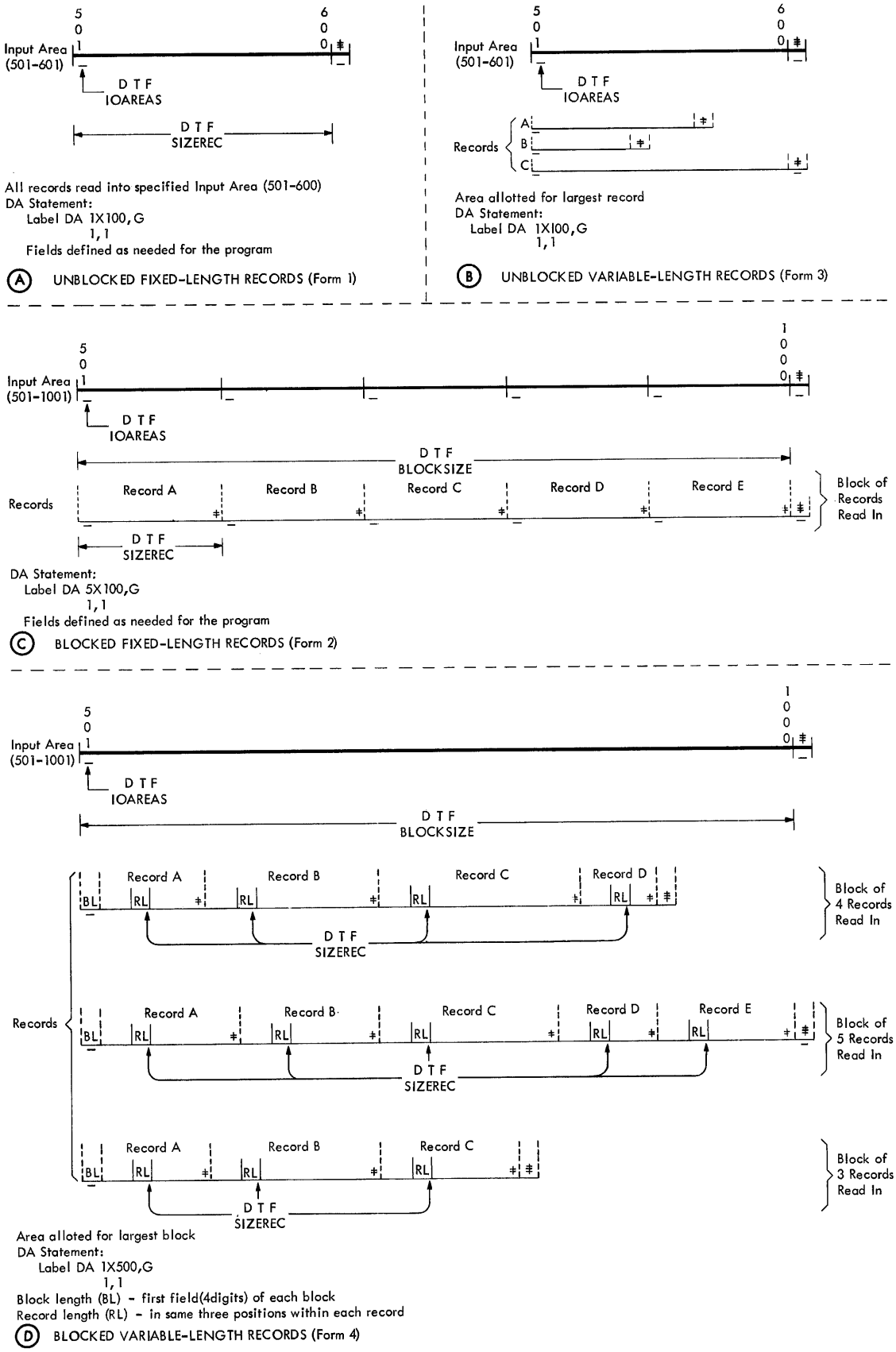


Figure 3. Schematic of Record Types and Input Areas

Card files that use the Read Release or Punch Release special feature, or that are processed in the *overlap mode,* require the use of a work area. All card output files require a work area.

Blocked Records

When blocked records are handled, two or more records at a time are read in from tape, or written on tape. The number of records in a block depends on the size of the records and the amount of storage that can be reserved for the block. The programmer must predetermine the record and block sizes and specify these in the proper DTF entries (SIZEREC and BLOCKSIZE). All records within a block may be the same length (fixed-length), or they may differ (variable-length). This affects the specifications written in the DTF entries (SIZEREC) and in the input, output, and work area DA statements.

When a block of records has been read, each record within the block, and the individual fields of data in that record, must be made available for processing according to the requirements of the job. With proper specifications, IOCS can automatically read the block of records and locate the individual record.

When fixed-length records (Figure 3C) are processed, the individual records and fields can be located by IOCS control in either of two ways:

1. by using the indexing feature to step over to the beginning of each record in the input area.
2. by moving each record, one at a time, to a work area.

One method or the other, but not both, can be specified. Therefore, in planning a job for fixed-length blocked records, the programmer must first determine whether he will process records directly in the input area or move them to one or more work areas for processing. For example, will he identify a control field, such as a part number, in the input area and compare it to another number, or will he move the whole record to a work area and then identify the individual control field within that area. The advisability of using work areas for fixed-length blocked records is affected by the complexities of the user's program to process data for the particular job being planned. Such factors as the amount of core storage used for the job and the index registers required for other functions must be considered. The IOCS routines for the input/output of data handle one method as readily as the other.

Also, the programmer must determine if he will use work areas for fixed-length *output* records, or if he will build them directly in the output area.

When variable-length blocked records (Figure 3D) are read, one or more work areas *must* be used. Indexing *cannot* be used to locate the individual records and fields. When variable-length blocked records are to be written, however, either work areas or the DTF entry VARBUILD can be used to build the blocks.

Requirements of Blocked Records

Several basic requirements of the records themselves must be met, to handle blocked records automatically by IOCS:

1. Each record in every block must contain a record mark as its last character. Therefore, the user must provide record marks in any records that will eventually be read or written by IOCS routines.
2. Fixed-length records must be padded so that all blocks are the same length. When output tape records are created by IOCS, they are automatically padded with blanks unless the user specifies some other character in the DTF entries (PADDING). Padded records are included in record counts and hash totals when these are specified (see *Control Totals).*
3. For variable-length records, a block-length field must be included in each block, and a record-length field in each record (see Figure 3D).

   Block length is a 4-position field and must always be recorded in the first four positions of the block. As the name implies, it contains the total number of characters in the block, including itself and record marks. It is used by IOCS for a wrong-length-record check. The units position of the field must contain AB bits. When output tape records are created by IOCS, this count and the AB bits are generated automatically.

   Record length is a 3-position field and must be located in the same positions within each record in the file. It contains the total number of characters in the record, including itself and the record mark, and is used to modify addresses. For this purpose, the location of the low-order position within the record must be specified in the DTF entries (SIZEREC). For example, the 15th position is specified if the record-length field is located in positions 13, 14, and 15 in each record. Furthermore, the high-order storage position of the record-length field must contain a word mark in the storage area referred to automatically by the IOCS routines. That is, the word mark must be in the work area whenever the work area is specified by the DTF entries (WORKAREA). When the work area is not DTF-specified, the word mark for this field must be in the input/output area. When output tape records are created, the programmer must develop the length of the record to be entered in this field. Unlike block-length, this is not developed automatically by IOCS routines.

Input Area

This area is equivalent in length to the size of the block of records. In the case of variable-length records, the allotted area must provide for the largest block. The area must be defined by a DA statement and followed by a group-mark with word-mark. The label of this DA statement must be included in the DTF entry (IOAREAS). The size of the input area (including the 4-position block-length field, if any) must be specified in the DTF entry (BLOCKSIZE). This does *not* include the group-mark with word-mark position or the extra position for record-length checking if that is specified in the DTF entry (WLRADDR).

When fixed-length records are processed in the input area (indexing specified) and reference must be made to individual fields, word marks must be provided to define these fields, as in any 1401 or 1460 operation. The fields can be labelled and defined with high-order word marks by the DA statement. Or, if the tape contains word marks, they can be inserted by reading the tape record in the load mode. The index register used for this operation is specified in the DTF entries (INDEX REG). It must also be specified in the DA statement if records are processed in the non-overlap mode. If they are processed in the *overlap mode,* the index register must be *omitted* from the DA statement.

When variable-length records are processed and the work area is not specified in the DTF entry (WORKAREA), the record-length fields must be defined with high-order word marks in the input area. This cannot be accomplished by the DA statement. Therefore, the word marks must be entered either by loading a tape record that contains the word marks, or by programming to insert the word marks in the proper positions after the block of records has been read in.

If blocked records are processed in the *overlap mode,* two input areas may be specified in the DTF entry (IOAREAS) and defined by DA statements.

When IOCS is to check the length of a block of either fixed- or variable-length records (as specified by the DTF WLRADDR entry), one extra position must be allotted in the input area. For example, if blocks of five 100-character records are handled, 501 positions must be allotted, followed by a group-mark with word-mark. This can be specified by two DA statements for the area:

```
LABEL     DA     5X100
                 1,1
                 .
                 .
                 .          } Field Definitions
                 .
                 .
          DA     1X1,G
```

After the block of records is read in from tape (fixed-length records or maximum-size variable-length records), the extra position is located between the last data record and the group-mark with word-mark position. When a correct-length record has been read, this position contains a group mark that was generated when the tape interrecord gap was sensed. If a wrong-length record was read, this position may contain any character other than a group mark.

Output Area

The same principles as described for blocked-record input areas apply to output areas. In addition, consideration must be given to the recording of record marks. Because newly developed records are written via the output area, record marks must be provided in this area. They can be specified in the output area DA statement for fixed-length records. For variable-length records, however, the locations of the record marks cannot be predetermined. Therefore the programmer must include program steps to insert a record mark immediately after each record.

Work Area

This area is equivalent in length to the size of an individual record. For variable-length records, the area must provide for the largest record. Each work area must be defined by a DA statement and followed by a group-mark with word-mark. The label of this DA statement must be included in the DTF entry (WORKAREA) or referred to in the GET or PUT macro instructions. Because the work area is used for processing records, the DA statement should also label and define the individual fields with high-order word marks.

Control Totals

The IOCS routines can provide three control totals for tape records with IBM standard labels: block count, record count, and hash total. These totals are accumulated during the run and may be recorded on (output tape) or checked against (input tape) the trailer label, according to the user's specifications. A block count is always taken, and it is checked or written when standard-label operation is specified. For unblocked records, this is the same as the record count. The record count and hash total are taken only if specified in the DTF entries (TOTALS). If a discrepancy is detected in any one of these totals when an input trailer is checked by IOCS (see DTF TOTALS), a programmed halt occurs.

When a hash total is to be taken, the particular field to be accumulated must be identified. For this, the low-order position of the field within the record must be

specified in the DTF entries (TOTALS), and the field must be defined by a high-order word mark in the storage area referred to automatically by the IOCS routines. That is, the word mark must be in the work area whenever the work area is specified by the DTF entries (WORKAREA). When a work area is not DTF-specified, the word mark for this field must be in the input/output area. The hash-total field can have a maximum of ten characters.

The record count and/or hash total may also be specified for tapes with nonstandard labels. In this case they are accumulated automatically, but the user must program to check (input) or write (output) them.

Record Input and Output

Once the programmer has determined the types of records to be handled and has planned the input, output, and work areas for his job, he writes only one instruction each time the program calls for a record to be read, written, or punched. For these operations, IOCS makes two macro instructions (GET and PUT) available to the programmer. Two others (RELSE and SPACE) are provided for special conditions.

GET Macro

This instruction locates the next single record for processing, and it can be written in either of two basic forms (Figure 4). In both forms, the term FILEA represents the symbolic name of the file assigned in the DTF entry. The term WORKA, in the second form, represents a work area used for the file and labelled in the DA statement. The form of the GET macro to use, and the specific functions performed, are determined by the types of records being handled and by the processing plans specified in the file descriptive entries (DIOCS and DTF).

Blocked Fixed-Length Records

PROCESSING IN THE INPUT AREA

Whenever records are to be processed in the input area, the first form of the GET macro (GET FILEA) is used.

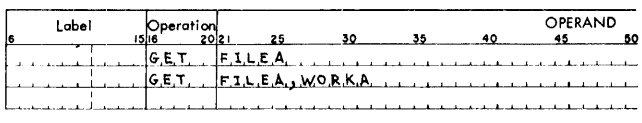


Figure 4. GET Macro Instruction

Blocked fixed-length records can be processed in the input area, only if indexing is specified in the DTF entries (INDEXREG). In this operation (Figure 5A), the first GET instruction causes a block of records to be read from tape to the input area, and it initializes the index register so that reference can be made to the first record in the block. Subsequent GET instructions increment the index register, and successive records can be processed. After all records in the block have been processed; the next GET again reads a block of records and initializes the index register.

When records are processed in the *non-overlap* mode, the particular index register specified in the DTF entry should be included in the input area DA statement. Also, the individual fields should be labelled in the DA statement, for reference throughout the program (Figure 6).

When records are processed with indexing in the *overlap* mode, however, two basic changes must be made in the DA statement for the input area (Figure 7, relate to Figure 6):

1. Field labelling *must* be omitted.
2. The index register *must* be omitted.

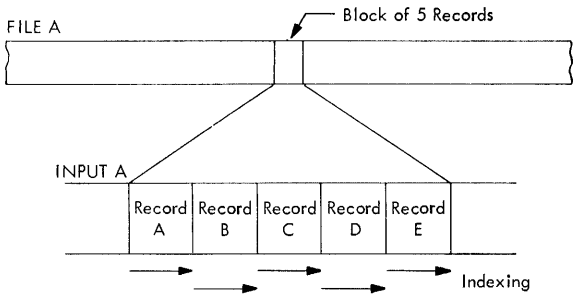
To assign labels to fields set up in the DA statement, the user must equate these labels to one-less-than the actual positions of the fields within the record. This is necessary because, with overlap, the index register always contains the *address* of the high-order position of the record to be processed. Indexing must be indicated either in the equate statements, as shown, or in the individual instructions throughout the program.

PROCESSING IN WORK AREAS

The first form of the GET macro (GET FILEA) is used whenever all records in an input file are to be processed in the same work area. For this, the label of the work area must be included in the DTF entries, and indexing must be omitted.

The second form of the GET macro (GET FILEA, WORKA) is used whenever records are to be processed in different work areas. It specifies the work area required for each separate record. It may be advantageous to set up two work areas, for example; and to specify each area in alternate GET instructions. This would permit the programmer to compare each record with the preceding one, for a control change. Whenever work areas are to be specified in GET instructions, both indexing and a work area specification must be omitted from the DTF entries.

When work areas are used (see Figure 5B, C), the first GET instruction in the program transfers a block of records from tape to the input area, and then moves the first record in the block directly to the specified work area. Each subsequent GET instruction moves

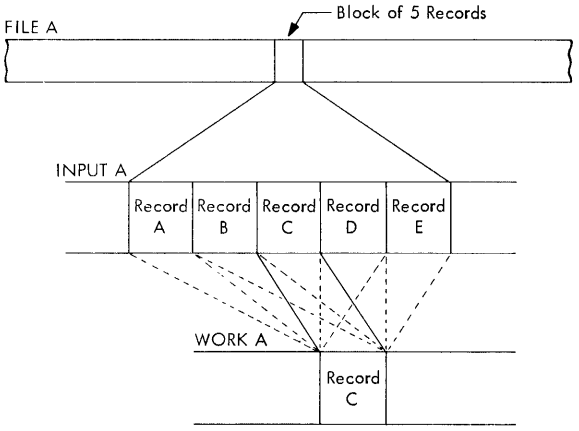


GET FILEA

(DTF INDEXREG Specified
DTF WORKAREA Not Specified)

Used for fixed-length records when processing can be performed in the input area.

(A)

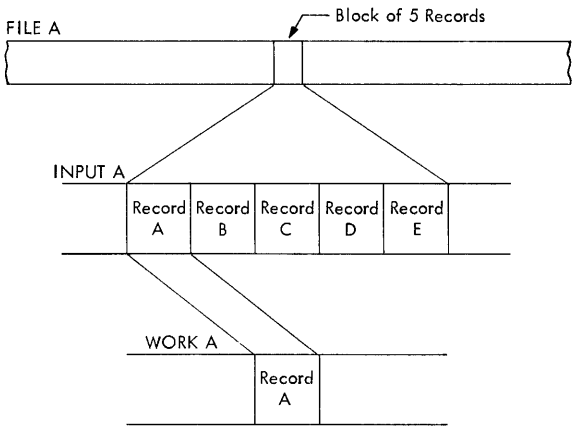


GET FILEA

(DTF INDEXREG Not Specified
DTF WORKAREA Specified)

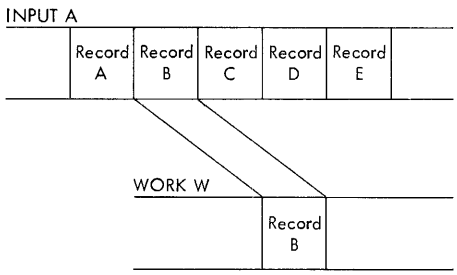
Used for fixed-length or variable-length records when one record work area is required.

(B)



GET FILEA,WORKA

Next GET Instruction



GET FILEA,WORKW

(DTF INDEXREG Not Specified
DTF WORKAREA Not Specified)

Used for fixed-length or variable-length records when records are moved to different record work areas.

(C)

Figure 5. Reading Blocked Records

Label	Operation	OPERAND
INPUTA	DA	5x100,,X2,,G
DATE		32,37
NAME		11,30
MANNO		4,,8
YRGROS		70,,76
YRWTAX		85,,89
YRFICA		90,,94

Figure 6. DA Statement for Indexed Records, Non-Overlap Mode.

Label	Operation	OPERAND	
INPUTA	DA	5x100,,G	
		32,,37	
		11,,30	
		4,,8	Input Area DA Statement
		70,,76	
		85,,89	
		90,,94	
MANNO	EQU	7+X2	Equate Statements
YRGROS	EQU	75+X2	
	C	MANNO,,EMPNO	Program Instructions
	A	CURRGR,,YRGROS	

Figure 7. Indexed Records, Overlap Mode

the next individual record from the input area to the work area, until all records in the block have been processed. Then a new block is automatically read in from tape, and the operation is repeated.

Work areas specified for input files may also be specified for corresponding output files.

The principle of specifying the work area in the GET instruction is also used when an *output area* is to be treated as a *work area.* When the output records are blocked in this operation, indexing must be specified in the DTF entry for the output area. If records are processed in the *non-overlap* mode, the index register is included in the DA statement for the output area, and the GET macro is written with the label of the output area in the operand field (Figure 8). If records are processed in the *overlap* mode, however, the index register cannot be specified in the output area DA statement, and must be included in each GET instruction. Instead of entering the label of the output area in the operand field, the output area is indicated by "0 + Xn" (zero + the number of the index register; Figure 9). This is necessary because, with overlap, the index register always contains the address of the high-order position of the record to be processed. The DA statement and fields for this output area must be set up in the same manner as described for the DA statement when records are processed in the input area with indexing and overlap (see Figure 7).

PROCESSING PADDED RECORDS

Input files which contain padded records must be programmed to prevent the padded records from being processed. This checking may be done immediately following each GET for that file. When the first padded record is detected, a simple GET loop may be entered which will cause all subsequent padded records for that file to be bypassed until an EOF condition is reached or a RELSE macro may be issued, followed by another GET. IOCS will then automatically branch to the user's EOF routine as specified under the DTF entry.

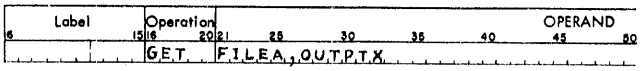


Figure 8. Output Area Used as Work Area, Non-Overlap Mode

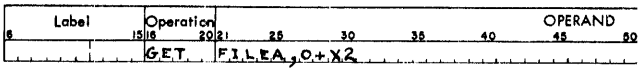


Figure 9. Output Area Used as Work Area, Overlap Mode

Blocked Variable-Length Records

One or more work areas must be used to process these records. They cannot be processed in the input or output area, and indexing must not be specified in the DTF entry for this type of record. The GET instructions used and the operations performed to transfer variable-length records from tape to a core-storage work area are the same as described for blocked fixed-length records, under *Processing in Work Areas.*

Unblocked Records

When unblocked records (both unit record and tape) are handled, each GET instruction transfers a single record to the input area. If a work area is specified in either the DTF entry or the GET instruction (see *Processing in Work Areas),* each GET then moves the record directly from the input area to that work area for processing. A work area is required for card input files whenever the read release special feature or the processing overlap special feature is used.

If unblocked tape records are to be processed in *two* input areas (overlap mode), indexing is required. The programming for this operation is the same as that described for blocked fixed-length records processed in the input area with indexing and overlap.

STACKER SELECTION

Card input files should be selected to stack in pocket 1 or 2, because the IOCS card-read error routine stacks any unreadable cards in the normal pocket. This selection (with cards processed in the non-overlap mode) may be specified in the DTF entries (CARDPOC) when all cards are to be stacked in the same pocket, or it may be specified in the GET macro, but not in both. With the read release special feature, stacker selection (if any) must be specified in the DTF entries.

To specify stacker selection in the GET instruction, the basic form of the macro instruction is modified to indicate the pocket number (Figure 10). In the first form of the GET macro, two commas separate "FILEA" and "2". In the second form, one comma separates "WKAREA" and "1" in the operand field. Because commas are always used to separate various operands, either form enters the number as the *third* operand so that it can be recognized as a pocket number by the IOCS routines.

Stacker selection cannot be specified for card input records processed in the overlap mode. With this feature, all cards are stacked in the normal read pocket.

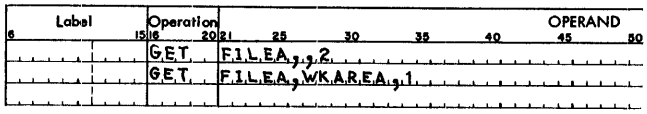


Figure 10. GET Instruction with Read Stacker Selection

PUT Macro

This instruction is used to write or punch a record that has been processed. It operates much the same as the GET macro, but in reverse. Similar to GET, it can be written in either of two basic forms (Figure 11). In both forms, the term FILEX represents the symbolic name of the file assigned in the DTF entry. The term WORKX, in the second form, represents a work area used for the file and labelled in the DA statement. The form of the PUT macro to use, and the exact functions performed, are determined by the processing plans and file specifications (DIOCS and DTF entries).

Blocked Fixed-Length Records

BUILDING IN THE OUTPUT AREA

Whenever records are built directly in the output area, the first form of the PUT macro (PUT ,FILEX) is used.

Blocked fixed-length records can be built in the output area, only if indexing is specified in the DTF entries (INDEXREG). In this operation (Figure 12A), each PUT instruction increments the index register so that the next record can be built in the next record-area within the output block. Also, each PUT tests the output area to see if it is filled. When it is, the block of records is automatically written on tape, and the index register is initialized so that the following record will be built at the beginning of a new block.

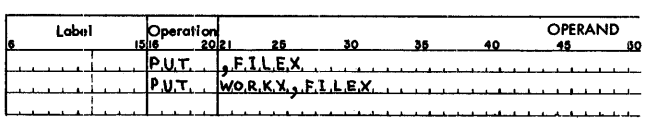


Figure 11. PUT Macro Instruction

When records are processed in the *non-overlap* mode, the particular index register specified in the DTF entry must be included in the output area DA statement, and the individual fields should be labelled in the DA statement for reference throughout the program. This is similar to the indexed *input area* DA statement (see Figure 6). However, when records are processed in the *overlap* mode, the index register and field labelling must be omitted from the DA statement. The DA statement and fields must be set up in the same manner as described for the input area when blocked fixed-length records are processed with indexing and overlap (see Figure 7). To initialize the index register and provide the address for the *first record* in a run, with overlap, a PUT ,FILEX instruction must be issued before the first record is built in the output area.

BUILDING IN WORK AREAS

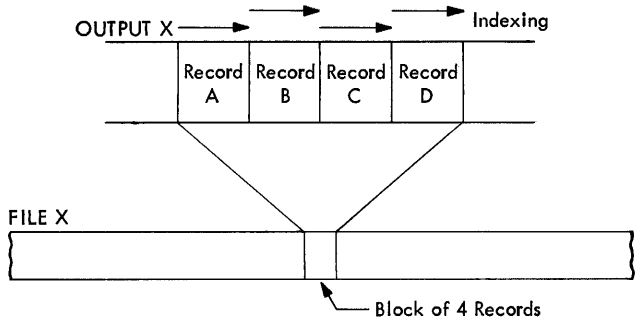
The first form of the PUT macro (PUT ,FILEX) is used whenever all records in an output file are to be built in the same work area. For this, the label of the work area must be included in the DTF entries and indexing must be omitted.

The second form of the PUT macro (PUT WORKX, FILEX) is used whenever records are to be built in different work areas. It specifies the work area required for each separate record. Both indexing and a work area specification must be omitted from the DTF entries, with this type of PUT instruction.

When work areas are used (Figure 12C, D), each PUT instruction moves an individual record from the specified work area to the proper location in the output area. Also, each PUT tests the output area to determine if it is filled. If it is, the completed block of records is automatically transferred to the output tape.

Work areas specified for output files may be the same work areas as specified for corresponding input files.

The principle of specifying the work area in the PUT instruction is also used when an *input area* is to be treated as a *work area.* That is, each record is processed (built for output) in the input area and then moved to the output area to be written on the output tape. When the input records are blocked in this operation, indexing must be specified in the DTF entry for the input area. If records are processed in the *non-overlap* mode, the index register must be included in the DA statement for the input area. Also, the PUT macro is written with the label of the input area in the operand field (Figure 13). If records are processed in the *overlap* mode, however, the index register cannot be specified in the input area DA statement, and it must be included in each PUT instruction. Instead of entering the

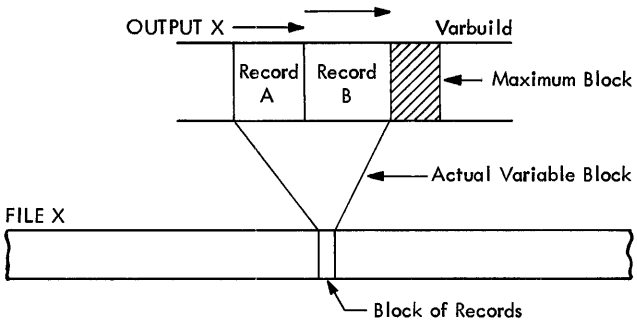


PUT ,FILEX

(DTF INDEXREG Specified
DTF WORKAREA Not Specified)

Used when fixed-length records can be built in the output area.

Ⓐ

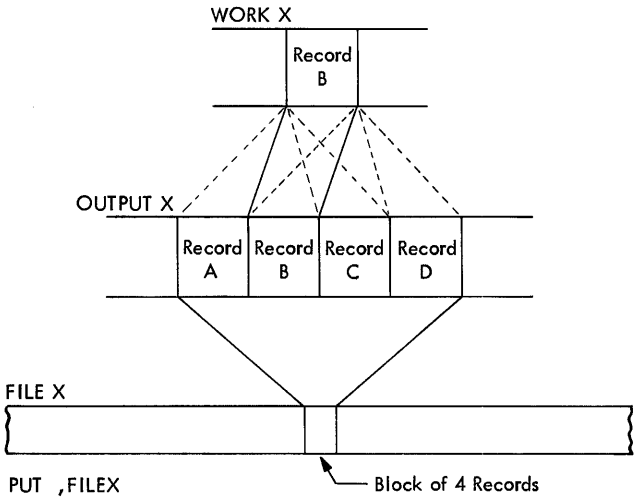


PUT ,FILEX

(DTF INDEXREG Not Specified
DTF WORKAREA Not Specified
DTF VARBUILD Specified)

Used when variable-length records can be built in the output area.

Ⓑ

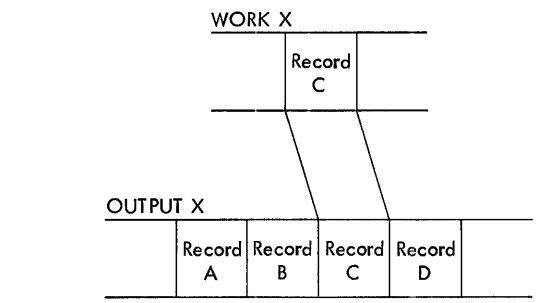


PUT ,FILEX

(DTF INDEXREG Not Specified
DTF WORKAREA Specified)

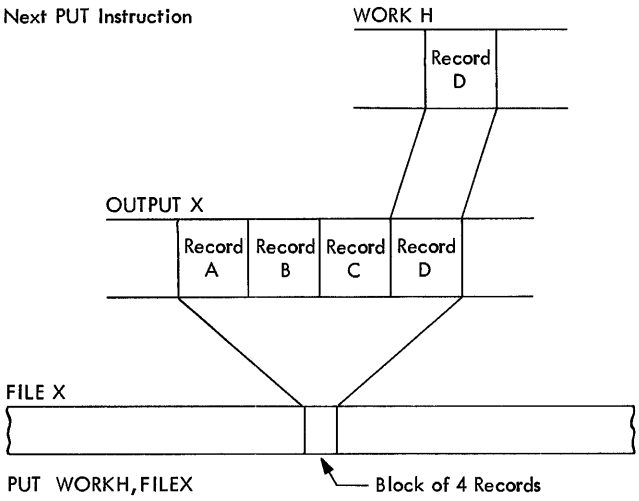
Used for fixed-length or variable-length records when one record work area is required.

Ⓒ



PUT WORKX,FILEX

Next PUT Instruction



PUT WORKH,FILEX

(DTF INDEXREG Not Specified
DTF WORKAREA Not Specified)

Used for fixed-length or variable-length records when records are built in different work area.

Ⓓ

Figure 12. Writing Blocked Records

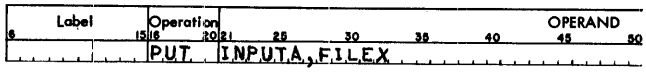


Figure 13. Input Area Used as Work Area, Non-Overlap Mode

label of the input area in the operand field, the input area is indicated by "0 + Xn" (zero + the number of the index register; Figure 14). This is necessary because, with overlap, the index register always contains the address of the high-order position of the record to be processed. The DA statement and fields for this input area must be set up in the same manner as described for processing records in the input area with indexing and overlap (see Figure 7).

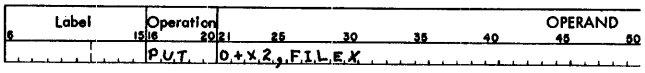


Figure 14. Input Area Used as Work Area, Overlap Mode

Blocked Variable-Length Records

One or more work areas may be used to build these records. The PUT instructions used to transfer variable-length records from a work area to tape are the same as those described for blocked fixed-length records, under *Building in Work Areas.* In addition, the length of each record must be determined by the user's program and included in the record-length field. This length is then used by IOCS in a test to see if the record fits in the allotted output-block area.

Blocked variable-length records can be built in the output area if the DTF entry labelled VARBUILD is specified. Indexing cannot be specified for variable-length records. VARBUILD performs functions for variable-length blocked records that are similar to those performed by INDEXREG for fixed-length blocked records. In this operation (see Figure 12B), each PUT instruction tests to see if the *next* record to be built will fit in the allotted output-area block. If it will, the record is built in the current block. If it will not, the block is written on tape and the next record is built at the beginning of a new block. For this operation and because the records are variable-length, the programmer must *determine* the length of the next record before a PUT instruction is issued at the end of building the present output record. That is, before he writes the PUT instruction for output record A, he must read the input data for output record B and determine its length. Variable-length output records are generally developed from variable-length input records, perhaps modified by current-period card or tape records. The variable-length input must contain a record-length field and this, with any current modification, will give the length of the output record to be built. The programmer enters this length in the VARBUILD field.

The DTF VARBUILD entry must contain the label of a three-position field, which may be index register 2, index register 3, or any other 3-character area defined by the programmer. The VARBUILD field performs two functions:

1. It makes the length of the next record to be built (entered by the programmer) available to the IOCS routines. IOCS can then determine if the present block of records should be written, or if the next record will fit in the block.
2. It makes available the address of the high-order position of the next record to be built. Thus, after the PUT instruction for record A, the address for record B is available for the programmer's use in building record B in the output area. If an index register is used for the VARBUILD field, the individual fields can be built readily by indexing each instruction. If a three-position field other than index register 2 or 3 is specified, it is used to modify each address as the fields are built.

A simplified program and three sample records (Figure 15) illustrate the use of VARBUILD. The illustration assumes this setup:

- DTF VARBUILD entry specifies X2 (index register 2) as the VARBUILD field.
- FILEA—Variable-length input file, with DTF-specified work area.
- FILEX—Variable-length output file.
- OUTPTX—100-Position output block (assembled address 501).
- LENGTH—Work area to accumulate length of next record.

Several factors should be especially noted in this illustration:

1. The *next* record is read and its length determined before the PUT instruction for a record is issued.
2. The length of the *next* record is moved into the VARBUILD field (X2) before the PUT instruction is issued for the completed record.
3. The PUT instruction tests to determine that the *next* record will fit in the current block.
4. After the PUT instruction, the address of the high-order position of the next record is available in the VARBUILD field (X2).
5. Comparable to factors 1-4 above, the first input record must be read and its length determined, and a PUT instruction must be given *before* the *first* output record is built. This makes the address for the first record available in the VARBUILD field. Similarly, if an output file is *released* (see *RELSE Macro),* a PUT instruction must be issued before the first record of the new block is built.
6. When the next record will not fit in the current block, the block is written and the address for the next record is provided.
7. A record-length field is built in each output record by the programmer for later use when this record becomes *input.* This must always be located in the same positions within each record.

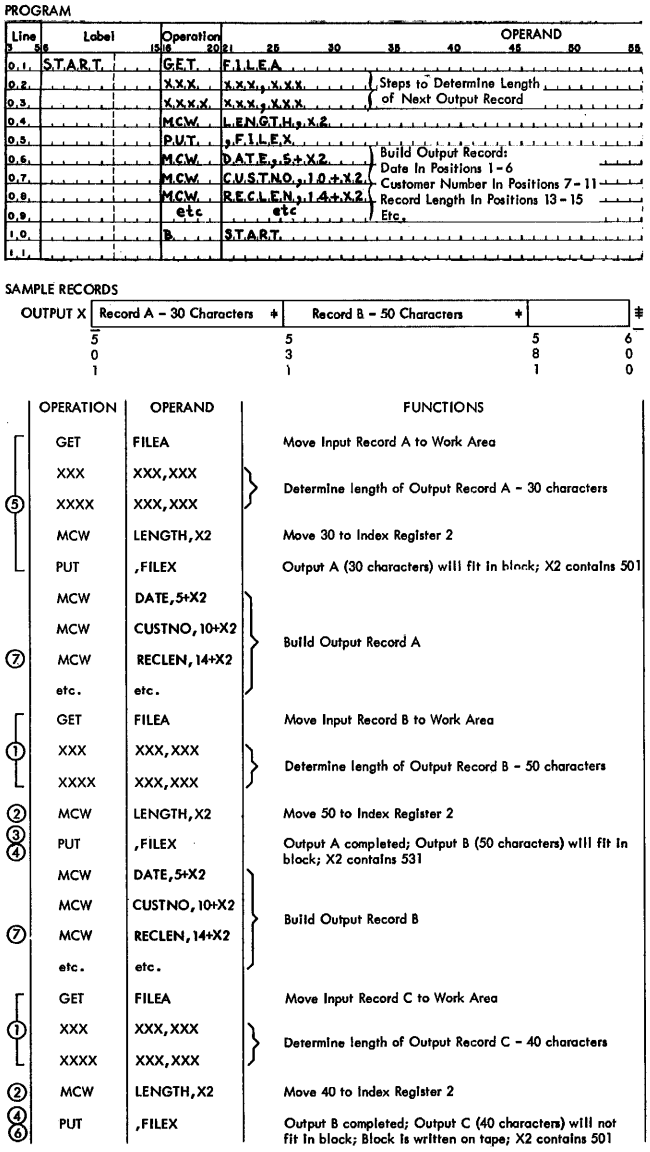


Figure 15. Variable-Length Records Built with VARBUILD

Unblocked Records

When unblocked records (card, tape, or printer) are handled, each PUT instruction transfers a single record from the output area to the output file. If a work area is specified in either the DTF entry or the PUT instruction (see *Building in Work Areas),* each PUT first moves the record from that work area to the output area, and then transfers it to the output file. A work area is required for card output records. They cannot be built directly in the output area because of the IOCS routines for card punch checking.

If unblocked tape records are to be built in *two* output areas (overlap mode), indexing is required. The programming for this operation is the same as that described for blocked fixed-length records processed with indexing and overlap.

STACKER SELECTION

Card output files may be selected to stack in pocket 4 or 8. This selection may be specified in the DTF entries (CARDPOC) when all cards are to be stacked in the same pocket, or it may be specified in the PUT macro, but not in both. It is specified in PUT by modifying the basic form of the macro and writing the pocket number as the third operand (Figure 16).

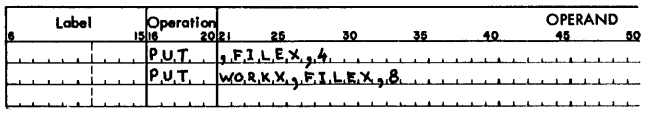


Figure 16. PUT Instruction with Punch Stacker Selection

Unlike card input files, this selection applies to card output records processed in either the non-overlap or overlap mode.

PRINTER FORMS CONTROL

Spacing and skipping of forms can be controlled by the IOCS routines. The operation may be specified in the DTF entries or in the PUT instruction, but not in both. In either case, the standard IBM 1401 d-character for forms control is used to indicate the desired operation (see *System Operation Reference Manual for* IBM *1401 and 1460,* Form A24-3067). The d-character is specified in the DTF entry (FORMCNTL) if the same operation is to be performed for each printed line, such as double-spacing. It is specified in the PUT instruction (Figure 17) whenever different spacing or skipping is to occur for different printed lines.

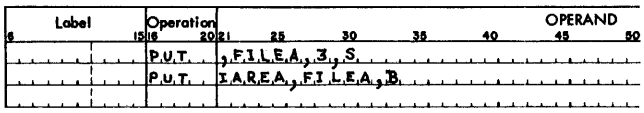


Figure 17. PUT Instruction with Forms Control

The layout of a form may require certain spacing (or skipping) either before *or* after a particular line of printing, or both before *and* after printing. When one control (before *or* after) is required, the d-character is entered as the third operand in the PUT instruction. For control both before *and* after, the d-character for *immediate* spacing or skipping (before printing) must be entered as the third operand, and the d-character for *after print* spacing or skipping as the fourth operand.

RELSE (Release) Macro

The release macro instruction is used in conjunction with blocked tape records. It allows the programmer to skip the remaining records in a block and continue processing with the first record of the next block. This function applies to a job in which records on tape are categorized and each category (perhaps a major grouping) is planned to start as the first record in a block. For example, sales data may be recorded and analyzed by division (major), district (intermediate), and branch office (minor). Then, if management frequently requires special analyses of sales for certain specified divisions, these analyses can be obtained quickly and efficiently with a system planned so that the records for each division start at the beginning of a new block of records. The specified division can be located readily by checking only the first record in each block. If that record is not in the specified division, the other records in the block can be ignored and the first record of the next block can be checked.

The symbolic name of the file, specified in the DTF entry, is entered in the operand field of the release instruction. In the overlap mode, the word OVERLAP must also be entered in the operand field (Figure 18). When this is an input file, the next GET instruction reads in a new block of records and makes the first record available for processing. If indexing is used, the index register is initialized. Because input records are skipped in this operation, record counts and hash totals cannot be taken.

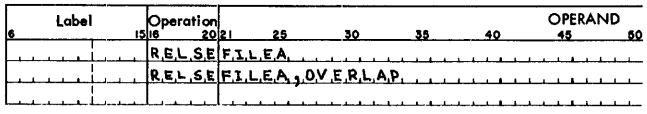


Figure 18. RELSE Macro Instruction

When an output file is released, the next PUT instruction causes the existing block of records to be written on tape. The new record becomes the first record of a new block. With blocked fixed-length records, any unfilled portion of the block is padded with the character specified in the DTF entry (PADDING), or with blanks. The padded records are included in any record counts or hash totals. With variable-length records and an unfilled portion, a short block is written.

When output records are built in work areas, the control number in each record can be examined (by the user's program) as the record is being built, to determine whether or not the record is the first one of a new major group. If it is, RELSE is issued and the PUT instruction for that record causes the existing block to be written *before* the record is moved from the work area to the output area. If records are built directly in the output area, however, examining the control number of a record as it is built would cause the first record of a new group to be erroneously included in the previous block when RELSE and PUT for that record are issued. Therefore, programming must be provided to make sure that the first record of a new group *is* in a new block. One method is to determine the control number of the *next* output record after the PUT instruction is issued at the end of building each record. If the *next* record belongs in a new group, RELSE is issued after the PUT instruction for the last record of the present group.

SPACE/SKIP Macro

When records are processed in the overlap mode, spacing and skipping of printer forms may be controlled by the SPACE/SKIP macro instruction, instead of the DTF entry (FORMCNTL) or the PUT instruction. The 1401 or 1460 d-character for the desired operation is entered in the operand field of the SPACE or SKIP instruction.

Tape Labels

Data processing installations using magnetic tape storage have many reels of tape to be stored and handled. To ensure that the correct tape reel of data is used in each job, it is common practice to identify the tape itself with *header* and *trailer* labels, in addition to a printed label on the outside of the reel. A header label is the first record on tape and contains such factors as identification numbers, identification name, and effective dates. A trailer label is at the end of data on tape and provides control totals and an indication that the reel is, or is not, the last reel for the job.

The format of both header and trailer labels may be the IBM standard form, or it may be a different arrangement (nonstandard) planned by the user. The IOCS routines can process tapes with either standard or nonstandard labels, or without labels. However, only standard labels are automatically written on output tapes and automatically checked on input (or output) tapes. Nonstandard labels are written or checked by the user's program, outside the IOCS routines. In each job the user must specify in the DIOCS entries (LABELDEF) and in the DTF entries (TYPELABEL) the type of labels to be processed.

Standard Header Label

A standard header label is 80 characters in length (Figure 19). The first 40 positions are used for seven fields that contain standard identifying information, and the remaining 40 positions are blank. The first two fields (Header and Tape Serial Number) permanently identify the reel itself. The other five fields identify the job and are automatically written (or checked) with information specified in the DTF entries. If a job requires more than one tape reel of data, the same header information (with the exception of tape and reel numbers) is repeated on all reels. Information may be written or checked in the last 40 positions, if desired, by the user's program (see *Label Operation*).

When an input tape with a standard header label is to be processed, the user must specify (in DTF CHECKLABEL) the checking he wants performed by the IOCS routines: complete (the standard identifying information), partial (name only), or none at all. If an error is detected in this check, a programmed halt occurs. After the halt, operation can be started using the same tape, if desired, or the tape can be replaced. In this case, the header of the new tape is checked before data is processed.

When an output tape is to be written, the effective dates in the *old* header should be checked to ensure that the data on the tape is no longer active and may be destroyed. Therefore, IOCS automatically checks the retention cycle in the old header against the time elapsed between the creation date and today's date. If an error is detected, a programmed halt occurs. Then, the tape may be changed, or operation may be started using the same tape.

The seven identifying fields in the standard header label are:

1. *Header*—consists of the digit 1, the letters HDR, and a blank position.

   When a new reel of tape is received in an installation and before it is used in an IOCS job, the characters 1BLNK should be written in this field as a temporary header identification. The first time records are written by IOCS, 1BLNK is automatically replaced with the standard header identification (1HDRb).

   Input Tape: This field identifies the first record as the header label.

   Output Tape: This field is written automatically by the IOCS routines.

2. *Tape Serial Number*—5 digits

   This is the sequential number of the tape reel within the whole installation. As soon as a new reel of tape is received in the installation, it should be identified in this field with the next available number. The IOCS routines do not affect this number.

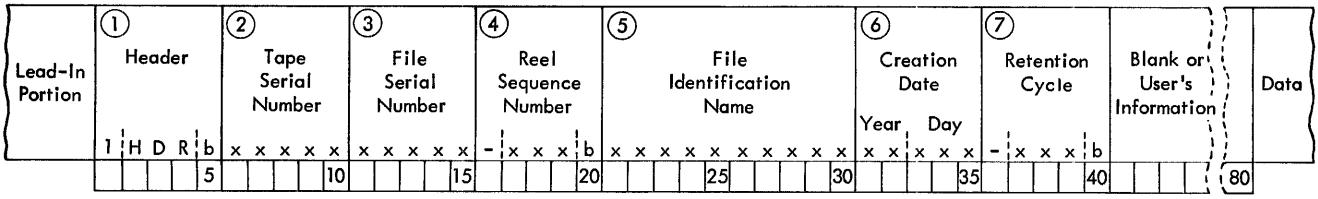


Figure 19. Schematic of Standard Header Label

3. *File Serial Number*—5 digits

   This generally represents a job number in an installation.

   Input Tape: When the header is to be completely checked, this number must be specified in the DTF entry SERIALNUM.

   Output Tape: The number to be recorded must be specified in the DTF entry SERIALNUM, unless the user plans to make this number the same as the *tape* serial number. If the tape serial number is used and the job requires two or more reels, the tape number of the *first* reel becomes the file serial number on all reels.

4. *Reel Sequence Number*—consists of a minus sign, 3 digits, and a blank position.

   This field is used to number the reels in a multi-reel job. The number of the first reel is 001 unless the user specifies some other number (3 digits) in the DTF entry REELSEQ.

   Input Tape: When the header is to be completely checked, reel sequence is checked automatically.

   Output Tape: Reels, after the first, are serially numbered automatically.

5. *File Identification Name*—10 characters

   This may be a job name in an installation. This identification *must* be ten positions long with a significant character, not a blank, in the tenth position. Also, no more than one blank may separate two significant characters within the field.

   Input Tape: When the header is to be checked (either completely or partially) this name must be specified in the DTF entry HEADER.

   Output Tape: The name to be recorded must be specified in the DTF entry HEADER.

6. *Creation Date*—5 digits

   The dating system consists of 2 digits for the year, followed by 3 digits for the day of the year.

   Input Tape: When the header is to be completely checked, this date must be specified in the DTF entry HEADER.

   Output Tape: The date is taken from today's date in storage. The user must load the date in storage positions 195-199. To do this, he punches a card with today's date and loading instructions, and inserts it in the object-program condensed deck following the IOCS portion:

COLUMNS	PUNCH
1-5	(Today's date)
40-46	L005199
47-53	N000000
54-60	N000000
61-67	N000000
68-71	1040

7. *Retention Cycle*—consists of a minus sign, 3 digits, and a blank position.

   This specifies the number of days after the creation date that the tape is to be kept active.

   Input Tape: When the header is to be completely checked, the number of days (3 digits) must be specified in the DTF entry HEADER.

   Output Tape: The number of days (3 digits) to be recorded must be specified in DTF HEADER.

The information for header fields 3-7 is specified in a RDLIN information card, whenever a RDLIN macro instruction is used in a program (see *RDLIN Macro).*

Standard Trailer Label

A standard trailer label is 80 characters in length (Figure 20). The first 30 positions are used for four fields that contain standard control checks, and the remaining 50 positions are blank. The first field indicates whether or not this is the last reel of data for a job. The other three fields provide control totals and are written (or checked) with totals accumulated during the run, if specified in the DTF entries. Information may be written or checked in the last 50 positions, if desired, by the user's program (see *Label Operation).*

As he does with header labels, the user must specify checking in the DTF entries (CHECKLABEL) whenever he wants the IOCS routines to check the standard trailer label of an input tape. The three control totals are checked, and if an error is detected a programmed halt occurs. Operation can be resumed, if desired, or the error can be displayed for the operator to investigate.

The four fields in the standard trailer label are:

1. *Trailer*—consists of the digit 1, the letters EOF or EOR, and a blank position.

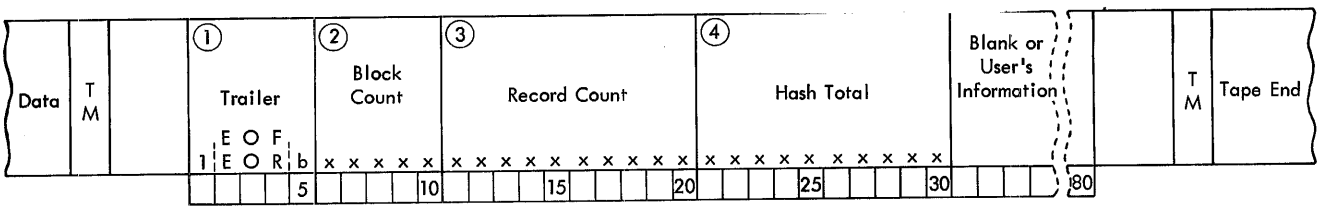


Figure 20. Schematic of Standard Trailer Label

The letters EOF (end of file) indicate that this is all the data for the job. EOR (end of reel) indicates that this is the end of a reel of tape but one or more reels follow for the same job.

Input Tape: The field identifies the record as a trailer label and indicates if another reel follows.

Output Tape: The letters EOF are written automatically when a CLOSE macro instruction is given. The letters EOR are written automatically when the tape reflective spot is sensed or when a FEORL macro instruction is given.

2. *Block Count*—5 digits

   A count of the number of blocks processed is always accumulated automatically by the IOCS routines. In the case of unblocked records, this count is the same as a record count.

   Input Tape: When the trailer label is to be checked, the count accumulated is compared to the count recorded on the tape when it was written.

   Output Tape: The accumulated count is written automatically.

3. *Record Count*—10 digits

   This is a count of the number of records processed. With unblocked records, this is the same as the block count.

   Input Tape: When a trailer that contains a record count is to be checked, this must be specified in the DTF entry TOTALS.

   Output Tape: This count is accumulated and written automatically if it is specified in the DTF entry TOTALS. If it is not specified, blanks are written in this field.

4. *Hash Total*—10 digits

   A hash total is the total of some item, such as customer number, for processing-control purposes. It is not intended to be a significant total such as the total dollar amount of sales would be.

   Input Tape: When a trailer that contains a hash total is to be checked, the low-order position of the same field selected when the tape was written must be specified in the DTF entry TOTALS.

   Output Tape: The hash total is accumulated and written automatically if the low-order position of the selected field in the record is specified in the DTF entry TOTALS. If it is not specified, blanks are written in this field.

Label Operation

Tape labels are processed by IOCS *open, close,* and *end-of-reel* routines. The operations performed when a file is opened or closed, or when an end-of-reel condition occurs, are affected by the types of labels and the specifications in the DIOCS and DTF entries.

If standard labels are specified, the first record read from the tape reel, or written on it, is assumed by IOCS to be a header label. Similarly, the last three records on the reel are assumed to be a tape mark, a trailer label, and another tape mark. The IOCS routines set up an 80-position area into which the label is read from tape for automatic checking, or in which information is built from the DTF entry specifications for writing an output label. This 80-position area has a symbolic address of IOCSLB (IOCS Standard Label). The programmer can use this address if he wishes to modify the standard label.

If nonstandard labels are specified, IOCS does not set up the 80-position IOCSLB area, but does provide *exits* for the programmer to read, write, and/or check the labels himself.

If no labels are specified, IOCS assumes that the first record on tape consists of data, *not* header information, and that no record follows the tape mark at the end of data.

Exits

In order to modify standard labels or to handle nonstandard labels, seven *exits* from the IOCS routines are provided to branch to the programmer's own routines. Each of the exits occurs at a specific time in the IOCS open, close, or end-of-reel routine and is used for a specific function. The programmer must specify the appropriate one for the operation he wishes to perform, by selecting the corresponding DTF entry (EX1ADDR, EX2ADDR, ---------EX8ADDR) and entering the symbolic address of his routine in the operand field. In his program, the user returns to the IOCS routines by branching to IOCSRE (IOCS Return) at the end of his own routine. An eighth exit is provided to permit including a user's routine *before* a trailer label is written, when the end of a reel of output tape is signalled by the reflective spot on the tape.

The specific functions of the eight exits (Figure 21) are:

- *Exit 1* allows the programmer to modify a *standard output trailer.* This is used to enter information in positions 31-80, or to change any information in positions 1-30. In his routine, the programmer moves the information to the desired positions in IOCSLB and returns to IOCS by IOCSRE. The trailer is then written on tape by the IOCS routines.
- *Exit 2* is used to build and write *nonstandard output trailers* or *additional labels* following a standard output trailer. The user must provide his own area for building the trailer information and must program to write the trailer on tape before returning to IOCS by IOCSRE.

EXIT NO.	HEADER		TRAILER		USE
	Standard	Non-Standard	Standard	Non-Standard	
1			Output		Modify standard trailer
2				Output	Build and write non-standard trailers
3	Output				User checks old standard header
4	Output				Modify standard header
5		Output			Read and check old nonstandard headers; build and write nonstandard headers
6			Input	Input	Additional check of standard trailer; read and check nonstandard trailers; modify reel count; change DTF REWIND specification
7	Input	Input			Additional check of standard header; read and check nonstandard headers
8					User's routine at end of reel (reflective spot) on output tape

Figure 21. Exits from IOCS Routines for Header and Trailer Labels

The totals accumulated by IOCS throughout the run are available at these symbolic addresses:

Hash Total —IOCHSH (10-position field)
Record Count—IOCRCT (10-position field)
Block Count —IOCBLK-1 (5-position field)

The hash total and record count are accumulated only if specified in the DIOCS COUNTS and DTF TOTALS entries.

- *Exit 3* permits the user to check a *standard header* on an *output* tape, instead of the IOCS *open* routine checking the retention cycle automatically. The programmer can obtain the header information for checking from IOCSLB.
- *Exit 4* allows the programmer to modify a *standard output header*. This is used to enter information in positions 41-80, or to change any information in positions 1-40. In his routine, the programmer moves the information to the desired positions in IOCSLB and returns to the open routine by IOCSRE. The header is then written on tape by the IOCS *open* routine.
- *Exit 5* is used to build and write *nonstandard output headers* or *additional labels* following a standard output header. The user must provide his own area for building the header information and must program to write the header on tape before returning to the *open* routine by IOCSRE. When nonstandard labels are specified, this exit can also be used to read and check the old header on an output tape.
- *Exit 6* permits:

A. additional checking of a *standard input trailer.*

The programmer obtains the information to be checked from IOCSLB.

B. reading and checking *additional labels* following a standard input trailer.

The user must provide his own area for entering the trailer information and he must program to read and check the information. Programming must be returned to IOCS by branching to IOCSRE.

C. reading and checking *nonstandard input trailers.*

The user must provide his own area for entering the trailer information and he must program to read and check the information. Programming is returned to IOCS by branching to IOCSRE on an *end-of-reel* condition, but a branch to the end-of-file address (specified in DTF EOFADDR) should be programmed for an *end-of-file* condition.

The totals accumulated by IOCS throughout the run are available at these symbolic addresses whenever Exit 6 is used:

Hash Total—IOCHSH (10-position field)
Record Count—IOCRCT (10-position field)
Block Count—IOCBLK-1 (5-position field)

The hash total and record count are accumulated only if specified in the DIOCS COUNTS and DTF TOTALS entries.

D. modifying the reel sequence count.

This count is stored in a three-position field with a symbolic address of IOCSEQ (IOCS Sequence). The count is not increased before branching from Exit 6, but is automatically increased by one after returning to IOCS by IOCSRE.

E. changing the DTF REWIND specification.

The desired rewind code for the file can be moved into IOCPSV-9. The codes are:

Blank — no rewind at the beginning or end of the tape reel
A — rewind at both the beginning and end of the tape reel
B — rewind at the beginning of the reel, and rewind and unload at the end of the tape reel.

The specification for a particular file can be altered only if the DIOCS RWDOPTION entry contains UNLOAD.

- *Exit 7* permits:
  A. additional checking of a *standard input header.*
     The programmer obtains the information to be checked from IOCSLB.
  B. reading and checking *nonstandard input headers* or *additional labels* following a standard input header.
     The user must provide his own area for entering the header information and must program to read and check the information before returning to the open routine by IOCSRE.

- *Exit 8* allows the programmer to enter his own routine when the reflective spot is reached, and before a tape mark is written, at the end of a reel of *output tape.* At the end of his routine, the programmer must issue either a FEORL macro instruction to write an end-of-reel trailer and process the header on the next reel, or a CLOSE macro instruction to write an end-of-file trailer. He cannot return by IOCSRE.

Nonstandard Labels

For completely automatic operation by IOCS routines, a header or trailer label may be considered *standard* only if it is 80 positions long and has the format described under *Standard Header Label* or *Standard Trailer Label.* Thus, a label is *nonstandard* if it:

1. has a different format.
2. is longer than 80 positions, regardless of whether or not the first 80 positions are standard format.
3. is shorter than 80 positions, regardless of whether or not the fields are in the standard format.
4. is an additional label after the first. That is, any additional labels are considered nonstandard regardless of format. IOCS considers only *one* header and *one* trailer label as standard. If the first label is standard length and format, *standard* can be specified in the DTF entries for automatic reading, writing, and/or checking of the first label. The additional labels are then read, written, and/or checked using the proper exits.

A modification of these rules can be made to gain some of the advantages of automatic standard-label operation for *nonstandard* labels. That is, an 80-position *nonstandard* label can be automatically read into, or written from, the IOCSLB area by specifying the labels as *standard* in the DTF entry (TYPELABEL). However, checking *must* be programmed by the user (via Exit 6 or 7) and *must not* be specified in the DTF entry (CHECKLABEL). Furthermore, the information for labels on an output tape *must* be entered in IOCSLB by use of Exit 1 or 4, and cannot be specified in the DTF entries.

File Opening and Closing

Before the first record can be read from any input file, or written on any output file, that file must be readied for use by the IOCS routines. Similarly, after all records have been processed for a file, that file must be removed from use. For these operations, IOCS makes two basic macro instructions (OPEN and CLOSE) available to the programmer. Three others (FEORL, DCLOS, and RDLIN) are provided for special conditions.

OPEN Macro

This instruction (Figure 22) is used to activate each file that is to be used: card reader, card punch, printer, tape input, and tape output. The symbolic name of the file (assigned in the DTF entry) is entered in the operand field. Two or more files may be opened with one instruction. The file names in the operand must be separated by commas.

For the card reader, card punch, or printer, OPEN simply makes the file available for reading, punching, or writing.

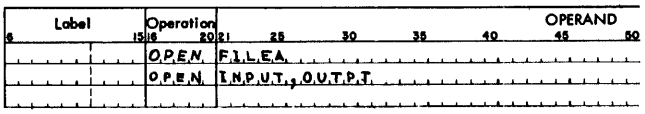


Figure 22. OPEN Macro Instruction

Tape Input File

The OPEN instruction rewinds the tape according to the specification in the DTF entry (REWIND). If the tape does not contain a header label, it is ready for the first GET instruction to read data.

When an input tape contains a *standard header label,* the header is automatically read and checked as specified in the DTF entry (CHECKLABEL). The header fields are checked with the information supplied by the user in the DTF entries (HEADER, SERIALNUM, and REELSEQ) or in RDLIN information cards (see *RDLIN Macro).* Additional checking of the standard header in the IOCSLB area, or reading and checking of additional (nonstandard) labels, may be done using Exit 7. In either case, programming must be returned to the *open* routine by branching to IOCSRE. A tape mark (if any) is passed if specified in the DTF entry (TYPELABEL), and the file is ready for the first GET instruction.

When the tape contains *nonstandard header labels,* the programmer can read and check the information by using Exit 7. At the end of his routine, he must return to the IOCS *open* routine by branching to

IOCSRE. Then a tape mark (if any) is passed if specified in the DTF entry (TYPELABEL) and the file is ready for the first GET instruction.

Tape Output File

The OPEN instruction rewinds the tape according to the specification in the DTF entry (REWIND). If header labels are not used, the tape is ready for data.

A reel of tape used for output generally contains data from a previous job. To make sure that this data is no longer active and may be destroyed, IOCS reads the header label and checks the retention cycle when *standard labels* are specified. If the user does not want automatic checking, however, he can program to check the retention cycle and any other fields of the standard header (in the IOCSLB area), by using Exit 3 and returning to the open routine via IOCSRE. The tape is then backspaced and ready for the new output header. This is written automatically, with the information supplied by the user in the DTF entries (HEADER, SERIALNUM, and REELSEQ) or in RDLIN information cards (see *RDLIN Macro).* The new standard header may be modified by using Exit 4, and additional (nonstandard) header labels may be built and written by using Exit 5. In either case, programming must be returned to the *open* routine by branching to IOCSRE. The tape is then ready for data.

When *nonstandard labels* are specified, the programmer can read and check the old header, backspace the tape, and build and write his output header using Exit 5. After the header is written on tape, programming must be returned to the IOCS *open* routine by branching to IOCSRE. The tape is then ready for data.

CLOSE Macro

This instruction (Figure 23) is used to deactivate each file that has been used: card reader, card punch, printer, tape input, and tape output. The symbolic name of the file (assigned in the DTF entry) is entered in the operand field. Two or more files may be closed with one instruction. The file names in the operand must be separated by commas.

For the card reader or printer, CLOSE simply makes the file unavailable for use. For the card punch, this instruction causes a final dummy card to be fed, so that the last card record can be checked. (Checking occurs on the card feed cycle following punching.) Then the file is made unavailable for use.

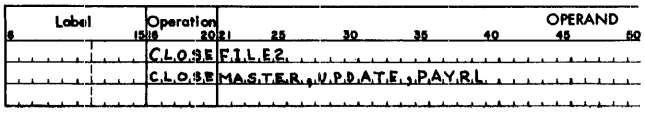


Figure 23. CLOSE Macro Instruction

Tape Input File

The CLOSE macro instruction rewinds, or rewinds and unloads, the input tape according to the DTF specification (REWIND), and it deactivates the file.

In most jobs, a file of input data is *ready* to be closed after all data in the file has been read. When *standard labels* are DTF-specified, IOCS automatically reads and checks the trailer label, and determines that this *is* an end-of-file condition. Then IOCS automatically branches to the end-of-file address specified in the DTF entry (EOFADDR). In this end-of-file routine the programmer issues the CLOSE macro instruction. He may also, in this routine, perform any processing of data that he requires to finish his job. If no processing is required and CLOSE is the only instruction, it must be labelled with the symbolic address specified in the DTF entry (Figure 24).

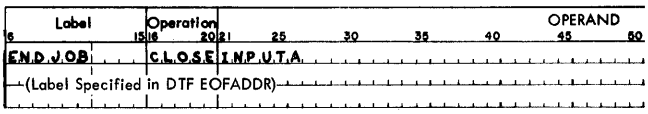


Figure 24. CLOSE Instruction for an Input File

Before closing an input file the programmer may, if he wishes, completely check a *standard trailer* himself, rather than checking automatically by IOCS (DTF CHECKLABEL not specified). In this case, he uses Exit 6 and he must program to branch to his end-of-file address (specified in DTF EOFADDR), where he issues the CLOSE instruction.

When *nonstandard labels* are specified, the trailer label may be read and checked using Exit 6. At the end of his checking routine, if an end-of-file condition exists, the programmer should branch to his end-of-file address (specified in DTF EOFADDR) where he issues the CLOSE instruction. If the user does not care to check his nonstandard trailer label (does not use Exit 6), or if labels are not DTF-specified, IOCS automatically branches to the DTF-specified end-of-file address, where CLOSE is issued.

If, for some reason, the user plans to close an input file before all the data and the tape mark have been read, he writes the CLOSE instruction at the desired place in his program. This merely rewinds, or rewinds and unloads, the tape and deactivates the file. No trailer labels are read or checked.

Tape Output File

When all the data has been processed for an output tape, the CLOSE instruction writes the last block if necessary (partially filled with data) and causes a tape mark to be written. Then, if labels are not specified, it

rewinds, or rewinds and unloads, the tape as specified by the DTF entry (REWIND) and deactivates the file.

When labels are specified:

1. Standard — IOCS writes the *EOF* indication and the totals accumulated during the run. The standard trailer can be modified by using Exit 1, and additional (nonstandard) trailers can be built and written by using Exit 2. In either case, programming must be returned to the *close* routine by branching to IOCSRE.
2. Nonstandard — The programmer can build and write an output trailer using Exit 2. This should include an end-of-file identification for later use on *input*. Programming must be returned to the *close* routine by branching to IOCSRE.

After a trailer label is written, IOCS writes another tape mark, rewinds or rewinds and unloads the tape as specified by the DTF entry (REWIND), and deactivates the file.

In some cases, the data to be written on an output tape may require slightly more tape footage than is available ahead of the reflective spot at the "end" of the tape. Because a few records can be written beyond a reflective spot still leaving enough tape for threading and feeding, the programmer can plan to pass this spot and finish his output data. For this he uses Exit 8 to branch to his routine where he writes the rest of his data and issues his CLOSE instruction.

End-of-Reel for Multi-Reel Files

When all the tape on a reel has been read or written but the job is not complete, additional reels must be processed. At the end of the reel, processing of data must be interrupted to process the trailer label, to change the reel, and to process the header of the new reel. These functions are completely handled by the IBM 1401 IOCS routines when standard labels are specified, and partially handled when nonstandard labels are specified. The end of a reel of tape is signalled by a *tape mark* for input files, and by a *reflective spot* on the tape for output files.

Tape Input File

When *standard labels* are DTF-specified, IOCS reads and checks the trailer label, according to the DTF specifications (CHECKLABEL and TOTALS), and determines that this is *not* an end-of-file condition. Thus, another reel must be processed. The completed tape reel is automatically rewound, or rewound and unloaded, as specified in the DTF entry. If an alternate tape-drive number is specified in the DTF entry (ALTTAPE), the change to the alternate drive is made automatically. If not, a halt occurs to allow the operator time to change the reel on the tape unit. Processing restarts automatically in the first case, or by pressing the start key after the halt. The header on the new reel is read and checked in the same manner as on the preceding reel, and the processing of data resumes.

When *nonstandard labels* are specified, the trailer label can be read and checked using Exit 6. As part of the checking, the programmer determines that this is *not* an end-of-file condition and then returns to IOCS by branching to IOCSRE. Then IOCS performs the same end-of-reel functions as for standard labels: rewinds, or rewinds and unloads, the tape as specified; changes the drive if DTF-specified, or halts to allow the operator to change the tape reels; and permits reading and checking the header of the new reel before processing of data continues.

If labels are not used, an end-of-reel condition cannot be distinguished from an end-of-file. IOCS always assumes an end-of-file condition and branches to the user's EOF address specified in the DTF entry.

Tape Output File

When the reflective spot indicates that the end of a reel of tape has been reached, a tape mark is written after the last block of data for that tape has been written. Then, if labels are not specified, the tape reel is rewound, or rewound and unloaded, according to the DTF specification, but the file is not deactivated. The operator may install another reel of tape and press the start key to continue operation.

When the reflective spot is sensed and *standard labels* are specified:

- the totals accumulated during the run (block, record, and hash) are written,
- the standard trailer can be modified using Exit 1,
- additional (nonstandard) labels can be built and written using Exit 2,
- a tape mark is written following the trailer label,
- the tape is rewound, or rewound and unloaded, as specified.

The EOR indication is automatically included in the trailer. Then, if an alternate tape drive is DTF-specified, the change to the alternate drive is made automatically. If not, a halt occurs to permit mounting another reel on the tape unit. Processing restarts automatically if an alternate drive has been specified, or it is restarted by pressing the start key after the halt. The header on the new reel is written in the same way as on the completed reel, and the processing of data is resumed.

When *nonstandard labels* are specified and the reflective spot is sensed, the programmer can build and

write an output trailer by using Exit 2 and returning to IOCS via IOCSRE. This label should include an end-of-reel identification for later use on *input*. Then IOCS performs the same end-of-reel functions as for standard labels: writes a tape mark; rewinds, or rewinds and unloads, the tape; changes the drive if DTF-specified, or halts to allow the operator to change the tape reel; and permits writing the header on the new reel before the processing of data continues.

As in an end-of-file operation (see *CLOSE Macro, Tape Output File),* the programmer can branch to his own routine to write additional records on the reel when the reflective spot is sensed, by using Exit 8. Because this is an end-of-reel condition rather than an end-of-file, however, he must cause an end-of-reel operation when he has finished writing his data records. For this he writes a FEORL (Forced End-of-Reel) macro instruction in his routine. Then label processing and reel changing are handled according to his specifications, before processing of data resumes.

FEORL Macro

The FEORL (Forced End-of-Reel) instruction is used for *output* tape whenever the user wishes to perform the end-of-reel operations at a time other than when the reflective spot is sensed. For example, it is used in conjunction with a programmer's Exit 8 routine for output tapes. That is, when the programmer plans to use Exit 8 to write records beyond the reflective spot in an end-of-reel condition (not end-of-file), he must *cause* the end-of-reel operations after all his data is written. The FEORL instruction writes the last block of records if necessary (partially filled with data) and writes a tape mark. Then it processes the trailer label, provides for a reel change, and processes the header label, as specified in the DTF entries.

This instruction is used for *input* tape if the programmer wants to discontinue reading data from one reel and switch to another before the tape mark signals the end of data on the first reel. For input tape, FEORL immediately rewinds, or rewinds and unloads, the tape, provides for a reel change, and processes the header label, as specified in the DTF entries.

Whenever blocked *input or output* records are processed, a RELSE macro instruction must be issued immediately ahead of a FEORL instruction. This permits IOCS to correctly process any padded records that may be involved.

The symbolic name of the file, specified in the DTF entry, is entered in the operand field of the FEORL instruction (Figure 25).

Label	Operation	OPERAND
	FEORL	MASTER

Figure 25. FEORL Macro Instruction

RDLIN Macro

Some time after the user's source program and the IOCS routines have been assembled to create the object program, it may be necessary, or advantageous, to change the tape header information from that originally specified in the DTF entries. For example, the date in the header for a recurring job, such as accounts receivable, changes constantly, and the correct date must be specified for checking when the tape is used as input. If the correct date is not specified, an error is signalled and the program halts. A change in the header information for output tapes would be needed whenever one standard program could be used to create tapes for several different jobs in which the type of processing, the blocking factor, etc., were the same, but the data was different. The header would have to vary in each run to properly identify the content of the particular tape records.

With preplanning and the use of the RDLIN (read label information) macro instruction, the information specified for checking or writing *standard header labels* can be altered without reassembling (at object time — when the program is run to process data). While the user is writing the source program, he determines which I/O files require a periodic change in the header information. He includes a RDLIN instruction in his program somewhere ahead of the OPEN instruction for each of these files, and also ahead of any initializing instructions that set word marks in the card read-in area. The symbolic name of the file, specified in the DTF entry, is entered in the operand field of this instruction. Two or more files may be included in one instruction, if the names are separated by commas (Figure 26).

Then, when the object program is run, a RDLIN information card must be available in the card reader at the time the RDLIN macro is executed. If two or more files are named in the RDLIN macro instruction, a separate card must be inserted for each file. These cards must be in the same sequence as the file names in the

Label	Operation	OPERAND
Macro Instruct.	RDLIN	INPUTA, INPUTB, OUTPUTA
Information Cards	RDLIN	2065 3,-025, ACCTS, RECV,6,2,2,2,9,-090,
	RDLIN	1825 9, MNTH, SALES,6,2,2,5 0,-005,
	RDLIN	2065 3,-030, ACCTS, RECV, -090,

Figure 26. RDLIN Macro Instruction and Information Cards

instruction. Each RDLIN information card contains the current information required to properly check or write the header label at this time. The RDLIN macro causes the card to be read and the information to be entered in the IOCS area reserved for header specifications. If a RDLIN information card is not included, a halt occurs. After inserting the card, processing can be resumed by pressing the start key.

The RDLIN information cards must contain the entire header information (except Header identification and Tape Serial Number), not just the fields that are changed. These cards (Figure 26) are punched in *Autocoder* format, with RDLIN in the operation field and the header information in the operand field in the same sequence as the header label:

COLUMNS	HEADER INFORMATION
16-20	RDLIN
21-25	File Serial No.
26-30	Reel Sequence No. (Col. 30 is always blank)
31-40	File Identification Name
41-45	Creation Date
46-50	Retention Cycle (Col. 50 is always blank)

DCLOS Macro

When an input tape is read, any block of records that contains a parity error can be "dumped" onto another tape for later investigation, if specified in the DIOCS entry (READERROR). This tape is activated by the *TAPE,n* specification in the READERROR entry, but it must be deactivated by using the DCLOS (Dump-Close) macro instruction (Figure 27). This instruction causes a tape mark to be written after the last record. The tape is rewound if REWIND is specified in the operand field of this instruction. It is rewound and unloaded if REWIND, UNLOAD is specified.

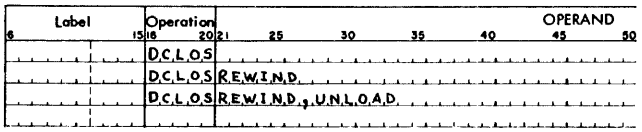


Figure 27. DCLOS Macro Instruction

DIOCS Entry

The DIOCS Entry (Descriptive IOCS) specifies generally the characteristics of all the files to be processed and the machine configuration on which the program will be run. During assembly, the DIOCS Entry cards must follow the JOB and CTL cards and precede the user's source program.

The entry (Figure 28) consists of a DIOCS header card followed by a series of detail cards. The header card contains only DIOCS, in the operation field; the label and operand fields are unpunched.

The succeeding detail cards may follow in any order and must be unpunched in the operation field. Their label and operand fields are punched with the specifications for the program. When multiple operands are required for a particular label, they may appear in any order and must be separated by commas. Any detail card may have comments in the operand field, provided two blanks separate the comments from the last operand. When a particular detail entry does not apply to the job, the card may be omitted, or it may be included with the operand field unpunched.

Altdrive

If an alternate tape drive is to be used for any multi-reel tape file, as specified in a DTF ALTTAPE entry, this card is included. The word *YES* is entered in the operand.

LABEL	OPERATION	POSSIBLE OPERANDS*
	DIOCS †	
ALTDRIVE		YES
COUNTS		HASH
		RECORD
DIOCSORG		Any actual or symbolic address (Max. of 10 Characters #)
EXITS		Any Exit Numbers 1-8 (1 Digit per Exit)
FEATURES		RELEASE
		STORAGE
		OVERLAP
IODEVICES †		READER
		PUNCH
		TAPE
		PRINTER
LABELDEF		STANDARD or NONSTANDARD or MIXED
		CHECK or IDENT
		TM
		RDLIN
READERROR		CLEAN
		TAPE,n (n=1 Tape Drive Number)
		PROCESS or SCAN
RWDOPTION		UNLOAD or NORWD
TAPEUSE		INPUT or OUTPUT

† Must be included. Other cards are included when applicable.

* Where two or more lines of operands are shown for a label, one item from each line may be entered.

Maximum size of the label is 6 positions. The other 4 characters allow for an indication of address adjustment.

Figure 28. Specifications for DIOCS Entry Cards

Counts

If a total is specified in the DTF TOTALS entry for writing or checking the trailer label of any tape file, this card is included. The operand field contains:

HASH, if any hash total is taken.

RECORD, if records are counted for any tape reel.

DIOCSORG

This card indicates the location in core storage at which the programmer wants the *Autocoder* processor to begin generating the IOCS subroutines. Any actual or symbolic address that is valid in an *Autocoder* ORG statement may be entered in the operand field. If this card is omitted, the IOCS subroutines start at storage location 333.

Exits

If any exit from the IOCS routines is used for any header or trailer label operation, this card is included. The number (1-8) of every DTF EXIT specification (EX1ADDR-----EX8ADDR) used in the program must be entered in the operand field.

Features

This card indicates any special features affecting input/output operations that are utilized by the program. The operand field contains:

RELEASE, if either the Read Release or Punch Release feature is used. This cannot be included with OVERLAP.

STORAGE, if the Print Storage feature is used.

OVERLAP, if the Processing Overlap feature is used. This cannot be included with RELEASE. Whenever this feature is used, all input and output operations must be handled by IOCS.

IODevices

This card must be included, and it specifies all the types of input and output files that are handled by IOCS. It summarizes the DTF FILETYPE entries. The operand field contains one or more of these:

READER, if any input data is in cards.

PUNCH, if any output data is punched in cards.

TAPE, if any input or output records are on magnetic tape.

PRINTER, if any printer operation occurs.

If READER and/or PUNCH is specified, the I/O check-stop switch on the console should be turned OFF at object time in order to utilize the IOCS error-checking procedures. If OVERLAP is specified in the DIOCS FEATURES entry, *all* input and output operations must be handled by IOCS.

Labeldef

Whenever one or more tape files with header and/or trailer labels are processed, this card summarizes the specifications in all the DTF TYPELABEL and CHECKLABEL entries. It also indicates any tape mark after a header, and the use of the RDLIN macro instruction.

For type of label, *one* of these three types is entered in the operand:

STANDARD, when *all* tape files have IBM 1401 standard header and trailer labels.

NONSTANDARD, when *all* tape files have nonstandard header and trailer labels.

MIXED, when some tape files have standard labels and some have nonstandard labels or are not labelled.

If any standard labels are checked, *one* of these two specifications is entered in the operand:

CHECK, whenever standard labels in *one or more* tape files are to be completely checked.

IDENT, when standard labels in one or more tape files are to be partially checked (name in header and all of trailer) and *none* of the files are to be completely checked. That is, CHECK takes precedence; if it is entered in the operand, IDENT is omitted.

If any input tape contains a tape mark between the header label and the first data record and the programmer wants IOCS to pass it automatically, *TM* must be entered in the operand.

When the RDLIN macro instruction is used to enter label information in any standard header label, *RDLIN* must be entered in the operand.

Readerror

If a parity error is detected when an input tape is read, the tape is automatically backspaced and reread ten times, before the block is considered an error block. After this, an error block is automatically passed without processing and the next block is read in and processing continues, if the READERROR entry is not included. By including this READERROR entry, the programmer can specify other procedures to be followed for an error block. He can enter these specifications in the operand:

CLEAN – to request the tape-cleaning procedure. That is, when an error block is detected the tape is backspaced six times and read five times, and then another attempt is made to read the block that was in error.

TAPE,n – to transfer an error block to another tape for later investigation. The letter "n" represents the number of the tape drive (1-6) used for this purpose.

PROCESS – to process, rather than bypass, an error block. This can be used with TAPE,n and/or with CLEAN, but not with SCAN.

SCAN – to halt the program and allow the operator to investigate the error. The error-stop switch on the tape-adapter unit must be OFF for this operation. When processing stops, the operator must follow this procedure:

1. Set the tape-select switch to D (Diagnostic).
2. Turn OFF the check-stop switch on the auxiliary console.
3. Press the start key to reread the error block. This stores the characters as they are written on tape, without internally correcting any parity error. After the block is stored, the program halts again, and STORAGE ADDRESS displays the address of the next sequential instruction.
4. If a parity error was detected on the reread (step 3), the stop, process, and tape lights on the console are ON. Perform a storage-scan operation to locate the character that contains the parity error, and correct if possible. If a parity error was not detected, only the stop light is ON.
5. Reset the tape-select switch to N.
6. Turn the check-stop switch ON.
7. Press CHECK RESET to turn off the process light, and press START RESET to turn off the tape light, if a parity error occurred on the reread (step 4).
8. Turn sense switch G on to process a corrected block or off to bypass an incorrect one, and press START.

This operand can be used with TAPE,n and/or with CLEAN, but not with PROCESS.

Rwdoption

If no specifications are given by the programmer, tape files are automatically rewound, but not unloaded, on an OPEN or CLOSE instruction and on an end-of-reel condition. This RWDOPTION entry must be included if other operations are required for any tape file, as specified in the DTF REWIND entries. Either of these specifications, but not both, is entered in the operand:

UNLOAD, whenever *one or more* tape files are to be rewound on OPEN and rewound and unloaded on CLOSE or on an end-of-reel condition.

NORWD, if one or more tape files are not to be rewound at all, and none are to be unloaded. That is, UNLOAD takes precedence; if it is entered in the operand, NORWD is omitted.

Tapeuse

This card summarizes the types of tape files specified in the DTF FILETYPE entries. One, but not both, of these operands is entered:

INPUT, if *all* tape files are used for input.

OUTPUT, if *all* tape files are used for output.

If both input and output tape files are used in the program, or if records are processed in the overlap mode, this card is omitted.

DTF Entry

The DTF Entry (Define The File) describes the file, indicates methods of processing the file, and specifies symbolic addresses of routines and areas unique to the file. A DTF Entry is included for *each* input or output file that is to be processed in the program: one to six tape files (each identified by tape drive number), and one file each for the card reader, card punch, and printer. During assembly, the DTF Entry cards must follow the DIOCS Entry cards and precede the user's source program.

Like the DIOCS, a DTF Entry (Figure 29) consists of a header card followed by a series of detail cards. The header card contains DTF in the operation field and the symbolic name of the file in the operand field. The label field is unpunched. This symbolic file name is entered in the IOCS macro instructions referring to this file.

The succeeding detail cards may follow in any order and must be unpunched in the operation field. Their label and operand fields are punched with the particular specifications for the file. When multiple operands are required for a label, they may appear in any order, unless specified otherwise, and they must be separated by commas. Any detail card may have comments in the operand field, provided two blanks separate the comments from the last operand. When a particular detail entry does not apply to the file, the card may be omitted, or it may be included with the operand field unpunched. All symbolic addresses entered as operands may be character-adjusted. Also, all symbolic addresses, except input/output area labels, may be indexed. An operand consisting of a symbolic address with character adjustment and/or indexing may not exceed ten characters in length.

Alttape

This card specifies the number (1-6) of a tape drive unit used as an *alternate* for a tape file that has two or more reels of data. The *original* drive unit is specified in DTF CHANDRIVE.

Blocksize

This card must be included for any tape file with blocked records. The size of the input, or output, storage area allotted for the block is entered in the operand field. This does *not* include either the group-mark with word-mark position or the extra position for record-length checking when that feature is specified by the DTF WLRADDR entry.

If two input or output areas are allotted when tape records are processed in the *overlap mode,* the size of *one area* is specified in this entry.

Cardpoc

When all the cards in a file are to be selected into the same pocket, the number of that pocket (1, 2, 4, or 8) is specified in the operand of this entry, if it is *not* specified in the GET or PUT macro instructions. If the read release special feature is used, any selection of card input files *must* be specified in this entry, not in the GET instruction. Cards from an input file *cannot* be selected when records are processed in the overlap mode.

Chandrive

The number (1-6) of the tape drive unit used for a tape input or output file is specified in the operand of this entry. In a multi-reel tape-file operation using two tape drives, this entry specifies the *original* drive unit, while the ALTTAPE entry specifies the *alternate* drive unit.

Checklabel

With this entry, the programmer specifies the checking procedure he desires for IBM standard header and trailer labels in a tape input file. He enters *one* of these two specifications in the operand:

ALL, to completely check the standard header and trailer labels.

IDENT, to partially check the standard header, and completely check the standard trailer. Only the ten-position file identification name is checked in the header.

EOFAddr

This entry must be included for any card or tape input file. The user specifies, in the operand, the symbolic address for his end-of-file routine. On a last-card indication or tape end-of-file condition, but not on an end-of-reel, IOCS automatically branches to this routine, where the programmer generally issues the CLOSE instruction for the input file.

EX1Addr through EX8Addr

These eight cards are used in conjunction with tape input/output files. They provide the means of branching from the IOCS routines to a user's routine for processing header or trailer labels. The symbolic address of the user's routine is entered in the operand of the EXIT card selected. Each exit (1-8) occurs at a specific time in the IOCS routines and is used for a specific function, as indicated briefly in Figure 29 and described under *Label Operation.* The programmer includes the appropriate EXIT cards according to the operations he wishes to perform.

Figure 29. Specifications for DTF Entry Cards (Part 1 of 2)

LABEL	OPERATION	POSSIBLE OPERANDS	MAX. CHAR*	APPLIES TO					MUST BE INCLUDED	REMARKS
				INPUT TAPE	OUTPUT TAPE	READER	PUNCH	PRINTER		
	DTF‡	Any File Name	6	x	x	x	x	x	Each file	
ALTTAPE		Any Tape Drive Number 1-6	1	x	x				Alternate tape drive	
BLOCKSIZE		Size of one Input or Output Storage Area	4	x	x				Blocked records	Omit group-mark with word-mark position. Area defined by DA statement.
CARDPOC		Any Card Pocket Number 1, 2, 4, or 8	1			x	x		Selection of card input file with Read Release Special Feature	Omit if selection specified in GET or PUT. With overlap, card input may not be selected.
CHANDRIVE		Any Tape Drive Number 1-6	1	x	x				Tape records	
CHECKLABEL		ALL or IDENT	—	x					Check IBM standard labels	Omit for nonstandard labels.
EOFADDR		Label of user's end-of-file routine	10	x		x			Input file	Routine entered on end-of-file only, not on end-of-reel.
EX1ADDR		Label of user's routine	10		x				Modify standard trailer	
EX2ADDR			10		x				Build and write additional/nonstandard trailers	
EX3ADDR			10		x				User checks old standard header	Eliminates IOCS-checking of retention cycle.
EX4ADDR			10		x				Modify standard header	
EX5ADDR			10		x				Build and write additional/nonstandard headers	Also read and check old nonstandard header.
EX6ADDR			10	x					Additional check of standard trailer Read and check additional/nonstandard trailers	May also use to modify reel-sequence count, or to change DTF REWIND specification.
EX7ADDR			10	x					Additional check of standard header Read and check additional/nonstandard headers	
EX8ADDR			10		x				User's routine before tape mark is written	Occurs at reflective spot on tape.
FILETYPE‡		READER or PUNCH or TAPE,INPUT or TAPE,OUTPUT or PRINTER or TAPE,INPUT,CHECKPOINT	—	x	x	x	x	x	Each file	
FORMCNTL		Any 1401 Space or Skip d-character	1					x		Omit if specified in PUT or SPACE/SKIP.
HEADER		File Identification Name	10	x	x				Check or write standard label	Operands must be entered in sequence shown.
		Creation Date	5							Omit Creation Date for output tape.
		Retention Cycle	3							Include Creation Date and Retention Cycle for input tape, only if ALL specified in DTF CHECKLABEL.
INDEXREG		X1 or X2 or X3	2	x	x				Index Register assigned to an input or output area.	Omit if DTF WORKAREA is included, or if work areas specified in GET or PUT.
IOAREAS		Label of input or output area	10	x	x				Tape records	Same as label of DA statement.
MODEPAR		LOAD or MOVE	—	x	x				LOAD mode	May be omitted for MOVE mode.
OVERFLOW		9 or 12 or 9,Label of user's routine or 12,Label of user's routine	10					x	Carriage overflow	If operand 9 or 12 is used, carriage restores to channel 1 and processing continues.

LABEL	OPERATION	POSSIBLE OPERANDS	MAX. CHAR*	APPLIES TO					MUST BE INCLUDED	REMARKS
				INPUT TAPE	OUTPUT TAPE	READER	PUNCH	PRINTER		
PADDING		Any character except * ¢ ≢ ≠ √‾ =	1		x				Pad with character other than blank	Applies only to blocked fixed-length records.
RECFORM		UNBLOCKED,FIXED or UNBLOCKED,VARIABLE or BLOCKED,FIXED or BLOCKED,VARIABLE	—	x	x				Blocked and/or variable-length records	
REELSEQ		Any 3-digit number	3	x	x				First-reel number other than 001	Applies only to IBM standard labels. Omit if first-reel number is 001.
REWIND		UNLOAD or NORWD	—	x	x				Unload at CLOSE or end of reel Prevent rewinding	Omit if rewind at OPEN,CLOSE, and end of reel.
SERIALNUM		Any 5-digit number	5	x	x				Automatic check on input File No. differs from Tape No. on output	Applies to File Serial Number only in IBM standard labels.
SIZEREC		Length of record (fixed-length) or Low-order position of record-length field (variable-length)	3	x	x				Blocked records Unblocked fixed-length records	
TOTALS		RECORD; Low-order position of hash-total field	3	x	x				Record and/or hash totals	Applies only to labelled tape files. Block count is taken automatically.
TYPELABEL		STANDARD or NONSTANDARD; TM	—	x	x				Labelled tape files	TM — to pass tape mark after header label.
VARBUILD		X2 or X3 or Label of 3-position field	10		x				Build blocked variable-length records in output area	
WLRADDR		Label of user's routine	10	x					Check tape-record length	Omit for unblocked variable-length records.
WORKAREA		Label of work area	10	x	x	x	x	x		Same as label of DA statement. Omit if DTF INDEXREG or VARBUILD included, or if work areas specified in GET or PUT. Include for card files with release special feature, or if processing in the overlap mode, when only one work area is used.

‡ Must be included. Other cards are included when applicable.

* Maximum number of characters allowed by IOCS for the operand. The 10-position operands allow for a maximum-size label of 6 positions and an indication of address adjustment and/or indexing.

Figure 29. Specifications for DTF Entry Cards (Part 2 of 2)

Filetype

This card must be included, and it states the type of file described in this set of DTF entries. The operand must be punched with one of these:

READER — for a card input file.

PUNCH — for a card output file.

TAPE,INPUT — for input tape records. If this file contains *checkpoint* records that should be bypassed by IOCS, the operand CHECKPOINT must also be included. The entry would then read TAPE,INPUT,CHECKPOINT. With this operand, the checkpoint records are not included in a hash total or counted in block or record counts. Checkpoint records are assumed by IOCS to consist of two records, and they are recognized by IOCS only if the first record is identified by **CHKPT in the first seven positions. Both records are then bypassed.

TAPE,OUTPUT — for output tape records.

PRINTER — for printed reports.

Formcntl

Spacing or skipping of forms is specified in this card if it is to be the same for every line written on a form and if it is *not* included in either the PUT or SPACE macro instructions. The standard IBM 1401 d-character for the desired control is entered in the operand.

Header

This card must be included for:

- an input tape file with standard labels when checking is DTF-specified.
- an output tape file with standard labels.

It specifies the information for checking three fields in an input header, or for writing two fields in an output header:

1. *File Identification Name:* Ten alphamerical character (see *Standard Header Label,* item 5). This is included for input and output files.
2. *Creation Date:* Five digits — two for year, followed by three for day. This is included only for an input file that is to be completely checked. It is not included for an output file, because the date is taken from today's date in storage.
3. *Retention Cycle:* Three digits. This is included as the third operand for an input file that is to be completely checked, or as the second operand for an output file.

When two (output tape) or three (input tape) of these operands apply, they *must* be written in the sequence listed and separated by a comma.

Indexreg

When an index register is assigned to the input, or output, area of a tape file, the number of the register is specified by X1, X2, or X3 in the operand of this entry. This card must be included for:

- blocked fixed-length records processed in the input area.
- blocked fixed-length records built in the output area.
- unblocked tape records processed in *two* input areas, in the overlap mode.
- unblocked tape records built in *two* output areas, in the overlap mode.

The index register specified in this entry must also be specified in the DA statement for the corresponding input or output area, for records processed in the non-overlap mode. For records processed in the overlap mode, however, the index-register specification must be omitted from the DA statement.

Whenever this entry is included for a file, the DTF WORKAREA entry must be omitted and work areas must not be specified in the GET or PUT macro instructions.

IOAreas

The symbolic address of the input, or output, area for a tape file is specified in this entry. This must be the same label as specified in the DA statement for the area. The address may not be indexed in this entry. Any indexing required is specified in the DTF INDEXREG entry and in the DA statement (non-overlap mode).

When tape records are processed in the *overlap mode,* one or two input, or output, areas may be assigned. If two input, or output, areas are assigned, both symbolic addresses are specified and separated by a comma in this entry.

Modepar

This card must be included if tape records are to be read or written in the load mode. *LOAD* is entered in the operand. If this card is omitted, IOCS automatically reads or writes in the move mode, but this card may be included with *MOVE* in the operand, if desired.

Overflow

This card defines the operation to be performed if an overflow condition occurs in the printer carriage. The number of the carriage-tape channel used to indicate the overflow is entered in the operand. This may be either channel 9 or channel 12. When an overflow occurs, processing of data is interrupted, the carriage is restored to channel 1, and processing resumes.

If he prefers, the programmer may branch to his own routine on an overflow. For this he enters the symbolic address of his routine as the second operand in this card. Then he must return to IOCS at the end of his routine by branching to IOCQUT.

Padding

If blocked fixed-length records in an *output* tape are to be padded with some character other than blanks, this card is included. It can specify any one character except: asterisk (*), cent sign (¢), group mark (≡), record mark (⊤), tape mark (√‾), or word separator (=).

Recform

This card specifies the type of records in a tape input or output file:

BLOCKED *must* be indicated if records are blocked.

UNBLOCKED may be included if desired, but it is not required. IOCS assumes unblocked records if BLOCKED is not specified.

VARIABLE *must* be indicated if records are variable-length.

FIXED may be included if desired, but it is not required. IOCS assumes fixed-length records if VARIABLE is not specified.

Thus, this card *must* be included if tape records are blocked and/or variable-length.

Reelseq

When a multi-reel tape file with standard labels is processed, the *Reel Sequence Number* field in the header label of the first reel is automatically numbered *001* by IOCS, if no other number is specified. If the programmer wants a different number in the first reel, he includes this REELSEQ entry and specifies a three-digit number. This is used to check the header in the first reel on input, or to write the header on output.

After the number of the first reel is determined (001 or some other number), it is automatically increased by one for each succeeding reel, or it may be increased or decreased by some other factor by using Exit 6.

Rewind

If no specifications are given by the programmer, tape files are automatically rewound, but not unloaded, on an OPEN or CLOSE instruction and on an end-of-reel condition. If other operations are desired for a tape input or output file, this card may specify:

UNLOAD, to rewind the tape on OPEN, and to rewind and unload on CLOSE or an end-of-reel condition.

NORWD, to prevent rewinding the tape at any time.

Serialnum

This card specifies the five-digit *File Serial Number* for a standard header label and must be included for:

- an input tape file with standard labels when checking is DTF-specified.
- an output tape file with standard labels if the *File Serial Number* is to differ from the *Tape Serial Number*. If this card is omitted, the pre-recorded tape number is automatically repeated in the *File Serial Number* field. With a multi-reel file, the tape number of the first reel becomes the file number in all reels.

Sizerec

This card must be included for any tape input or output file with blocked records, or with unblocked fixed-length records.

For fixed-length records (blocked or unblocked), this card specifies the number of characters in the data record, including the record mark, if any. (This does *not* include the extra position allotted in core storage for record-length checking, when that feature is specified by the DTF WLRADDR entry.)

For variable-length blocked records, this card specifies the low-order position of the record-length field. For example, if the record-length field is located in positions 13, 14, and 15 of the record, "15" is entered in the operand of this card. The record-length field is included in each record, and it specifies the number of characters in the record, including itself and the record mark. It must be defined by a word mark in the storage area referred to by IOCS. This is the work area if it is specified in the DTF entries; otherwise, it is the input (or output) area.

Totals

This card specifies the totals (record or hash) required. To accumulate a record count, *RECORD* is entered in the operand. To accumulate a hash total, the low-order position of the chosen field, within the record, is specified. For example, if the field is located in record positions 16-21, "21" is entered in the operand. This location cannot be a symbolic reference. This field must be defined by a word mark in the storage area referred to by IOCS. This is the work area if it is specified in the DTF entries; otherwise, it is the input (or output) area.

These totals may be specified for tape files with standard or nonstandard labels, but not for unlabelled files.

Typelabel

Whenever an input or output tape file contains header and trailer labels, this card must be included to specify

the type of label:

STANDARD, whenever standard header and trailer labels are used. The standard label may be followed by additional (nonstandard) labels.

NONSTANDARD, when nonstandard labels are used.

On input tape, this entry is also used to indicate that a tape mark (if any) between a header label and the first data record should be passed by IOCS. For this, *TM* is entered in the operand.

Varbuild

If blocked variable-length records are to be built directly in the output area, this card must be included. It performs functions for variable-length output records similar to those performed by DTF INDEXREG for fixed-length output records.

This card must specify a three-position field, which is used by IOCS for two functions in building each variable-length record: (1) checking the length of the record to determine if it fits in the current block, and (2) making available the address of the high-order position of the record. (See the description of VARBUILD under *PUT Macro, Blocked Variable-Length Records.)* The field specified in this entry may be index register 2 (X2), index register 3 (X3), or the symbolic address of any other 3-position area defined by the programmer.

WLRAddr

This entry causes IOCS to check the length of records read from tape. If a wrong-length record (unblocked record or block of records) is read, programming branches to the user's routine specified by the symbolic address in the operand.

At the end of his routine, the user must return to IOCS by issuing another GET macro for the same file.

This card may be included for input tape files that contain blocked records (fixed- or variable-length) or unblocked fixed-length records. Unblocked variable-length records cannot be checked for correct record-length.

Whenever this entry is included for a tape file, one extra core storage position must be provided in the input area for that file. This is specified by the DA statement(s).

Workarea

This card is included for an *input* tape file if records are to be processed in one work area, or for an *output* tape file if records are to be built in one work area. It specifies the symbolic address of the work area. This must be the same as the label of the DA statement for the work area.

The same work area may be specified in the DTF entries for an input file and in the DTF entries for a corresponding output file.

This card is also included for an *input card* file if the read release or processing overlap special feature is used and records are processed in one work area. *Output card* files always require the use of a work area. If all output card records are to be built in the same work area, this DTF entry should be used. If the records are to be built in two or more work areas, each work area must be specified in the corresponding PUT instruction.

Whenever this entry is included for a file, the DTF INDEXREG and VARBUILD entries must be omitted, and work areas must not be specified in the GET or PUT macro instructions.

This section summarizes the major points that the programmer should consider when he is:

- allotting core-storage areas for IOCS operation.
- writing macro instructions.
- planning to use the processing overlap special feature.

Core-Storage Areas

The factors pertaining to input/output and work areas are itemized in Figure 30. The programmer must specify one or more input, or output, storage areas for each tape file used in the program, but he does not specify input/output areas for card or printer operations since these are fixed areas in the 1401 or 1460. A work area may be specified for IOCS operation whenever the DTF INDEXREG entry is not specified. A work area is required:

1. to process blocked variable-length input records.
2. for card files whenever the read release or punch release special feature is used.
3. for card files whenever the processing overlap special feature is used.

Macro Instructions

The macro instructions are summarized in Figure 31, which shows how each instruction may be written, its major function and operation, and the type of input/output file to which it applies.

Processing Overlap Special Feature

When this feature is installed in the IBM 1401 or 1460 and OVERLAP is specified in the DIOCS FEATURES entry, these factors should be considered:

1. For greatest efficiency, two input or output areas should be allotted for each tape file. This requires:
   a. two DA statements
   b. two operands in the DTF IOAREAS entry.
2. To process tape records in the input/output areas with a two-area operation, an index register is required for either blocked or unblocked records. If an index register is not available, records must be processed in a work area.
3. Whenever an index register is used in conjunction with a tape input/output area(s), it always contains the address of the high-order position of the record to be processed. The programmer must:
   a. Specify the index register in the DTF INDEXREG entry.
   b. Omit reference to the index register in the DA statement(s).
   c. Omit field labelling in the DA statement(s).
   d. Include indexing in the program instructions or in equate statements, to process fields within a record.
   e. Use a GET FILEA,0+Xn instruction to make an input record available for processing in the output area.
   f. Use a PUT 0+Xn,FILEX instruction to make a record that has been processed in the input area directly available for output.
   g. Issue a PUT ,FILEX instruction before building the *first* record in a run in an output area, to initialize the index register.
4. If blocked records are to be released, the release instruction must be written in the form RELSE FILEA,OVERLAP.
5. Whenever card files are processed:
   a. A work area is required for processing.
   b. Cards in an input file cannot be selected. All cards in the card reader stack in the NR pocket.
   c. The read release or punch release special feature cannot be used.
6. For printer operations, an additional SPACE/SKIP macro instruction is available for forms control.

STORAGE FACTORS	TAPE INPUT OR OUTPUT AREA	WORK AREA
Size of Area for:		
Unblocked fixed-length records	One record	One record
Unblocked variable-length records	Largest record	Largest record
Blocked fixed-length records	One block	One record
Blocked variable-length records	Largest block	Largest record
DA Statement	Required	Required
Group-Mark with Word-Mark	Required	Required
Field labels and word marks	If process in I/O area	If process in work area
Record Marks	Blocked records	Blocked records
Word-Mark — Record-Length Field	If DTF WORKAREA not specified	If DTF WORKAREA specified
Word-Mark — Hash-Total Field		

Figure 30. Summary of Factors Related to Core Storage Assignment

Figure 31. Summary of Macro Instructions (Part 1 of 2)

MACRO INSTRUCTION	HOW WRITTEN	MAJOR FUNCTION	USE FOR	OPERATION	REMARKS
GET	GET FILEA*	Make individual input record available for processing	Any file	Make record available in input area or DTF-specified work area	If tape records processed in the input area, specify DTF INDEXREG for: (1) blocked records, (2) unblocked records with 2-area operation
	GET FILEA,WORKA*		Any file	Make record available in work area specified in GET	Do not specify index register or work area in DTF entry
	GET FILEA,OUTPTB		Tape file	Use output area as work area in non-overlap mode	With blocked output records, specify DTF INDEXREG for output area
	GET FILEA,0+Xn		Tape file	Use output area as work area with blocked records in overlap mode	0 + Xn = zero plus DTF-specified index register assigned to output area
	GET FILEA, ,n		Card file	Transfer input card record and select card	n = pocket number for stacker selection. Use only in non-overlap mode.
	GET FILEA,WORKA,n		Card file		
PUT	PUT ,FILEA*	"File" individual output record after processing completed	Any file	Make record in output area or DTF-specified work area available for output	If tape records built in the output area, specify DTF INDEXREG for: (1) fixed-length blocked records, (2) unblocked records with 2-area operation If blocked variable-length records built in output area, specify DTF VARBUILD
	PUT WORKA,FILEA*		Any file	Move record from work area specified in PUT	Do not specify index register, VARBUILD, or work area in DTF entry
	PUT INPUTA,FILEB		Tape file	Use input area as work area in non-overlap mode	With blocked input records, specify DTF INDEXREG for input area
	PUT 0+Xn,FILEB		Tape file	Use input area as work area with blocked records in overlap mode	0 + Xn = zero plus DTF-specified index register assigned to input area
	PUT ,FILEA,n		Card file	Transfer output card record and select card	n = pocket number for stacker selection
	PUT WORKA,FILEA,n		Card file		
	PUT ,FILEA,d		Printer file	Transfer record and space or skip	d = 1401 d-character for spacing or skipping
	PUT ,FILEA,di,da		Printer file		di = 1401 d-character for *immediate* space or skip da = 1401 d-character for *after print* space or skip
OPEN	OPEN FILEA,FILEB,FILEX	Activate file	Any file	Tape file: per DTF specs — rewind tape and process header	
CLOSE	CLOSE FILEA,FILEB,FILEX	Deactivate file	Any file	Tape input file: rewind tape per DTF specs. Tape output file: write last block and tape marks; process trailer and rewind tape per DTF specs	

MACRO INSTRUCTION	HOW WRITTEN	MAJOR FUNCTION	USE FOR	OPERATION	REMARKS
RELSE	RELSE FILEA*	Skip over remaining records in a block without processing	Tape file with blocked records	Input file: record count and hash total not taken Output file: fixed-length records padded and included in record count and hash total	
	RELSE FILEA,OVERLAP				
SPACE/SKIP	SPACE d	Space or skip printer form	Printer file	Available only with overlap special feature	d = any 1401 d-character for spacing or skipping
	SKIP d				
FEORL	FEORL FILEA	Force end-of-reel operations	Tape file	Input file: per DTF specs — rewind tape, provide for reel change, and process new header Output file: write last block and tape marks; process trailer, provide for reel change, and process header per DTF specs	Output file: generally used in conjunction with Exit 8
RDLIN	RDLIN FILEA,FILEB	Change standard-header information without reassembling	Tape file	Transfer factors from RDLIN information cards to IOCS area for header specs	RDLIN information cards: *Autocoder* format with factors in same order as standard header label; cards fed in same order as file names in operand
DCLOS	DCLOS*	Deactivate file	"Dump" tape file		Use for tape file specified in DIOCS READERROR for parity-error records
	DCLOS REWIND			Rewind the tape	
	DCLOS REWIND,UNLOAD			Rewind and unload the tape	

* Basic form of this macro instruction.

Figure 31. Summary of Macro Instructions (Part 2 of 2)

The tape IOCS program consists of a set of library routines in *Autocoder* format. These library routines contain model statements that are selected and tailored at assembly time to produce the IOCS routines applicable to the particular job being programmed. The selection and tailoring are based on the IOCS descriptive entries (DIOCS and DTF) and macro instructions.

The IOCS library routines are inserted in the *Autocoder* system tape by a librarian run. A listing of these routines is also obtained by a librarian run. A deck of cards containing the IOCS library routines is supplied in a format that allows direct insertion in the *Autocoder* system. The deck is distinguished by identification in columns 1-5 and 76-80:

COLUMNS	IDENTIFICATION
1-5	Program Instruction Sequence Number. Numbered consecutively starting with 0000b (the HEADR statement) for each library routine. The INSER statement is blank in column 1-5.
76-77	IOCS Program Identification Number (65).
78-79	Library Routine Identification Number.
80	IOCS Version Number (Overpunched with an "X" for subsequent changes.)

The tape IOCS system must be used in conjunction with IBM 1401 *Autocoder* program, and therefore requires the same machine configuration as *Autocoder*. Sense switches are also required if tape scanning will be performed.

IOCS Library Routines

The identification numbers, names, and use of the IOCS library routines are:

IDENT NO. (COL. 78-79)	NAME (COL. 6-10)	USE
05	CLOSE	Linkage to CLOSE routine
10	DCLOS	Close Dump tape
15	FEORL	Linkage to forced end-of-reel routine
20	GETXX	Linkage to DTF input routine (referred to as GET in the user's routine)
25	OPENX	Linkage to OPEN routine (referred to as OPEN in the user's routine)
30	PUTXX	Linkage to DTF output routine (referred to as PUT in the user's routine)
35	RDLIN	Linkage to RDLIN routine
40	RELSE	Linkage to DTF routine for block release
45	SPACE	Forms control
50	STACK	Stacker selection
55	15000	Closed subroutine used to bypass checkpoint records
60	18000	Closed subroutine used to convert 5-character addresses to 3-character addresses
65	33333	DTF routines
70	44444	To dump IOCS subroutines
75	55555	OPEN, CLOSE, FEORL, RDLIN, End-of-Reel, read/write and error routines
80	66666	To dump IOCS literals and establish value of IOCEND

The user may not identify any library routines of his own with names whose first three characters are the same as those used in any of these IOCS routines.

The library routines with a numerical character in the high-order position of the name (15000-66666) may be used only by IOCS. These routines are not available to the user.

Source Program

A symbolic source program that uses IOCS must be organized in this specific manner:

1. The first two statements (as in any *Autocoder* program) must be a JOB statement and a CTL statement, in that order.
2. The next statements must be the DIOCS header and detail entries.
3. Following the DIOCS entry must be the DTF header and detail entries for each file being specified for IOCS operation.
4. The user's source program follows the last DTF entry.

After the *Autocoder* processor has encountered the DIOCS and DTF entries, the generation of IOCS routines begins. The processor generates all the routines, areas, constants, and literals required by IOCS (with the exception of the input/output and work area DA's) before assembling the user's program. The last statement generated by IOCS is an *origin* statement with the label IOCEND.

If the user has no overlays in his program, he may write his source program without an origin statement. The instructions and constants will be assigned starting at IOCEND. If the user has overlays, however, each overlay may be given the origin IOCEND. This prevents overlaying the IOCS routines, which must be present in core storage for proper operation.

Object Program

Organization

The organization of an assembled object program that includes the generated IOCS routines is:

1. The entry to the OPEN routine
2. The entry to the FEORL routine
3. The entry to the CLOSE routine
4. The End-of-Reel routine
5. The read/write routine
6. The routines pertaining to each DTF in the same order as the DTF's. Each DTF routine is preceded by a number of constants containing information about that file. These constants are available to the OPEN, CLOSE, FEORL, End-of-Reel and DTF routines.
7. The user's program.

If a program utilizes a tape output file with standard labels, *Today's Date* must be loaded into positions 195-199 in storage. To accomplish this, a card may be punched (as specified here) and inserted in the condensed deck following the IOCS portion of the object program:

COLUMNS	PUNCH
1-5	XX|XXX Yr|Day
40-46	L005199
47-53	N000000
54-60	N000000
61-67	N000000
68-71	1040

IOCS Labels

IOCS generates two kinds of labels in the model statements that form the subroutines in the assembled object program.

1. Internal labels as defined by *Autocoder* for use in model statements. Each of these is a six-character alphamerical label starting with a lozenge (□).
2. Labels that communicate between different IOCS subroutines, and between IOCS subroutines and the user's program. Each of these is a six-character label starting with the letters IOC. The labels to which the user may have access are:

IOCXR1, IOCXR2, IOCXR3 — The designation of index registers. The first seven statements generated by IOCS initialize the three index registers with word marks in positions 087, 092, and 097, and zeros in 087-100. In addition, the names IOCXR1, IOCXR2, and IOCXR3 are assigned to index registers 1, 2, and 3.

IOCSRE — The address to which the user should branch when returning to IOCS from his program, after using Exits 1-7.

IOCRDX — The address of a closed subroutine that causes a card to be read if a *start-read-feed* command has been given by IOCS (DIOCS FEATURES entry specifies RELEASE). The user should branch to IOCRDX if his own processing time between a *GET card* and any other *GET* or *PUT* may cause over-extension of the release time. (In computing allowable process time, allow 2 milliseconds for IOCS processing.) These routines may not be used with process overlap.

IOCPNX — The address of a closed subroutine that causes a card to be punched if a *start-punch-feed* command has been given by IOCS (DIOCS FEATURES entry specifies RELEASE). The user should branch to IOCPNX if his own processing time between a *GET card* and any other *GET* or *PUT* may cause over-extension of the release time. (In computing allowable process time, allow 2 milliseconds for IOCS processing.) These routines may not be used with process overlap.

IOCSEQ — The address of a three-character field in which the reel sequence number is stored for the particular file concerned when IOCS branches to the user's routine from Exit 6. The reel sequence number is not increased before branching. If the user desires to modify this number, he must remember that upon return to the IOCS routines, IOCS increases the number by 1.

IOCPSV-9 — The address at which the rewind option is stored for the particular file concerned when IOCS branches to the user's routine from Exit 6. The rewind codes are: blank — no rewind at the beginning or end of the reel; *A* — always rewind; *B* — rewind at the beginning of the reel, and rewind and unload at the end of the reel. The rewind option can be changed only if the DIOCS RWDOPTION entry contains UNLOAD.

IOCSLB The high-order address of an 80-character area that contains the standard header or trailer label of the particular file concerned when IOCS branches to the user's routine from any exit. IOCS generates this area only if the DIOCS LABELDEF entry specifies STANDARD or MIXED.

IOCQUT The address to which the user should branch after he has completed the processing in his routine specified by the DTF OVERFLOW entry.

IOCTDY The address in which *Today's Date* is stored (location 195-199) when creating standard header labels for tape output files.

IOCSRW The address of a closed subroutine that reads and writes tape records and performs error-checking operations (DIOCS READERROR entry). To utilize this routine, the user must employ the following linkage:

```
                B           IOCSRW
                NOP         (any label)
                .           .
                .           .
                .           .
(any label)     DCW         @K@
                DCW         @29@ or @49@
                DCW         (End-of-File address)
                normal tape instruction
```

Use @29@ for tape output, and @49@ for tape input. This routine is generated for either read-only mode or write-only mode, or both, depending on the DIOCS TAPEUSE specification. This routine may not be used when operating in the overlap mode.

IOCHSH The address of a field of ten characters in which the hash total is stored for the particular file concerned when IOCS branches to the user's routine from Exit 2 or 6. The hash total is accumulated only if specified in the DIOCS COUNTS and DTF TOTALS entries.

IOCRCT The address of a field of ten characters in which the record count is stored for the particular file concerned when IOCS branches to the user's routine from Exit 2 or 6. The record count is accumulated only if specified in the DIOCS COUNTS and DTF TOTALS entries.

IOCBLK-1 The address of a field of five characters in which the block count is stored for the particular file concerned when IOCS branches to the user's routine from Exit 2 or 6.

Error Indications

IOCS has been designed so that most errors in the user's specifications are detected at assembly time and are indicated on the *Autocoder* listing. These include errors such as:

- Specification errors within the DIOCS or DTF entries. For example, specification of the release feature without using either a reader or punch file.
- A substitution-type parameter greater in length than the maximum allowed. When the *Autocoder* processor picks up parameters from the DIOCS and DTF entries, each substitution-type parameter is assumed to have a maximum length. If one is longer than the maximum, a shift within a table used by *Autocoder* occurs and causes other parameters to be lost. For instance, the INDEXREG parameter cannot be greater than two characters (Xn). The maximum lengths are indicated in the *Specifications for DIOCS Entry Cards* (Figure 28) and *Specifications for DTF Entry Cards* (Figure 29).
- A misspelling of a non-substitution-type parameter (for example, TAPE spelled TPAE). If the spelling is incorrect in the first three positions, the parameter is not picked up.

On the *Autocoder* listing, the specific error is printed as a *comments card* and the legend *MACRO ERROR* appears to the right of the *CARD* column. This error is counted in the total errors. The comment describes the specification violation in detail. In most cases, the presence of any one of the error messages requires reassembly because the instructions and constants that have been generated are incorrect.

Errors that may cause incorrect operation when the object program is run cannot be detected, and are the responsibility of the user. This type includes such errors as missing word marks, record marks, etc.

Halts Identified by B-Address Register		
B-Address	Reason	Procedure
3001	RDLIN information card missing	Run cards out, place proper cards in the card reader, and press Start.
3010	Ten consecutive erases while attempting to write on tape.	Press Start to continue the attempt.
3027	End-of-reel condition when no alternate tape drive has been specified	Mount new reel on the same tape drive and press Start.
3030	Thirty parity-error detections while writing on a tape.	Press Start to continue.
3050	Fifty parity-error detections while reading from a tape.	Press Start to continue.
3111	Card read error — error card is last card in normal-read (NR) pocket (The I/O check-stop switch must be off.)	Run cards out, replace the read hopper (error card first), and press Start.
3134	First halt (parity error) when DIOCS READERROR specifies SCAN (The error-stop switch on the tape-adapter unit must be off.)	Set the Tape-select switch to "D", Turn off the check-stop switch on the 1401 auxiliary console, and Press start to reread the error block for scanning. (Second halt is 3234.)
3222	Printer error on last line printed	Press Start to continue.
3234	Second halt when DIOCS READERROR specifies SCAN	Scan for parity error and correct if possible, Reset the tape-select switch to "N", Turn the check-stop switch on, Press check reset and start reset, and Turn sense switch G on to process a corrected block, or off to bypass an incorrect block. Press Start.
3345	RDLIN macro used for unlabelled file	Press Start to continue processing.
3444	Ten punch errors detected while attampting to punch a card. (The I/O check-stop switch must be off.)	Press Start to continue.
3555	Program branched to IOCSRE after using Exit 8	Correct the program and restart.
3666	Ten consecutive erases while attempting to write on a dump tape	Press Start to continue the attempt.
3777	Ten parity-error detections while attempting to read an old header label from a tape output reel (when there are no tape input files)	Press Start to continue.
3800	Standard input trailer does not check	Processing cannot be restarted from this halt. (See halt located 13 positions past label ¤ 3L001, under I-address below.)
3900	Tape-error condition detected at the end of a reel of output tape (reflective spot)	Press Start to continue. If no alternate tape drive has been specified, halt 3027 will occur after the tape record is written correctly.
3999	Neither a work area nor VARBUILD specified for a blocked variable-length tape output file	Correct the program and reassemble.
Halts Identified by I-Address Register		
I-Address	Reason	Procedure
13 positions past label ¤ 3J001	Standard input header does not check	Press Start to process this reel anyway. If you mount a different reel, press Start Reset and Start to check the header of this reel.
13 positions past label ¤ 3L001	Standard input trailer does not check	Press Start to continue processing. If Start Reset and Start are pressed, a second halt occurs with 3800 in the B-address register and processing cannot be restarted.
40 positions past label ¤ 3R001	Retention cycle in the old header of a tape output file does not check	Press Start to use this reel anyway. If you mount a different reel, press Start Reset and Start to check the old header of this reel.
At label IOCERC	Attempt made to close a tape file that has not been opened (overlap mode).	Press Start to continue processing.

Figure 32. IOCS Generated Halts

Reassembly

If a program using IOCS must be reassembled, the normal rules described in the *Autocoder,* Version 3, documentation should be followed. The generated IOCS instructions and constants will be regenerated only if sense switches B and G are ON.

If the DIOCS and DTF specifications were incorrect, they should be altered and the IOCS routines should be regenerated. However, if the IOCS section is correct but the user's program is incorrect, substantial time is saved, on reassembly, by not regenerating the IOCS routines.

IOCS Generated Halts

Several halts are generated by IOCS. Each halt can be identified by displaying either the B-address register or the I-address register, as indicated in Figure 32. This figure lists the reason for each halt and a procedure to be followed if it occurs.

Index

IBM 1401/1460 Printed in U.S.A. C24-1462-3

IBM®

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

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]