SPERRY RAND

UNIVAC

9200/9200 II
9300/9300 II
SYSTEMS

UNIVAC
8410 DISC
IOCS

PROGRAMMERS
REFERENCE

UP-7639 Rev. 1

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware or software changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

UNIVAC is a registered trademark of Sperry Rand Corporation.

PRINTED IN U.S.A.

CONTENTS

TABLES

1. INTRODUCTION

1.1. GENERAL

This manual describes the Input/Output Control System (IOCS) provided for the UNIVAC 8410 Direct Access Subsystem used with the UNIVAC 9200/9200 II/9300/9300 II Systems.

A knowledge of the *UNIVAC 9200/9200 II/9300/9300 II Systems Minimum Operating System Programmers Reference, UP-7547* (current version) and the *UNIVAC 9200/9200 II/9300/9300 II Systems Tape/Disc Assembler Programmers Reference, UP-7508* (current version) is helpful in using this manual.

The UNIVAC 8410 Disc IOCS includes the following:

- The Sequential IOCS – used for sequential processing of a disc file.
- The Direct Access IOCS – used for random processing of records in a file as performed by direct addressing of records in a file.
- The Indexed Sequential IOCS – used for either sequential processing or non-sequential processing of a file based on a key field in a record.

Each of the preceding sections describes the declarative and imperative macro instructions required for the input and output operations.

A macro instruction is similar in form to a source code instruction; it can require a label, but it must have an operation code and an operand field containing one or more parameters.

The parameters used with the declarative macro instructions describe all aspects of the file to be processed, whereas the parameters used with the imperative macro instructions point to the file described by a declarative macro instruction, and sometimes add additional details specifying processing action to be taken.

1.1.1. Declarative Macro Instructions

A problem program must inform the system of the parameters, special conditions, current status, and options pertaining to a file. This is accomplished by including a declarative (file definition) macro instruction for each file required by the problem program. As implied by the term declarative, these macro instructions generate nonexecutable code, such as constants and storage areas for variables. Therefore, these macro instructions must be separated from the inline file processing coding. The declarative macro instruction and the selected keyword parameters in the operand define the file. The first three characters of the operation code must be DTF, meaning Define The File. The last two characters usually indicate the type of device or method of accessing. A keyword parameter consists of a word or code immediately followed by an equals sign (=) which is, in turn, followed by one specification.

The format of the declarative macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
filename	DTFcc	keyword–1=x,keyword–2=y,...,keyword–n=z

The symbolic name of the file must appear in the label field. It has a maximum of eight characters and must begin with an alphabetic character. The appropriate DTF designation must appear in the operation field. The keyword parameters are written in any order in the operand field and must be separated by commas. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation statements.

NOTE: Any program, even one which cannot be executed on a card system, can be assembled using the card assembler. However, if UNIVAC 8410 Disc Subsystem IOCS programs are assembled, the card assembler recognizes only the first four characters of the symbolic name of the file.

1.1.2. Imperative Macro Instructions

A problem program must be able to communicate with the IOCS in order to accomplish the processing of files that have been defined by declarative macro instructions. This is accomplished by including imperative (file processing) macro instructions in the problem program which, in turn, communicate with the modules of the IOCS coding. The imperative macro instructions are expanded as inline executable code. Not all macro instructions are available for use on all devices. Some are specifically input type macro instructions and cannot be used for a device that is exclusively used for output; the opposite is true, also.

The format of the imperative macro instructions is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	xxxx	yyyy,...,zzzz

A symbolic name can appear in the label field. It can have a maximum of eight characters and must begin with an alphabetic character. The appropriate verb or code must appear in the operation field. The positional parameters (as signified by the name) must be written in the specified order in the operand field and separated by commas. When a positional parameter is omitted, the comma must be retained to indicate the omission except in the case of omitted trailing parameters. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation statements.

NOTE: Any program, even one which cannot be executed on a card system, can be assembled using the card assembler. However, if UNIVAC 8410 Disc Subsystem IOCS programs are assembled, the card assembler recognizes only the first four characters of the symbolic name of the file.

1.2. STATEMENT CONVENTIONS

The conventions used to illustrate statements in the manual are:

- Capital letters and punctuation marks (except braces, brackets, and ellipses) are information that must be coded exactly as shown.
- Lowercase letters and terms represent information that must be supplied by the programmer.
- Information contained within braces represents necessary entries one of which must be chosen.
- Information contained within brackets represents optional entries that (depending on program requirements) are included or omitted. Braces within brackets signify that one of the entries must be chosen if that operand is included.
- An ellipsis set indicates the presence of a variable number of entries.
- Commas are required after each parameter, except after the last parameter specified. When a positional parameter is omitted from within a series of parameters, the comma must be retained to indicate the omission.

2. SEQUENTIAL IOCS

2.1. GENERAL

The Sequential Access Method (SAM) of addressing files allows the user to create, retrieve, process, and maintain sequential files which reside on the UNIVAC 8410 Direct Access Subsystem (DAS). SAM also provides the interface between the user problem programs and the 8410 disc dispatcher. The 8410 disc dispatcher provides a software interface between SAM and/or user-developed software and the UNIVAC 8410 disc hardware.

This section describes the minimum hardware and software configurations of SAM, the input/output record size and format, the user interfacing procedures, and the declarative and imperative macro instructions.

2.2. MINIMUM HARDWARE AND SOFTWARE CONFIGURATION

The minimum hardware configuration required for SAM is a central processor unit with a main storage capacity of at least 12K, a card input device, a printer, and one 8410 disc drive.

The minimum software configuration required for SAM is the Minimum Operating System (MOS).

2.3. INPUT/OUTPUT RECORD SIZE AND FORMAT

The physical record size processed by SAM is specified by the user in the DTFSA declarative macro instruction (see 2.5). However, the user is limited to a maximum record length of 80 bytes for blocked records and 160 bytes for unblocked records; SAM will process single or multivolume files containing fixed-length blocked or unblocked record formats.

2.4. USER INTERFACING PROCEDURES

The procedures required to interface the user problem program and SAM are as follows:

(1) Advise the IOCS as to which phases of the I/O control routine are required. This is accomplished by specifying either the DSDS1 and DSDS2, or the DSDIO declarative macro instruction (see 2.5.1.).

(2) Provide the IOCS with the information required to construct a file description table defining the file. This is accomplished by specifying the DTFSA declarative macro instruction (see 2.5.2). The user must specify a DTFSA macro instruction for each file accessed.

(3) Initiate specific sequential processing by specifying various imperative macro instructions (see 2.6).

2.4.1. Operator Interface

The operator must react to specific error halt/displays which signify that SAM has encountered either a hardware or software error. If an error is not directly related to a specific disc unit, the physical unit number of the first volume of the file is displayed.

Table 2-1 lists the error halt/displays and the corresponding replies.

HEXADECIMAL DISPLAY	SIGNIFICANCE	REPLY
23x1	Expiration date of output file is less than creation date, or volume number specified in DTFis not one less than specified in VTOC.	Key a zero into location 4 to repeat check. Key a one into location 4 to cancel job. Press START.
23x2	OPEN/CLOSE requested for a previously OPENED/CLOSED file.	Press START to cancel job.
23x3	File label cannot be read.	
23x4	Specified label not found in VTOC directory.	Press START to search VTOC again. Key a nonzero into location 4 to cancel job. Press START.
23x5	Invalid function or catastrophic error during I/O.	Press START to cancel job.
23x6	UOB or UAL I/O error and keyword ERRO is not specified.	
23x7	GET/PUT requested for unopened file.	
23x8	Input file label check failed.	Press START to ignore error. Key a one into location 4 to re-check. Press START. Key a nonzero into location 4 to cancel job. Press START.
23x9	PUT request after EOF detected.	Press START to cancel job.
23xA	All extents are full. The current output request cannot be completed.	
23xB	PUT request for input file.	
23xC	Get request for output file.	
23xD	TTSS>9999	
23xE	File type is specified incorrectly.	
23xF	Attempt was made to execute two consecutive PUTs to an update file without an intervening GET.	
23x0	End of volume detected.	Press start to continue or replace volume and press start.

NOTE: x indicates physical unit number.
TTSS=Track Track Sector Sector.

Table 2-1. Error Halt Displays and Replies

2.5. DECLARATIVE MACRO INSTRUCTIONS

In order to use SAM, the programmer must specify macro instructions to describe the I/O routine and define the file characteristics. The DSDS1, DSDS2, and DSDIO macro instructions describe the I/O routine, and the DTFSA macro instruction defines the direct access storage device file.

The symbolic name of the file has a maximum length of eight characters and must begin with an alphabetic character. The keyword parameters may appear in any order, and they must be separated by commas.

2.5.1. Generating the I/O Routine

Because of the size limitations of a 12K card loaded system, two macros (DSDS1 and DSDS2) are available to indicate which phases of the SAM I/O file control routine are required to process the program. For either a 16K card, tape, or disc loaded system, the DSDIO macro instruction generates the complete I/O routine.

The formats of the DSDS1, DSDS2, and DSDIO declarative macro instructions follow. The formats list, in alphabetical order, the optional keyword parameters which may appear in the operand of the macro instructions in any order. A description of the individual parameters follows each illustration. A summary of the keyword parameters is given in Table 2-2.

The format of the DSDS1 macro instruction, used for 12K card loaded systems, and the DSDIO macro instruction, used for 16K card, tape, or disc loaded systems, is identical in form:

LABEL	OPERATION	OPERAND
[name]	{ DSDS1 } { DSDIO }	[INPT=NO] [,OUPT=NO] [,UPDT=NO] [,WORK=NO]

- INPT Keyword Parameter

  If input capability is not required, the following keyword parameter must be specified:

  INPT= NO

  If this keyword parameter is omitted, the coding necessary to process input files is generated.

- OUPT Keyword Parameter

  If output capability is not required, the following keyword parameter must be specified:

  OUPT = NO

If this keyword parameter is omitted, the coding necessary to process output files is generated.

- UPDT Keyword Parameter

  If updating capability is not required, the following keyword parameter must be specified:

  UPDT =NO

  If the UPDT keyword parameter is not specified, updating capability is provided.

  If an update operation is to be performed using a 12K card loaded system, updating capability must be provided for both the DSDS1 and DSDS2 macro instructions; otherwise, both macro instructions should contain the parameter UPDT = NO.

  NOTE: Input capability must be provided if updating capability is provided.

- WORK Keyword Parameter

  If a work file is not to be used by the problem program, the following keyword parameter must be specified:

  WORK =NO

  If this keyword parameter is omitted, work files are utilized.

The format of the DSDS2 macro instruction, used with a 12K card loaded system, is as follows:

LABEL	ↂ OPERATION ↂ	OPERAND
[name]	DSDS2	[UPDT=NO]

- UPDT Keyword Parameter

  If updating capability is not required, the following keyword parameter must be specified:

  UPDT = NO

  If the UPDT keyword parameter is not specified, updating capability is provided.

  If an update operation is to be performed using a 12K card loaded system, updating capability must be provided for both DSDS1 and DSDS2 macro instructions; otherwise, both macro instructions should contain the parameter UPDT = NO.

  NOTE: Input capability must be provided if updating capability is provided.

  The preassembly macro pass for a 12K card loaded system requires two input decks; one deck is associated with the DSDS1 macro, and the other is associated with the DSDS2 macro. Two output decks are produced; these decks are combined to form one I/O program deck by removing the END card from the DSDS1 deck, and placing the deck in front of the DSDS2 deck. The resultant deck is then assembled as one program.

KEYWORD	SPECIFICATION	12K CARD DSDS1	12K CARD DSDS2	16K CARD, TAPE, DISC DSDIO	REMARKS
INPT	NO	X		X	Specifies that input files will not be processed.
OUPT	NO	X		X	Specifies that output files will not be processed.
UPDT	NO	X	X	X	Specifies that input files will not be updated.
WORK	NO	X		X	Specifies that a work file will not be used by the problem program.

Table 2-2. Summary of DSDS1, DSDS2, and DSDIO Macro Instruction Keyword Parameters

2.5.2. Defining Direct Access Storage Device Files

The DTFSA declarative macro instruction is required to define all files which are to be processed by SAM.

Following is the format of the DTFSA macro instruction which shows, in alphabetical order, the required and optional keyword parameters which may appear in the operand of the DTFSA macro instruction. A description of the individual parameters follows the format. A summary of the keyword parameters is given in Table 2-3 following the descriptions.

The format of the DTFSA macro instruction is:

LABEL	OPERATION	OPERAND
filename	DTFSA	CRDT = label, DEVA = nn, EOFA = label, FLID = label, GENO = label, IOA1 = label, VOLN = label, XPDT = label [,BLCK = YES] [,CHCK = NO] [,EOVA = label] [,ERRO = {IGNORE / label / SKIP}] [,IOA2 = label] [,RCSZ = n] [,TYPE = {OUTPUT / WORK}] [,UPDT = YES]

- File Identification

The following five keyword parameters are required to identify the file area.

- CRDT = label

  This required keyword parameter allows the programmer to insert a creation date (YYMMDD), where label is the address of a six-byte area in the problem program containing the creation date.

- FLID = label

  This required keyword parameter allows the programmer to insert file identification data, where label is the address of an eight-byte area in the problem program containing user file identification.

- GENO = label

  This required keyword parameter allows the programmer to insert a generation number, where label is the address of a four-byte area in the problem program containing the generation number.

- VOLN = label

  This required keyword parameter allows the programmer to insert volume numbers, where label is the address of a two-byte storage area in the problem program containing the volume number. The numbering of volumes starts with zero. Thus, if a file consists of three volumes, and the user wishes to process only the third volume, the number specified in the storage area of the problem program is 2. If the user wishes to process all three volumes, the number specified in the storage area of the problem program is 0.

- XPDT = label

  This required keyword parameter allows the programmer to insert an expiration date, where label is the address of a six-byte area in the problem program containing the expiration date.

■ Device Address

The logical unit number of the device on which volume 1 of the file resides is specified by the following required keyword parameter:

DEVA = nn

where nn is a decimal number denoting the logical unit number (00 to 63) of the device.

■ Type of File

A file is considered an input file by SAM unless one of the following optional keyword parameters, specified by the programmer, indicates otherwise.

- TYPE = OUTPUT

  This keyword parameter specifies an output file (one that is to be written).

- TYPE = WORK

  This keyword parameter specifies a work or scratch file.

  A work file must be opened using an OPEN macro. When the file is opened the IOCS performs a limited label check. This label check for a work file differs from that of an input or output file in that only the filename and expiration date are checked to ensure that the file has been defined for this disc and that it may be written on.

  After the work file is opened, it can then be treated as either an input file or an output file.

■ Input/Output Buffer Area

An I/O buffer area must be assigned to each file in a problem program. The programmer defines the location of the buffer area by specifying the following required keyword parameter:

IOA1 = label

where label is the address of the most significant byte of a 165-byte area.

- Update Buffer Area

  When updating a file, a buffer area must be defined to contain a record to be updated. The programmer defines the location of the buffer area by specifying the following keyword parameter:

  IOA2 = label

  where label is the address of the most significant byte of a 165-byte area. This parameter is required for updating only.

- End of Input File

  The following keyword parameter is required for all input files and specifies the address to which control is transferred. Upon detecting an end-of-file condition, the IOCS transfers program control to a location in the problem program which is specified in the following parameter:

  EOFA = label

  where label is the address of the first byte of the instruction to which program control is transferred.

- End of Input Volume

  The IOCS notifies the problem program when an end-of-volume condition is detected if the following optional keyword parameter is specified:

  EOVA = label

  where label is the address in the problem program to which the IOCS transfers program control. The return address to the IOCS is contained in register 14.

- Error Conditions

  If an I/O request is unsuccessful, one of the following error types is sent by the 8410 disc dispatcher to SAM:

  - Nonoperative Channel
  - Invalid Function
  - Catastrophic Failure
  - Unrecoverable Output Bus Check
  - Unrecoverable Abnormal Line

  If a Nonoperative Channel error occurs, the Sequential File Processor cancels the job. If any of the other errors occur, an error display and halt results. If START is then pressed, the Sequential File Processor cancels the job.

  If the IOCS detects an Unrecoverable Output Bus Check (UOB) or an Unrecoverable Abnormal Line (UAL), an error display and program halt occur. The program is then terminated when START is pressed. If other action is to be taken, the programmer must specify one of the following optional keyword parameters:

- ERRO=IGNORE

  This keyword parameter indicates that the IOCS ignores either the UAL or UOB error condition and continues program execution.

- ERRO=label

  This keyword parameter indicates that the IOCS transfers control to the address specified by label, where label is the address of the first byte of the error routine in the problem program.

- ERRO=SKIP

  This keyword parameter (for input operation only) indicates that the IOCS skips the disc sector in which the error occurred and continues processing.

■ Blocked Records

The processing of blocked records is indicated by specifying the following optional keyword parameter:

BLCK=YES

The blocking factor (number of records in a sector or block) is computed automatically by the sequential IOCS. The largest possible number of records is placed in each sector. In order to have blocked records, the record size must be 80 bytes or less. If the record size is greater than 80, and blocking is requested, the calculated blocking factor will be 1 and no blocking will take place.

■ Write Check

A write check is performed on all records transferred to disc unless the following optional keyword parameter is specified:

CHCK=NO

This parameter is used by the programmer when saving execution time and storage space is a factor.

■ Record Size

The maximum record size is 160 bytes. If a record size of less than 160 bytes is to be used, the programmer must specify the following optional keyword parameter:

RCSZ=n

where n is the number of bytes expressed as a decimal number.

When writing unblocked records, a record size of more than one byte but less than 160 bytes results in the transfer of the record. However, the unused portion of the sector is filled with zeros. If the record size is one byte, the contents of the remainder of the sector is unpredictable.

■ Update File

To update a file, the programmer must specify the following optional keyword parameter:

UPDT=YES

This specification allows the program to read a record, perform the necessary processing, and then write the record into the disc area from which it was originally read.

Work files cannot be updated.

Table 2–3 summarizes the DTFSA macro instruction keyword parameters.

KEYWORD	SPECIFICATION	FILES		REMARKS
		INPUT	OUTPUT	
BLCK	YES	X	X	Indicates blocked record processing.
CHCK	NO		X	Indicates no write check is to be performed.
CRDT	label	R	R	Inserts a creation date.
DEVA	nn = logical unit number	R	R	Specifies logical unit number of device on which file resides.
EOFA	label	R		Identifies EOF routine address.
EOVA	label	X	X	Identifies EOV routine address.
ERRO	IGNORE	X	X	Indicates ignored error and continued execution.
	label	X	X	Indicates address of user's error routine.
	SKIP	X		Indicates skipping the sector containing the error.
FLID	label	R	R	Inserts file identification data.
GENO	label	R	R	Inserts generation number.
IOA1	label	R	R	Indicates the address of the input/output buffer area.
IOA2	label	R		Indicating the address of the updating buffer area, this parameter is required when updating only.
RCSZ	n = number of bytes in record	X	X	Indicates record size less than 160 bytes.
TYPE	OUTPUT		X	Indicates an output file.
	WORK	X	X	Indicates a work or scratch file.
UPDT	YES	X		Indicates that records are to be read, updated, and written back to their original locations.
VOLN	label	R	R	Inserts volume numbers.
XPDT	label	R	R	Inserts an expiration date.

LEGEND:
R = Required
X = Optional

Table 2-3. Summary of DTFSA Macro Instruction Keyword Parameters

Example:

```
LABEL      OPERATION      OPERAND                                  72  80
           DTFSA          CRDT=DATE,                               X
                          DEVA=ADDR,                               X
                          EOFA=END,                                X
                          FLID=FILE,                               X
                          GENO=GNUM,                               X
                          IOA1=BEGIN,                              X
                          VOLN=VNUM,                               X
                          XPDT=XDATE,                              X
                          ERRO=CHECK,                              X
                          IOA2=BUFF,                               X
                          UPDT=YES,                                X
                          RCSZ=120
```

2.5.2.1. DTFSA Error Messages

During the generation of the DTFSA macro instruction, tests are made to ensure the validity of the instruction parameters. If an error is found, an appropriate error message is produced. This message produces an I flag from the assembler, and is accompanied by a 4-byte zerofill (assembler produces 4 bytes of zeros). In order to facilitate the user's modifying the DTFSA expansion without having to reassemble (REP cards), each error message is followed by an ORG*-4, thus, allowing the extra 4-byte pad to be overlaid by the proper specification. DTFSA error messages are illustrated in Table 2-4.

MESSAGE	SIGNIFICANCE
CHCK-INVALID-IGNORED	CHCK specified for input file.
EOFA-NOT-SPECIFIED	EOFA = 0 or when EOFA is not specified for INPT.
EOFA-INVALID-IGNORED	EOFA ≠ 0 or when specified for an output file.
RCSZ-TRUNCATED-TO-MAX-160	RCSZ > 160
IOA1-NOT-SPECIFIED	IOA1 = 0
IOA2-NOT-SPECIFIED	IOA2 = 0
LABEL-INFO-INCORRECT	CRDT/XPDT/FLID/GENO not specified.
UPDT-INVALID-IGNORED	File type not input.

Table 2-4. DTFSA Error Messages

2.6. IMPERATIVE MACRO INSTRUCTIONS

The imperative macro instructions supplied for SAM are OPEN, GET, PUT, and CLOSE. These macro instructions are used by the problem program to open, process, and close files. The filename of an imperative macro instruction may have a maximum of eight characters.

2.6.1. OPEN Macro Instruction

This instruction must be used to initialize the file before any other imperative macro instruction can be performed. It is also used for label checking.

The format of the OPEN macro instruction is:

LABEL	b OPERATION b	OPERAND
[name]	OPEN	filename

- Positional Parameter 1

  filename - is the label of the corresponding DTFSA declarative macro instruction in the program defining the file to be opened.

Operational Conditions:

An OPEN macro issued to a file previously opened results in an error display and termination of the job.

For an input operation, the execution of the OPEN macro results in a comparison between the file ID in the VTOC (Volume Table of Contents) and the file ID specified in the DTFSA macro. If the values are not equal, a program halt occurs. The operator can then either load a different disc and press START, or cancel the job by keying a binary nonzero value into location 4 by means of the ALTER switch and pressing START.

After the correct file ID is established, the IOCS compares the creation date, generation number, volume number, and expiration date contained in the VTOC against the corresponding information specified in the DTFSA macro defining the file. If any part of the comparison is unsuccessful, an error display results and a program halt occurs. The operator can then either load a different disc into the handler and initiate the search again by pressing START, inform the IOCS to ignore the error by keying a binary 1 into location 4 by means of the ALTER switch and pressing START, or cancel the job by keying a binary nonzero value other than 1 into location 4 by means of the ALTER switch and pressing START.

If the error is ignored, IOCS replaces the creation date, generation number, volume number, and expiration date defined by the user DTF, with the corresponding information contained in the VTOC.

For an output operation, the IOCS compares the volume number and expiration date in the label to the corresponding information in the DTFSA macro defining the file. The expiration date must be less than or equal to the new creation date to allow the current file information to be overlaid. If the result of either comparison is incorrect, an error display results and a program halt occurs. The operator can then either load a different disc or cancel the job.

If the results of the comparisons are correct, the IOCS writes a label containing the creation date, generation number, expiration date, and volume number as specified in the DTFSA macro defining the file.

For output work files, the IOCS checks the expiration date in the label against the creation date specified in the DTFSA macro defining the file.

The creation date must be greater than or equal to the expiration date. If the results of the comparison are incorrect, the operator can either load a different disc or cancel the job.

Example:

```
LABEL    ↹ OPERATION ↹        OPERAND                    ↹
1        10              16
         OPEN    INPUT
         OPEN    OUTPUT
```

2.6.2. GET Macro Instruction

This instruction retrieves the next logical record in sequence from a disc file during sequential retrieval of records.

The format of the GET macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	GET	filename, workarea

- Positional Parameter 1

  filename – is the label of the DTFSA declarative macro instruction defining the file from which the record is to be read.

- Positional Parameter 2

  workarea – is the label of the first byte of a work area defined by the user.

Operational Conditions:

When an end-of-volume condition is detected by the IOCS, automatic unit swapping occurs unless the programmer has indicated control is to be returned to the problem program as specified in the EOVA parameter.

Example:

LABEL	OPERATION	OPERAND
	GET	INPUT,INPWA

2.6.3. PUT Macro Instruction

If updating is specified, this instruction indicates that a record retrieved by the GET macro instruction has been updated and is to be written on the direct access storage device. If updating is not specified, the PUT macro instruction indicates that a new record is to be written on the direct access storage device. Updating, in a sequential retrieval operation, takes place only through the execution of the PUT macro instruction.

The format of the PUT macro instruction is:

LABEL	OPERATION	OPERAND
[name]	PUT	filename,workarea

- Positional Parameter 1

  filename – is the label of the DTFSA declarative macro instruction defining the file to which the record is to be written.

- Positional Parameter 2

  workarea – is the label of the first byte of a work area defined by the user.

Operational Conditions:

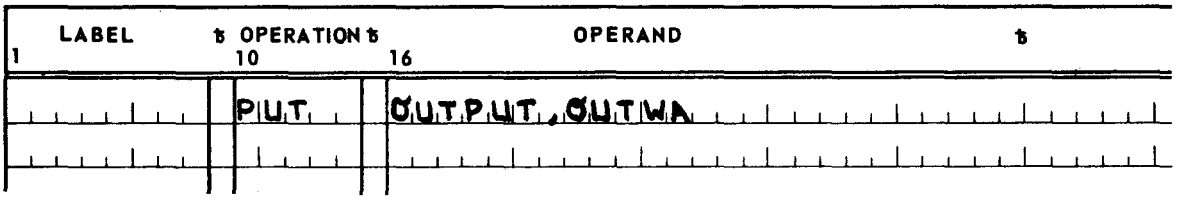
An error condition, resulting from the unsuccessful execution of a PUT macro, is processed after the next PUT or CLOSE is issued. The error condition is handled by a routine in the problem program located at the address specified in the ERRO parameter. Subsequent PUT macros are processed until all sectors in the file have been filled. (The file boundaries on disc are defined in the VTOC.)

A PUT macro following the transfer of data into the last sector of a file does not transfer data and results in an error display and termination of the job.

A PUT macro issued after an end-of-volume condition transfers control to the end-of-volume routine in the problem program, if requested, or the IOCS executes a unit swap and continues processing the records.

To update a record, the program must issue a PUT macro before issuing a second GET macro.

Example:



2.6.4. CLOSE Macro Instruction

This instruction must be used to terminate the processing of the file. Once the file is closed, no other macro instruction may be executed for the file until it is reopened by the OPEN macro instruction.

The format of the CLOSE macro instruction is:

LABEL	OPERATION	OPERAND
[name]	CLOSE	filename

- Positional Parameter 1

  filename – is the label of the corresponding DTFSA declarative macro instruction in the program defining the file to be closed.

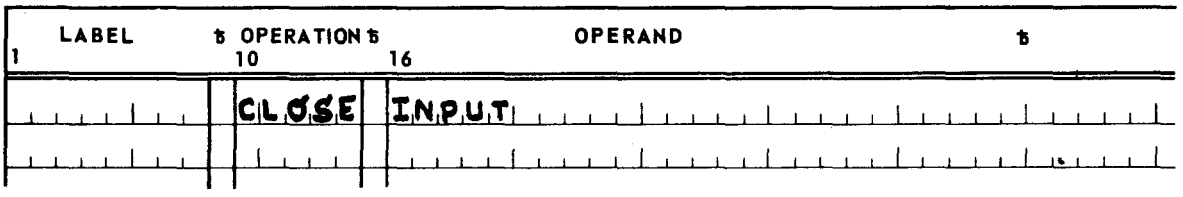
Operational Conditions:

A CLOSE macro must be executed before program control is returned to the operating system by means of the EOJ routine.

An attempt to close a file that has been previously closed or has not been opened results in an error display and termination of the problem program.

A file can be reopened after it has been closed.

Example:



2.7. SOURCE CODE LINK INSTRUCTIONS

The DTFSA macro instruction generates a series of Branch And Link instructions and a file description table defining the file and serving as a link between the program and the I/O routines of SAM.

The Branch And Link instructions generated by the DTFSA macro instruction take the form:

```
BAL     15,N?OP     OPEN
BAL     15,N?CL     CLOSE
BAL     15,N?GT     GET
BAL     15,N?PT     PUT
```

The imperative macros produce the following source code:

```
OPEN -      BAL     14,filename + 0
CLOSE -     BAL     14,filename + 4
GET -       BAL     14,filename + 8
            DC      Y (workarea)
PUT -       BAL     14,filename + 16
            DC      Y (workarea)
```

Register 14 contains the return address to the problem program.

3. DIRECT ACCESS IOCS

3.1. GENERAL

The Direct Access Method (DAM) of addressing files allows the user to create, retrieve, and process sequentially and nonsequentially ordered direct access files which reside on the UNIVAC 8410 Direct Access Subsystem (DAS). DAM also provides the interface between the user problem programs and the 8410 disc dispatcher. The 8410 disc dispatcher provides a software interface between DAM and/or user-developed software and the UNIVAC 8410 disc hardware.

DAM selects records for processing based on a different numeric value associated with each record. The numeric value indicates the logical position of a record in a file.

This section describes the minimum hardware and software configurations of DAM, the input/output record size and format, the user interfacing procedures, and the declarative and imperative macro instructions.

3.2. MINIMUM HARDWARE AND SOFTWARE CONFIGURATION

The minimum hardware configuration required for DAM is a central processor unit with a main storage capacity of at least 12K, a card input device, a printer, and one 8410 disc drive.

The minimum software configuration required for DAM is the Minimum Operating System (MOS).

3.3. INPUT/OUTPUT RECORD SIZE AND FORMAT

The physical record size processed by DAM is specified by the user in the DTFDA declarative macro instruction (see 3.5). DAM will process only single or multi-volume files containing fixed-length unblocked record formats; the user is limited to a maximum record length of 160 bytes for unblocked records.

3.4. USER INTERFACING PROCEDURES

The procedures required to interface the user problem program and DAM are as follows:

(1) Provide the IOCS with the information required to construct a file description table defining the file. This is accomplished by specifying the DTFDA declarative macro instruction (see 3.5.1). The user must specify a DTFDA macro instruction for each file accessed.

(2) Initiate specific direct access processing by specifying various imperative macro instructions (see 3.6).

3.4.1. Operator Interface

The operator must react to specific error halt displays which signify that DAM has encountered either a hardware or software error. If an error is not directly related to a specific disc unit, the physical unit number of the first volume of the file is displayed.

Table 3-1 lists the error halt displays and the corresponding replies.

HEXA-DECIMAL DISPLAY	SIGNIFICANCE	REPLY
24x2	OPEN/CLOSE requested for a previously OPENED/CLOSED file.	Press START to cancel job.
24x3	I/O errors prevent reading of VTOC or file labels.	
24x4	Cannot find VTOC label or cannot find filename in VTOC label.	Press START to cancel job. Key a nonzero into location 4 to repeat search. Press START.
24x6	I/O error occurred during PUT/GET and keyword ERRO is not specified.	Press START to cancel job.
24x7	PUT/GET issued for an unopened file.	Press START to cancel job. Unit number in this display identifies the first volume of the file.
24x8	File label check failed during OPEN.	Press START to cancel job. Key a nonzero into location 4 to repeat label check. Press START.
24xF	Relative record number requested is larger than the largest record number in the file.	Press START to cancel job. Unit number in this display identifies the first volume of the file.

NOTE: x indicates physical unit number.

Table 3-1. Error Halt Displays and Replies

3.5. DECLARATIVE MACRO INSTRUCTIONS

In order to use DAM, the programmer must specify a macro instruction to define the file characteristics. The DTFDA macro instruction defines the file characteristics of the direct access storage device.

The symbolic name of the file has a maximum length of eight characters and must begin with an alphabetic character. The keyword parameters may appear in any order, and they must be separated by commas.

3.5.1. Defining Direct Access Storage Device Files

The DTFDA declarative macro instruction is required to define all files which are to be processed by DAM.

Following is the format of the DTFDA macro instruction which shows, in alphabetical order, the required and optional keyword parameters which may appear in the operand of the DTFDA macro instruction. A description of the individual parameters follows the format. A summary of the keyword parameters is given in Table 3-2 following the descriptions.

The format of the DTFDA macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
filename	DTFDA	CRDT = label, DEVx = nn, FLID = label, GENO = label, NODV = n, TABL = label, XPDT = label [,CHCK = NO] [,ERRO = {IGNORE / label}] [,RCSZ = n]

- File Identification

  The following four keyword parameters are required to identify the file area:

  - CRDT = label

    This required keyword parameter allows the programmer to insert a creation date, where label is the address of a six-byte area in the problem program containing the creation date.

  - FLID = label

    This required keyword parameter allows the programmer to insert file identification data, where label is the address of an eight-byte area in the problem program containing user file identification.

  - GENO = label

    This required keyword parameter allows the programmer to insert a generation number, where label is the address of a four-byte area in the problem program containing the generation number.

  - XPDT = label

    This required keyword parameter allows the programmer to insert an expiration date, where label is the address of a six-byte area in the problem program containing the expiration date.

- Device Address

  The logical unit number of the device on which the file resides is specified by the following required keyword parameter:

  DEVx = nn

  where x is a decimal number ranging from 1 to 8 specifying the volume contained on the logical unit, and nn is a decimal number denoting the logical unit number (00 to 63) of the device.

One DEV parameter must be specified for each volume in a file; each logical unit number must be unique. A file defined by the DTFDA macro instruction must begin with volume 1.

- Number of Disc Devices

  The number of disc devices containing a file must be specified by the following required keyword parameter:

  NODV=n

  where n is the number of devices (1–8) and is equal to the number of volumes of the file.

- Error Conditions

  If an I/O request is unsuccessful, one of the following error types is sent by the 8410 disc dispatcher to DAM:

  - Nonoperative Channel
  - Invalid Function
  - Catastrophic Failure
  - Unrecoverable Output Bus Check
  - Unrecoverable Abnormal Line

  If the IOCS detects an Unrecoverable Output Bus Check or an Unrecoverable Abnormal Line, an error display and program halt occur. The error condition reflects the status of the last PUT or GET issued by the problem program. The program is then terminated when START is pressed. If other action is to be taken, the programmer must specify one of the following optional keyword parameters:

  - ERRO=IGNORE

    This keyword parameter indicates that the IOCS ignores either the UOB or UAL error condition and continues program execution.

  - ERRO=label

    This keyword parameter indicates that the IOCS transfers control to the address specified by label, where label is the address of the first byte of the error routine in the problem program.

- Write Check

  A write check is performed on all records transferred to disc unless the following optional keyword parameter is specified:

  CHCK=NO

  This parameter is used by the programmer when saving execution time and storage space is a factor.

- Record Size

DAM handles only unblocked records (one record per sector); maximum record size is 160 bytes. If a record size of less than 160 bytes is to be used, the programmer must specify the following optional keyword parameter:

RCSZ = n

where n is the number of bytes expressed as a decimal number.

During a write operation which transfers a record containing more than one byte but less than 160 bytes, the IOCS clears the unused area of the sector to zeros. However, if the record size is one byte, the contents of the remainder is unpredictable.

- Extent Table

An extent refers to the limits of the area on disc assigned to a file (the VTOC contains the limits). The programmer must reserve a 24-byte area in the problem program for DAM to use for file extent definitions. The table area information is supplied by the following required keyword parameter:

TABL = label

where label is the address of the user area which contains descriptive information for the file extent. For information concerning file extent definitions see *UNIVAC 9200/9300 Systems 8410 Direct Access Subsystem Disc Utility Programmer Reference, UP-7668* (current version).

KEYWORD	SPECIFICATION	FILES		REMARKS
		INPUT	OUTPUT	
CHCK	NO		X	Indicates no write check is to be performed.
CRDT	label	R	R	Inserts a creation date.
DEVx	nn=logical unit number	R	R	Specifies logical unit number of device on which file resides.
ERRO	IGNORE	X	X	Indicates ignored error and continued execution.
	label	X	X	Indicates address of user's error routine.
FLID	label	R	R	Inserts file identification data.
GENO	label	R	R	Inserts generation number.
NODV	n=number of devices	R	R	Indicates number of devices per file.
RCSZ	n=number of bytes	X	X	Indicates record size less than 160 bytes.
TABL	label	R	R	Indicates address of extent table.
XPDT	label	R	R	Inserts an expiration date.

LEGEND:
R = Required
X = Optional

Table 3-2. Summary of DTFDA Macro Instruction Keyword Parameters

Example:

LABEL	OPERATION	OPERAND	72	80
	DTFDA	CRDT=DATE,	X	
		DEVI=01,	X	
		FLID=FILE,	X	
		GENO=GNUM,	X	
		NODV=1,	X	
		TABL=EXTAB,	X	
		XPDT=EXDAT,	X	
		ERRO=CHECK,	X	
		RCSZ=115		

3.5.1.1. DTFDA Error Messages

During the generation of the DTFDA macro instruction, tests are made to ensure the validity of the instruction parameters. If an error is found, an appropriate error message is produced. This message produces an I flag from the assembler, and is accompanied by a 4-byte zerofill (assembler produces 4 bytes of zeros). In order to facilitate the user's modifying the DTFDA expansion without having to reassemble (REP cards), each error message is followed by an ORG *-4, thus allowing the extra 4-byte pad to be overlaid by the proper specification.

DTFDA error messages are illustrated in Table 3-3.

MESSAGE	SIGNIFICANCE
FILE-LABEL-ERROR	FLID/CRDT/GENO not specified.
NODV-NOT-SPECIFIED	NODV=0
TABL-NOT-SPECIFIED	Extent table address not specified.
DEVICE 1 IS INCORRECTLY SPECIFIED	DEV1>63
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV2	More than one volume assigned to same logical unit or DEVx>63.
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV3	
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV4	
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV5	
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV6	
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV7	
DEVICE-ASSIGNMENT--GENERATION-STOPPED AT DEV8	
MAXIMUM-OF-EIGHT-DEVICES--REMAINING DEVICES IGNORED	NODV>8

Table 3.3. DTFDA Error Messages

3.6. IMPERATIVE MACRO INSTRUCTIONS

The imperative macro instructions supplied for DAM are OPEN, GET, PUT, and CLOSE. These macro instructions are used by the problem program to open, process, and close files. The filename of an imperative macro instruction may have a maximum of eight characters.

3.6.1. OPEN Macro Instruction

This instruction must be used to initialize the file before any other imperative macro instruction can be performed. It is also used for label checking.

The format of the OPEN macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	OPEN	filename

- Positional Parameter 1

  filename – is the label of the corresponding DTFDA declarative macro instruction in the program defining the file to be opened.

Operational Conditions:

The execution of the OPEN macro results in the following comparisons:

The volume number specified in the DEV parameter is compared to that in the VTOC;

The file identification, creation date, and generation number contained in the disc label are compared to the corresponding information specified in the DTFDA macro.

If any of the items do not match, an error is displayed and the processor halts.

The operator can then either load a new disc and reinitiate the search by pressing START, or cancel the job by keying a binary nonzero value into location 4 by means of the ALTER switch and then pressing START.

The operator must mount a new disc if a file search is to be reinitiated, because the IOCS assumes that each file identifier contained in the VTOC is unique.

If the comparisons are valid, the IOCS moves all extent definitions, including any contained in secondary file labels, to the save area specified by the TABL parameter.

DAM then repeats the entire open procedure for the next logical unit specified by the DEV parameter in the DTFDA macro defining the file, and continues this repetition until all of the specified units have been opened.

Example:

```
LABEL    ↹ OPERATION ↹    OPERAND                    ↹
1        10               16
         OPEN             INPUT
         OPEN             OUTPUT
```

3.6.2. GET Macro Instruction

This instruction retrieves a logical record from a disc file during random retrieval of records.

The format of the GET macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	GET	filename,workarea

- Positional Parameter 1

  filename – is the label of the DTFDA declarative macro instruction defining the file from which the record is to be read..

- Positional Parameter 2

  workarea – is the label of the first byte of a work area defined by the user.

Operational Conditions:

The work area must be five bytes larger than the record size because the first five bytes are reserved for the unit, track, and sector address of the record to be read.

The define constant statement DC CL6 'dddddd', where 'dddddd' is the number of the record within the file, must be specified immediately following the GET macro instruction.

The IOCS uses the constant 'dddddd' with the GET macro to calculate the track and sector address of the record. This address is stored in the first five bytes of the work area. The IOCS issues a call to the disc dispatcher to read a record and waits until a status is returned. If the status does not indicate an error condition, the IOCS returns control to the problem program for processing; otherwise, the error is processed by the routine in the problem program indicated by the label in the ERRO parameter. If ERRO is not specified, the job will be canceled when START is pressed. In either case, the IOCS executes the I/O operation and transfers a record to the work area.

Example:

LABEL 1	ƀ OPERATION ƀ 10	OPERAND 16	ƀ
	GET	INPUT,INWA	
RECNO	DC	CL6'000725'	

3.6.3. PUT Macro Instruction

This instruction indicates that a new or updated record is to be written on the direct access storage device.

The format of the PUT macro instruction is:

LABEL	ƀ OPERATION ƀ	OPERAND
[name]	PUT	filename,workarea

- Positional Parameter 1

  filename – is the label of the DTFDA declarative macro instruction defining the file to which the record is to be written.

■ Positional Parameter 2

workarea – is the label of the first byte of a work area defined by the user.

Operational Conditions:

The work area must be five bytes larger than the record size because the first five bytes are reserved for the unit, track, and sector address of the record to be read.

The define constant statement DC CL6 'dddddd', where 'dddddd' is the number of the record within the file, must be specified immediately following the PUT macro instruction.

The IOCS uses the constant 'dddddd' supplied with the PUT macro to calculate the track and sector address of the record. This address is stored in the first five bytes of the work area. The IOCS issues a call to the disc dispatcher to write a record and waits until a status is returned. If the status does not indicate an error condition, the IOCS returns control to the user's program for processing; otherwise, the error is processed by the routine in the problem program indicated by the label in the ERRO parameter if ERRO is not specified, the job will be canceled when START is pressed. In either case, the IOCS executes the I/O operation and transfers a record to the work area.

A PUT macro issued for a record in an unopened file, or for a record (address) not in the file, results in an error display and termination of the job.

Example:

1 LABEL	↹ OPERATION ↹ 10	16 OPERAND ↹
	PUT	OUTPUT,OUTWA
RECX	DC	CL6'000545'

3.6.4. CLOSE Macro Instruction

This instruction must be used to terminate the processing of the file. Once the file is closed, no other macro instruction may be executed for the file until it is reopened by the OPEN macro instruction.

The format of the CLOSE macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	CLOSE	filename

■ Positional Parameter 1

filename – is the label of the corresponding DTFDA declarative macro instruction in the program defining the file to be closed.

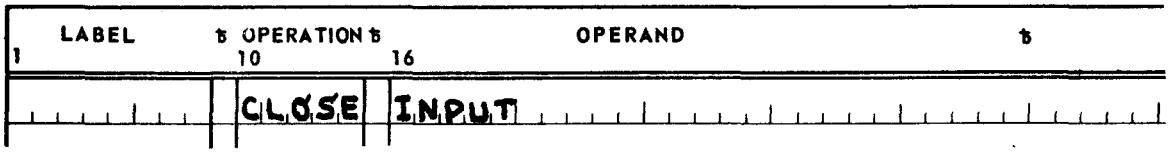
Operational Conditions:

A CLOSE macro must be executed before program control is returned to the operating system by way of the EOJ.

An attempt to close an unopened file or a file that has been previously closed results in an error display and termination of the problem program.

A file can be reopened after it has been closed.

Example:



3.7. RESTART INFORMATION

DAM provides the problem program with access to a restart control word generated by DTFDA containing address information necessary for a restart procedure.

DAM provides the following four-byte restart control word at filename-4:

FILENAME-4	Logical unit number pointer.
FILENAME-3	Label pointer. This is the actual displacement from FILENAME.
FILENAME-2	Access method. C'R'
FILENAME-1	File status X'00' - File Open X'FF' – File Closed

The file status indicator is updated at open and close time without regard to the restart specification.

The layout of label information is as follows:

FILENAME + LABEL POINTER

FLID	ADDRESS	CRDT	ADDRESS
GENO	ADDRESS		

The layout of the restart control word is as follows:

LU# POINTER	LABEL POINTER	ACCESS METHOD	FILE STATUS

Note the negative displacement from the filename. At location 'Filename-14' a list of logical unit numbers, one byte per number, will be found. The end of the list is indicated by a byte containing X'FF'.

4. INDEXED SEQUENTIAL IOCS

4.1. GENERAL

The Indexed Sequential Access Method (ISAM) of addressing files allows the user to create, retrieve, process, and maintain, in either a random or sequential manner, indexed sequential files which reside on the UNIVAC 8410 Direct Access Subsystem (DAS). ISAM also provides the interface between the user problem programs and the 8410 disc dispatcher. The 8410 disc dispatcher provides a software interface between ISAM and/or user-developed software and the UNIVAC 8410 disc hardware.

ISAM selects records for processing based on the value in a key field in each record in the file.

This section describes the indexed sequential file characteristics, the minimum hardware and software configurations of ISAM, the input/output record size and format, the user interfacing procedures, and the declarative and imperative macro instructions.

4.1.1. Indexed Sequential File

An indexed sequential file includes three areas: an index area (on fastband), a prime data area, and an optional overflow area. Each record in the file must contain a key; and, at the time the file is loaded, the records must be delivered in ascending order by key. ISAM writes the records in the prime area; the records are loaded in sequence starting with the lowest numbered sector in the first volume of the prime area. When the first volume is full, loading continues on the next volume, and so on, up to a maximum of eight volumes. On any one volume of an indexed sequential file, the prime area must comprise only one extent which must begin in the second logical track, that is, track zero sector 55 (0055), and must end in sector 54 or sector 99 of any track in the volume.

Each volume of an indexed sequential file may include as many as ten overflow extents. The overflow area(s) may begin and end anywhere on the disc.

Records may be blocked in the prime area; that is, more than one record may be written in the same sector. If records are blocked, each block of records in a sector is preceded by a key field which contains the key of the last data record associated with that sector. If records are unblocked, each sector contains one record preceded by a key field. The 8410 Disc Index Sequential Processor for RPG and the assembler utilizes a staggering effect on each physical track used in the prime area; for example, a master key located on track 00 sector 55 will have its associated record on track 00 sector 73. The record in track 00 sector 55 will be associated with the master key located on track 00 sector 82. RPG has a

staggering effect of 18. The assembler for the 9300 Central Processor Unit may have a staggering effect of either 12 or 18 depending on what is stated as the system identification parameter (12 for PROC = 9300, 18 for PROC = 9200) in the DTFIA macro instruction. The DTFIA system identification parameter for the 9200 CPU should be PROC = 9200. A program generated on a UNIVAC 9200 or 9300 CPU may function on either computer, but if RPG and the assembler language are used with the same file, PROC = 9200 must be stated in the DTFIA macro instruction.

ISAM considers a disc to be comprised of 200 logical tracks (100 physical tracks). Each set of sector addresses in the range n00 through (n+1) 99 is considered to comprise four tracks. The range of sector addresses n00 through n54 is one track, n55 through n99 is a second track, (n+1)00 through (n+1)54 is a third track, and (n+1)55 through (n+1)99 is a fourth track. For example, the range of sector addresses 0000 through 0199 comprises four logical tracks (two physical tracks): 0000-0054, 0055-0099, 0100-0154, and 0155-0199.

As an indexed sequential file is loaded on a disc, ISAM constructs an index of keys on the fastband. Each fastband sector contains four entries or keys corresponding to four logical tracks with sector addresses in the range n00 through (n+1)99; therefore, each fastband entry corresponds to one particular track. Since a maximum of 199 keys (0000-0054 comprises the VTOC area) must be available for storage on the fastband, maximum key size is limited to 40 bytes. ISAM ensures that all unused sectors of the fastband contain binary zeros, and is also responsible for staggering the high key of the fastband sector from the remaining three keys of a sector, enabling them to be read after a magnitude search without an intervening disc revolution. The staggering effect on the fastband is the same as that of the prime area.

After an indexed sequential file is loaded, insert additions cause higher prime data records to be placed in the overflow area(s) and are chained into the proper sequence in the file by means of a linking procedure. Blocking is accomplished automatically in the overflow area(s) if possible, even though the prime area may be unblocked. High adds are placed in the prime area if it is not full; otherwise they will be placed in the overflow area. If a group of records with high keys is to be added to the prime area, they must be presented in ascending sequence. If records are blocked, more than one record may be written in an overflow area sector, but each record carries with it a link field. As a mechanism used in the linking procedure, each sector in the prime area contains one link field. Thus, maximum record size is sector size less the sum of key field size and link field size. Records may not be blocked unless sector size is at least as large as link field size plus key field size plus two times record size. When an indexed sequential file has been loaded, each prime area link field contains zeros. Link fields are always five bytes in length.

If an indexed sequential file is subject to insert additions, it should be reorganized periodically, since additions increase the time required to access records; if the overflow area(s) becomes full, reorganization is necessary. An indexed sequential file can be reorganized by reading it sequentially and then loading it onto another file. ISAM does not handle deletions from an indexed sequential file. The user must tag the records to be deleted and omit them when the file is reorganized.

4.2 MINIMUM HARDWARE AND SOFTWARE CONFIGURATION

The minimum hardware configuration required for ISAM is a central processor unit with a main storage capacity of at least 12K, a card input device, a printer, and one 8410 disc drive.

The minimum software configuration required for ISAM is the minimum operating system (MOS).

4.3 INPUT/OUTPUT RECORD SIZE AND FORMAT

The physical record size processed by ISAM is specified by the user in the DTFIA declarative macro instruction (see 4.5). The maximum length for unblocked records is 160 bytes less the sum of key field size plus link field size. However, when using blocked records, sector size must be large enough to accomodate link field size plus key field size plus two times record size. ISAM will process single or multi-volume files containing fixed-length unblocked or blocked record formats.

4.4 USER INTERFACING PROCEDURES

The procedures required to interface the user problem program and ISAM are as follows:

(1) Advise the IOCS as to which phases of the I/O control routine are required. This is accomplished by specifying either the DISIO declarative macro instruction, or the DISMN, DISOC, DISRD, DISLD, DISA1, and DISA2 declarative macro instructions.

(2) Provide the IOCS with the information required to construct a file description table defining the file. This is accomplished by specifying the DTFIA declarative macro instruction (see 4.5.2). The user must specify at least one DTFIA macro instruction for each file type accessed.

(3) Initiate specific indexed sequential processing by specifying various imperative macro instructions (see 4.6).

4.4.1. The operator must react to specific error halt displays which signify that ISAM has encountered either a hardware or software error. If an error is not directly related to a specific disc unit, the physical unit number of the first volume of the file is displayed.

Table 4-1 lists the error halt displays and the corresponding replies.

HEXA-DECIMAL DISPLAY	SIGNIFICANCE	REPLY
25x0	Any I/O error except nonoperational channel.	Press START to continue. Key a nonzero into location 4 to cancel job. Press START.
25x1	Update macro (PUT/WRITE) issued and either not preceded by an input macro (GET/READ) or key of record to be updated was changed.	Press START to cancel job.
25x2	An attempt was made to close or access a file that had not been opened, or to open an opened file.	
25x3	An imperative macro was issued and file does not support this type of macro instruction.	
25x4	An imperative macro was issued and the indexed sequential generated I/O control routines (DISIO) do not support this type of macro instruction.	
25x5	An attempt was made to open a file that was not marked as a LOAD, ADD, READ, or ADDRTR file.	
25x6	An I/O error occurred when OPEN/CLOSE accessed the file label(s).	
25x7	During OPEN the disc did not contain the VTOC label.	
25x8	During OPEN the disc did not contain a file label for this file.	
25x9	During OPEN a label check failure occurred (either CRDT or GENO) for a READ, ADD, or ADDRTR file.	To retry, mount another disc unit. Press START. Key a nonzero other than one into location 4 to cancel job. Press START. Key a one into location 4 to ignore. Press START.
25xA	During OPEN, a nonempty LOAD file contained a creation date which was less than the expiration date written on the label on the disc, or the FLID or volume number did not match.	To continue, mount another disc unit. Press START. Key a nonzero into location 4 to cancel job. Press START.
25xB	During OPEN the file was not marked as an indexed sequential file in the file label.	Press START to cancel job.
25xC	The programmer tried to enter the IOCS after the ERRO routine was entered, but before making special return to the IOCS.	Press START to cancel job (file not closed before cancel).
25xD	The programmer-generated DTFIS macro supports one type of LOAD and the generated I/O routine supports the other type.	Press START to cancel job.

NOTE: x indicates physical unit number.

Table 4-1. Error Halt Displays and Replies

4.5. DECLARATIVE MACRO INSTRUCTIONS

In order to use ISAM, the programmer must specify macro instructions to describe the I/O routine and define the file characteristics. Seven declarative macro instructions are available to describe the I/O routine; the DTFIA macro instruction defines the direct access storage device.

The symbolic name of the file has a maximum length of eight characters and must begin with an alphabetic character. The keyword parameters may appear in any order, and they must be separated by commas.

4.5.1. Generating the I/O Routine

For either a 16K card, tape, or disc loaded system, the DISIO macro instruction generates the complete I/O routine. Because of the size limitations of a 12K card loaded system, the DISMN and DISOC macro instructions must be specified to generate the I/O routine; in addition, the DISRD, DISLD, DISA1, and DISA2 macro instructions are specified in conjunction with the DISMN and DISOC macros when their associated options are desired.

Table 4-2 lists the macro instructions required for generating an I/O routine.

SYSTEM	MACRO INSTRUCTION	FUNCTION
9300 Tape or Disc System	DISIO	Generates ADD, READ, LOAD
9200/9300 Card Systems	DISMN	Generates generalized routines
	DISOC	Generates OPEN/CLOSE
	DISRD	Generates sequential retrieve and/or random read routines.
	DISLD	Generates LOAD (high speed) routine.
	DISA1 DISA2	Generates ADD, LOAD (normal) routine.

Table 4-2. I/O Routine Generating Macro Instructions

Because of the storage limitations of a card loaded system, each macro instruction must be generated by the preassembly macro pass separately. The decks produced during these preassembly macro passes are combined into one deck and assembled as one program. Before the deck is assembled into the final source program, it is necessary to remove all but the last END card. The order in which the final deck must be assembled is as follows:

(1) DISMN coding

(2) DISOC coding

(3) DISRD and/or DISLD coding and/or DISA1/DISA2

The I/O routine generating macros produce the coding necessary to support the user's imperative macro instructions. This coding is interfaced to the user's instructions and provides for the entering of ISAM at a unique label. In order to provide greater flexibility, the I/O routine generating macros need not be assembled in the same program as the DTFIA macro instruction generating the interface. ISAM provides appropriate EXTRN and ENTRY instructions to allow for the linking of the two separately assembled entities by means of the linker.

The input/output declarative macro instruction formats follow. The formats list, in alphabetical order, the optional keyword parameters which may appear in the operand of the macro instructions in any order. A description of the individual parameters follows the illustrations. A summary of the keyword parameters is given in Table 4-3.

The format of the DISIO macro instruction, used to generate the I/O routine for 16K card, tape, or disc loaded systems, and the formats of the DISMN, DISOC, DISRD, DISLD, DISA1, and DISA2 macro instructions, used to generate the I/O routine for 12K card loaded systems, are as follows:

LABEL	♭ OPERATION ♭	OPERAND
[name]	DISIO	[ADD = YES] [,BLK = YES] [,HSPD = YES] [,LOAD = YES] [,READ = NO] [,RAND = YES] [,SEQ = NO] [,SETL = NO] [,UPDT = YES]

LABEL	♭ OPERATION ♭	OPERAND
[name]	DISMN	[ADD = YES] [,LOAD = YES] [,RAND = YES] [,READ = NO] [,SEQ = NO] [,SETL = NO] [,UPDT = YES]

LABEL	↔ OPERATION ↔	OPERAND
[name]	DISOC	[ADD = YES] [,LOAD = YES] [,READ = NO]

LABEL	↔ OPERATION ↔	OPERAND
[name]	DISRD	[ADD = YES] [,BLK = YES] [,RAND = YES] [,SEQ = NO] [,SETL = NO] [,UPDT = YES]

LABEL	↔ OPERATION ↔	OPERAND
[name]	DISLD	[BLK = YES]

LABEL	↔ OPERATION ↔	OPERAND
[name]	DISA1	[ADD = YES] [,BLK = YES] [,LOAD = YES]

LABEL	↔ OPERATION ↔	OPERAND
[name]	DISA2	[ADD = YES] [,BLK = YES] [,LOAD = YES]

- ADD Keyword Parameter

  If the user desires to add records, the following keyword parameter must be specified:

  ADD = YES

- BLK Keyword Parameter

  If the user desires to allow blocked records, the following keyword parameter must be specified:

  BLK = YES

- HSPD Keyword Parameter

  If the user desires high speed loading, the following keyword parameter must be specified:

  HSPD = YES

- LOAD Keyword Parameter

  If the user desires to load records, the following keyword parameter must be specified:

  LOAD = YES

- RAND Keyword Parameter

  If the user desires random retrieval of records, the following keyword parameter must be specified:

  RAND = YES

- READ Keyword Parameter

  If the user desires the elimination of retrieval coding, the following keyword parameter must be specified:

  READ = NO

- SEQ Keyword Parameter

  If the user desires the elimination of sequential retrieval of records, the following keyword parameter must be specified:

  SEQ = NO

- SETL Keyword Parameter

  If the user desires the elimination of SETL coding, the following keyword parameter must be specified:

  SETL = NO

- UPDT Keyword Parameter

  If the user desires to update records, the following keyword parameter must be specified:

  UPDT = YES

KEYWORD	SPECIFICATION	16K CARD, TAPE, DISC	12K CARD					REMARKS
		DISIO	DISMN	DISOC	DISRD	DISLD	DISA1 DISA2	
ADD	YES	X	X	X	X		X	Specified if ADD function is desired.
BLK	YES	X			X	X	X	Specified if blocked records are used.
HSPD	YES	X						Specified if high speed LOAD is desired.
LOAD	YES	X	X	X			X	Specified if LOAD is desired.
RAND	YES	X	X		X			Specified if random retrieval is desired.
READ	NO	X	X	X				Specified to eliminate retrieval coding.
SEQ	NO	X	X		X			Specified to eliminate sequential retrieval.
SETL	NO	X	X		X			Specified to eliminate SETL coding.
UPDT	YES	X	X		X			Specified to update records.

Table 4-3. Summary of I/O Routine Generating Keyword Parameters

4.5.1.1. Indexed Sequential I/O Error Messages

During the generation of indexed sequential I/O macro instructions, tests are made to ensure the validity of the instruction parameters. If an error is found, an appropriate error message is produced. This message produces an I flag from the assembler; however, the generated coding is usable. Indexed sequential I/O error messages are illustrated in Table 4-4.

MESSAGE	SIGNIFICANCE
READ-INVALID-READ-GENERATED	READ≠0 and READ≠NO
ADD-INVALID-ADD-NOT-GENERATED	ADD≠0 and ADD≠YES
LOAD-INVALID-LOAD-NOT-GENERATED	LOAD≠0 and LOAD≠YES
UPDT-INVALID-UPDT-NOT-GENERATED	UPDT≠0 and UPDT≠YES
UPDT-INCONSISTENT-IGNORED	READ=NO and UPDT≠0
RAND-INVALID-RAND-NOT-GENERATED	RANDOM≠0 and RANDOM≠YES
RAND-INCONSISTENT-IGNORED	READ=NO and RANDOM≠0
SEQ-INVALID-SEQ-GENERATED	SEQ≠0 and SEQ≠NO
SEQ-INCONSISTENT-IGNORED	READ=NO and SEQ≠0
BLK-INVALID-BLK-NOT-GENERATED	BLK≠0 and BLK≠YES
SETL-INVALID-SETL-GENERATED	SETL≠0 and SETL≠NO
SETL-INCONSISTENT-IGNORED	SETL≠0 and (either READ=NO or READ=0 and SEQ=NO)
HSPD-INVALID-HSPD-LOAD-NOT-USED	LOAD=YES and HSPD≠0 and HSPD≠YES
HSPD-INVALID-IGNORED	HSPD=YES and LOAD≠YES
PARAMETERS-INCONSISTENT-NOT-GENERATED	READ≠NO and SEQ≠NO and RANDOM=YES
PARAMETERS-INCONSISTENT-NOTHING-GENERATED	READ=NO and ADD≠YES and LOAD≠YES

Table 4-4. Indexed Sequential I/O Error Messages

4.5.2. Defining Direct Access Storage Device Files

The DTFIA declarative macro instruction is required to define all files which are to be processed by ISAM.

Following is the format of the DTFIA macro instruction which shows, in alphabetical order, the required and optional keyword parameters which may appear in the operand of the DTFIA macro instruction. A description of the individual parameters follows the format. A summary of the keyword parameters is given in Table 4-5 following the descriptions.

The format of the DTFIA macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
filename	DTFIA	CRDT=label, DEVx=nn, DVNO=n, ERRO=label, FLID=label, GENO=label, KLEN=n, KLOC=n, PROC={9200 / 9300}, RCSZ=n [,ADAR=label] [,CHCK=NO] [,DTAR=label] [,DUPR=label] [,EQFA=label] [,IOA1=label] [,IOA2=label] [,IORT={ADD / ADDRTR / LOAD}] [,KARG=label] [,LDTM=HSPD] [,RCFM=BLK] [,RTRV=label] [,SQCH=label] [,TYPE=RANDOM] [,UPDT=YES] [,WORK=label] [,XPDT=label]

- File Identification

  The following four keyword parameters are used to identify the file area.

  - CRDT = label

    This required keyword parameter allows the programmer to insert a creation date, where label is the address of a six-byte area in the problem program containing the creation date.

  - FLID = label

    This required keyword parameter allows the programmer to insert file identification data, where label is the address of an eight-byte area in the problem program containing user file identification.

  - GENO = label

    This required keyword parameter allows the programmer to insert a generation number, where label is the address of a four-byte area in the problem program containing the generation number.

  - XPDT = label

    This keyword parameter allows the programmer to insert an expiration date, where label is the address of a six-byte area in the problem program containing the expiration date. The XPDT keyword parameter is required only when the IORT = LOAD keyword parameter is used.

- Device Address

  The volume number of the disc pack and the logical unit number of the device on which the file resides are specified by the following required keyword parameter:

  DEVx = nn

  where x is the volume number (1-8) of the disc pack, and nn is a decimal number denoting the logical unit number (00 to 63) of the device.

- Device Number

  The number of disc devices is specified by the following required keyword parameter:

  DVNO = n

  where n is the number of disc devices (1-8), and is equal to the number of volumes of the file.

- Unrecoverable Errors

  If an I/O request is unsuccessful, one of the following error types is sent by the 8410 disc dispatcher to ISAM:

- Nonoperative Channel
- Invalid Function
- Catastrophic Failure
- Unrecoverable Output Bus Check
- Unrecoverable Abnormal Line

If a nonoperational channel error occurs, ISAM issues a Branch And Link instruction, through register 14, to the label specified by the ERRO keyword parameter:

ERRO = label

This required keyword parameter indicates that the IOCS transfers control to the address specified by label, where label is the address of the first byte of the error routine in the problem program.

If any of the other error conditions occur, the unit, track, and sector numbers are stored in the status block along with the status indicator, and are passed to the user through register 15; an error display and halt results. The user may then optionally key a binary nonzero value into location four before pressing START to cancel the job. In either case, the user must return control through register 14 (see 4.8.1).

■ Key Length

The following required keyword parameter is used to specify the key length:

KLEN = n

where n is the number of bytes in the key. Maximum key size is 40 bytes.

■ Key Location

The following required keyword parameter is used to specify the location of the key within the record:

KLOC = n

where n is the number of the most significant byte of the key field, and bytes in the record are numbered from zero.

■ System Identification

The following required keyword parameter must be specified because the staggering effect of ISAM (number of sectors separating the key from the prime data block) is dependent upon processor speed:

$$\text{PROC} = \left\{ \begin{matrix} 9200 \\ 9300 \end{matrix} \right\}$$

The user should specify PROC = 9200 for the 9200 processor; for the 9300 processor, either PROC = 9200 or PROC = 9300 may be specified.

- Full Overflow Area

  If the overflow area becomes full, ISAM branches unconditionally to the label specified by the following keyword parameter in the DTFIA declarative macro instruction when the next record is submitted to the file:

  ADAR = label

  where label is the address of a routine in the problem program. The ADAR keyword parameter must be specified if the ADD function is employed.

- Write and Check

  ISAM assumes that all records are to be written with the WRITE-CHECK command. If the user desires records to be written without CHECK, the following keyword parameter must be specified:

  CHCK = NO

- Full Prime Area

  If the prime area becomes full during the process of loading an indexed sequential file, the next execution of a WRITE macro instruction causes ISAM to branch unconditionally to the label specified by the following keyword parameter:

  DTAR = label

  where label is the address of a routine in the problem program (see 4.8). The DTAR keyword parameter must be specified if the loading function is employed.

- Duplicate Keys

  If, while adding records to an indexed sequential file, a key is presented to ISAM which is equal in value to a key which previously existed on that file, ISAM branches unconditionally to the label specified by the following keyword parameter:

  DUPR = label

  where label is the address of a routine in the problem program (see 4.8). The DUPR keyword parameter must be specified for an add operation.

- End of File

  When reading an indexed sequential file in a sequential manner, ISAM assumes the responsibility of detecting the end of file and notifies the problem program by branching unconditionally to the label specified by the following keyword parameter:

  EOFA = label

  where label is the address of a routine in the problem program (see 4.8). The EOFA keyword parameter must be specified if the reading function is employed.

- Input/Output Areas

To handle an indexed sequential file, ISAM must have an input/output area(s) to read and write sectors on a disc; ISAM requires that the length of its I/O area(s) be 165 bytes for all files.

The following keyword parameters are used to specify the input/output areas:

- IOA1 = label

This keyword parameter must be included in the DTFIA declarative macro instruction when reading an indexed sequential file, where label is the address of a 165-byte I/O area.

- IOA2 = label

This keyword parameter must be included in the DTFIA declarative macro instruction when an indexed sequential file is being loaded or if records are to be added to the file, where label is the address of a 165-byte I/O area.

If the user is both reading and adding records, he must specify both IOA1 and IOA2.

The user need not specify an I/O area when loading a file using the high-speed method; LDTM=HSPD generates all the buffers needed for this mode.

- File Processing Function

ISAM can be used to load an indexed sequential file, add records to an indexed sequential file, read records from an indexed sequential file, or both read and add records in the same program. ISAM assumes that it is to be used to read records; if otherwise, one of the specifications of the IORT keyword parameter must be implemented as follows:

$$\text{IORT} = \left\{ \begin{matrix} \text{ADD} \\ \text{ADDRTR} \\ \text{LOAD} \end{matrix} \right\}$$

If ISAM is used to read records from an indexed sequential file, the records can be read sequentially or randomly; ISAM assumes that it is to read records sequentially. If ISAM is used to add records, IORT=ADD should be specified. If ISAM is used to both read and add records, IORT=ADDRTR should be specified.

If ISAM is used to load an indexed sequential file, IORT=LOAD should be specified; two methods of loading are possible (normal and high speed) depending on the amount of storage available. The normal loading procedure, in which the staggering effect results in the key being separated from its associated data by a prescribed number of sectors, and in which sectors are loaded one at a time, is assumed by ISAM. If the user desires high-speed loading, the following keyword parameter must be included:

LDTM = HSPD

A buffer area equal to the number of sectors between data and key is reserved, and a write to the disc is not performed until the buffer area is filled. This is the most efficient method of loading; however, it requires a considerable amount of storage.

- File Processing Type

  ISAM assumes that it is to read records sequentially. If the user desires to read records randomly from an indexed sequential file, the following keyword parameter must be included in the DTFIA declarative macro instruction:

  TYPE=RANDOM

- Retrieval Search Argument

  If records are to be retrieved randomly from an indexed sequential file, the user must specify the key of the record to be read before executing the READ imperative macro instruction. The user specifies this key by moving it to a key area before the execution of the ensuing READ instruction; therefore, if records are are to be read randomly, the following keyword parameter must be specified:

  KARG=label

  where label is the address of the first byte of the key area. In addition, the key must be followed by a sentinel byte (X'FF').

  If records are to be added to an indexed sequential file, the KARG keyword parameter must be specified; the key of the record to be added must be placed in the key area before executing the WRITE NEWKEY imperative macro instruction. The KARG keyword parameter must also be specified if the SETL instruction is used.

- Record Blocking

  ISAM assumes that records in an indexed sequential file are unblocked. If the user desires to block records, the following keyword parameter must be specified:

  RCFM=BLK

- Record Size

  ISAM handles only fixed-length records. Maximum record size for unblocked records is equal to 160 bytes minus the sum of key field size and link field size. However, when using blocked records, the user must remember that sector size must be large enough to accommodate link field size plus key field size plus two times record size. ISAM assumes that record size is maximum record size. The following required keyword parameter must be specified for all files:

  RCSZ=n

  where n is the number of bytes in the record.

- Record Not Found

  If ISAM is unable to find a record during the setting of a lower limit or during the reading of an indexed sequential file in a random manner, ISAM branches unconditionally to the label specified by the following keyword parameter:

  RTRV=label

  where label is the address of the user's routine; (see 4.8); therefore, whenever the READ KEY or the SETL KEY macro instruction is employed, the RTRV keyword parameter must be specified.

- Sequence Error

  During a load operation, if a WRITE macro instruction presents a record whose key is not in ascending sequence from the last WRITE instruction, ISAM branches unconditionally to the label of the following keyword parameter:

  SQCH=label

  where label is the address of the user's routine (see 4.8). This keyword parameter must be specified for a load file.

- File Updating

  ISAM assumes that an indexed sequential file to be read is not to be updated. If a record is to be updated, the following keyword parameter must be specified:

  UPDT=YES

  After a record has been read and updated, the execution of a PUT or WRITE macro instruction (depending upon whether the file is to be processed sequentially or randomly) causes the record to be written back into the file at the position from which it was retrieved, destroying the original record. The PUT or WRITE instruction must follow the execution of the GET or READ instruction which delivered the record for processing, and these two instructions must not be separated by the execution of any other GET or READ instructions for the file; moreover, the execution of two PUT or WRITE instructions must be separated by the execution of at least one GET or READ instruction.

- Work Area

  Whenever loading, adding, or reading records randomly, ISAM must know the location of the work area in which the record is to be found or placed. This information is provided by specifying the following keyword parameter:

  WORK=label

  where label is the address of the work area. Work area size should be the same as max record size.

KEYWORD	SPECIFICATION	FILES		REMARKS
		INPUT	OUTPUT	
ADAR	label	X	X	Specifies address of user's routine for full overflow area. Required when adding records.
CHCK	NO		X	Must be specified for write without check.
CRDT	label	X	X	Specifies creation date. Required for all files.
DEVx	nn	X	X	Supplies volume number and the logical unit number of the disc. Required for all files.
DTAR	label	X	X	Specifies address of user's routine for full prime area. Required to load a file.
DUPR	label	X	X	Specifies user's routine for duplicate keys. Required to add records.
DVNO	n	X	X	Specifies the number of disc devices. Required for all files.
EOFA	label	X		Indicates the end of file. Required to read records sequentially.
ERRO	label	X	X	Specifies address of error routine. Required for all files.
FLID	label	X	X	Specifies file identification area. Required for all files.
GENO	label	X	X	Specifies generation number area. Required for all files.
IOA1	label	X		Specifies 165-byte I/O area. Required for reading.
IOA2	label		X	Specifies 165-byte I/O area. Required for add or normal load. If read and add are both used, IOA1 and IOA2 must be specified.
IORT	LOAD		X	Required to load a file.
	ADD		X	Required to add records to a file.
	ADDRTR	X	X	Required to add and read records.
KARG	label	X	X	Required for random processing, adding, or when SETL is used.
KLEN	n	X	X	Specifies key length. Required for all files.

Table 4-5. Summary of DTFIA Macro Instruction Keyword Parameters
(Part 1 of 2)

KEYWORD	SPECIFICATION	FILES		REMARKS
		INPUT	OUTPUT	
KLOC	n	X	X	Specifies position of key within record. Required for all files.
LDTM	HSPD		X	Required to specify high-speed load.
PROC	9200 or 9300	X	X	Required for all files.
RCFM	BLK	X	X	Required if blocked records are desired.
RCSZ	n	X	X	Specifies number of bytes in the record. Required for all files.
RTRV	label	X		Specifies address of user's routine for unfound records. Required for reading an indexed sequential file randomly and when SETL is used.
SQCH	label		X	Specifies address of user's routine for records out of sequence. Required for loading.
TYPE	RANDOM	X	X	Required for random processing.
UPDT	YES	X	X	Required to update records.
WORK	label	X	X	Specifies address of work area. Required for loading, adding, and reading randomly.
XPDT	label		X	Specifies expiration date. Required when IORT=LOAD is used.

Table 4-5. Summary of DTFIA Macro Instruction Keyword Parameters (Part 2 of 2)

Example:

```
LABEL      OPERATION      OPERAND                                  72  80
           DTFIA          CRDT=DATE,                               X
                          DEVI=01,                                 X
                          DVNO=2,                                  X
                          ERRO=CHECK,                              X
                          FLID=FILE,                               X
                          GENO=GNUM,                               X
                          KLOC=10,                                 X
                          PROC=9300,                               X
                          ADAR=OVFL,                               X
                          DUPR=DKEYS,                              X
                          IOA1=BEGIN,                              X
                          IOA2=USE,                                X
                          IORT=ADDRTR,                             X
                          KARG=RASET,                              X
                          RTRV=UNFND,                              X
                          TYPE=RANDOM,                             X
                          WORK=WAREA
```

The following required keyword parameters are common to all file processing operations:

*CHCK
CRDT
DEVx
DVNO
ERRO
FLID
GENO
KLOC
PROC
**RCFM

*Specified when write without check is desired.

**Specified when blocked records are desired.

Table 4-6 lists the additional keyword parameters required for loading, adding, sequential retrieval, random retrieval, sequential retrieval and addition, and random retrieval and addition of records, respectively.

LOAD	ADD	SEQUENTIAL RETRIEVAL	RANDOM RETRIEVAL	SEQUENTIAL RETRIEVAL AND ADDITION	RANDOM RETRIEVAL AND ADDITION
DTAR	ADAR	EOFA	IOA1	ADAR	ADAR
IOA2	DUPR	IOA1	KARG	DUPR	DUPR
IORT	IOA2	**KARG	RTRV	EOFA	IOA1
SQCH	IORT	**RTRV	TYPE	IOA1	IOA2
WORK	KARG	*UPDT	*UPDT	IOA2	IORT
XPDT	WORK		WORK	IORT	KARG
				KARG	RTRV
				**RTRV	TYPE
				*UPDT	*UPDT
				WORK	WORK

*Specified when updating is desired.

**Specified when using SETL.

Table 4-6. Summary of Indexed Sequential File Processing Operations

4.5.2.1. DTFIA Error Messages

During the generation of the DTFIA macro instruction, tests are made to ensure the validity of the instruction parameters. If an error is found, an appropriate error message is produced. This message produces an I flag from the assembler, and is accompanied by a four-byte zerofill (assembler produces four bytes of zeros). In order to facilitate the user's modifying the DTFIA expansion without having to reassemble (REP cards), each error message is followed by an ORG *-4, thus allowing the extra four-byte pad to be overlaid by the proper specification. DTFIA error messages are illustrated in Table 4-7.

MESSAGE	SIGNIFICANCE
IORT-INVALID-READ-ASSUMED	IORT not equal to either LOAD, ADD, or ADDRTR
DVNO-INVALID-OR-MISSING-8-USED	DVNO<1 or DVNO>8
UPDT=YES-INVALID-IGNORED	UPDT YES is valid only when IORT=0 or IORT=ADDRTR
TYPE=RANDOM-INVALID-IGNORED	TYPE RANDOM is only valid for IORT=0 or IORT=ADDRTR
RCFM-INVALID-UNBLK-ASSUMED	RCFM not equal to BLK
CHCK-INVALID-WRT-CHECK-ASSUMED	CHECK not equal to NO
FLID-MISSING-ZERO-USED	FLID=0
CRDT-MISSING-ZERO-USED	CRDT=0
GENO-MISSING-ZERO-USED	GENO=0
XPDT-MISSING-ZERO-USED	XPDT=0 and IORT=LOAD
RCSZ-INVALID-(155-KLEN)-USED	RCSZ<KLEN or RCSZ>(155-KLEN)
KLOC-INVALID-ZERO-USED	KLOC MISSING or >RCSZ-KLEN
KLEN-INVALID-OR-MISSING-40-USED	KLEN MISSING or KLEN>40
ERRO-MISSING-ZERO-USED	ERRO=0
WORK-MISSING-ZERO-USED	WORK=0 and TYPE≠"SEQUENTIAL"
WORK-INVALID-IGNORED	WORK≠0 and TYPE="SEQUENTIAL"
KARG-MISSING-ZERO-USED***	IORT≠LOAD and KARG=0
DTAR-INVALID-IGNORED	IORT≠LOAD and DTAR≠0
SQCH-INVALID-IGNORED	IORT≠LOAD and SQCH≠0
RTRV-MISSING-ZERO-USED***	IORT=0 or IORT=ADDRTR and RTRV=0
EOFA-MISSING-ZERO-USED	EOFA=0 and TYPE="SEQUENTIAL"
EOFA-INVALID-IGNORED	EOFA≠0 and TYPE≠"SEQUENTIAL"
ADAR-INVALID-IGNORED	ADAR≠0 and IORT≠(ADD or ADDRTR)
DUPR-INVALID-IGNORED	DUPR≠0 and IORT≠(ADD or ADDRTR)
RTRV-INVALID-IGNORED	RTRV≠0 and IORT≠(0 or ADDRTR)
IOA1-MISSING-ZERO-USED	IOA1=0 and IORT=(READ or ADDRTR)
IOA1-INVALID-IGNORED	IOA1=0 and IORT=(ADD or LOAD)
IOA2-MISSING-ZERO-USED	IOA2=0 and IORT=(ADD, ADDRTR, or LOAD*)
IOA2-INVALID-IGNORED	IOA2≠0 and IORT=(READ or LOAD**)
PROC-INVALID-9200-USED	PROC≠9200 and PROC≠9300
LDTM-INVALID-HSPD-LOAD-NOT-USED	LDTM≠0 and LDTM≠HSPD and IORT=LOAD
LDTM-INVALID-IGNORED	LDTM≠0 and IORT≠LOAD
DTAR-MISSING-ZERO-USED	IORT=LOAD and DTAR=0
SQCH-MISSING-ZERO-USED	IORT=LOAD and SQCH=0
KARG-INVALID-IGNORED	IORT=LOAD and KARG≠0
ADAR-MISSING-ZERO-USED	ADAR=0 and IORT=(ADD or ADDRTR)
DUPR-MISSING-ZERO-USED	DUPR=0 and IORT=(ADD or ADDRTR)
XPDT-INVALID-IGNORED	XPDT≠0 and IORT≠LOAD

*LDTM≠ HSPD
**LDTM≠ HSPD
***This message appears whenever the indicated parameter is missing for random and sequential retrieval files.
For a sequential file, the parameter is only required when the user intends to issue the SETL imperative macro instruction. If SETL is not to be used, the message will be produced, but should be ignored; the generated DTFIA file table will yield valid results for this case.

Table 4-7. DTFIA Error Messages

4.6. IMPERATIVE MACRO INSTRUCTIONS

The imperative macro instructions supplied for ISAM are OPEN, WRITE NEWKEY, READ KEY, WRITE KEY, SETL, GET, PUT, and CLOSE. These macro instructions are used by the problem program to open, process, and close files. The filename of an imperative macro instruction may have a maximum of eight characters.

4.6.1. OPEN Macro Instruction

This instruction must be used to initialize the file before any other imperative macro instruction can be performed. It is also used for label checking.

The format of the OPEN macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	OPEN	filename

- Positional Parameter 1

  filename – is the label of the corresponding DTFIA declarative macro instruction in the program defining the file to be opened.

Operational Conditions:

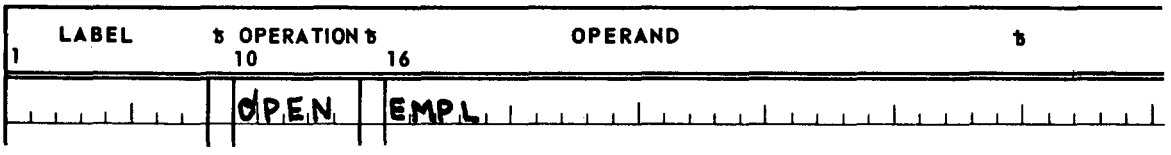
In both reading operations and writing operations, the file-ID in each volume of a file is compared to the file-ID indicated by FLID in the DTFIA macro, and the volume number is compared to that specified in the DEV parameter. If either of the compared values do not match, a display is produced indicating an error. The operator then has the option of either mounting another volume or canceling the job.

Before loading data into an area on disc containing data, the IOCS performs a comparison between the expiration date in the label on disc and the creation date indicated by the CRDT parameter in the DTFIA macro. If the expiration date is greater than the creation date, an error display is produced. The operator can then either mount another volume or cancel the job.

In loading data into an area on disc defined by the VTOC but not previously loaded, the creation date (CRDT), generation number (GENO), and the expiration date (XPDT) are written into the disc file label.

In read and add operations, the generation number (GENO) and creation date (CRDT) indicated by parameters in the DTFIA macro are compared to the corresponding information in the label. If any one of the values do not match, an error display is produced. The user can ignore the error, mount another volume, or cancel the job.

Example:



4.6.2. WRITE NEWKEY Macro Instruction

This instruction inserts a new record into an existing file.

The format of the WRITE NEWKEY macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	WRITE	filename,NEWKEY

- Positional Parameter 1

  filename – is the label of the DTFIA declarative macro instruction defining the file to which the record is to be written.

- Positional Parameter 2

  NEWKEY – indicates that a new record is to be written in an ISAM file.

Operational Conditions

The record to be written must be moved to the work area specified by the WORK= label parameter before execution of the WRITE.

For an add operation, the user must transfer the key to the key area specified by the KARG=label parameter before execution of the WRITE.

The key must be followed by the sentinel byte consisting of all one bits (FF_{16}). Example:

```
LABEL      ↹ OPERATION ↹      OPERAND                    ↹
1                 10          16
           WRITE    EMPL,NEWKEY
```

4.6.3. READ KEY Macro Instruction

This instruction initiates the retrieval of a single logical record from an ISAM file for the problem program during random retrieval of records.

The format of the READ KEY macro instruction is:

LABEL	↹ OPERATION ↹	OPERAND
[name]	READ	filename,KEY

- Positional Parameter 1

  filename – is the label of the DTFIA declarative macro instruction defining the file from which the record is to be read.

- Positional Parameter 2

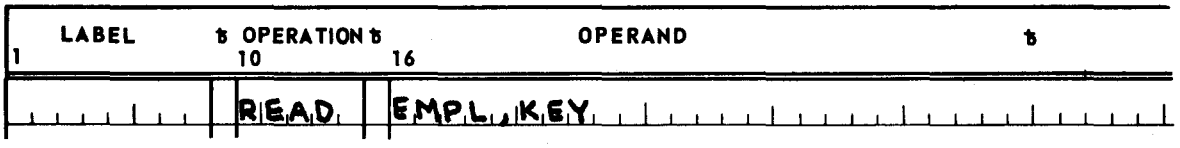
  KEY – indicates that a random retrieval (by key) is to be performed.

Operational Conditions:

The key of the record to be retrieved must be placed in the location specified by the KARG= label parameter before execution of the READ.

The key must be followed by a sentinel byte consisting of all ones (FF_{16}).

Example:



4.6.4. WRITE KEY Macro Instruction

This instruction initiates the rewriting (updating) of the last record retrieved with a READ KEY macro instruction during nonsequential file processing.

The format of the WRITE KEY macro instruction is:

LABEL	OPERATION	OPERAND
[name]	WRITE	filename,KEY

- Positional Parameter 1

  filename – is the label of the DTFIA declarative macro instruction defining the file to which the record is to be written.

- Positional Parameter 2

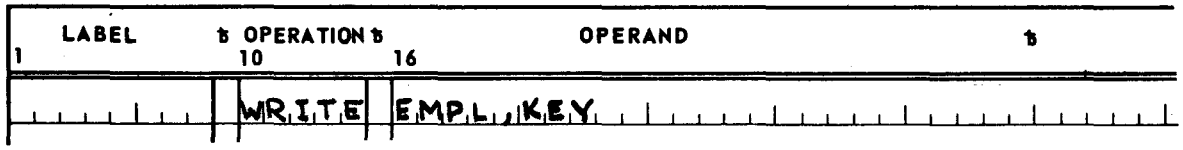
  KEY – indicates that the last record retrieved by a READ KEY macro instruction is to be rewritten in the file.

Operational Conditions:

The record must be in the work area specified by the WORK= label parameter before execution of the WRITE.

The key in the record must not be altered.

Example:



4.6.5. SETL Macro Instruction

This instruction directs the processor to begin a sequential retrieval with a specific record which is not necessarily the first record in the file. The SETL macro instruction must be issued before and after the WRITE NEWKEY macro instruction if the user is both sequentially retrieving and adding to a file.

The format of the SETL macro instruction is:

LABEL	OPERATION	OPERAND
[name]	SETL	filename,KEY

- Positional Parameter 1

  filename – is the label of the DTFIA declarative macro instruction defining the file from which records are to be read.

- Positional Parameter 2

  KEY – indicates that the area equated to the KARG=label keyword parameter holds the key of the first logical record to be retrieved.

Operational Conditions:

The key field of the first record to be processed must be transferred to the key area (KARG=label) before the SETL macro is executed. The key must be followed by a sentinel byte consisting of all ones (FF_{16}).

Example:

1 LABEL	OPERATION 10	16 OPERAND
	SETL	EMPL,KEY

4.6.6. GET Macro Instruction

This instruction retrieves the next logical record in sequence from a disc file during sequential retrieval of records.

The format of the GET macro instruction is:

LABEL	OPERATION	OPERAND
[name]	GET	filename,workarea

- Positional Parameter 1

  filename – is the label of the DTFIA declarative macro instruction defining the file from which the record is to be read.

- Positional Parameter 2

  workarea – is the label of the first byte of a work area defined by the user.

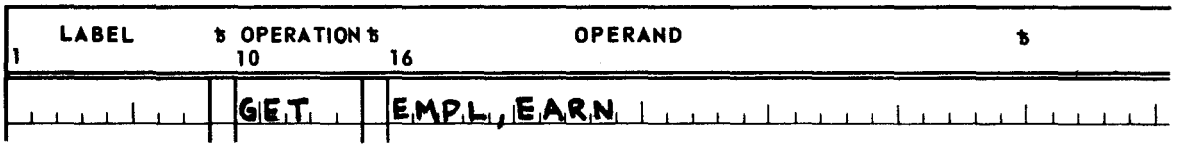
Operational Conditions:

For updating, the I/O instruction following a GET macro must be a PUT macro instruction.

The execution of a GET macro delivers one record into the work area defined in the problem program.

The first record delivered to the work area contains the lowest key in the file. If a SETL macro instruction is executed before a GET macro instruction, the record containing the key specified in the SETL macro instruction is delivered to the work area.

Example:



4.6.7. PUT Macro Instruction

This instruction indicates that the last record retrieved by a GET macro instruction has been updated and is to be rewritten on the direct access storage device. The PUT macro instruction must be part of a valid retrieval sequence initiated by a SETL macro instruction, and must follow a GET macro instruction. Updating, in a sequential retrieval operation, takes place only through the execution of the PUT macro instruction; if a record retrieved by a GET instruction is not to be updated, there is no need to execute a PUT instruction.

The format of the PUT macro instruction is:

LABEL	OPERATION	OPERAND
[name]	PUT	filename,workarea

- Positional Parameter 1

  filename – is a label of the DTFIA declarative macro instruction defining the file to which the record is to be written.

- Positional Parameter 2

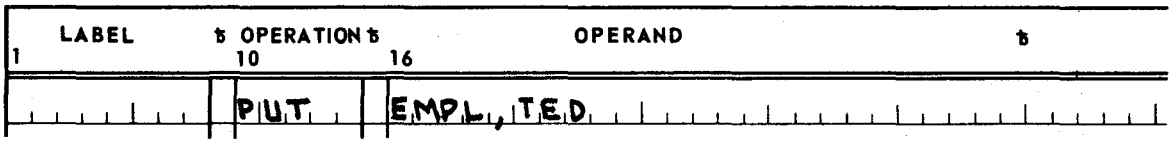
  workarea – is the label of the first byte of a work area defined by the user.

Operational Conditions:

The key in the updated record must not be changed.

The I/O instruction preceding a PUT macro instruction must be a GET macro.

Example:



4.6.8. CLOSE Macro Instruction

This instruction must be used to terminate the processing of the file. Once the file is closed, no other macro instruction may be executed for the file until it is re-opened by the OPEN macro instruction.

The format of the CLOSE macro instruction is:

LABEL	OPERATION	OPERAND
[name]	CLOSE	filename

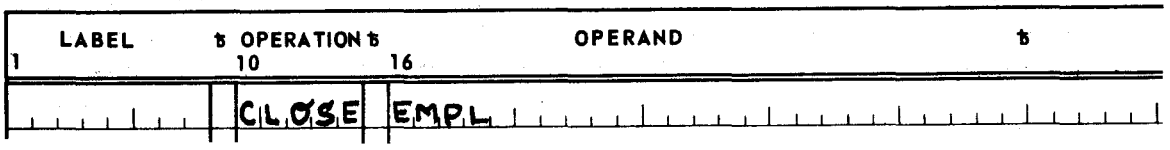
- Positional Parameter 1

  filename – is the label of the corresponding DTFIA declarative macro instruction in the program defining the file to be closed.

Operational Condition:

A CLOSE macro instruction must be executed before program control is returned to the operating system by means of the EOJ.

Example:



4.7. SOURCE CODE LINK INSTRUCTIONS

The DTFIA macro instruction generates a series of Branch and Link instructions and a file description table defining the file and serving as a link between the program and the I/O routines of ISAM.

The Branch and Link instructions generated by the DTFIA macro instruction take the form:

```
BAL     15,I?A4     OPEN
BAL     15,I?A5     CLOSE
BAL     15,I?A0     GET,READ
BAL     15,I?A1     WRITE NEWKEY
BAL     15,I?A2     PUT,WRITE
BAL     15,I?A3     SETL
```

The imperative macro instructions produce the following source code:

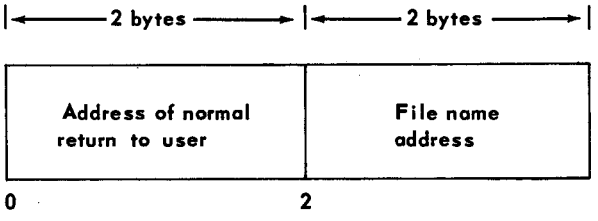
OPEN	filename	BAL	14,filename
CLOSE	filename	BAL	14,filename+4
GET	filename, workarea	BAL	14,filename+8
		DC	Y(workarea)
PUT	filename,workarea	BAL	14,filename+16
		DC	Y(workarea)
READ	filename,KEY	BAL	14,filename+8
WRITE	filename,KEY	BAL	14,filename+16
WRITE	filename,NEWKEY	BAL	14,filename+12
SETL	filename,KEY	BAL	14,filename+20

Register 14 contains the return address to the problem program.

4.8. EXIT HANDLING

A status block is created when a contingency condition transfers control to a user label (DTAR, DUPR, SQCH, RTRV, EOFA, ADAR). The status block allows the programmer to identify the error and continue processing if desired.

The file status block contains the address of the file for which the condition was encountered, and the address of the normal return to the user, had the condition not occurred. The IOCS does not assume any responsibility for the recovery from any of these conditions; it is therefore possible for the user to take any action he deems appropriate. For example, if control is transferred to the user ADAR label, there is no way the IOCS can enlarge the overflow area; however, the user may continue to issue macro commands to the IOCS. The address of the file status block is passed to the user in general register 15. The file status block is four bytes and consists of the following:

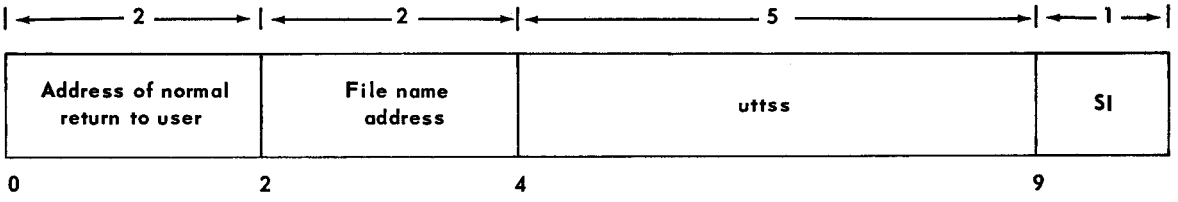


If the user wishes to return to the normal point of processing (instruction following the one for which the condition occurred), it is suggested that he load register 14 with the first two bytes (0–1) of the file status block and branch by way of register 14 [BC 15,0(,14)].

4.8.1. ERRO Exit Handling

Whenever the user enters the IOCS, general registers 8 through 13 are saved, and restored for him when the IOCS returns to the user. In the case of the transfer to the ERRO label, the user may use any registers that he chooses, and his original register contents will not be affected. However, the IOCS does not load the registers with their original values prior to the transfer to the ERRO label.

The processor creates a special file status block before transferring to the label specified by the ERRO parameter. The first four bytes of this file status block are as defined in 4.8. The other six bytes of this file status block contain information which readily identifies the cause and location of the I/O error. Bytes 4 through 8 contain the sector address where the error occurred, byte 9 contains the status indicator as found in the packet as passed to IOCS by the disc dispatcher. The address of this file status block is passed to the user in register 15.



where:

u is the physical unit number of the disc.

tt is the track number.

ss is the sector number.

SI is the status indicator, a one-byte indicator created by the disc dispatcher to describe the specific type of I/O error.

It is important to note that all I/O errors correspond to the handling of sectors, not necessarily to the handling of a user record. It is, therefore, impossible to determine which record or records are in error when the error occurs with respect to an output macro command (WRITE NEWKEY, WRITE KEY, PUT). This information may be obtained when the user attempts to read his file. IOCS assumes no responsibility for the corrections of the error, but does allow the user the capability of keeping track of which sectors are bad and which records are unreadable (input macro commands). It is further expected that the user will, at a later time, copy his file to a new disc unit, and perhaps reconstruct his file at that time.

Because of the nature of the processing techniques employed, it is necessary for the IOCS to regain control after transferring to the user ERRO label. This allows the IOCS to complete the specific cycle it was processing, and to reset itself for additional entries. IOCS transfers control to the user ERRO label by a: BAL 14, label. It is mandatory for the user to return control to IOCS by way of a branch on register 14[BC,15,0(,14)] if he desires to continue processing. The IOCS restricts any access to itself (by way of any imperative macro, for any file) until this return is made. Should the user attempt to issue another imperative macro to IOCS before returning control from his ERRO routine, the job will be cancelled without that particular file being closed. If the user does not return to the IOCS, he may cancel the job, or continue processing without accessing IOCS again. However, any processing that added records to his files in this job, would be wasted. These records would be lost because it is impossible to read any records written to a file unless the file is first closed, and then reopened. It is also possible that some records which have been updated (PUT macro instruction) may be lost due to the buffering scheme for blocked records. It is the function of CLOSE to assume that any non-empty buffers are written to the disc, and it is the function of both OPEN and CLOSE to update the end-of-file indicator which the IOCS employs in order to determine to which sector the file extends.

4.9. STORAGE REQUIREMENTS AND TIMING CONSIDERATIONS

Coding generated by the DISIO and/or DISLD macro instructions require the number of storage locations shown below:

Routine Type	Number of Bytes
ADD	5048
LOAD-YES	3234
HSPD-YES	4578
READ-SEQ	2492
READ-RANDOM	2666
* ADD-READ-LOAD(HSPD)	8000
* ADD-READ-LOAD(NORM)	5724
** UPDT	152
***SETL	406

NOTES:

*Read includes Random, Sequential-SETL.

**This must be added to requirement for any READ case.

***This must be added to requirement for any SEQUENTIAL READ case.

For handling of blocked records, add approximately 10% more bytes.

The following formulas may be used to determine the number of bytes required in the DTFIA file table for each type of file supported.

N = number of volumes, $1 \leq N \leq 8$

Y = 12 – if 9300 processed

18 – if 9200 processed

K = key length, $1 \leq K \leq 40$

R = record size $1 \leq R \leq (155-K)$

S = block size = $K+10+nR$, $12 \leq S \leq 165$

where n = # records/block

X = number of bytes required

1. LOAD-YES

   $X = 290+8N=Y(2K+S+165)+S$

2. HSPD-YES

   $X = 109+8N$

3. READ

   $X = 98+N(K+3)$

4. ADD

   $X = 120+N(K+28)$

5. ADDRTR

   $X = 138+N(K+28)$

The following information provides a guide for estimating the time used by IOCS. Times are specified in milliseconds and are listed for the UNIVAC 9300 System (for a 9200 System, times should be doubled).

These times, given in Table 4-8, include only the instruction time used by the IOCS. They do not include dispatcher time or physical I/O time. Record advance refers to the amount of time used by IOCS if the block desired is already in memory. Block advance refers to the amount of time used by IOCS to locate the block and manipulate the record in the block.

MACRO-MODE	EXECUTION TIME	
OPEN-ADD	8.1ms/vol	
OPEN-READ	7.2ms/vol	
OPEN-LOAD	7.0ms/vol	
CLOSE-ADD	(57.8+3.7/vol)ms	
CLOSE-READ	5.6ms	
CLOSE-LOAD-YES	(55.5+3.7/vol)ms	
CLOSE-HSPD-YES	(128.5+3.7/vol)ms	
	RECORD ADVANCE	BLOCK ADVANCE
SETL	–	16.8ms *
READ	–	16.8ms *
WRITE	–	5.7ms
GET	4.2ms	6.3ms
PUT	4.4ms	5.9ms
WRITE NEWKEY HSPD-YES	4.1ms	6.2ms
LOAD-YES	4.3ms	9.9ms
ADD-EXTEND	6.2ms	11.4ms
ADD-INSERT	23.0ms *	26.7ms *

*This is the amount of time required to perform the function to the first element in the chain. (For example, time for READ is amount of time to locate the first record in the prime data block only. Additional time may be needed if the desired record is in another position in the block or in the overflow area.)

Table 4-8. Timing Specifications

4.10. RESTART INFORMATION

ISAM provides the problem program with access to a restart control word generated by DTFIA containing address information necessary for a restart procedure.

ISAM provides the following four-byte restart control word at filename-4:

FILENAME-4 Logical unit number pointer. This value must be subtracted from the address of filename to obtain the address of the Logical Unit Table.

FILENAME-3 Table pointer. This value must be added to the address of filename to obtain the address of the file information.

The label information is formatted as follows:

filename + DL : address of file identification (FLID)

filename + DL + 2 : address of file creation date (CRDT)

filename + DL + 4 : address of file generation number (GENO)

filename + DL + 6 : address of file expiration date (XPDT)

FILENAME-2 Access method. Indicates that the file is an indexed sequential file.

I (X'C8')

FILENAME-1 File Status

X'OO' – File Open

X'FF' – File Closed

The file status indicator is updated at open and close time without regard to the restart specification.

The layout of label information is as follows:

FILENAME + LABEL POINTER

FLID	ADDRESS	CRDT	ADDRESS
GENO	ADDRESS		

The layout of the restart control word is as follows:

LU # POINTER	LABEL POINTER	ACCESS METHOD	FILE STATUS

Note the negative displacement from the filename. At location 'Filename-14' a list of logical unit numbers, one byte per number, will be found. This list indicates the logical unit numbers of all volumes used for the file. The first byte begins on a half-word boundary and the list always contains an even number of bytes (one byte of zero may be needed). The end of the list is indicated by a byte containing X'FF'.

UP-7639 Rev. 1