Use DAINIT to initialize a new
D write data set, otherwise it appears as empty.

IBM SYSTEM/360

GENERAL I/O PACKAGE

MARCH 1973

Prepared By

Charles I. Dickman

National Aeronautics and Space Administration
Goddard Space Flight Center
Laboratory for High Energy Astrophysics
Data Management and Programming Office
Code 664
Greenbelt, Maryland 20771

Abstract

This document describes a set of general I/O routines, for the IBM System/360 computers, which were originally developed by Alan R. Thompson of the Federal Systems Division of the IBM Corporation under contract number NAS5-10022 to the Goddard Space Flight Center. Extensive modification of the original package has warranted the publication of new documentation and this document obsoletes any and all previously issued documentation intended to describe this package.

These routines are intended to replace FORTRAN I/O routines in situations where the FORTRAN routines are too inefficient or restrictive. Although intended for use by FORTRAN programmers, these routines may also be called from PL/1 or ALC programs.

Introduction

In the process of writing data reduction or data analysis programs for the IBM/360 computer, it often becomes necessary or desirable to write specialized assembly language I/O routines because FORTRAN or PL/1 I/O is either too restrictive or too inefficient. After much experience with specialized I/O routines, it was decided to develop a generalized 360 I/O package which could be used in FORTRAN or PL/1 programs and which would provide a maximum of efficiency with a minimum of restrictions. The routines written for this package are divided in two groups, the FTIO routines for sequential I/O and the DAIO routines for direct access I/O. While each set is a separate module (CSECT) and each may be used independently, they use the same default DD card naming scheme (the same as FORTRAN's) and the same unit table. The same unit number may, therefore, not be used for both sequential and direct I/O. Although FTIO checks for internal consistency, the myriad of options available to the user allows for situations in which FTIO may not be able to discern when restrictions are being violated. Results occurring from violations of restrictions as set forth herein which can not be trapped by FTIO are generally unpredictable and, therefore, the user should be careful to heed the documented restrictions.

Fortran I/O Package (FTIO)

When FORTRAN unformatted READ and WRITE statements are used for I/O operations, a considerable amount of CPU time is spent collecting data items from the list and assembling them into data records. This time is wasted if the data is already contigous in core and can be eliminated by using the FORTRAN I/O package (FTIO) described here. In addition to saving time, there are several other advantages when using these routines. These advantages are:

1) Capability of reading and writing multiple file tapes without multiple DD cards.

2) Capability of using the read backward mode for tapes.

3) Record format is not restricted to variable or variable blocked.

4) It is not necessary to know the length of a record before it is read.

5) Program may select tape to be mounted.

6) Partitioned data sets may be read or written.

7) Data set name may be altered by the program.

8) Capability of processing several labeled tapes with different DCB attributes on the same unit.

9) Ability to override Fortran DD naming conventions.

10) Ability to read tapes BLP.

11) Prevention of wait state timeouts (completion code S522) due to failure of operator to mount the requested tape.

12) Automatic interrupt preprocessing to insure retention of all 16 general registers and the PROGRAM OLD PSW in user core upon the occurrence of system OCO-OCF interruptions while maintaining any user (eg. FORTRAN) error handling routines.

The FTIO routines use QSAM LOCATE mode for all I/O operations.

Before they can be used the following conditions must be satisfied:

1) The data to be contained in a record must be located con-

tiguously in core. (This can be done by placing it all in a common area or array, or by using the EQUIVALENCE statement.

2) Backspacing must not be needed.

If these conditions are satisfied, any FORTRAN unformatted READ or WRITE statement may be replaced with an appropriate call to an FTIO routine.

FTIO Routines:

1.	FREAD	-Reads a record
2.	FREADB	-Reads a record backwards
3.	FWRITE	-Writes a record
4.	POSN	-Positions to specified file
5.	REWIND	-Closes DCB (REREAD)
6.	UNLOAD	-Closes DCB (DISP)
7.	LEAVE	-Closes DCB (LEAVE)
8.	MOUNT	-Mounts specified tape
9.	MEMBER	-Selects a PDS member to be read or written

Detailed Parameter Description

A = Beginning of the area into which data is to be moved.

N = Fullword integer value corresponding to the FORTRAN logical unit number being addressed.

If a negative unit number is passed, its absolute value is used as the unit number and a special locate mode switch is set. When this switch is set, FREAD will not move the data into the users area. It will instead store the address of the data in the users area. The first parameter, A, passed in this case should be the full word in which the address will be stored. The record will be available in the buffer until FREAD is called

again. If this switch is set when FWRITE is called, the first parameter passed will be taken as a pointer to the data record. This type of call to FWRITE can be used in conjunction with a similar type call to FREAD to save movement of data in core. Records with record format VBS can not be read in locate mode. (ie. with a negative unit number).

DDN = Eight character DD name of the unit being addressed. If the high order byte of the unit parameter contains an EBCDIC alphabetic character, it is assumed that the user wishes to address a unit with the DDNAME specified in the unit parameter instead of following the FORTRAN naming conventions. In this case the DDNAME is passed instead of the FORTRAN unit number. The name is assumed to be 8 EBCDIC characters, left justified, and must be padded with blanks to 8 characters. It is not necessary, however, to place the DDNAME on a doubleword boundary.

L = Fullword location in which the length in bytes of the record read or to be written is returned to the caller or passed to FWRITE respectively.

LC = Locate mode switch under user DD name format. If a user DDNAME is passed, a fourth argument, LC, must be included in the call. This fullword integer variable is used to determine whether the data or the address of the data should be moved to the user's area as noted above. If LC is negative, FREAD functions as though a negative unit number were passed. If LC is positive or zero, the program functions as though a positive unit number were passed.

END = Statement number to which control is to be given when an end of file or end of volume is encountered.

ERR = Statement number to which control is to be given when a permanent I/O error is encountered.

IO = A fullword integer which contains a value corresponding to the next I/O operation to be performed on the subject unit as follows:

IO = 1; read forward

IO =-1; read forward BLP (valid only in a call to MOUNT)

IO = 2; write

IO = 3; read backward

IO =-3; read backward BLP (valid only in a call to MOUNT)

NF = A fullword integer which contains the file sequence number to be processed next on the subject unit. Only the low order halfword of this parameter is used but the parameter must be 4 bytes long and reside on a fullword boundary.

DSN = The starting location of an area in core which contains the fully qualified EBCDIC data set name of the data set next to be processed on the subject unit. This area must contain at least one non-blank, alphabetic, left justified character at the location passed. A maximum of 44 characters are allowed. The parameter NC must accompany DSN in all calls to an FTIO routine. No boundary alignment is assumed for DSN.

NC = A fullword integer which contains the number of characters in DSN beginning at DSN, one character per byte. NC must be greater than zero and less than 45. NC must accompany DSN in all calls to an FTIO routine.

DT = The starting location of an area in core which contains a six character tape volume identifier packed one EBCDIC character per byte, left justified, and padded with blanks to six characters if necessary. If more than six non-blank characters are present at the address passed, only the first six will be used. No boundary alignment is assumed for DT.

DM = The starting location of a core area containing an eight character member name packed one EBCDIC character per byte, left justified, and padded with blanks to eight characters if necessary. If more than eight non-blank characters are present at the address passed, only the first eight will be used. No boundary alignment is assumed for DM.

Description of Routines

1. Call FREAD (A, N, L, &END, &ERR)

   Call FREAD (A, DDN, L, LC, &END, &ERR)

   This routine reads a data record from the unit specified into contiguous locations starting with the first parameter. The length of the record read is returned in the third parameter. The first alternate return is used if the end-of-file is reached and the second if an I/O error is encountered.

   The first call format is used when FORTRAN DD naming conventions are being followed and the second format is used when the user desires to provide his own DD name.

2. Call FREADB (A, N, L, &END, &ERR)

   Call FREADB (A, DDN, L, LC, &END, &ERR)

   This routine reads a data record backwards from the logical unit specified, into contiguous locations starting with the first parameter. This routine can only be used for reading tapes with fixed length (blocked or unblocked) or undefined records. Disk data sets and data sets with variable length records may not be read backwards. The records read from tape using this routine are presented to the calling program in an order opposite to that in which they were written, (i.e. the last one written is the first read). The order in core of data from an individual record, however, is the same as it was when written. The data in a record is not reversed, but the order of the records is. The length of the record read is returned in the third parameter. The first alternate return is used if the beginning of file (or beginning of tape) is reached, and the second if an I/O error is encountered.

The first call format is used when FORTRAN DD naming conventions are being followed and the second format is used when the user desires to provide his own DD name.

3. Call FWRITE (A, N, L)

   Call FWRITE (A, DDN, L, LC)

This routine writes a record on the logical unit specified. The first parameter gives the address of the beginning of the record, and the third gives its length (variable and undefined records only). The third parameter is ignored if fixed length (F or FB) records are being written. OS/MVT will not allow user processing of the end or error conditions. If a permanent I/O error occurs during a write operation, the task issuing the operation will be abnormally terminated with a completion code of S001. If the system is able to sense an end of volume condition (i.e. by encountering the reflective strip) a scratch tape will be called for. If the reflective strip is missing, the system will write on the tape until it is pulled from the reel whereupon the next access to the unit will cause the issuing task to abend with a completion code of S001.

The first call format is used when FORTRAN DD naming conventions are being followed and the second format is used when the user desries to provide his own DD name.

4. Call POSN (IO, N, NF)

   Call POSN (IO, DDN, NF)

   Call POSN (IO, N, NF, NC, DSN)

   Call POSN (IO, DDN, NF, NC, DSN)

This routine positions the tape on the unit specified so that the next I/O operation on this unit will access the file of the data set specified. Care must be taken to avoid passing a file number which does not exist on the tape. For input operations there must be at least NF files on the tape. For output operations there must be at least NF-1 files on the tape. If this restriction is violated, a completion code of S613 will result. Care must also be exercised in passing a data set name which does not match that on the data set label of an input tape. If the DD card which pertains to the specified unit has DISP = (OLD, ----, ----) coded and the data set name on the data set label does not agree with that passed to POSN, a completion code of S813 will result. Any data set name passed to POSN which is defined as a NEW data set on the corresponding DD card will have the data set name written in the HDR1 record of the label contained on the data set. This allows one to define multi-file data sets of which more than one such data set may reside on the same tape volume. One may also define single file multi-volume data sets and multi-file single volume data sets. This routine should not be used for disk data sets.

The first call format is used when FORTRAN DD naming conventions are being followed and the data set name specified on the DD card is desired. The second call format is for use when specifying a user DD name and using the data set name as it appears on the DD card. The third format is used to specify a data set name which overrides that on the DD card where the DD name follows the FORTRAN DD naming conventions. The fourth call format is used to pass both a user

defined DD name and a data set name which overrides that on the DD card.

5. Call REWIND (N)

   Call REWIND (DDN)

   This routine positions the unit specified to the beginning of the data set (the physical end of a data set being read backwards). If the data set is being written, a file mark will be written and data set labels will be processed before rewinding. The first call format is used if FORTRAN DD naming conventions are being followed and the second is used when a non-FORTRAN DD name is required.

6. Call UNLOAD (N)

   Call UNLOAD (DDN)

   This routine logically disconnects the unit specified and frees all core associated with it. If the unit is used subsequently, core storage will be reallocated. If the data set is being written, a file mark will be written and data set labels will be processed before disconnecting the unit. The first call format is used if FORTRAN DD naming conventions are being followed and the second is used when a non-FORTRAN DD name is being used.

7. Call LEAVE (N)

   Call LEAVE (DDN)

   This routine positions the unit to the end of the data set (the physical beginning of a data set being read backwards). If the data set is being written a file mark is written and data set labels are processed before positioning. The first call format is for use when FORTRAN DD naming conventions are being followed and the second is for use with non-FORTRAN DD names.

8. Call MOUNT (IO, N, DT)

Call MOUNT (IO, DDN, DT)

Call MOUNT (IO, N, DT, NF)

Call MOUNT (IO, DDN, DT, NF)

Call MOUNT (IO, N, DT, NC, DSN)

Call MOUNT (IO, DDN, DT, NC, DSN)

Call MOUNT (IO, N, DT, NC, DSN, NF)

Call MOUNT (IO, DDN, DT, NC, DSN, NF)

This routine mounts the tape indicated on the specified unit. If NF is specified, the tape is positioned to the specified file. If NF is not specified, the file number last passed to POSN or MOUNT for the specified unit will be used. If POSN or MOUNT have not been called previously for the specified unit and NF is not coded, the file number specified in the LABEL parameter on the DD card for this unit will be used. If NC and DSN are coded, the data set name as specified on the DD card of the subject unit are overridden. The same precautions with respect to file number and data set name as were described for POSN apply to MOUNT as well.

Call formats 1, 3, 5 and 7 are used when FORTRAN DD naming conventions are being followed while formats 2, 4, 6 and 8 are used for non-FORTRAN DD names. Formats 1 and 2 are for specifying the volume serial number only. Call formats 3 and 4 are used to pass both volume number and file number while formats 5 and 6 are used to pass volume number and data set name. Formats 7 and 8 are used to pass volume number, data set name, and file number.

9. Call MEMBER (IO, N, DM)

   Call MEMBER (IO, DDN, DM)

   This routine is used to process partitioned data sets. When used for input, the member specified is read with subsequent calls to FREAD. If the member requested does not exist, a call to MEMBER will result in a completion code of SF13. If MEMBER is used for output, subsequent calls to FWRITE create a member with the specified name. The member thus created must not already exist within the data set.

DD Card Considerations

The FORTRAN convention for naming DD cards means that if logical unit number xx is used, a DD card named FTxxF001 must be supplied. The following points should be kept in mind when specifying DD card parameters:

1. DSNAME:

   a) Must not be a temporary name if more than one tape is to be processed on this unit by calling MOUNT.

   b) Must specify a dummy member name if MEMBER is used to process a PDS.

2. Volume Serial Number:

   Even if MOUNT will be used to specify the proper volume serial number, a dummy number should be specified on the DD card.

3. UNIT:

   Must indicate deferred mounting if MOUNT is called to specify volume serial number.

4. DISP:

   Must specify KEEP if multiple tape volumes are to be processed using MOUNT.

5. LABEL:

   If POSN is used to specify file sequence number, the sequence number specified in this parameter is ignored. If POSN is not used, this number is.

6. DCB:

   a) RECFM - If not specified and not contained in data set labels will default to U. If V, VB, or VBS is specified, the appropriate control words will be attached by FWRITE and stripped off by FREAD.

   b) LRECL - Required only for FB records. Ignored if RECFM ≠ FB.

   c) BLKSIZE - If not specified and not contained in data set labels, defaults to 32767. If VB or VBS records are being used, the BLKSIZE should be at least as big as the largest logical record plus 4.

   d) RECFM - FTIO has EROPT = ACC hard coded; the user can not modify this parameter.

I/O Errors

If a permanent I/O error is detected during a read operation, certain information about the error is made available in a common area called FERMSG before the alternate return is taken. This information can be printed or used by the user if desired. The common area contains a full word array, 26 words long.

The contents of this array is as follows:

Word	Contents
1	Buffer address
2	Number of bytes read
3-22	EBCDIC message for printing
23	IOBFLAG1, IOBFLAG2, Sense bytes 0 and 1
24	Sense bytes 2, 3, 4 and 5
25	ECB completion code and CCW address from the CSW
26	Last 4 bytes of the CSW

To make the information available in the users program, the following common statement may be used:

```
COMMON/FERMSG/IMES(26)
```

The message can then be written when an I/O error occurs with the following write statement:

```
WRITE (6,1000)IMES
1000 FORMAT (1X, Z8, I6, 20A4, 4(1X,Z8))
```

A message written in this manner will contain the following items, separated by blanks and/or commas:

1. Buffer address (hex)
2. Number of bytes (decimal)
3. Jobname
4. Volume serial number (from UCB)
5. Unit address
6. Device type
7. DD name
8. Operation attempted

9. Error description

10. Tape: block number (decimal)

    Direct access: track address BBCCHHR (hex)

    Unit Record: asterisks

11. Access method

12. IOBFLAG1, IOBFLAG2, UCB sense bytes 0 and 1.

13. Sense bytes 2-5 from the UCB. For a description of these, see the component description manual for the I/O device.

14. ECB completion code and Channel Status Word (CSW) command address portion. See System Control blocks for the ECB code.

15. Last 4 bytes of CSW (status and residual count field.) See the System/360 reference card (green card) for the format of the CSW. The residual count indicates how many bytes were remaining to be read or written when the operation was terminated.

If a permanent I/O error is detected during a write operation, the same information is placed in the common area. The system, however, will abend the program with a completion code of 001 before control is returned to the user.

User Abends

U201 = Fortran logical unit number greater than 50 or equal to zero.

U210 = Illegal first parameter (IO) in a call to POSN, MOUNT, or MEMBER.

U202 = Subject unit is currently being used by DAIO for direct access.

U220 = Invalid record length detected.

U203 = Length of data set name is less than or equal to zero or greater than 44.

U230 = A DD card is missing for the subject unit.

U204 = An attempt was made to open a tape for output with BLP specified.

U205 = An attempt was made to read a VBS record in locate mode.

Programming Description

1. FREAD

   a) Gets core and sets up the DCB if this is the first use of the unit, or if UNLOAD was called previously for this unit.

   b) If DCB is open for OUTPUT or RDBACK, it is closed (REREAD).

   c) If DCB is closed it is opened (INPUT).

   d) Record is read using locate mode GET.

   e) If EODAD exit is taken, the program returns with a 4 in GR15. (First alternate return)

   f) If record format is variable (V, VB, or VBS) LRECL is obtained from the control word and the data portion of the record is moved to the user's area.

   g) If record format is not variable (F, FB, or U), LRECL is obtained from the DCB and the record is moved to the user's area.

   h) If no error was encountered reading the record the program returns with 0 in GR15. (normal return)

   i) If an I/O error was encountered, the program returns with an 8 in GR15. (second alternate return)

2. FREADB

Processing in this routine is the same as in FREAD except that in step b), if DCB is open for OUTPUT or INPUT it is closed (LEAVE). And in step c) DCB is opened (RDBACK).

3. FWRITE

a) Gets core and sets up the DCB if this is the first use of the unit, or if UNLOAD was called previously for this unit.

b) If the DCB is open for INPUT or RDBACK it is closed (LEAVE).

c) If the DCB is closed it is opened (OUTPUT).

d) If the record format is not fixed or fixed blocked, the DCBLRECL field is set from the length parameter passed.

e) A locate mode PUT is used to obtain an output buffer.

f) If the record format is fixed, fixed blocked or undefined, the data is moved into the buffer and the routine returns.

g) If the record format is V or VB, a control word is built at the beginning of the buffer area and the data is moved in following this word. The routine then returns.

h) If the record format is VBS, the record is segmented to fit in the block size specified by the user on the DD card, control words are attached and the data is moved to the buffer area. The routine then returns.

4. POSN

a) Gets core and sets up the DCB if this is the first use of the unit, or if UNLOAD was called previously for this unit.

b) If the DCB is open, it is closed (LEAVE)

c) The Job File Control Block for this unit is read.

d) The file number passed is placed in the JFCB.

e) The DCB is opened for INPUT, OUTPUT, or RDBACK, depending on the first parameter.

f) The routine returns.

5. REWIND

   a) If the DCB is open, it is closed (REREAD).

   b) Core storage occupied by buffers is freed.

6. UNLOAD

   a) If the DCB has not been acquired, the routine returns immediately.

   b) If the DCB is closed, it is opened for INPUT.

   c) The DCB is closed (DISP).

   d) The storage occupied by the DCB, the JFCB, and buffers is released and the routine returns.

7. LEAVE

   a) If the DCB is open, it is closed (LEAVE).

   b) Core storage occupied by buffers is freed.

8. MOUNT

   a) Gets core and sets up DCB if this is the first use of the unit, or if UNLOAD was called previously for this unit.

   b) If DCB is open, it is closed (DISP).

   c) The JFCB is read.

   d) The volume serial number is moved to the JFCB.

   e) The file number (if specified) is moved to the JFCB.

   f) The DCB is opened for INPUT, OUTPUT, or RDBACK depending on the first parameter.

   g) The routine returns.

9. MEMBER

   a) Gets core and sets up DCB if this is the first use of the unit or if UNLOAD was called previously for this unit.

   b) If the DCB is open, it is closed (DISP).

c) The JFCB is read.

d) The member name is moved to the JFCB.

e) The DCB is opened for INPUT, OUTPUT, or RDBACK depending on the first parameter.

10. Prevention of S522 Abends

Before an OPEN SVC is issued anywhere within FTIO, the following events occur:

a) If this is the first OPEN issued by the task, the jobname is retrieved from the TIOT and saved.

b) From the JFCB for the unit about to be opened, the serial number of the volume expected to be on the device at the completion of OPEN is extracted.

c) The unit is checked to determine if DD DUMMY has been coded on the DD card pertaining to that unit and, if so, a flag is set so that FWRITE will not attempt to move data into a non-existent buffer.

d) If the unit name for this unit has not yet been retrieved, it is obtained from the UCB and saved.

e) If the unit is not a tape drive, a return is effected and the OPEN is issued.

f) If the unit is a tape drive and the volume serial number of the tape expected to be mounted on the unit after OPEN is different from the volume serial number of the tape which is currently mounted on the unit, a real time interval timer is started.

g) A return is effected and the OPEN is issued.

If the interval timer expires before the OPEN has been accomplished,

the following events occur:

a) If this is not the first timer expiration for the specific OPEN, the message sent to the operator's scope is deleted.

b) From previously retrieved information, a message is sent to the operator's scope, the tape library console, the system log, and the programmer. This message has the following form;

JJJJJJJJ: - FTIO - YY/DDD @ HH:MM:SS.S -

PLEASE MOUNT TAPE # TTTTTT ON DRIVE UUU.

where YY = year

DDD = day of year

HH = hour

MM = minute

SS.S = seconds

TTTTTT = tape volume serial number

UUU = unit name

JJJJJJJJ = jobname

c) The message identifier is saved and a new interval timer is started. As long as a message is sent at least once every 15 minutes, the job will not get a S522 abend while waiting for a tape mount.

If the OPEN is accomplished before an interval timer expires, the following events occur:

a) The outstanding interval timer and exit routine are cancelled.

b) Any message sent to the operator's scope is deleted.

c) Normal processing continues.

11. Preprocessing of System Interrupts OCO-OCF

a) Upon the first entry to an FTIO control section, a SPIE SVC is issued to inform the system that FTIO wishes to process all OCO-OCF interrupts.

b) The pointer to any previously issued SPIE is saved and normal processing continues.

Upon the occurrence of an interrupt, the following events occur:

a) A valid copy of the 16 general registers and the program old PSW are made.

b) If no SPIE other than the one issued by FTIO was specified for the task, the task is abnormally terminated (see steg g).

c) If this type of interrupt is maskable and the previously issued SPIE chose to ignore it, a return is effected and normal processing continues.

d) If this type of interrupt is not maskable, or the interrupt is maskable and the previously issued SPIE chose to process it, the recovery procedure address is determined.

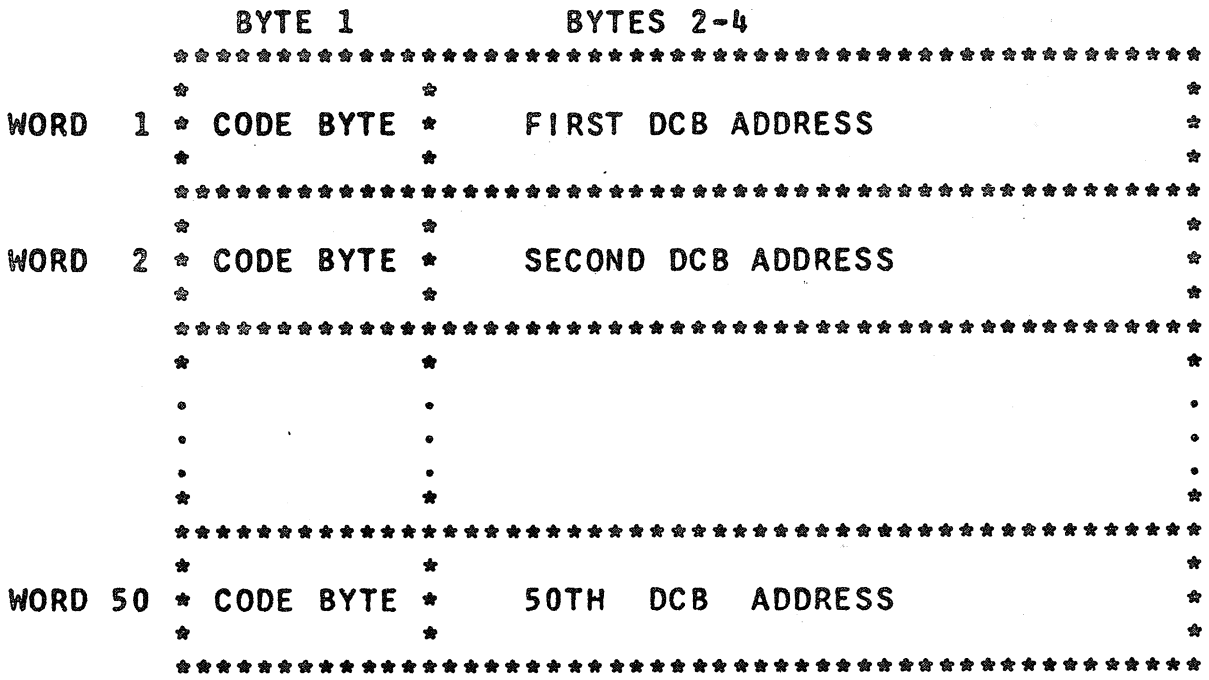
e) If such a recovery routine exists, control is passed to it for interrupt processing. When the recovery routine completes, a return is effected and normal processing continues.

f) If such a recovery routine does not exist, the task is abnormally terminated (see step g).

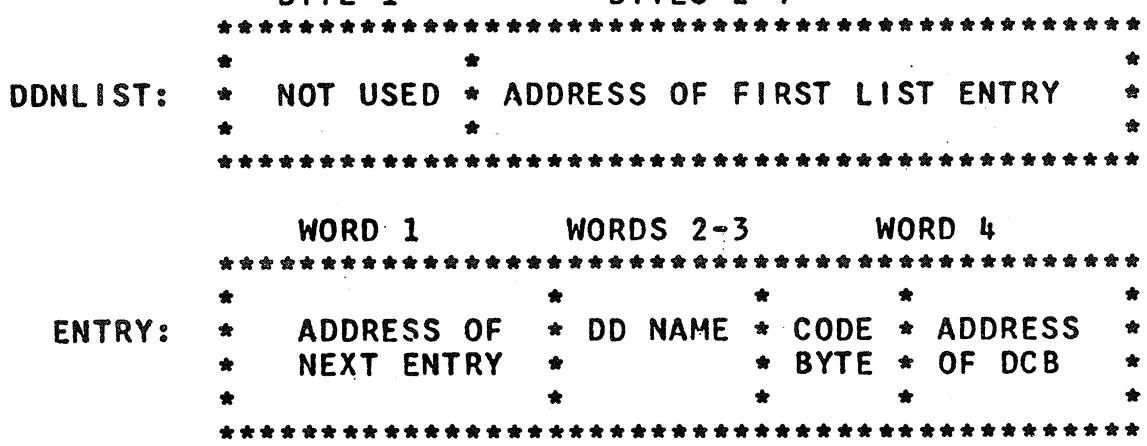
g) When it is determined that the task should be abnormally terminated (steps b and f) and either a SYSUDUMP or a SYSABEND DD card is supplied for the step, FTIO formats the interrupt type, the program old PSW, and the contents of the

16 general registers as they were when the interrupt occurred. This information is printed on the dump data set before the dump formatted by the system. After this information is printed, an ABEND SVC is issued with the appropriate completion code. If neither the SYSUDUMP nor the SYSABEND DD card is available for the step, the ABEND is issued without formatting the aforementioned information for output.

FORTRAN UNIT TABLE FORMAT



DD NAME LIST FORMAT



WHEN THE LIST IS EMPTY, DDNLIST POINTS TO ITSELF

THE LAST LIST ENTRY POINTS TO DDNLIST

THE CODE BYTE IS SET AS FOLLOWS :

NUMERIC = 0- STORAGE FOR DCB NOT ACQUIRED

NUMERIC = 1- DCB IS OPEN FOR INPUT

NUMERIC = 2- DCB IS OPEN FOR OUTPUT

NUMERIC = 4- DCB IS OPEN FOR READING BACKWARDS

NUMERIC = 8- DCB IS CLOSED

ZONE = 1- JFCB HAS BEEN READ FOR SUBJECT UNIT

ZONE = 2- UNIT DOES NOT FOLLOW FORTRAN NAMING CONVENTION

ZONE = 4- PREVIOUS VOLUME SERIAL NUMBER IS AVAILABLE FOR THIS UNIT

ZONE = 8- UNIT NAME IS AVAILABLE FOR THIS UNIT

CONTROL BLOCK ITEMS USED

```
ABS 16          = CVT ADDRESS
CVT + 0         = ADDRESS OF TCB ADDRESS
TCB + 12        = TIOT ADDRESS
UCB + 13        = UNIT NAME
UCB + 18        = UNIT TYPE
DCB + 36        = RECFM  (BITS 0 AND 1) 10=F,01=V,11=U
DCB + 40        = DDNAME
DCB + 48        = DCBOFLGS
DCB + 62        = BLKSIZE
DCB + 82        = LRECL
JFCB + 0        = DATA SET NAME
JFCB + 44       = MEMBER NAME  (FOR PDS PROCESSING)
JFCB + 68       = FILE SEQUENCE NUMBER
JFCB + 76       = JFCBFLGS  (MARK JFCB FOR RETURN TO JOB Q)
JFCB + 117      = NUMBER OF VOLUME SERIAL NUMBERS
JFCB + 118      = FIRST VOLUME SERIAL NUMBER
TIOT + 0        = JOBNAME
TIOT + 24       = FIRST DD ENTRY
DD ENTRY + 0    = LENGTH OF DD ENTRY
DD ENTRY + 4    = DDNAME
DD ENTRY + 16   = ADDRESS OF UCB
```

Direct Access I/O Package (DAIO)

The Direct Access I/O Package (DAIO) was written to provide programmers with a simple and efficient means of using direct access devices. It consists of two subroutines, one for reading and one for writing, which can be called from FORTRAN, PL/I and Assembly Language programs.

The direct access data set can be thought of as a series of sequentially numbered locations each of which can contain one fixed length data record. The number of locations and the size of the records is determined by the DD card that defines the data set. When the user writes a record into the data set, he specifies the location in which he wants the record written. Likewise when reading a record, he specifies the location he wants the record read from. The record most recently written in the specified location will be read. Records may be read and written in any order. It is the users responsibility to only read records that have previously been written.

The standard OS/360 access methods require that a direct access data set be initially created sequentially. This means that if records are missing when the data set is created, skeleton records (dummy records) must be written in place of them. FORTRAN direct access I/O routines write skeleton records for the entire data set when the first WRITE statement for a new data set is executed. The DAIO routines, however, were written so that skeleton records are written only where absolutely required. Skeleton records are required only when all the following conditions are met:

1) The data set is new

2) There is more than one record per track. (For a 2314 this means that the record size is less than 3521 bytes.)

3) Records are not created in sequential order.

When these three conditions are met, skeleton records are written for any records skipped while creating the data set. For example if record 3 is the first record written, skeleton records are written for records 1 and 2. If subsequently record 5 is written, a skeleton record will be written for record 4. If at a later time record 1, 2 or 4 is written, the actual record will replace the skeleton record. (A skeleton record consists of all binary zeros.)

CALL DCLOSE(NUNIT)

Description of Routines

1. CALL DWRITE (NUNIT, NRECNO, AREA)

   NUNIT - Logical unit number

   NRECNO = Location into which record is to be written

   AREA = Beginning of area from which data is to be written

   This routine writes a record on the logical unit specified. The second parameter gives the location into which the record is to go. This must be an integer between 1 and N where N is the maximum number of records in the data set. The record is taken from consecutive core locations starting with the third parameter. The size of the record is taken from the BLKSIZE parameter on the DD card. The record written will replace any record previously written into the same location.

2. CALL DREAD (NUNIT, NRECNO, AREA, & err)

   NUNIT = Logical unit number

   NRECNO = Location from which record is to be read

   AREA = Beginning of area into which data is to be read

err = Statement number to which control is to be passed if an I/O error is encountered.

This routine reads a data record from the logical unit specified into contiguous locations starting with the third parameter. The second parameter gives the location within the data set from which the record is to be read. If an I/O error is encountered, control is transferred to the statement number given as the alternate return.

DD Card Considerations

The FORTRAN convention for naming DD cards is used by these routines. This means thatif logical unit number xx is used, a DD card named FTxxF001 must be applied.

For a temporary data set, the minimum requirements on the DD card are the SPACE parameter and the UNIT parameter. Normally the record size (BLKSIZE) will also be specified. If it is omitted, however, the maximum record size for the device (without keys) will be used. For a 2314 this is 7294 bytes. In the example below, the data set is to contain 150 records, each of which is 800 bytes long:

```
//FT10F001  DD UNIT = 2314, SPACE = (800,150), DCB = BLKSIZE = 800
```

Restrictions

A maximum of 50 units can be used unless the program is reassembled with NUMENTS set higher.

The DAIO routines may be used with the FTIO routines (for sequential I/O), however, the same unit number may not be used for both sequential and direct I/O.

Programming Notes

The most efficient method of using these routines is to create the data set sequentially and then access it directly. When created in this manner no skeleton records are needed and therefore some I/O time is saved.

I/O Errors

If a permanent I/O error is detected during a read (or read backward) operation, certain information about the error is made available in a common area called FERMSG before the alternate return is taken. This information can be printed or used by the user if desired. The common area contains a full word integer array, 26 words long. The contents of this array is as follows:

WORD	CONTENTS
1	Buffer address
2	Number of bytes read
3-22	EBCDIC message for printing
23	IOBFLAG1, IOBFLAG2, Sense bytes 0 and 1
24	Sense bytes 2, 3, 4 and 5
25	ECB completion code and CCW address from the CSW
26	Last 4 bytes of the CSW

The first 22 words of this array are derived from the message created by the SYNADAF macro-instruction (see Supervisor and Data Management Macro Instructions C28-6647) and from the UCB. (Volume serial number from the UCB is stored in the message in place of stepname.) The remaining 4 words are derived from the first 4 words of the IOB and the sense information in the UCB. (See System Control Blocks - C28-6628).

To make the information available in the users program the following common statement may be used:

```
COMMON/FERMSG/IMES (26)
```

The message can then be written when an I/O occurs with the following write statement:

```
      WRITE (6, 1000) IMES
1000  FORMAT (1X, Z8, I6, 20A4, $(1X,Z8))
```

A message written in this matter will contain the following items, separated by blanks and/or commas:

1. Buffer address (hex)
2. Number of bytes read (decimal)
3. Jobname
4. Volume serial number (from UCB)
5. Unit address
6. Device type
7. DD name
8. Operation attempted
9. Error description
10. Tape: block number (decimal)

    Direct access: track address BBCCHHR (hex)

    Unit Record: asterisks
11. Access method
12. IOBFLAG1, IOBFLAG2, IOSENS0, IOSENS1 from the IOB. For a description of the first two see System Control Blocks, IOB description. For the second two, see the component description manual for the I/O device.
13. Sense bytes 2-5 from the UCB. For a description of these, see the component description manual for the I/O device.

14. ECB completion code and Channel Status Word (CSW) command address portion. See System Control blocks for the ECB code.

15. Last 4 bytes of CSW (status and residual count field.) See the System/360 reference card (green card) for the format of the CSW. The residual count indicates how many bytes were remaining to be read or written when the operation was terminated.

User Abends

101 = BLKSIZE specified is too large for device

102 = Unit number is being used for sequential I/O

103 = System error at EOV (should not occur)

001 = I/O error during output operation

201 = Invalid unit number

002 = Invalid record number

Unit Table Format:

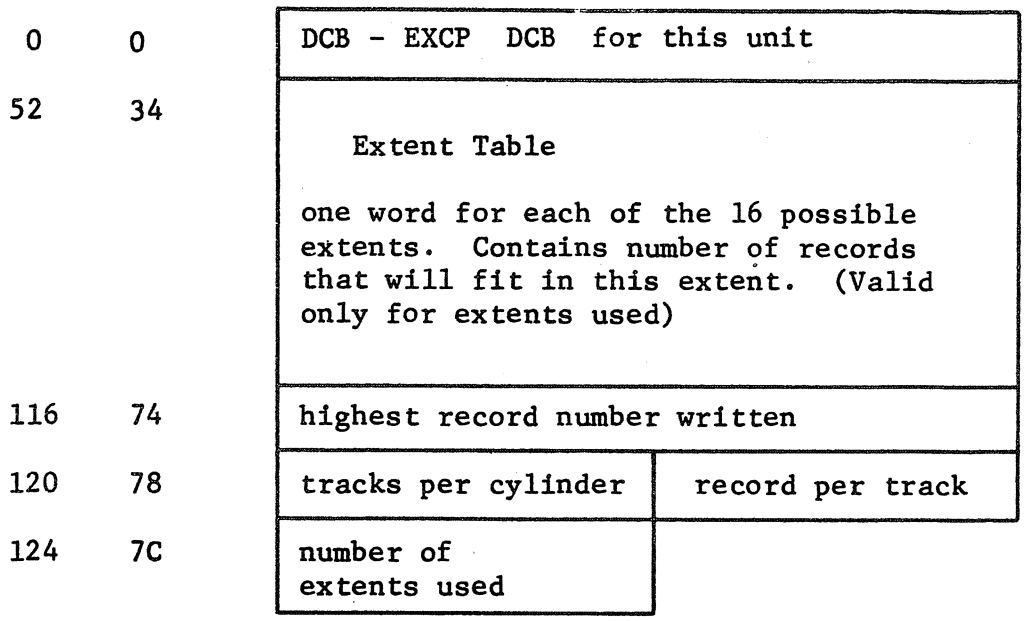
Word 1	Code Byte	1st DCB Address
Word 2	Code Byte	2nd DCB Address
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
Word 50	Code Byte	50th DCB Address

The unit table is in a CSECT called FUNITABL and contains one 4 byte entry for each logical unit number, 1 - 50. The first byte of each entry contains a code indicating the status of the unit as follows:

00 - Unit is not currently active
01 - DCB is open for input
02 - DCB is open for output
03 - DCB is open for reading backwards
04 - DCB is allocated but not open
FF - DCB is open for direct access

If core storage for the DCB has been allocated (any code except 00) the low order 3 bytes of the entry contain the DCB address. (In the case of a direct access unit this is also the address of the direct access control block.) If the code byte is 00 and the remainder of the entry contains an address, this unit was previously used but UNLOAD has been called and the DCB area has been freed.

Direct Access Control Block Format:



The Direct Access Control Block (DACB) for any unit can be found by looking up its address in the unit table (csect FUNITABL). This table has one 4 byte entry for each unit. To find the address for logical unit N, look up V(FUNITABL) + (N-1) *4.