FDRUM and FDRMB

1.0.0 Contents:

5.6.0 FSRCH To Search for an Item.
5.6.1 Entrance requirements.
5.6.2 Normal exit conditions.
5.6.3 Other exit conditions.
5.6.4 Example.

5.7.0 FDELC To Delete Item n and its Chain.
5.7.1 Entrance requirements.
5.7.2 Normal exit conditions.
5.7.3 Example.

5.8.0 FDELM To Delete the Item at Sector Address m.
5.8.1 Entrance requirements.
5.8.2 Normal exit conditions.
5.8.3 Other exit conditions.
5.8.4 Example.

5.9.0 FDELL To Delete Item Last Read by a FGETN, FGET, or FSRCH.
5.9.1 Entrance requirements.
5.9.2 Normal exit conditions.
5.9.3 Example.

5.10.0 FOVRM To Overwrite the Item at Sector Address m.
5.10.1 Entrance requirements.
5.10.2 Normal exit conditions.
5.10.3 Example.

5.11.0 FWRNL To Chain an Item to a Chain Whose Last Item is in Memory as a Result of a FGETN, FGET, FSRCH.
5.11.1 Entrance requirements.
5.11.2 Normal exit conditions.
5.11.3 Other exit conditions.
5.11.4 Example.

5.12.0 FCLOS To Close a File.
5.12.1 Entrance requirements.
5.12.2 Normal exit conditions.
5.12.3 Example.

2.0.0 General:

The FDRUM and FDRMB input/output routines and directives provide for the control of data-files on the FASTRAND drum(s), and must be used with the operating system (OPR). Control is provided for files on FASTRAND I, II, and MODULAR drum(s) configurations.

Before these routines can be used, the Initial-Drum-Set-Up routine (IDMS) must be run. Basically, IDMS "maps" the drum(s) by writing a series of directory entries, and provides the drum-locator-loader (DLL) which is written in the first track of drum zero. Through the use of parameter cards, the user can "map" the drum(s) according to his needs. IDMS must be run before the FASTRAND drum software can be used.

The following paragraphs (2.1.0 thru 2.1.5) define terms as to their meaning within this document:

2.1.0 Item (sector):

(1) The smallest unit of data which can be accessed by the directives within this document is one FASTRAND drum sector. A sector is 168 characters in length of which the last eight (8) are used for control purposes (see link-addresses). The user may wish to house more than one record (data-record) within each sector, or for that matter a record may be multiple sectors in length; but the file control routines will regard item and sector as being equal in meaning; therefore, within this document 'item' will mean FASTRAND drum sector.

(2) The largest unit of data which can be accessed by the directives within this document is one FASTRAND drum sector. (refer to the BDRUM PROCedure and directives for multiple sector functions).

2.1.1 Chaining data-sectors,
Link-addresses,
Forward link - Backward link:

(1) Data-sectors can be chained together; the length of the chain is unlimited, but only one level of chaining is allowed.

(2) The last eight (8) characters of each data-sector is reserved for control purposes; namely: forward and backward link addresses.

(3) Forward-link address (characters 164-167):

(a) if = binary zeroes: this sector is the last of a chain; an unused sector; or this is the only sector for this item. (has no links).

(b) if ≠ binary zeroes: these 4 characters are the forward link address (24 bits)(absolute drum address of next link).

(4) Backward-link address (characters 160-163):

(a) if = 077777777: this sector is the first sector; it may or may not have links (see forward link address).

(b) if ≠ 077777777; and ≠ to binary zeroes: these 4 characters are the backward link address (24 bits)(absolute drum address of previous link).

(c) if = binary zeroes: this is an unused data-sector.

(5) An item (sector) which is not used and is between address one and three of the directory will have both link-addresses set to binary zeroes.

2.1.2 nth item:

(1) The nth item is relative to the file within which it exists; the first item of a data-file is item Ø (where n = Ø), the second item of a data-file is item 1 (where n = 1), etc..

2.1.3 m (item m):

(1) m is the absolute drum address of the sector.

2.1.4 File directory:

(1) The following is the format of the file directories on the FASTRAND drum(s):

Characters	contents
(a) 0 - 7	file name (alpha-numeric name)
(b) 8 - 11	address 1: the sector address of the first item in the file.
(c) 12-15	address 2: the sector address of the first sector beyond the file area.
(d) 16- 19	address 3: the sector address of the next free sector available within the file area.
(e) 20- 23	address 4: the sector address of the next free sector available within the file area for chain links.
(f) 24	head-indicator. If = binary 1, then this file starts at head zero.

(2) Each class of drum area has its own file-directory-descriptor; each file has its own file-directory entry within the limits of this descriptor. These entries on the drum(s) are maintained by the FDRUM directives.

2.1.5 CLASS name:

(1) This is the name which was supplied to the Initial-Drum-Set-Up routine (IDMS), within which the file name exists.

This name must be eight (8) characters in length; the left most character must be alphabetic.

3.0.0 FDRUM and FDRMB Input/Output PROCedures:

The FDRUM or FDRMB call directs the assembler to include the drum input/output control routines in the worker program, with a linkage to the operating system. These routines conform to the FASTRAND drum data-file conventions established in the FASTRAND drum software specifications document.

The directives which are normally used in conjunction with the BDRUM PROCedure (refer to the BDRUM document) can also be used with the directives in this document; this is the reason for the two call lines: FDRUM and FDRMB.

(1) FDRUM PROCedure:
This PROCedure does not allow the use of any of the BDRUM directives, only directives within this document can be called by the worker program.

(2) FDRMB PROCedure:
This PROCedure allows both types of directives: BDRUM directives, and the directives within this document.

3.1.0 The FDRUM and FDRMB Call Lines:

(Only one of either must appear in the worker program).

LABEL	OP'N	OPERANDS	COMMENTS
	FDRUM	, P2, P3, P4, P5	P1 not required on FDRUM.
	FDRMB	P1, P2, P3, P4, P5	

Where:

P1 — total number of files to be used by the worker program (max. 63).

P2, P3 — two index registers which can be used by the directives. Whenever a directive is executed, the contents of these index registers will be changed. (1,2.... not X1, X2....)

P4 — the label of a subroutine to which a JR will be executed in the event of an unrecoverable FASTRAND drum error; the contents of tetrads 56, 57, 58, and 59 will be in AR1. If control is returned, the order will be considered successful.

P5 — RT, if this program's FASTRAND drum orders are to be given priority (when running concurrently). Both programs in core cannot be so designated.
Otherwise, this parameter is omitted.

3.1.1 Example:

(as the FDRUM or FDRMB call line might appear in the worker program).

	LABEL	OP'N	OPERANDS	COMMENTS
(1)		FDRUM	, 6, 7, ERROR, RT	
(2)		FDRMB	20, 6, 7, ERROR, RT	

. Comments: (1) this call line permits the use of the directives which are included in this document.

. 6,7 index registers (X6, X7).

. ERROR user's subroutine which will be entered in the event of an unrecoverable FASTRAND drum error.

. RT this program's FASTRAND drum order are to be given priority.

. (2) this call line permits the use of the directives which are included in this document, and also those which are included in the BDRUM document.

. 20 the total number of files which will be used by the worker program.

4.0.0 FFILE PROCedure:

The purpose of the FFILE PROCedure is to describe data-file(s) which will be accessed through the FDRUM directives by the worker program. An FFILE PROCedure call line should appear once for each file which is to be used, but this PROCedure is not to be used to describe files which will be accessed by the BDRUM directives.

The FFILE PROCedure generates a sixty-four (64) character table which is used by the FDRUM directives when accessing the file(s). The MSL of this table is the label on the FFILE call line.

4.1.0 The FFILE Call Line:

LABEL	OP'N	OPERANDS	COMMENTS
label	FFILE	P1, P2, P3, P4	New file
	FFILE	P1, P2	Existing file

Where:

P1 — file name. An eight (8) character alpha-numeric name bounded by apostrophes, the left-most character of which must be alphabetic.

P2 — the label to which control will be transferred if: EOF (end of file on input) or EOA (end of area on output) is detected; or if an input-output function is attempted on a file which has not been opened.

P3 — the number of FASTRAND drum sectors required for a new file.

P4 — HEAD, if a new file is to start at head ∅. Otherwise, this parameter is left blank.

Note: If this file exists on the FASTRAND drum(s), P3 and P4 are omitted.

label: this label becomes the MSL of a sixty-four (64) character file table.

4.2.0 Example:

```
        LABEL     OP'N      OPERANDS
(1)     FILE1     FFILE     'FILE 001', END01
(2)     FILE2     FFILE     'FILE 002', END02, 1000, HEAD
```

. Comments: (1) this call line describes a file currently existing on the FASTRAND drum(s).

. (2) this call line describes a new data-file.

. 'FILE 002' - this is the file name which will be used in referencing this file.

. END02 EOF and EOA routine.

. 1000 drum area is to allow for 1000 sectors.

. HEAD this file must start at head ∅.

5.0.0 FDRUM Directives:

The following directives work in conjunction with the FDRUM or FDRMB PROCedure.

5.1.0 FOPEN directive: To open a data-file.

LABEL	OP'N	OPERANDS
	FOPEN	P1, P2, P3, ~~P4~~

Where:

P1 the label of the FFILE call line for this file (here-after referred to as the file-id).

P2 the address (LSC) of the CLASS name, within which this file currently exists; or, if a new file, within which this file is to be assigned.

P3 the address to which control will be transferred if the file cannot be opened.

The LSC of AR1 will indicate the following possible reasons: (a binary character is stored in Location 15 by FDRUM)

if binary 1, directory is full.
2, data area is full.
3, class cannot be found.
4, supposed existing file cannot be found.
5, file is already open.

P4 HEAD, if this is a new file and if it is to start at head zero.
Otherwise, this parameter is omitted.

5.1.1 Entrance requirements:

(1) this file must have been described by an FFILE PROCedure.

5.1.2 Normal exit conditions:

(1) if this is a new file: a new directory entry has been written on the FASTRAND drum(s) within the class of drum area requested.

(2) if this is an existing file: its file directory was searched for and found on the FASTRAND drum(s).

(3) the file directory has been stored in the FFILE table the MSL of which is P1 (file-id).

(4) the file is open, and can be accessed by the other FDRUM directives.

5.1.3 Example:

	LABEL	OP'N	OPERANDS
(1)		FOPEN	FILEA, CNAME, ERR01, HEAD
(2)		FOPEN	FILEA, CNAME, ERR01

. Comments: (1) this is a new file because of the HEAD parameter.

. (2) FILEA the label of the FFILE call line for this file.

. CNAME address (LSC) of an eight character field containing the CLASS name within which this file exists or is to exist.

. ERR01 address to which control will be returned if the file cannot be opened.

5.2.0 . FGETN directive: Read the nth item.

LABEL	OP'N	OPERANDS
	BAl	mmmm, 4
	FGETN	P1, P2, P3, P4

Where:

P1	file-id
P2	core memory address of input area (MSL).
P3	EOC (end-of-chain) exit.
P4	NOCHAIN, if chained sectors are not to be read by this call; Otherwise, this parameter is omitted.
mmmm	the address (LSC) of a four character field containing n.

5.2.1 Entrance requirements:

(1) the 4 LSC of AR1 must contain the value of n, in binary.

(2) the file must be open.

5.2.2 Normal exit conditions:

(1) the nth item has been read, and:

(a) the value of m is in the 4 LSC of AR2.
(b) the value of n is in the 4 LSC of AR1.

(2) if P4 is omitted, and if this command is repeated with the same value of n in AR1, then the first chained sector will be read; if repeated again with the same value of n, the second chained sector will be read, etc.... until the EOC is reached.

5.2.3 Other exit conditions:

(1) if n is equal to or greater than address three (3) of the file directory, control will be transferred to the EOF (end-of-file) address specified by the FFILE call.

(2) if the nth item is not valid, control will be transferred to P3 (the EOC address).

Note: for a data-sector to be valid, the backward-link address must be unequal to binary zeroes (24 bits).

5.2.4 Example:

	LABEL	OP'N	OPERANDS
(1)		BAl	FLEAN, 4
		FGETN	FILEA, INPUT, EOCHN, NOCHAIN
(2)		BAl	FLEAN,4
		FGETN	FILEA, INPUT, EOCHN

. Comments: (1) no chained sectors are to be read by this call (even though this command is repeated with the same value of n in AR1).

. (1), (2) FLEAN - the address (LSC) of a 4 character field which contains n.

. FILEA - the label of the FFILE call line for this file.

. INPUT - MSL of input area.

. EOCHN - EOC (end-of-chain) address.

. NOCHAIN - (on 1) - no chained sectors are to be read.

5.3.0 FGET directive: Read the next item.

LABEL	OP'N	OPERANDS
	FGET	P1, P2

=

Where: P1 file-id.

P2 core memory address of input area (MSL).

5.3.1 Entrance requirements:

(1) the file must be open.

5.3.2 Normal exit conditions:

(1) the next item has been read (if the last item read was the nth item, then this call will read the n+1 item).

(2) no chaining is considered, and other exit conditions are as for the FGETN directive.

5.3.3 Example:

LABEL	OP'N	OPERANDS
(1)	FGET	FILEA, INPUT

. Comments: See FGETN (5.2.4).

5.4.0 FPUTN directive: Write the nth item.

LABEL	OP'N	OPERANDS
	BAl	mmmm, 4
	FPUTN	P1, P2

Where:

P1	file-id.
P2	core memory address of output area (MSL).
mmmm	the address (LSC) of a four character field containing n.

5.4.1 Entrance requirements:

(1) the 4 LSC of AR1 must contain the value of n.

(2) the file must be open.

5.4.2 Normal exit conditions:

(1) the nth item has been written, and:

(a) the value of m is in the 4 LSC of AR2.
(b) the value of n is in the 4 LSC of AR1.

(2) if the nth sector was already occupied, then a chain is formed, or linked to an existing chain. If n was equal to or greater than address three (3) of the file directory, then address three (3) is incremented to n+1 and sectors between the old address three (3) and n (if any) are cleared to binary zeroes (link-addresses only) indicating unused sectors.

5.4.3 Other exit conditions:

(1) If n is more than address four (4) of the file directory, the EOA (end-of-area) exit is given. The EOA is specified in the FFILE call for this file.

5.4.4 Example:

LABEL	OP'N	OPERANDS
(1)	BAl	FLEAN, 4
	FPUTN	FILEA, OUTPT.

. Comments: FLEAN 4 character field containing n.

. FILEA file-id.

. OUTPT core memory output area.

5.5.0 FPUT directive: Write the next item.

LABEL	OP'N	OPERANDS	
	FPUT	P1, P2	
	Where:	P1	file-id.
		P2	core memory address of output area (MSL).

5.5.1 Entrance requirements:

(1) file must be open.

5.5.2 Normal exit conditions:

(1) if the last item written was the nth, then this command will write the n+1th item. If the n+1th was already occupied, then a chain is formed.

(2) other exit conditions are as for FPUTN (5.4.2).

5.5.3 Example:

LABEL	OP'N	OPERANDS	
	FPUT	FILEA, OUTPT	
. Comments:	FILEA		file-id.
.	OUTPT		core memory address of output area.

5.6.0 FSRCH directive: Search for an item.

LABEL	OPCODE	OPERANDS

Move the search key to the first 8, 16, or 32 locations of the input area.

FSRCH p1, p2, p3, p4

Where:

p1	file-id.
p2	core memory address of input area (MSL). This address must be an even multiple of 64.
p3	search key length (8, 16, or 32).
p4	address to which control will be transferred if item cannot be found.

5.6.1 Entrance requirements:

(1) the search key must be stored in the first 8, 16, or 32 characters of the input area.

(2) the file must be open.

(3) Note: a search for "equal to" OR "equal to/greater than" can be ordered:

If an "equal to" search is to be made, P3 should equal: 8, 16, 32.

If an "equal to/greater than" search is to be made, P3 should equal: 8G, 16G, or 32G.

5.6.2 Normal exit conditions: (if find).

(1) the searched for sector has been read-in, and:

(a) the value of m is in AR2 (4 LSC).
(b) the value of n is in AR1 (4 LSC).

5.6.3 Other exit conditions:

(1) if sector is not found, control is given to P4.

(2) Note: only the first sector of a chain can be searched for.

5.6.4 Example:

LABEL	OP'N	OPERANDS
	BAl	KEY, 8
	SAl	INPT+7, 8
	FSRCH	FILEA, INPT, 8, NFIND

. Comments:

KEY	eight (8) character field containing the key.
INPT	core memory input area.
FILEA	file-id.
NFIND	address to which control will be given if sector cannot be found.

5.7.0 FDELC directive: Delete item n and its chain (if any).

LABEL	OP'N	OPERANDS	
	BAl	mmmm, 4	
	FDELC	P1	
	Where:	P1	file-id.
		mmmm	the address (LSC) of a 4 character field containing n.

5.7.1 Entrance requirements:

(1) the 4 LSC of AR1 must contain the value of n.

(2) the file must be open.

5.7.2 Normal exit conditions:

(1) the link addresses of the item n have been cleared to binary zeroes. The value of n and m in the file table are unchanged.

5.7.3 Example:

LABEL	OP'N	OPERANDS	
	BAl	FLEAN, 4	
	FDELC	FILEA	
. Comments:	FLEAN		address of 4 character field containing the value of n.
.	FILEA		file-id.

5.8.0 FDELM directive: Delete the item at sector address m.

LABEL	OP'N	OPERANDS	
	BA2	mmmm, 4	
	FDELM	P1, P2	
	Where:	P1	file-id.
		P2	invalid exit.
		mmmm	address (LSC) of a 4 character field containing the value of m.

5.8.1 Entrance requirements:

(1) the 4 LSC of AR2 must contain m.

(2) the file must be open.

5.8.2 Normal exit conditions:

(1) the backward and forward link addresses are cleared to binary zeroes. If there were preceding and/or succeeding sectors in the chain, then these are now linked directly. The values of n and m in the file table are unchanged.

5.8.3 Other exit conditions:

(1) if m is not valid (within the file specified), control is transferred to P2.

5.8.4 Example:

LABEL	OP'N	OPERANDS
	BA2	FLEAM, 4
	FDELM	FILEA, INVLD

Comments:	FLEAM	4 character field containing the value of m.
.	FILEA	file-id.
.	INVLD	address to which control will be given if m is not valid.

5.9.0 FDELL directive: Delete item last read by a FGETN, FGET, or FSRCH call.

LABEL	OP'N	OPERANDS
	FDELL	P1

Where: P1 file-id.

5.9.1 Entrance requirements:

(1) file must be open.

5.9.2 Normal exit conditions:

(1) same as for FDELM (refer to 5.8.2).

5.9.3 Example:

LABEL	OP'N	OPERANDS
	FDELL	FILEA

. Comments: FILEA file-id.

5.10.0 FOVRM directive: Overwrite the item at sector address M.

LABEL	OP'N	OPERANDS	
	BA2	mmmm, 4	
	FOVRM	P1, P2	
	Where:	P1	file-id.
		P2	core memory address of item to be written.
		mmmm	address (LSC) of 4 character field containing the value of m.

5.10.1 Entrance requirements:

(1) the 4 LSC of AR2 must contain the value of m.

(2) the file must be open.

5.10.2 Normal exit conditions:

(1) the item is written in sector m. Any backward and forward link addresses are retained. The values of m and n in the file table are unchanged.

5.10.3 Example:

LABEL	OP'N	OPERANDS	
	BA2	FLEAM, 4	
	FOVRM	FILEA, AREA	
. Comments:	FLEAM		address of 4 character field containing m.
.	FILEA		file-id.
.	AREA		output.area.

5.11.0 FWRNL directive: Chain an item to a chain whose last item is in memory as a result of a FGETN, FGET, FSRCH call.

LABEL	OP'N	OPERANDS	
	FWRNL	P1, P2, P3, P4	
	Where:	P1	file-id.
		P2	the memory address of the last item read.
		P3	the memory address of the item to be written.
		P4	error return address if the item last read is not the last of a chain. (forward link-address is $\neq 0$.)

5.11.1 Entrance requirements:

(1) file must be open.

5.11.2 Normal exit conditions:

(1) the item is added to the chain which begins with the nth item, and:

(a) the value of m is in the 4 LSC of AR2.
(b) the value of n is in the 4 LSC of AR1.

5.11.3 Other exit conditions:

(1) if there is no more space within the file area, control will be returned to the EOA address (see FFILE).

5.11.4 Example:

LABEL	OP'N	OPERANDS	
	FWRNL	FILEA, INPT, OUTPT, RETRN	
. Comments:		FILEA	file-id.
.		INPT	input area, of item last read.
.		OUTPT	output area, of item to be written.
.		RETRN	error return address.

5.12.0 FCLOS directive: Close a file.

LABEL	OP'N	OPERANDS	
	FCLOS	P1, P2, P3, P4	
	Who	P1	file-id.
		P2	the address (LSC) of an eight character field containing the CLASS name within which this file exists.
		P3	address to which control will be trans... if the file cannot be found. The LSC ... will indicate the following:
			if binary 3, CLASS cannot be f...
			4, File cannot be f...
		P4	REL, if the drum area associated wi... file is to be released; Otherwise, this parameter is omitted.

5.12.1 Entrance requirements:

(1) the file must be open.

5.12.2 Normal exit conditions:

(1) the open-indicator in the file table is reset, indicating the file is now closed, and the directory entry on the dr... re-written. If P4 = REL, then the file area is released.

5.12.3 Example:

LABEL	OP'N	OPERANDS	
	FCLOS	FILEA, CNAME, NFIND, REL	
. Comments:		FILEA	file-id.
.		CNAME	address (LSC) of field con... name.
.		NFIND	address to which control wi... the class or file cannot be ... drum(s).
.		REL	release the drum area.