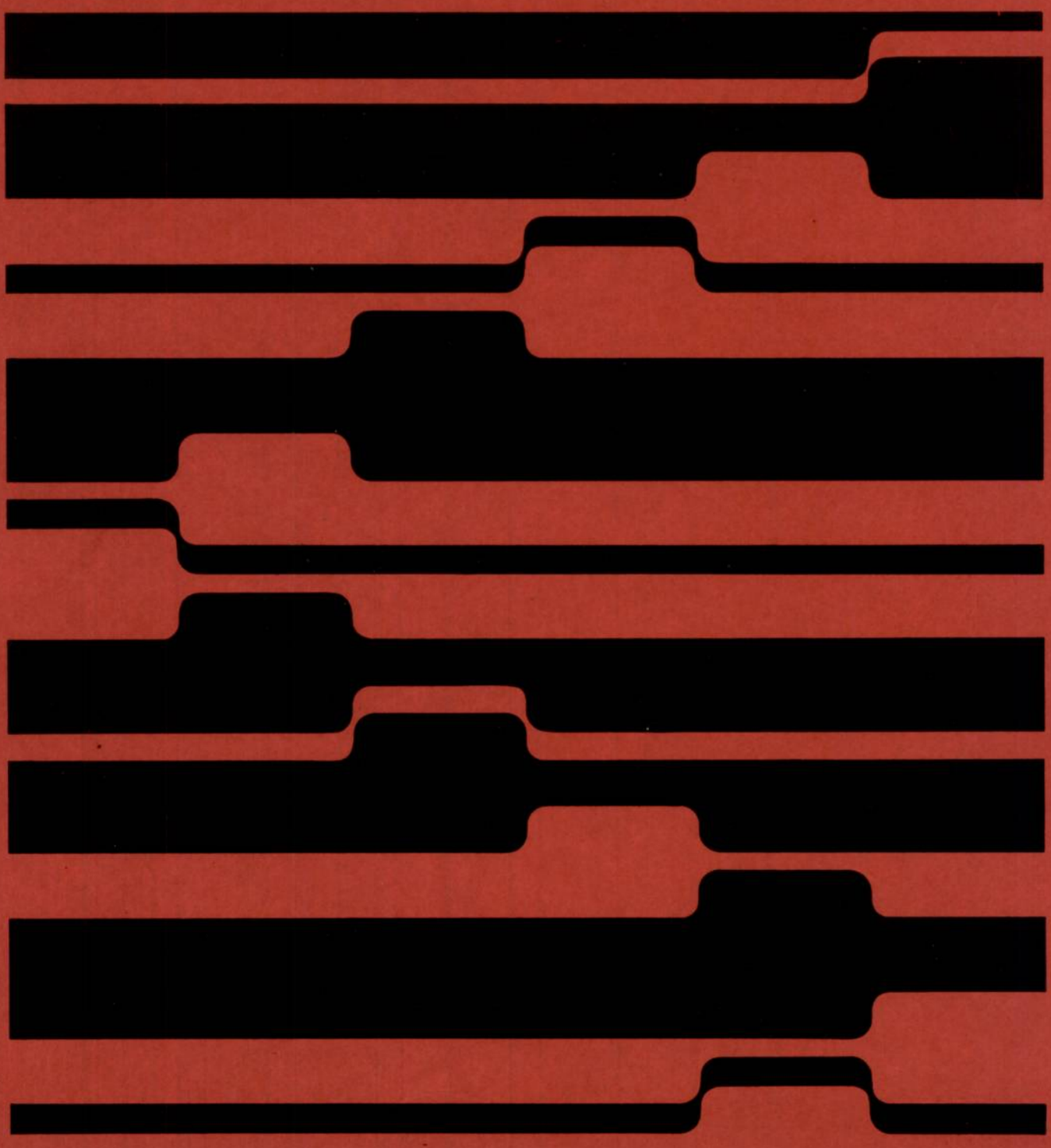
ICL

Find-2 Multiple Enquiry System

1900 Series



4187(2)

ICL

1900 Series

FIND-2 Multiple Enquiry System

First Edition December 1969

Amendment list 2
incorporating User Notices 4, 5 and 6

Each amendment list contains one or more numbered instructions to replace one or more existing pages or to add one or more new pages.

When a page is amended, significant technical changes on the re-issued page will be indicated by a vertical line in the margin against the changed passages. Any lines on the re-issued page which remain from a previous amendment, will be removed. New chapters or completely revised chapters will not be marked with amendment lines.

The date of issue appears at the foot of all new pages and re-issued pages in the form (month, year).

1	Contents and Illustrations	Remove and destroy pages v to xii. Insert new pages v to xii.
2	Chapter 4	Remove and destroy pages 15/16. Insert new pages 15/16.
3	Chapter 5	Remove and destroy pages 17 to 26. Insert new pages 17 to 26.
4	Chapter 6	Remove and destroy pages 33/34. Insert new pages 33/34.
5	Chapter 7	Remove and destroy pages 41 to 46. Insert new pages 41 to 46.
6	Chapter 8	Remove and destroy pages 51/52. Insert new pages 51/52.
7	Chapter 8	Remove and destroy pages 54.1 to 64. Insert new pages 55 to 64.9.
8	Chapter 8	Remove and destroy pages 67 to 68. Insert new pages 67 to 68.4.
9	Chapters 8 and 9	Remove and destroy pages 71 to 78. Insert new pages 71 to 78.
10	Chapter 9	Remove and destroy pages 83 to 90. Insert new pages 83 to 90.
11	Chapter 10	Remove and destroy pages 93 to 102. Insert new pages 93 to 102.1.
12	Chapter 11	Remove and destroy pages 105 to 110.2. Insert new pages 105 to 110.3.
13	Chapters 12 and 13	Remove and destroy pages 111 to 116. Insert new pages 111 to 116.
14	Appendix 2 and 3	Remove and destroy pages 119 to 124. Insert new pages 119 to 124.
15	Appendix 4, 5 and 6	Remove and destroy pages 125 to 130. Insert new pages 125 to 128.
16	Index	Remove and destroy pages 131 to 134. Insert new pages 129 to 132.
17		Destroy User Notices 4, 5 and 6.
18		Update the amendment record and file this list at the back of the manual.

4187(4.71)

ICL House, Putney, London, S.W.15

ICL International Computers Limited

User notice

MANUAL (NOTICE NO.)

25/3/70

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (3)

File one copy of this notice with each of the manuals indicated.

FORM 1/230/45(3.69)

FIND-2 MULTIPLE ENQUIRY SYSTEM (3)

The following programs have been modified:

#X63C/3: Overlaid from E.D.S. or Twin Disc

#X63D/3: Overlaid from Magnetic Tape

The changes are described in the following sections.

Modifications

All reported errors in the above programs up to and including those in Software Notice File Management and Reporting/5 have now been corrected.

The programs' core store sizes have now been reduced to:

X63X: 12100 words

X63D: 12000 words

Enhancements

One enhancement has been incorporated into the programs. It is now possible, under the #FORM directive, to specify 0X (zero X) as the output format requirement for a numeric field or arithmetic result, that is, to assign effectively no space in the line for a field or result to be written. This enables intermediate results of arithmetic operations to be omitted from an output line or record, even though such intermediate results must appear, each on its own 'field card'.

For example:

```
1TAG,FIELDA + FIELDB
2TAG,FIELDC - FIELDD
,1TAG * 2TAG
```

By assigning in the format line corresponding to these 'field cards' the items, say:

....0X 0X 10X

the outputting of the intermediate results 1TAG and 2TAG may be avoided.

Facilities not available in this issue

Dump and Restart is not yet available.

ICL International Computers Limited

User notice

RECEIVED 2 2 NOV 1971

MANUAL (NOTICE NO.) 10/11/71

4187

FIND-2 MULTIPLE ENQUIRY SYSTEM (7)

File one copy of th notice with each o manuals indicated.

The following programs have been modified:

#X63C/7: Overlaid from E.D.S. or twin disc
#X63D/7: Overlaid from magnetic tape
#X63F/4: Issued in semicompiled form with overlay steering segment in source.

All errors reported in the FIND-2 Multiple Enquir System up to and including those in Software Noti FILE MANAGEMENT AND REPORTING/60 are corrected ir this issue or are covered by this User Notice.

The information contained in this User Notice is divided into three sections:

1 Part 1 Enhancements in programs #X63C/7, #X63D/7, #X63F/4

2 Part 2 Enhancements in program #X63F/5 or

3 Part 3 Amendments to the FIND-2 Multiple Enquiry System manual.

PART 1: ENHANCEMENTS IN ALL PROGRAMS: #X63C/7, #X #X63F/4

The enhancements listed below have been made in a three programs:

1 TYPE V (TERMINATOR) VARIABLE LENGTH FIELDS
The programs have been enhanced in two ways respect to Type V (terminator) variable leng fields:

(a) A space is now permissible as a termina character. This should be indicated u the #DICTIONARY directive by leaving at one space after the solidus in the defi of the particular field, for example:

VFIELD7 6/*, VFIELD8 7/ , VFIELD9 8/$

FORM 1/230/45(3.69)

Issued by Technical Publications

(b) The programs will now handle variable length records as set up by the COBOL subroutine COBPACKM (1900 Series COBOL, Chapter 13, Edition 2, TP4279), that is, those records which do not contain the terminating characters of fields omitted from the end of the record. The programs will continue to process in exactly the same way as before, records set up with such trailing terminators present. A slightly different response is now required to the absence from a record of a terminator defined in the dictionary, however, since this is no longer automatically an error situation arising from a faulty record or incorrectly specified dictionary. The programs will no longer halt with messages of the form: *n*th TERMINATOR MISSING. Instead, a warning display message: TERMINATOR MISSING : CHECK FILE FORMAT will be output for the first record during each file pass in which this situation is encountered.

Thereafter this record and any others in which terminators are missing will be treated as if the field, or fields, were absent. Whenever this display message occurs, users should check carefully that the dictionary correctly describes the records, this is particularly important, since a wrongly specified terminator may cause all following fields to be regarded as missing.

The enquiry logic has been amended to deal with this situation and now conforms to the following table.

Relationship	*Hit*	*Miss*
EQL		✓
NEQ	✓	
GTR		✓
GEQ		✓
LSS	✓	
LEQ	✓	

The programs have been enhanced to allow a minimum record size of two words (instead of five) in the #OPTION parameter, for example: #OPT ED 2 LIST will now be valid.

The #FORM parameter has been enhanced to allow numeric constants to be directly defined in the field allocations line. Thus, if the user wishes to output the constant 99 together with the value in the binary field, WFIELD, his #FORM directive would be:

```
#FORM
99  8X
,WFIELD
```

4 The #OPTION parameter has been enhanced to allow a paper size to be specified. Page headings can therefore be produced regularly when off-lining under GEORGE2 or GEORGE3, or part page reports can be produced. The format of the #OPTION parameter is:

#OPTION ▽, *(Output device type)* ▽, *(Record size)* ▽,
[*Page size*▽,] { PRINT / LIST / TOTAL / TABLE } ▽, [[HIT*n*▽,]
(Record selection criteria)]

If the Page size field is omitted, the program will continue to use the line printer control loop as previously.

Note: For batched output reports the page size is not carried over from one option to the next.

PART 2: ENHANCEMENTS IN #X63F/5 ONLY

The enhancements listed below have been made in #X63F/5 only.

1 FINDUE2 has been enhanced to give access to the hit-words, enabling the user to determine which enquiry or enquiries a hit record satisfies. The new parameters to which X3 points are:

Word 0 : pointer to start address of record

Word 1 : 0 if record is a MISS for all enquiries.
Pointer to hitword(s) if a hit for at least one enquiry

Word 2 : Number of enquiries in the batch.

Note: This will have no adverse effect on user own-coding using the previous parameters.

2 FINDUE5 has been enhanced to give access to lines of totals in the LIST and TOTAL options as well as LIST detail lines. The new parameters to which X3 points are

Words 0 - 3 : as previously

Word 4 : negative signifies a LIST detail line
0-7 signifies the level of a totals line (where 0 = GRAND)

Note: Existing user own-coding may need to be amended to allow for the totals line.

3 A new user entry point FINDUE9 has been incorporated; this is called in the TABLE option at the end of processing the main file, after the GRAND TOTALS line has been printed. The line printer LP0 remains allotted and X3 points to one parameter:

Word 0 : pointer to start address of trailer label or E0F bucket. (Exactly as in FINDUE8).

Exit Condition: EXIT 0 0, the contents of X3 are immaterial.

4 A new user entry point FINDUE10 has been inserted to give access to the program during parameter validation. Thus the user may use own-coding to read parameters from backing store or read in parameters for his-own coding at validation. At each call, the parameter buffer passed to FINDUE10 contains the last card validated, to allow the user to monitor through the parameters. On exit from FINDUE10 the option is given to replace the validated parameter by a parameter 'card' supplied through own-coding for validation, or to continue reading and validating normally. (The user may specify that this parameter should not be printed unless in error.) The first call is before validation has commenced and a space-filled parameter card is passed into FINDUE10, thus all parameters may be supplied through own-coding. If this is the case the user can reply to this effect and FIND-2 will not allocate a slow input peripheral.

The parameter to which X3 points is:

Word 0 : Pointer to start address of buffer containing last parameter validated. Parameter 'cards' supplied by own-coding must overwrite this buffer.

Exit conditions:

EXIT 0 0 Read and validate next parameter as usual.

EXIT 0 1 The next parameter for validation has been placed by own-coding into the parameter buffer.

X3 = 0 Do not print this parameter unless in error.

X4 = 0 (first call only) All parameters will be supplied by own-coding and no slow input peripheral is required.

PART 3: AMENDMENTS TO THE FIND-2 MULTIPLE ENQUIRY SYSTEM MANUAL

The following amendments should be made to the FIND-2 Multiple Enquiry System Manual.

Chapter	*Page*	*Line*	
5	17	-12	Insert "Comment cards may not appear before the first major directive".
5	18	25	Alter "1 No or field blank" to "1 NO or field blank"
6	37	-4	Replace existing line with "The limit to the number of different constants is dealt with in Table 7 on page 94."
8	59	5	Alter "CHANNELn" to "CHANNELVn" and "[Comment]" to "[[comment]".
8	59	-2	Replace existing line by "1,YEAR,PAGE[YEARLY TOTAL"
8	60	2	Insert "The use of predefined constants in enquiries is described on page 32".
8	64.3	9	Delete "Fields defined as type B in the dictionary may not be output in the print option".
8	64.6	-9	Delete sentence beginning "The special reserved field name TODAY..."
8	64.6	-0	Insert "Only fields defined in the dictionary as B,W,H,D, M,P or F may be used in arithmetic operations."
8	64.7	-17	Insert "Only fields defined in the dictionary as B,W,H, D,M,B or F may be totalled."
8	64.7	-8	Replace existing line with "[COLUMN%]V [ROW]
8	64.8	8,-9, -2	Delete "[Field Tag]" but not the following comma from beginning of each line.
8	64.9	3	Insert "After #HALT the re-entry procedure described in Figure 12 on page 53 should be used."

Chapter	*Page*	*Line*	
9	88	7,8	AREA and DEPOT should have length 12C
9	88	-3	This line should begin - "4B 5A 2B 12A 2B 12A..."
9	90	-2,-3, -4,-5	Insert "," at beginning of each line.
10	94	24	Preset size in words for code number 2 is 97 words.
11	109	7	Add new line "5 run has been terminated by a #HALT."
Appendix 3	124	-4	Replace "less than 12 characters long" by "12 characters long or less".
Appendix 3	124	-1	Add new line "(h) When using enquiry selection on the X or Y axes the enquiries must be mutually exclusive, that is, a record must not be capable of satisfying more than one enquiry.".
Appendix 3	122	5	Insert "There is a limit, unalterable by #STORAGE, of approximately 100 relations in any one enquiry."
Appendix 4	126	-15	Amend to read "(basic unit determined from value of y. The maximum value of y is 6)".

ICL International Computers Limited

User notice

MANUAL (NOTICE NO.) 10/11/71

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (7)

File one copy of this notice with each of the manuals indicated.

The following programs have been modified:

#X63C/7: Overlaid from E.D.S. or twin disc
#X63D/7: Overlaid from magnetic tape
#X63F/4: Issued in semicompiled form with overlay steering segment in source.

All errors reported in the FIND-2 Multiple Enquiry System up to and including those in Software Notice FILE MANAGEMENT AND REPORTING/60 are corrected in this issue or are covered by this User Notice.

The information contained in this User Notice is divided into three sections:

1 Part 1 Enhancements in programs #X63C/7, #X63D/7, #X63F/4

2 Part 2 Enhancements in program #X63F/5 only

3 Part 3 Amendments to the FIND-2 Multiple Enquiry System manual.

PART 1: ENHANCEMENTS IN ALL PROGRAMS: #X63C/7, #X63D/7, #X63F/4

The enhancements listed below have been made in all three programs:

1 TYPE V (TERMINATOR) VARIABLE LENGTH FIELDS
The programs have been enhanced in two ways with respect to Type V (terminator) variable length fields:

(a) A space is now permissible as a terminating character. This should be indicated under the #DICTIONARY directive by leaving at least one space after the solidus in the definition of the particular field, for example:

VFIELD7 6/*, VFIELD8 7/ , VFIELD9 8/$

FORM 1/230/45(3.69)

Issued by Technical Publications Service, 30 - 31 Friar Street, Reading, Berkshire. RG1 1JP

(b) The programs will now handle variable length records as set up by the COBOL subroutine COBPACKM (1900 Series COBOL, Chapter 13, Edition 2, TP4279), that is, those records which do not contain the terminating characters of fields omitted from the end of the record. The programs will continue to process in exactly the same way as before, records set up with such trailing terminators present. A slightly different response is now required to the absence from a record of a terminator defined in the dictionary, however, since this is no longer automatically an error situation arising from a faulty record or incorrectly specified dictionary. The programs will no longer halt with messages of the form: *n*th TERMINATOR MISSING. Instead, a warning display message: TERMINATOR MISSING : CHECK FILE FORMAT will be output for the first record during each file pass in which this situation is encountered.

Thereafter this record and any others in which terminators are missing will be treated as if the field, or fields, were absent. Whenever this display message occurs, users should check carefully that the dictionary correctly describes the records, this is particularly important, since a wrongly specified terminator may cause all following fields to be regarded as missing.

The enquiry logic has been amended to deal with this situation and now conforms to the following table.

Relationship	*Hit*	*Miss*
EQL		✓
NEQ	✓	
GTR		✓
GEQ		✓
LSS	✓	
LEQ	✓	

2 The programs have been enhanced to allow a minimum record size of two words (instead of five) in the #OPTION parameter, for example: #OPT ED 2 LIST will now be valid.

3 The #FORM parameter has been enhanced to allow numeric constants to be directly defined in the field allocations line. Thus, if the user wishes to output the constant 99 together with the value in the binary field, WFIELD, his #FORM directive would be:

```
#FORM
99  8X
,WFIELD
```

4 The #OPTION parameter has been enhanced to allow a paper size to be specified. Page headings can therefore be produced regularly when off-lining under GEORGE2 or GEORGE3, or part page reports can be produced. The format of the #OPTION parameter is:

#OPTION ∇, *(Output device type)* ∇, *(Record size)* ∇,
[*Page size*∇,] { PRINT / LIST / TOTAL / TABLE } ∇, [[HIT*n*∇,]
(Record selection criteria)]

If the Page size field is omitted, the program will continue to use the line printer control loop as previously.

Note: For batched output reports the page size is not carried over from one option to the next.

PART 2: ENHANCEMENTS IN #X63F/5 ONLY

The enhancements listed below have been made in #X63F/5 only.

1 FINDUE2 has been enhanced to give access to the hit-words, enabling the user to determine which enquiry or enquiries a hit record satisfies. The new parameters to which X3 points are:

Word 0 : pointer to start address of record

Word 1 : 0 if record is a MISS for all enquiries.
Pointer to hitword(s) if a hit for at least one enquiry

Word 2 : Number of enquiries in the batch.

Note: This will have no adverse effect on user own-coding using the previous parameters.

2 FINDUE5 has been enhanced to give access to lines of totals in the LIST and TOTAL options as well as LIST detail lines. The new parameters to which X3 points are

Words 0 - 3 : as previously

Word 4 : negative signifies a LIST detail line
0-7 signifies the level of a totals line (where 0 = GRAND)

Note: Existing user own-coding may need to be amended to allow for the totals line.

3 A new user entry point FINDUE9 has been incorporated; this is called in the TABLE option at the end of processing the main file, after the GRAND TOTALS line has been printed. The line printer LP0 remains allotted and X3 points to one parameter:

Word 0 : pointer to start address of trailer label or E0F bucket. (Exactly as in FINDUE8).

Exit Condition: EXIT 0 0, the contents of X3 are immaterial.

4 A new user entry point FINDUE10 has been inserted to give access to the program during parameter validation. Thus the user may use own-coding to read parameters from backing store or read in parameters for his-own coding at validation. At each call, the parameter buffer passed to FINDUE10 contains the last card validated, to allow the user to monitor through the parameters. On exit from FINDUE10 the option is given to replace the validated parameter by a parameter 'card' supplied through own-coding for validation, or to continue reading and validating normally. (The user may specify that this parameter should not be printed unless in error.) The first call is before validation has commenced and a space-filled parameter card is passed into FINDUE10, thus all parameters may be supplied through own-coding. If this is the case the user can reply to this effect and FIND-2 will not allocate a slow input peripheral.

The parameter to which X3 points is:

Word 0 : Pointer to start address of buffer containing last parameter validated. Parameter 'cards' supplied by own-coding must overwrite this buffer.

Exit conditions:

EXIT 0 0 Read and validate next parameter as usual.

EXIT 0 1 The next parameter for validation has been placed by own-coding into the parameter buffer.

X3 = 0 Do not print this parameter unless in error.

X4 = 0 (first call only) All parameters will be supplied by own-coding and no slow input peripheral is required.

PART 3: AMENDMENTS TO THE FIND-2 MULTIPLE ENQUIRY SYSTEM MANUAL

The following amendments should be made to the FIND-2 Multiple Enquiry System Manual.

Chapter	*Page*	*Line*	
5	17	-12	Insert "Comment cards may not appear before the first major directive".
5	18	25	Alter "1 No or field blank" to "1 NO or field blank"
6	37	-4	Replace existing line with "The limit to the number of different constants is dealt with in Table 7 on page 94."
8	59	5	Alter "CHANNELn" to "CHANNEL▽n" and "[Comment]" to "[[comment]".
8	59	-2	Replace existing line by "1,YEAR,PAGE[YEARLY TOTAL"
8	60	2	Insert "The use of predefined constants in enquiries is described on page 32".
8	64.3	9	Delete "Fields defined as type B in the dictionary may not be output in the print option".
8	64.6	-9	Delete sentence beginning "The special reserved field name TODAY..."
8	64.6	-0	Insert "Only fields defined in the dictionary as B,W,H,D, M,P or F may be used in arithmetic operations."
8	64.7	-17	Insert "Only fields defined in the dictionary as B,W,H, D,M,B or F may be totalled."
8	64.7	-8	Replace existing line with "[COLUMN%]▽ [ROW]
8	64.8	8,-9, -2	Delete "[Field Tag]" but not the following comma from beginning of each line.
8	64.9	3	Insert "After #HALT the re-entry procedure described in Figure 12 on page 53 should be used."

Chapter	*Page*	*Line*	
9	88	7,8	AREA and DEPOT should have length 12C
9	88	-3	This line should begin "4B 5A 2B 12A 2B 12A..."
9	90	-2,-3, -4,-5	Insert "," at beginning of each line.
10	94	24	Preset size in words for code number 2 is 97 words.
11	109	7	Add new line "5 run has been terminated by a #HALT."
Appendix 3	124	-4	Replace "less than 12 characters long" by "12 characters long or less".
Appendix 3	124	-1	Add new line "(h) When using enquiry selection on the X or Y axes the enquiries must be mutually exclusive, that is, a record must not be capable of satisfying more than one enquiry.".
Appendix 3	122	5	Insert "There is a limit, unalterable by #STORAGE, of approximately 100 relations in any one enquiry."
Appendix 4	126	-15	Amend to read "(basic unit determined from value of y. The maximum value of y is 6)".

ICL International Computers Limited

User notice

RECEIVED 2 0 APR 1972

MANUAL (NOTICE NO.) 5/4/72

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (8)

File one copy of this notice with each of the manuals indicated.

The FIND-2 Multiple Enquiry System User Notice (7) contains references to a program #X63F(5) (pages 1,3). This is in error; the references are in fact to #X63F(4).

The following manual amendments were also printed incorrectly and should have read:

Chapter	*Page*	*Line*	
8	64.7	-17	Insert "Only fields defined in the dictionary as B,W,H,D,M,P or F may be totalled."
9	88	-13	This line should begin "4B 5A 2B 12A 2B 12A ..."

The following manual amendments should also be noted:

Chapter	*Page*	*Line*	
4	15	-17	Replace this line by "This file is used as a work file to hold the validated enquiry parameters (if Switch 7 is ON) and output parameters."
App 1	117	-6,-5	Replace 11 by "11 A constants dictionary may be used to make the values of constants in enquiries redefinable when using the re-entry facility at interrogation."

Issued by Technical Publications Service, 30 - 31 Friar Street, Reading, Berkshire. RG1 1JP

FORM 1/230/45(3.69)

ICL International Computers Limited

User notice

Library RECEIVED 4 DEC 1972

MANUAL (NOTICE NO.)

22/11/72

4187 FIND 2 MULTIPLE ENQUIRY SYSTEM (9)

File one copy of this notice with each of the manuals indicated.

The following amendments should be made to the *FIND-2 Multiple Enquiry System* manual:

Chapter	*Page*	*Line*	
2	4	27	This should read: The modular design ...
3	9	17	Delete the sentence: If the user.... end of the run.
3	11	-	Add a new paragraph at the bottom of the page: FIND2 can handle variable length Decode records provided that the code is immediately followed by the fixed length decoded field.
4	15	17	Delete: but only one file may be accessed in one run.. and replace with: and up to 6 files may be accessed in one run.
4	15	19	Delete: The file created ... standard indexes. and replace with: The file should be created from cards or paper tape using the standard ICL utility program #XJEZ. It must be loaded sequentially with standard indexes and should contain one record of maximum key value in each bucket.

FORM 1/230/45(3.69)

Issued by Technical Publications Service, 30 - 31 Friar Street, Reading, Berkshire. RG1 1JP

Chapter	*Page*	*Line*	
4	15	27	Delete: Records are stored....fixed length. and replace with: Both the code and the decoded entry must be fixed length. Records must be stored in ascending sequence according to their code.
4	15	33	Delete: and will be contracted at the end of the job.
4	15	37	Add new sentence: A parameter file created using one version of FIND2 cannot be used with a different issue of the program.
4	20	-9	Add a comma at the end of the line.
4	20	-8	Delete: a space. and insert: one or more spaces.
4	23	3 (after table)	Add new sentence: If variable length fields are being both interrogated and output, the dictionary should be respecified at the start of the output phase.
4	23	6	Delete: except a space.
4	23	-3	Add: ,unless (for type V fields only) the field is omitted from the end of the record.
4	23	-	Add at bottom of page: 8. If the terminating character is a space, this should be indicated by leaving at least one space after the solidus.

Chapter	*Page*	*Line*	
6	31	27	After this line insert new paragraph: When a variable length field is found to be absent from a record the enquiry conditions operate according to the following table:

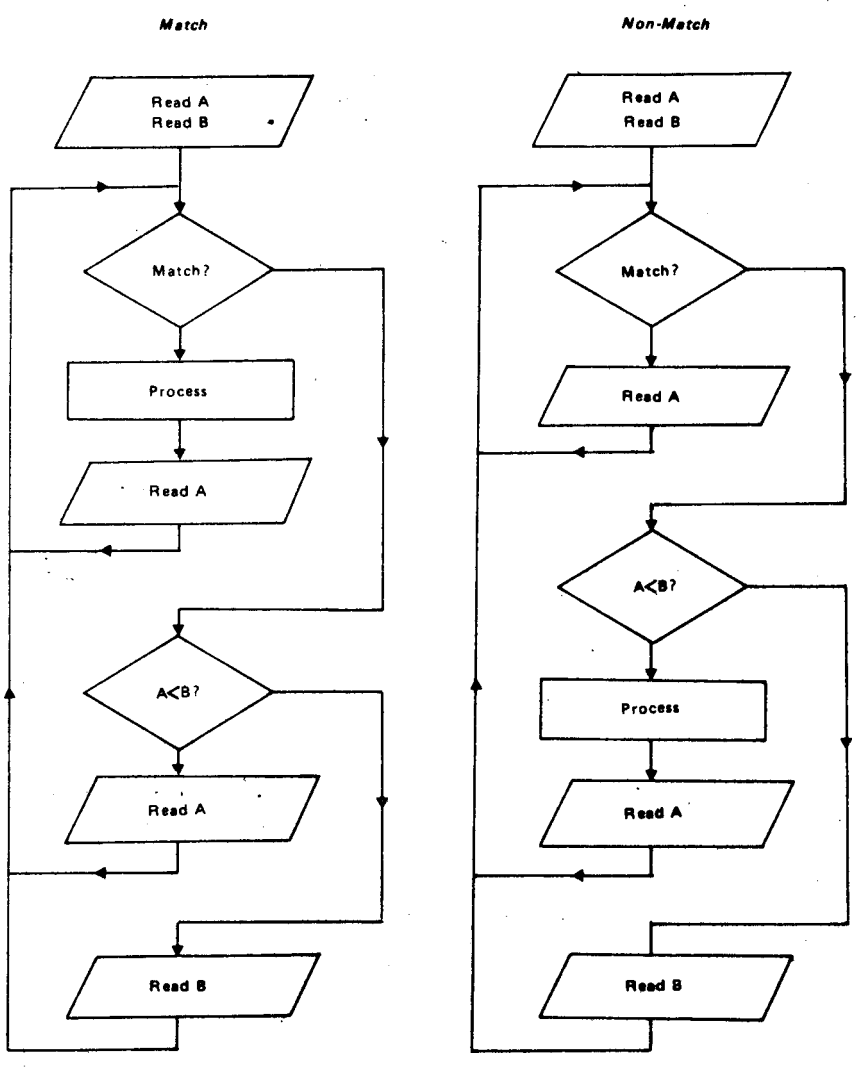
Relationship	*Hit*	*Miss*
EQL		✓
NEQ	✓	
GTR		✓
GEQ		✓
LSS	✓	
LEQ	✓	

6	32	12/13	Delete: Note: Ordinary decimal currencies ..... or mixed numbers.
6	33	-4	Delete: each word of an enquiry .... basic element. and replace with: that is each field name, relationship or logical operator.
8	63	9	Insert: ▽ after: #DICTIONARY and after: '(length)' [W C B]
8	64.3	-4	Add new sentence: When several #OPTIONs are specified with no #FORMs, another parameter, #HEAD or #PAGE, must separate the #OPTIONs.

Chapter	*Page*	*Line*	
8	64.4	-10	Before: When outputting insert a new sentence: If insufficient space is specified for numeric fields this is indicated in printed output by upward pointing arrows.
8	64.4	-5	Insert: only after: For the LIST and TOTAL options
8	64.4	-3	Insert: or consist only of numeric characters after: with a non numeric character
8	64.5	12	After: 4 STERLING insert: (see Appendix 4 pages 125 and 126)
8	64.5	-4	After: 7 DECIMAL CURRENCY insert: (see Appendix 4 pages 125 and 126)
8	64.6	20	Add new sentence: No field definition is required for a B or Z type field in the format line.
8	64.7	15	Delete: OX or OA and replace with: 0X or 0A Add new sentence: The results of arithmetic operations performed between

Chapter	*Page*	*Line*	
			items on one print line (defined by a #FORM) are not available for any purpose on another print line.
8	64.8	10	Add new sentence: The decode file opened will be the file with the highest generation number online.
8	64.8	-7	Add new sentence: Record counts may not be used in arithmetic operations.
8	64.8	-3	After: accumulated in the table delete the full stop and add: but may not be used in arithmetic operations.
8	66	3	Insert: ▽ , after: #MERGE
8	66	-3	Add new sentence: The merge facility can only be used with the **LIST** and TOTAL options.
8	66	-1	Insert a comma after: #MERGE
8	66		Add the diagram on page 6.
8	67	4	After: (Record size)▽, insert: [Page size ▽,]
8	67	14/15	Delete: on stationery .... characters. and insert: output for the second record beginning at the 69th character.

Flowchart of the merge process
A=main file B=merge file



Chapter	*Page*	*Line*	
8	67	18	Add new sentence: The minimum size is 2 words.
8	67	21	Delete: 42 and insert: 43 After line 21 insert the following paragraphs: Page size specifies the number of lines to be printed per page, after which a throw to head of page will occur and new page headings will be printed. If this field is omitted head of page throwing will be controlled by the line printer control loop. For batched out ut records the page size is not carried over from one option to the next.
8	69	2 and -1	Insert: ; after: CHANNEL Add new sentence: #SPACE may only follow #HEAD, #PAGE or #FORM.
8	73	20	Delete lines 20 and 21 and replace with: 3. The block size is mandatory and must be 512 for the parameter file. Other values may be specified for the hit file or an output file.
9	80	-	Add at foot of page: #WRI,MT,FIND2HITFILE,,,0,512.
9	81	5	Add new line after line 5: #REA,MT,FIND2HITFILE.

Chapter	*Page*	*Line*	
10	94	14	Add under *Description* column: This buffer is also used to hold the first record when two records per line are being output. The area must not be set to zero in this case.
10	94	25	Add under *Description* column: Each entry consists of 1 word holding the enquiry name and 1 word holding the hit count. Under *Preset size in words* column, delete: 97 and replace with: 193.
10	94	28	Add under *Description* column: Numeric constants are held in whole words; character constants are packed into a minimum number of words.
10	94	-12	Under *Description* column delete the full stop and add: space allocation in the #FORM directive. If a print line is to contain constants only then one entry is still required in this table.
10		-10	Add the following: If a print line is not to contain any constants then one entry is still required in this table.
10	95	14	After: #STORAGE insert: ;
10	97	1	Delete: 13 after Fig and insert: 14

Chapter	*Page*	*Line*	
10	1000	4	Delete complete line and replace with: Non 0 = A pointer to the hit word if a hit for at least one enquiry. Insert new line: (3) number of enquiries in the batch.
10	100	21	After line 21 insert new line: (5) 0 signified a LIST detail line. 0 to 7 signifies the level of a totals line (0=Grand)
11	108	-14	Delete: DISPLAY and replace with: HALTED
11	108	-11	Delete: DISPLAY and replace with: HALTED
11	109	-1	Add under *Reason* column: An error has occurred on writing or extending disc file N.
12	112	-	Add new section to table headed *RECOVERABLE ERRORS IN INPUT DATA*:

TERMINATOR MISSING: CHECK FILE FORMAT	The end of a record has been encountered before reaching the *n*th variable length field (type V) during interrogation or output	1	GO at this point will cause spaces to be inserted for this field
		2	Abandon run and check file format and Dictionary definitions

Chapter	*Page*	*Line*	
12	112	-	Delete the first entry in the table headed *IRRECOVERABLE ERRORS IN INPUT DATA.*
Appendix 3	121	-11	Delete existing line and replace with: 11 If variable length fields are being both interrogated and output the dictionary should be redefined at the start of the output phase.
Appendix 3	121	-10	Delete the comma and add: unless the field is omitted from the end of the record (V type fields only).
Appendix 3	124	7 (after table)	Delete existing line and replace with: (e) #CONTROL may not be used when 2 records per line are being printed.
Appendix 3	124	-10	Insert new lines as follows: (r) when totalling to tape or disc a zero filled record is written to the end of the file if GRAND TOTALS have not been specified. (s) A control field cannot be totalled unless it is defined twice in the dictionary with a different name and type.
Appendix 3	124	-4	After: (including insert: percentages, field suppressed by use of 0X and

ICL International Computers Limited

User notice

L1

PUBLICATION (NOTICE NO.) 30/1/78

4187 FIND 2 MULTIPLE ENQUIRY SYSTEM (10)

File one copy of this notice with each of the publications indicated.

NEW ISSUE

The following new versions of the programs will shortly be available:

#X63C/8 Overlaid from disc
#X63D/8 Overlaid from tape
#X63F/8 Issued in source (for further details see *Manual amendments* below).

Errors reported in the following Software Notices are corrected in this issue:

1900 File Management and Reporting

Notice No.	*Item No.*
65	231
67	238
69	242
69	243
71	248
71	251
71	255
71	257
73	271
75	277
75	278
79	287
79	288
80	293
92	327
93	330
97	345
99	352
99	353
100	355
100	356

Issued by Software Distribution Department, 30 - 31 Friar Street, Reading, Berkshire RG1 1JP

Notice No.	*Item No.*
101	363
101	364
101	365
101	366
102	372
102	373
102	375
107	396
107	397
108	399

Restriction

Sterling fields may no longer be output in £.s.d. format. An attempt to do so will cause the program to halt QQ.

MANUAL AMENDMENTS

Chapter	*Page*	*Line*	
4	15	-10	Delete "..... 4- or ....."
7	44		Add a new sentence at the foot of the page: "Variable length fields may not be output to a magnetic tape or disc file."
8	64.3	-9	Add a new sentence: "Only one record per line may be printed when using the TOTAL option."
8	64.8		Add a note at the foot of the page: "Note: The COUNT facility will produce a zero total at a control change (LIST or TOTAL option), or for a Row total (TABLE option), unless a field defined in the dictionary is also being totalled in the same way".
10	97	-6	Delete "..... semi-compiled ....." and insert "..... source ....."
10	102	21	Delete the paragraph beginning: "One version of .....". Substitute the following paragraph:

Chapter	*Page*	*Line*	
			"One version of the FIND 2 Multiple Enquiry System, namely #X63F, is issued in PLAN source in COSY format. The filename is X63F SOURCE and the subfile name is FIND-2SOURCE. This file may be updated by #XPMR, etc., in the normal way, allowing the user to insert his own PLAN segments, or to change the overlay medium (present to disc in segment OVERLAYM), or program name."

ICL

International Computers Limited

User notice

L.

PUBLICATION (NOTICE NO.) 15/10/75

4187 FIND2 MULTIPLE ENQUIRY SYSTEM (11)

File one copy of this notice with each of the publications indicated

New issue

The following new versions of the programs will shortly be available:

#X63C/8B Overlaid from disc
#X63D/8B Overlaid from tape
#X63F/8B PLAN source

Errors and restrictions reported in the following Software Notices are corrected in this issue.

1900/File Management and Reporting

Notice number	*Item number*
71	249
71	262
77	280
79	289
80	291
111	408
111	409
112	412
113	413
114	417
114	418
114	419
114	421
116	440
119	453
119	454
119	455
119	456
119	457

Issued by Software Distribution Department, 30-31 Friar Street, Reading, Berkshire RG1 1JP

International Computers Limited

User notice

· Library

PUBLICATION (NOTICE NO.) 2/2/77

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (12)

File one copy of this notice with each of the publications indicated

AMENDMENT TO USER NOTICE 9

User Notice 9, page 10, lists an amendment to Appendix 3, page 124, line 7 (after the table) of the FIND-2 Multiple Enquiry System manual.

The amendment should read:

"Add the following to paragraph 4(e):

#CONTROL may not be used when 2 records per line are being printed"

PROGRAMS X63C, X63D

In the Output Phase when the TOTAL option is being used it is not possible to specify a FORM parameter with a format line composed entirely of constants (that is, without field definitions). Looping will result if such a #FORM parameter is specified.

(See page 64.4 of the manual *FIND-2 Multiple Enquiry System* (Edition 1, TP4187))

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

International Computers Limited

User notice

PUBLICATION (NOTICE NO.) 4/1/78

4187 FIND2 MULTIPLE ENQUIRY SYSTEM (13)

File one copy of this notice with each of the publications indicated

New issue

The following new versions of the programs will shortly be available:

#X63C/8D Overlaid from disc
#X63D/8D Overlaid from magnetic tape
#X63F/8D PLAN source

Errors reported in the following Software Notices have been corrected in this issue.

1900/File Management and Reporting

Notice number	*Item number*
124	7
131	1
135	2
135	3
135	4
135	5
138	1

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND 2 MULTIPLE ENQUIRY SYSTEM (14) 3/1/79

File one copy of this notice with each of the publications indicated

New issue

The following new version of X63F will be available shortly. It eliminates the corruption announced in software notice number 142.

X63F/8E PLAN source

Amendments given in the following software notices have been incorporated into the issue in order to keep it in line with Mark 8D.

1900/File Management and Reporting

Notice number	*Item number*
140	2
140	3
141	2

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND 2 MULTIPLE ENQUIRY SYSTEM (14) 3/1/79

File one copy of this notice with each of the publications indicated

New issue

The following new version of X63F will be available shortly. It eliminates the corruption announced in software notice number 142.

X63F/8E PLAN source

Amendments given in the following software notices have been incorporated into the issue in order to keep it in line with Mark 8D.

1900/File Management and Reporting

Notice number	*Item number*
140	2
140	3
141	2

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (15) 18/7/79

File one copy of this notice with each of the publications indicated

This user notice covers a number of items previously published in software notices.

Active items for Mark 8D of the FIND-2 Multiple Enquiry System begin in Software Notice 1900 File Management and Reporting, number 143.

Users are requested to make the following changes:

Field types

There is a limit of 64 X-type fields which can be defined in a record. The following changes should be made to the manual:

TYPE X (SUBRECORD TYPE) (page 24)

Note 5 should read:

5 Up to 64 type X variable length fields ...

INPUT LIMITATIONS (page 121)

Item 10 should read:

10 Maximum of either 36 type V or 64 type X variable length fields are allowed in a record

Arithmetic relationships in enquiries

The following note should be added at the end of the section on arithmetic relationships (page 37).

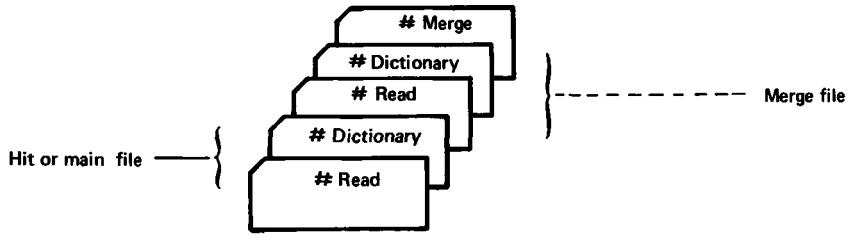
Note: This facility is restricted to whole word binary fields namely types D, F, H, M, P and W.

File definitions for a MERGE run

The following should be added to page 66 regarding the defining of the files for report(s) using the MERGE facility:

The MERGE parameter requires that both the main/hit file and the merge file are defined by a READ parameter and associated dictionaries as:

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR



This group of parameters will be active for all subsequent report OPTIONS until one of the following three parameters is read:

1	#MERGE	to replace the original/previous MERGE parameter. It will still act on the same two files
2	#READ plus #DIC	to introduce a new pair of files and a MERGE parameter or to revert/change to simple file processing
3	#STOP	to end the run

Compatibility of output facilities with output options

Page 43 and page 50 should be replaced by the new page 43 and pages 50 to 50.2.

Replacement pages contain new version of tables and figures which give a better representation of both LIST and TOTAL OPTIONs when output is to magnetic tape or exchangeable disc but do not override restrictions detailed under the parameters themselves or those enumerated in Appendix 3, page 124.

Facilities	*Options*					
	PRINT		LIST	TOTAL		TABLE
LP output	✓	✓		✓		✓
MT/Disc output			✓		✓	
Print two records per line		✓				
Print more than 1 line per record		✓	✓	✓	✓	
Merging		✓	✓	✓	✓	
Decoding		✓	✓			
Variable line spacing		✓		✓		✓
Optional headings	✓	✓		✓		✓
Automatic page headings	✓					
Automatic formatting	✓					
Printing variable length fields	✓	✓				
Arithmetic		✓	✓	✓	✓	✓
Column totalling		✓		✓	✓	✓
Row totalling						✓
Percentages						✓
Paper line-up by program		✓		✓		

Table 4 Compatibility of output facilities with output options

Headings

Headings are optional and may be specified for all printed output. Main headings will appear once only at the start of the tabulation; page headings will be repeated at the start of each page. Each type of heading is limited to 800 characters and so the maximum number of lines of heading which can be printed varies from 5 to 8 according to the size of the line printer.

Record selection

Various types of record selection are available so that results can be presented as efficiently as possible.

ENQUIRY SATISFACTION

Records may be selected for output on the basis of enquiries which they have satisfied. This applies only when output is from a hit file, where each record has the hit word(s) attached.

FIELD VALUES

With the TABLE option, records may be selected on the values of X and Y control fields.

MATCH CONDITIONS

When merging a main and secondary file on input, the user can specify that records be output when either a match or non match is obtained.

PART-FILE ACCESS

With this facility, the user can specify that only records from a certain part of his file are to be output.

Merging

The user may merge two files together to combine the information into a new file or report. Merging takes place at field level and it is possible to produce a new record format for an output file comprising information from two input records held on the main and merge files. These input files can be either the user's files or files produced

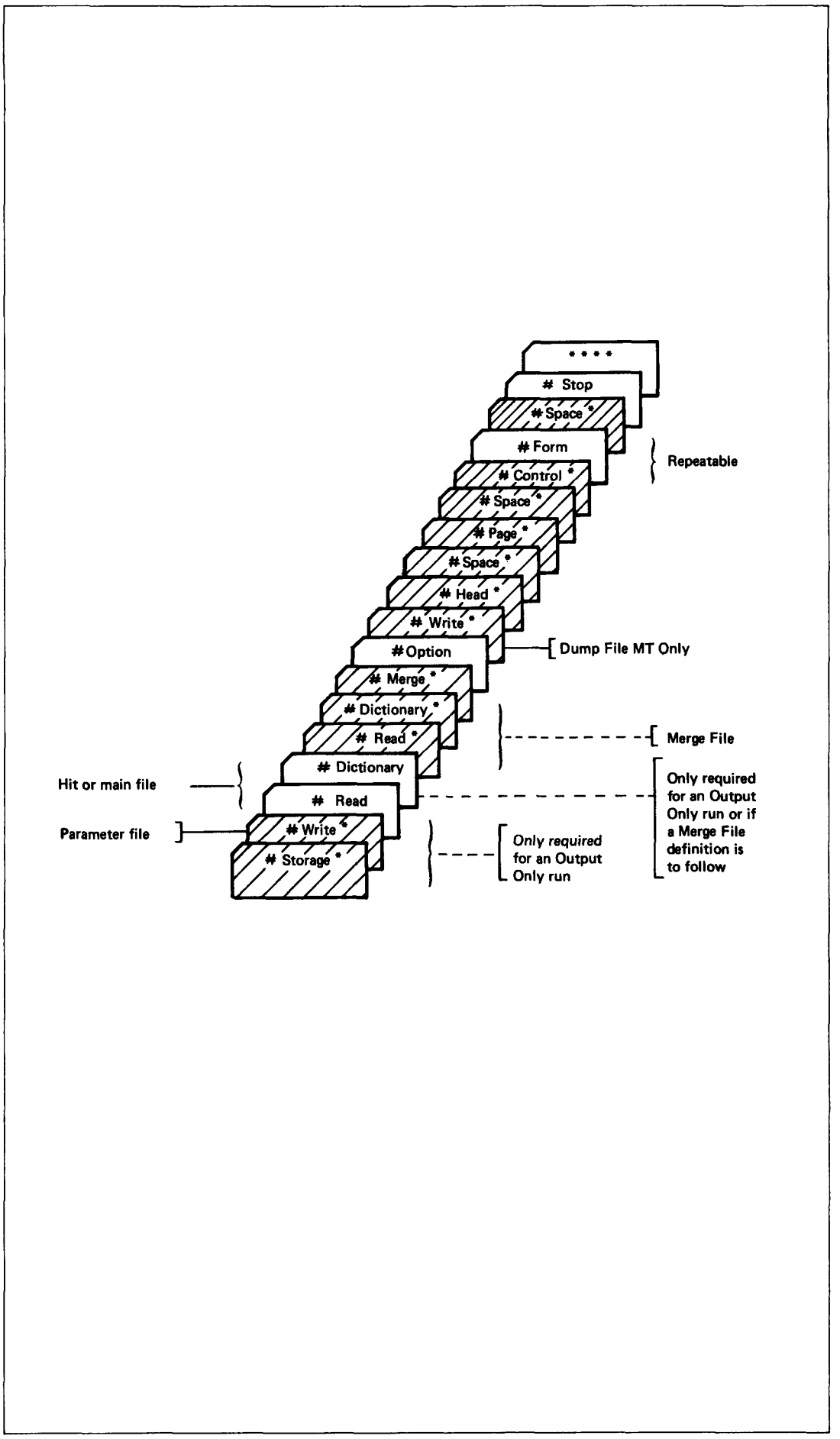


Figure 8a Output Phase List or Total (LP output)

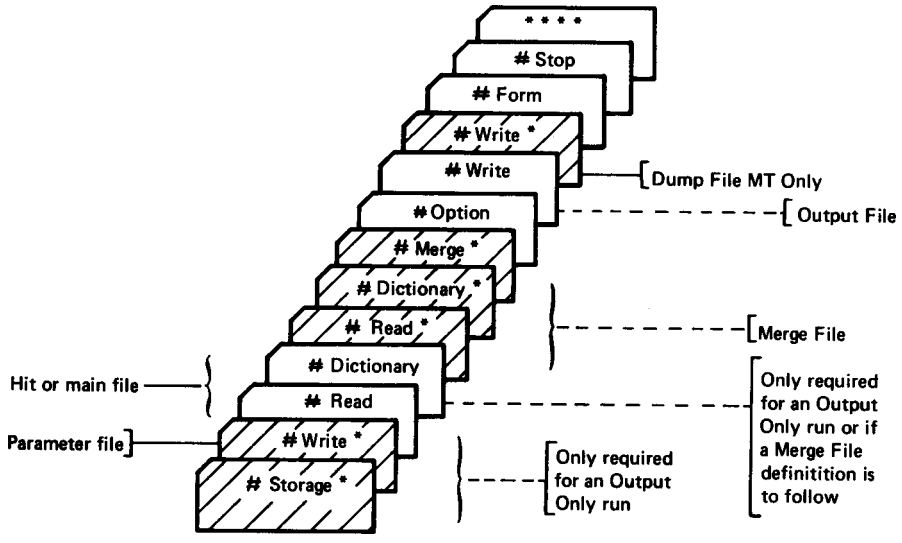


Figure 8b Output Phase List (MT or disc output)

(b.iii) TOTAL option (MT or disc output)

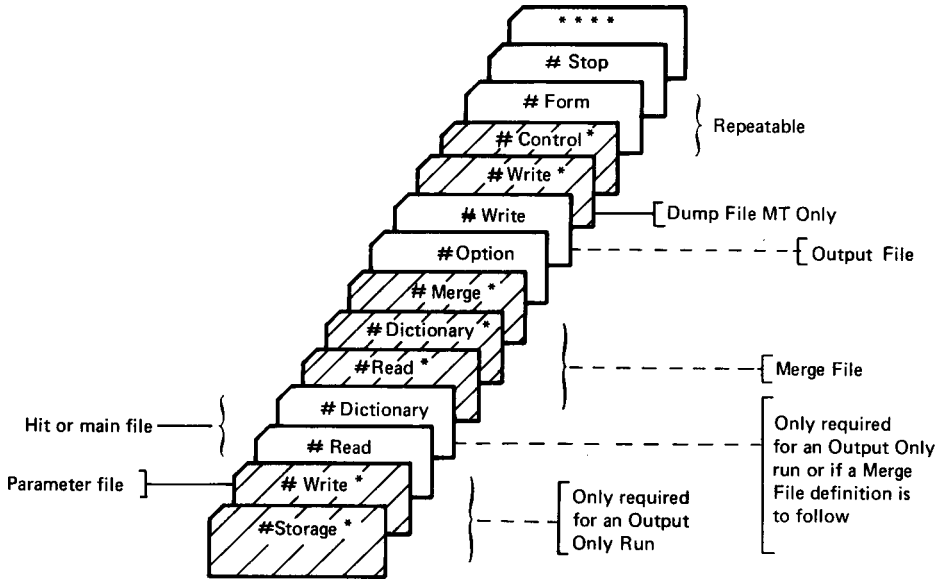


Figure 8c Output Phase Total (MT or disc output)

Line numbering under #HEAD and #PAGE

The following should be added to page 65:

Only a single numeric digit, in the range 1 to 8 inclusive, is allowed for the line number; the program will ignore any characters after the first non-space one encountered.

Page size

A further note should be added to the specification of the #OPTION parameter (page 67) to the effect that:

If a page size has been specified in an OPTION parameter, but a printer control loop is used with a smaller page size, then the loop takes precedence.

#YCONTROL in the TABLE OPTION

The following note replaces the sentence immediately before the examples on page 74.1:

The program is pre-set to accept up to 27 field values (or enquiry identities) for an unsorted file and the rules for their specification are the same as those for the XCONTROL parameter. If more (or fewer) values are required for any report, the allowable number can be varied by using the STORAGE parameter (see pages 71, 93 and following) to reset the number of entries in areas 19 and 20.

Note: If no values are being specified for a TABLE report, a single entry is still required in both areas 19 and 20.

Area sizes controlled by #STORAGE

Two additional notes should be added to Table 7 on page 94.

Code number 9:	This area is also used in conjunction with the outputting of variable-length fields. If such fields are to be output then the area should not be reduced to less than 10 words
Codes 19 and 20:	If these entries are changed then both should be changed to the same value. If no Y control field values are specified for a TABLE, a single entry is still required in both areas (that is, they cannot both be set to zero in an attempt to save space) to accommodate the accumulation of the current control value totals ready for printing

Program X63E - decode file identification

The device identity of the decode file (page 110) should be EDO.

Character constants in output options

Item 4(p), page 124 should read:

*(p) Store space is available for between 400 and 700 characters of constant information to be put in the output print lines (or the output record) from each input record (and can be changed by the use of #STORAGE).

In addition to the constants in the format line, as defined on page 64.4, zeros (for example, 4Z) when output is to a line printer and spaces (for example, 7B) when output is to magnetic tape or disc will be treated as constants.

The number of characters of constant information per line (that is, per #FORM group) is limited (and cannot be changed by the use of #STORAGE) to between 200 and 300 characters. This is adequate for line printer reports but may necessitate the use of several #FORM groups for a report being output to a magnetic tape or disc file

ICL

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (15) 18/7/79

File one copy of this notice with each of the publications indicated

This user notice covers a number of items previously published in software notices.

Active items for Mark 8D of the FIND-2 Multiple Enquiry System begin in Software Notice 1900 File Management and Reporting, number 143.

Users are requested to make the following changes:

Field types

There is a limit of 64 X-type fields which can be defined in a record. The following changes should be made to the manual:

TYPE X (SUBRECORD TYPE) (page 24)

Note 5 should read:

5 Up to 64 type X variable length fields ...

INPUT LIMITATIONS (page 121)

Item 10 should read:

10 Maximum of either 36 type V or 64 type X variable length fields are allowed in a record

Arithmetic relationships in enquiries

The following note should be added at the end of the section on arithmetic relationships (page 37).

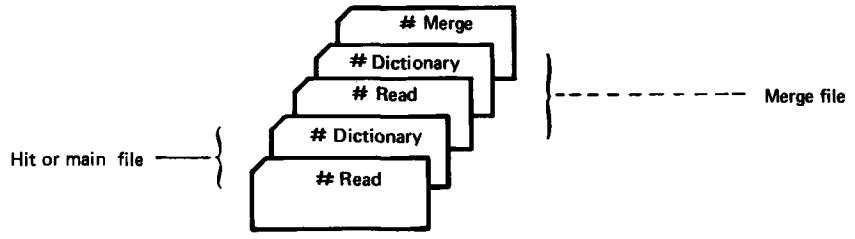
Note: This facility is restricted to whole word binary fields namely types D, F, H, M, P and W.

File definitions for a MERGE run

The following should be added to page 66 regarding the defining of the files for report(s) using the MERGE facility:

The MERGE parameter requires that both the main/hit file and the merge file are defined by a READ parameter and associated dictionaries as:

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR



This group of parameters will be active for all subsequent report OPTIONS until one of the following three parameters is read:

1	#MERGE	to replace the original/previous MERGE parameter. It will still act on the same two files
2	#READ plus #DIC	to introduce a new pair of files and a MERGE parameter or to revert/change to simple file processing
3	#STOP	to end the run

Compatibility of output facilities with output options

Page 43 and page 50 should be replaced by the new page 43 and pages 50 to 50.2.

Replacement pages contain new version of tables and figures which give a better representation of both LIST and TOTAL OPTIONs when output is to magnetic tape or exchangeable disc but do not override restrictions detailed under the parameters themselves or those enumerated in Appendix 3, page 124.

Facilities	*Options*					
	PRINT		LIST	TOTAL		TABLE
LP output	✓	✓		✓		✓
MT/Disc output			✓		✓	
Print two records per line		✓				
Print more than 1 line per record		✓	✓	✓	✓	
Merging		✓	✓	✓	✓	
Decoding		✓	✓			
Variable line spacing		✓		✓		✓
Optional headings	✓	✓		✓		✓
Automatic page headings	✓					
Automatic formatting	✓					
Printing variable length fields	✓	✓				
Arithmetic		✓	✓	✓	✓	✓
Column totalling		✓		✓	✓	✓
Row totalling						✓
Percentages						✓
Paper line-up by program		✓		✓		

Table 4 Compatibility of output facilities with output options

Headings

Headings are optional and may be specified for all printed output. Main headings will appear once only at the start of the tabulation; page headings will be repeated at the start of each page. Each type of heading is limited to 800 characters and so the maximum number of lines of heading which can be printed varies from 5 to 8 according to the size of the line printer.

Record selection

Various types of record selection are available so that results can be presented as efficiently as possible.

ENQUIRY SATISFACTION

Records may be selected for output on the basis of enquiries which they have satisfied. This applies only when output is from a hit file, where each record has the hit word(s) attached.

FIELD VALUES

With the TABLE option, records may be selected on the values of X and Y control fields.

MATCH CONDITIONS

When merging a main and secondary file on input, the user can specify that records be output when either a match or non match is obtained.

PART-FILE ACCESS

With this facility, the user can specify that only records from a certain part of his file are to be output.

Merging

The user may merge two files together to combine the information into a new file or report. Merging takes place at field level and it is possible to produce a new record format for an output file comprising information from two input records held on the main and merge files. These input files can be either the user's files or files produced

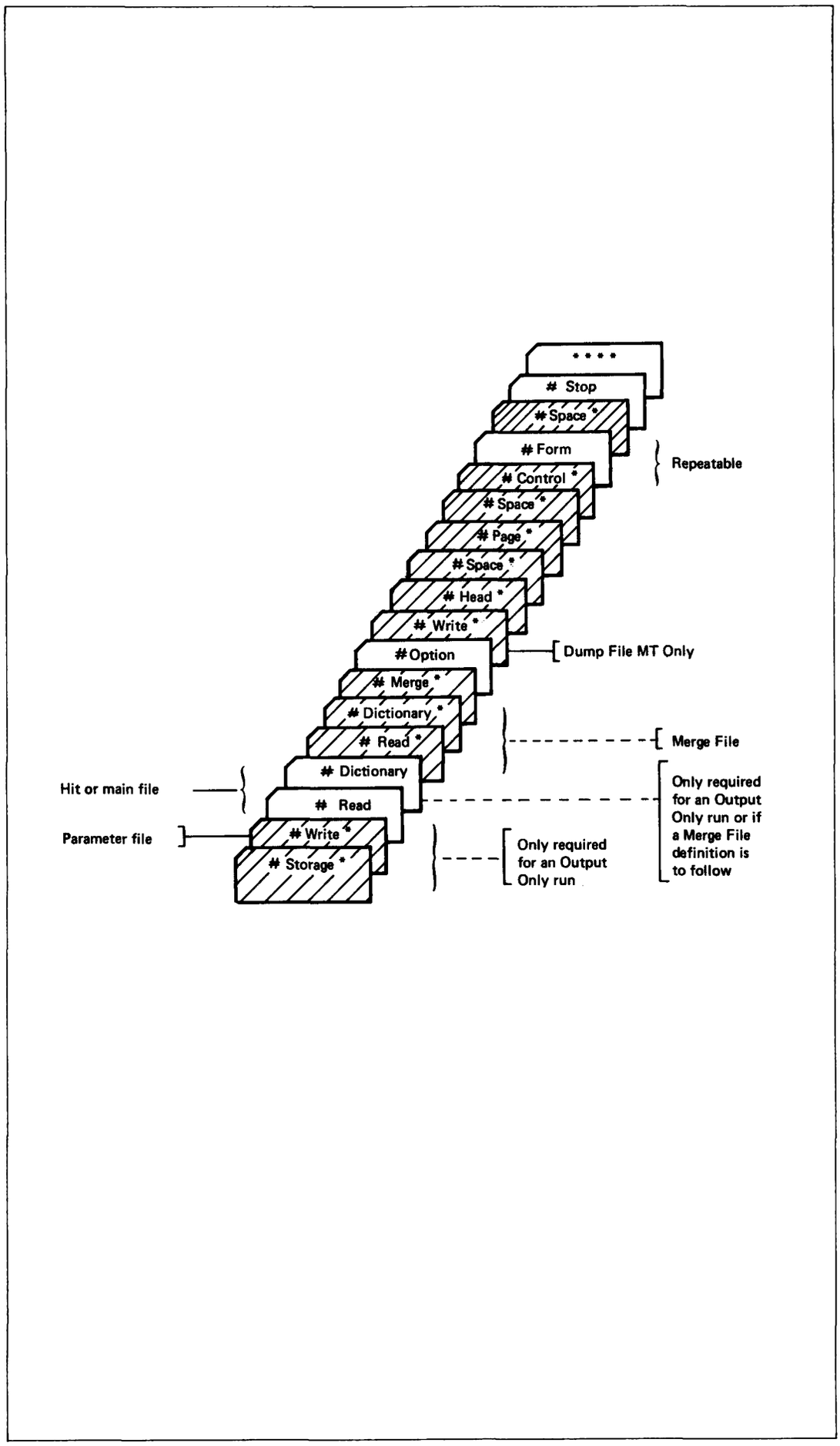


Figure 8a Output Phase List or Total (LP output)

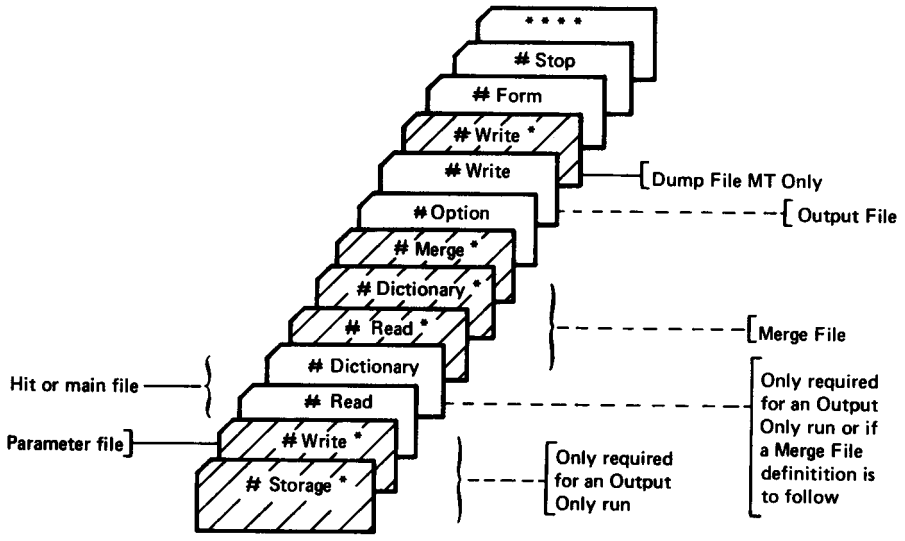


Figure 8b Output Phase List (MT or disc output)

(b.iii) TOTAL option (MT or disc output)

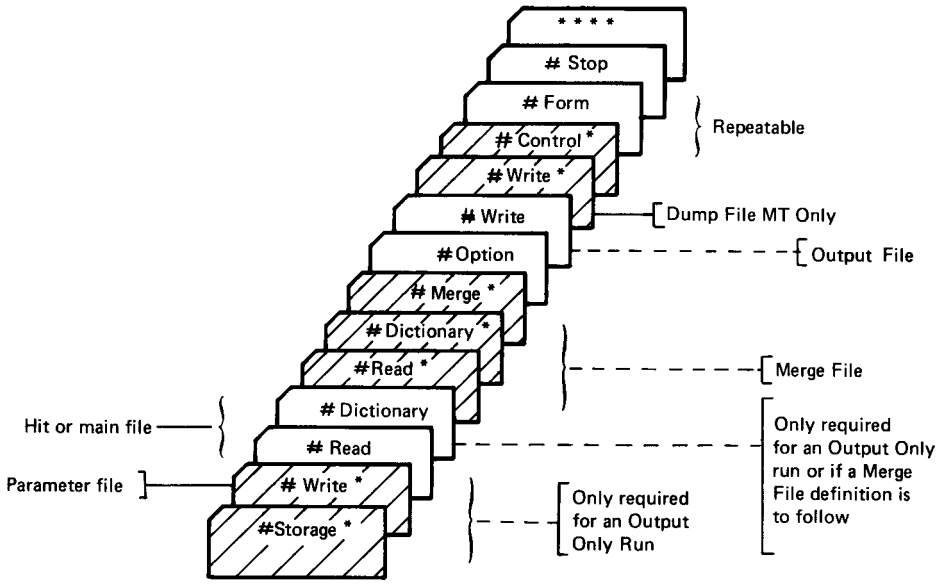


Figure 8c Output Phase Total (MT or disc output)

Line numbering under #HEAD and #PAGE

The following should be added to page 65:

Only a single numeric digit, in the range 1 to 8 inclusive, is allowed for the line number; the program will ignore any characters after the first non-space one encountered.

Page size

A further note should be added to the specification of the #OPTION parameter (page 67) to the effect that:

If a page size has been specified in an OPTION parameter, but a printer control loop is used with a smaller page size, then the loop takes precedence.

#YCONTROL in the TABLE OPTION

The following note replaces the sentence immediately before the examples on page 74.1:

The program is pre-set to accept up to 27 field values (or enquiry identities) for an unsorted file and the rules for their specification are the same as those for the XCONTROL parameter. If more (or fewer) values are required for any report, the allowable number can be varied by using the STORAGE parameter (see pages 71, 93 and following) to reset the number of entries in areas 19 and 20.

Note: If no values are being specified for a TABLE report, a single entry is still required in both areas 19 and 20.

Area sizes controlled by #STORAGE

Two additional notes should be added to Table 7 on page 94.

Code number 9:	This area is also used in conjunction with the outputting of variable-length fields. If such fields are to be output then the area should not be reduced to less than 10 words
Codes 19 and 20:	If these entries are changed then both should be changed to the same value. If no Y control field values are specified for a TABLE, a single entry is still required in both areas (that is, they cannot both be set to zero in an attempt to save space) to accommodate the accumulation of the current control value totals ready for printing

Program X63E - decode file identification

The device identity of the decode file (page 110) should be EDO.

Character constants in output options

Item 4(p), page 124 should read:

*(p) Store space is available for between 400 and 700 characters of constant information to be put in the output print lines (or the output record) from each input record (and can be changed by the use of #STORAGE).

In addition to the constants in the format line, as defined on page 64.4, zeros (for example, 4Z) when output is to a line printer and spaces (for example, 7B) when output is to magnetic tape or disc will be treated as constants.

The number of characters of constant information per line (that is, per #FORM group) is limited (and cannot be changed by the use of #STORAGE) to between 200 and 300 characters. This is adequate for line printer reports but may necessitate the use of several #FORM groups for a report being output to a magnetic tape or disc file

ICL International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (15) 18/7/79

File one copy of this notice with each of the publications indicated

This user notice covers a number of items previously published in software notices.

Active items for Mark 8D of the FIND-2 Multiple Enquiry System begin in Software Notice 1900 File Management and Reporting, number 143.

Users are requested to make the following changes:

Field types

There is a limit of 64 X-type fields which can be defined in a record. The following changes should be made to the manual:

TYPE X (SUBRECORD TYPE) (page 24)

Note 5 should read:

5 Up to 64 type X variable length fields ...

INPUT LIMITATIONS (page 121)

Item 10 should read:

10 Maximum of either 36 type V or 64 type X variable length fields are allowed in a record

Arithmetic relationships in enquiries

The following note should be added at the end of the section on arithmetic relationships (page 37).

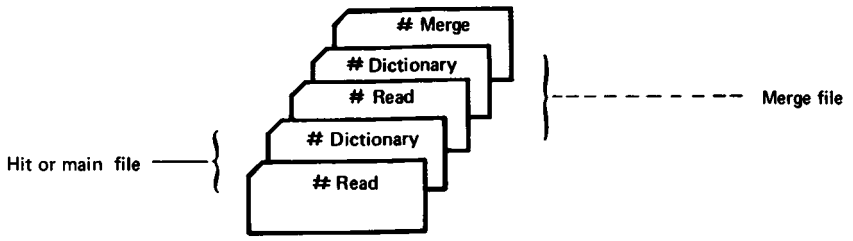
Note: This facility is restricted to whole word binary fields namely types D, F, H, M, P and W.

File definitions for a MERGE run

The following should be added to page 66 regarding the defining of the files for report(s) using the MERGE facility:

The MERGE parameter requires that both the main/hit file and the merge file are defined by a READ parameter and associated dictionaries as:

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR



This group of parameters will be active for all subsequent report OPTIONS until one of the following three parameters is read:

1	#MERGE	to replace the original/previous MERGE parameter. It will still act on the same two files
2	#READ plus #DIC	to introduce a new pair of files and a MERGE parameter or to revert/change to simple file processing
3	#STOP	to end the run

Compatibility of output facilities with output options

Page 43 and page 50 should be replaced by the new page 43 and pages 50 to 50.2.

Replacement pages contain new version of tables and figures which give a better representation of both LIST and TOTAL OPTIONs when output is to magnetic tape or exchangeable disc but do not override restrictions detailed under the parameters themselves or those enumerated in Appendix 3, page 124.

Facilities	*Options*					
	PRINT		LIST	TOTAL		TABLE
LP output	✓	✓		✓		✓
MT/Disc output			✓		✓	
Print two records per line		✓				
Print more than 1 line per record		✓	✓	✓	✓	
Merging		✓	✓	✓	✓	
Decoding		✓	✓			
Variable line spacing		✓		✓		✓
Optional headings	✓	✓		✓		✓
Automatic page headings	✓					
Automatic formatting	✓					
Printing variable length fields	✓	✓				
Arithmetic		✓	✓	✓	✓	✓
Column totalling		✓		✓	✓	✓
Row totalling						✓
Percentages						✓
Paper line-up by program		✓		✓		

Table 4 Compatibility of output facilities with output options

Headings

Headings are optional and may be specified for all printed output. Main headings will appear once only at the start of the tabulation; page headings will be repeated at the start of each page. Each type of heading is limited to 800 characters and so the maximum number of lines of heading which can be printed varies from 5 to 8 according to the size of the line printer.

Record selection

Various types of record selection are available so that results can be presented as efficiently as possible.

ENQUIRY SATISFACTION

Records may be selected for output on the basis of enquiries which they have satisfied. This applies only when output is from a hit file, where each record has the hit word(s) attached.

FIELD VALUES

With the TABLE option, records may be selected on the values of X and Y control fields.

MATCH CONDITIONS

When merging a main and secondary file on input, the user can specify that records be output when either a match or non match is obtained.

PART-FILE ACCESS

With this facility, the user can specify that only records from a certain part of his file are to be output.

Merging

The user may merge two files together to combine the information into a new file or report. Merging takes place at field level and it is possible to produce a new record format for an output file comprising information from two input records held on the main and merge files. These input files can be either the user's files or files produced

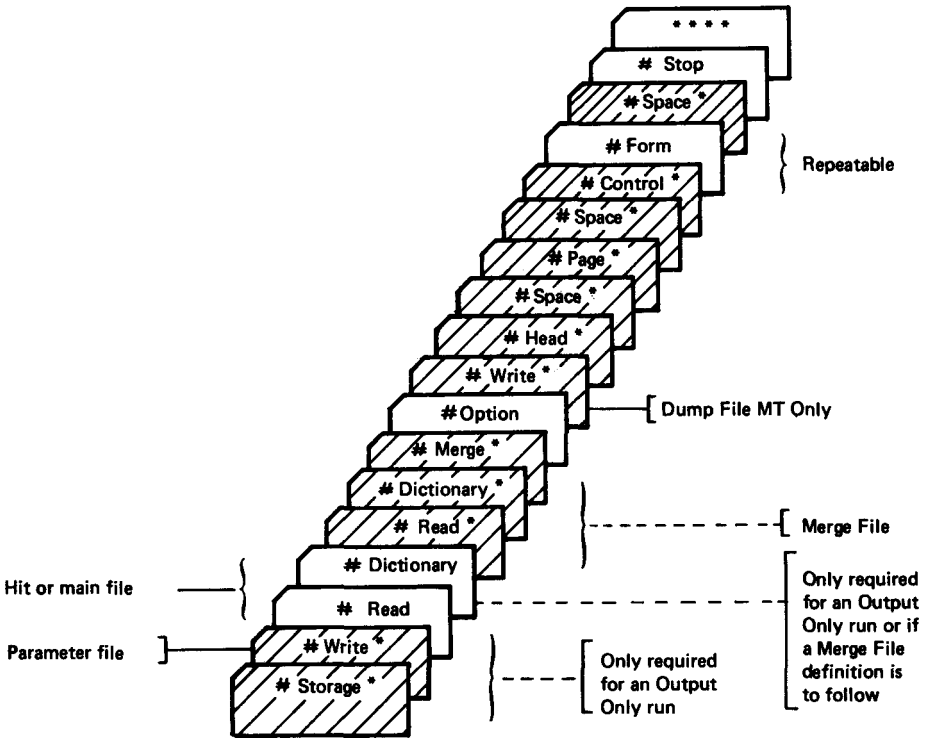


Figure 8a Output Phase List or Total (LP output)

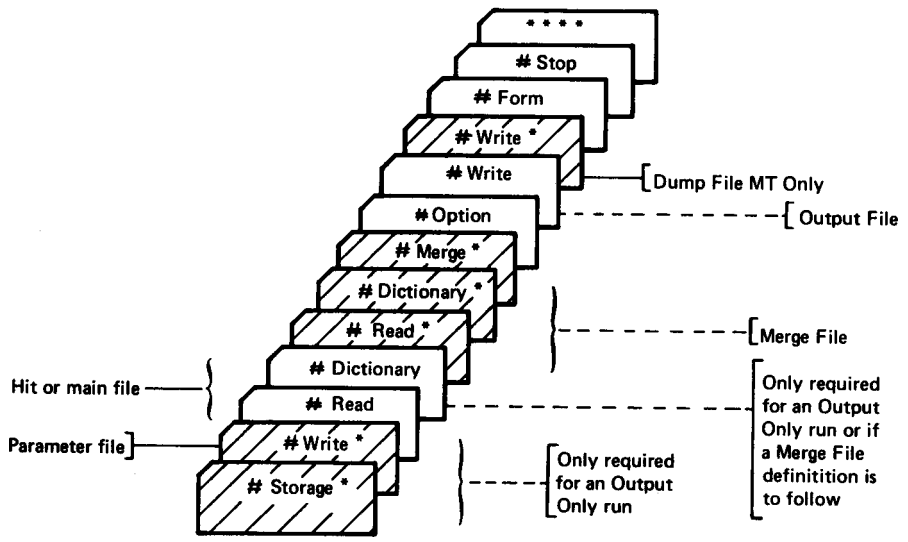


Figure 8b Output Phase List (MT or disc output)

(b.iii) TOTAL option (MT or disc output)

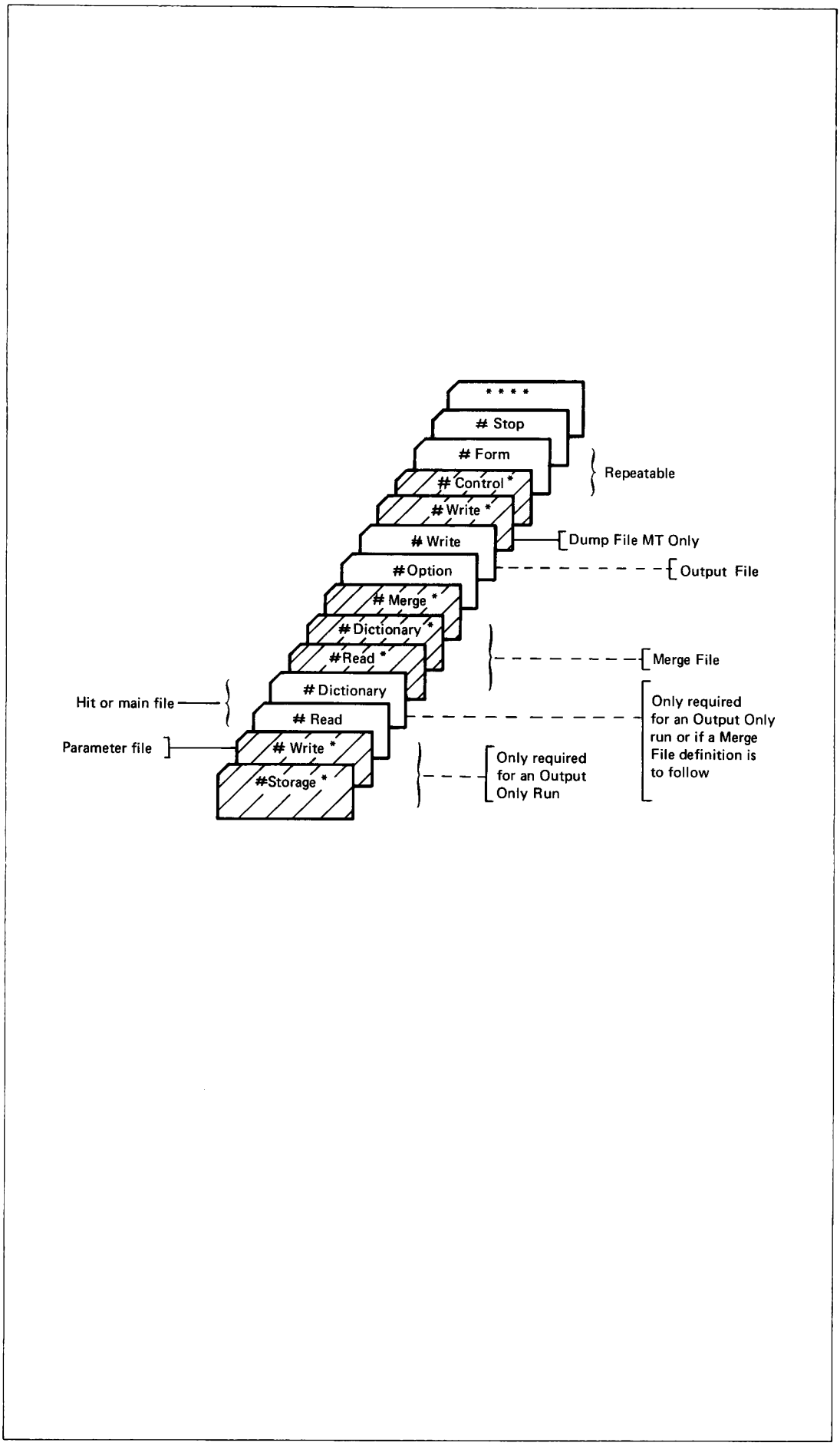


Figure 8c Output Phase Total (MT or disc output)

Line numbering under #HEAD and #PAGE

The following should be added to page 65:

Only a single numeric digit, in the range 1 to 8 inclusive, is allowed for the line number; the program will ignore any characters after the first non-space one encountered.

Page size

A further note should be added to the specification of the #OPTION parameter (page 67) to the effect that:

If a page size has been specified in an OPTION parameter, but a printer control loop is used with a smaller page size, then the loop takes precedence.

#YCONTROL in the TABLE OPTION

The following note replaces the sentence immediately before the examples on page 74.1:

The program is pre-set to accept up to 27 field values (or enquiry identities) for an unsorted file and the rules for their specification are the same as those for the XCONTROL parameter. If more (or fewer) values are required for any report, the allowable number can be varied by using the STORAGE parameter (see pages 71, 93 and following) to reset the number of entries in areas 19 and 20.

Note: If no values are being specified for a TABLE report, a single entry is still required in both areas 19 and 20.

Area sizes controlled by #STORAGE

Two additional notes should be added to Table 7 on page 94.

Code number 9:	This area is also used in conjunction with the outputting of variable-length fields. If such fields are to be output then the area should not be reduced to less than 10 words
Codes 19 and 20:	If these entries are changed then both should be changed to the same value. If no Y control field values are specified for a TABLE, a single entry is still required in both areas (that is, they cannot both be set to zero in an attempt to save space) to accommodate the accumulation of the current control value totals ready for printing

Program X63E - decode file identification

The device identity of the decode file (page 110) should be EDO.

Character constants in output options

Item 4(p), page 124 should read:

*(p) Store space is available for between 400 and 700 characters of constant information to be put in the output print lines (or the output record) from each input record (and can be changed by the use of #STORAGE).

In addition to the constants in the format line, as defined on page 64.4, zeros (for example, 4Z) when output is to a line printer and spaces (for example, 7B) when output is to magnetic tape or disc will be treated as constants.

The number of characters of constant information per line (that is, per #FORM group) is limited (and cannot be changed by the use of #STORAGE) to between 200 and 300 characters. This is adequate for line printer reports but may necessitate the use of several #FORM groups for a report being output to a magnetic tape or disc file

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (16) 12/9/79

File one copy of this notice with each of the publications indicated

New Enquiry types

The two new enquiry types, TERM and GROUP, described in Amendment list 3, are not yet available. They will be available in a new release of the system in the near future.

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

International Computers Limited

User notice

PUBLICATION (NOTICE NO.)

4187 FIND-2 MULTIPLE ENQUIRY SYSTEM (16) 12/9/79

File one copy of this notice with each of the publications indicated

New Enquiry types

The two new enquiry types, TERM and GROUP, described in Amendment list 3, are not yet available. They will be available in a new release of the system in the near future.

Issued by UK Software & Literature Distribution Centre, 60 Portman Road, Reading, Berkshire RG3 1NR

ICL

Find-2 Multiple Enquiry System

1900 Series

The policy of International Computers Limited is one of continuous development and improvement of its products and services, and the right is therefore reserved to alter the information contained in this document without notice. ICL makes every endeavour to ensure the accuracy of the contents of this document but does not accept liability for any error or omission. Any equipment or software performance figures and times stated herein are those which ICL expects to be achieved in normal circumstances. Wherever practicable, ICL is willing to verify upon request the accuracy of any specific matter contained in this document.

Technical Publication 4187

First Edition December 1969

Reprinted July 1970
(incorporating Amendment List 1)

Issued by Technical Publications Service
International Computers Limited,
Head Office: ICL House, Putney, London SW15
Produced by ICL Printing Services
at Letchworth, Hertfordshire

Preface

FIND–2 (File Interrogation of Nineteen hundred Data) is a general purpose information retrieval and reporting package designed for use on customers' 1900 Series data files held on E.D.S., twin disc, F.D.S. or magnetic tape. It has been developed from the FIND–1 system, the original FIND package, and incorporates much of the experience gained from users of that initial package. FIND–2 offers many new facilities for both the interrogation of files and presentation of report information.

Two systems are available within the FIND–2 package, known as the Single Enquiry System and the Multiple Enquiry System.

The Single Enquiry System has been designed so that it can be used by non-computer specialists and will be particularly useful for handling ad-hoc enquiries. It deals with one enquiry at a time, includes all the interrogation facilities and produces results directly on to the line printer, automatically formatted by the program. It requires an 8K central processor. A separate manual is available for this system.

The Multiple Enquiry System handles batches of up to 96 enquiries at a time. The system provides extensive report generation facilities which may be used independently of the enquiry facilities, and allows greater freedom in the presentation and manipulation of information. The scope of the system can also be extended by the introduction of users specialised routines at controlled points in the program.

The minimum configuration for the Multiple Enquiry System is as follows:

A 16 K central processor

A card or paper tape reader

A line printer and

3 magnetic tape decks or

One E.D.S. or twin disc drive

This manual deals specifically with the Multiple Enquiry System and has been organised into the following three parts:

Part 1, an introduction to the Multiple Enquiry System is written particularly for the user who knows little or nothing about computers. It outlines information retrieval problems and shows how this system offers a solution.

Part 2 contains details on file conventions and enquiry and output specification. It includes complete information on parameter formats and gives examples of their use.

Part 3 provides the necessary information for running the system in the computer installation.

For users with a minimum of 16K core store, FIND–2 will replace completely the original FIND package. The Single Enquiry System manual summarises the differences between the single enquiry version of FIND–2 and the original FIND package. Appendix 1 of this manual compares the Single with the Multiple Enquiry System.

Documentation available on FIND–2 comprises:

Single Enquiry System Manual T.P.4155

Multiple Enquiry System Manual

Publicity Brochure

Format Reference Card

Contents

Illustrations

Tables

ICL

Find-2 Multiple Enquiry System

Part 1 Introduction

4187(12.69)

Chapter 1 Introduction

As today's organisations diversify and increase in size and complexity, so the need for information becomes greater. Decision-making and control grow more important, yet more difficult as undertakings expand and control can only be exercised efficiently when those in authority are in contact with the facts. Possession of relevant information is essential if contact is to be maintained within the various departments of any organisation.

Information is required at all levels within any organisation, by manager or clerk, machine operator or administrator. In general the higher the level of management concerned, the more summarised and better organised the information must be. At a lower level, information is generally more specialised, that is, relating in great detail to a small area of activity, and also more immediate, or near to the event described, for example, invoices and stock records. The manager or chief accountant tends to require information that has been selected and organised by others.

As well as being provided at various levels, information can vary according to the basis on which it is prepared. At least three bases can generally be discerned, regular, ad hoc and exception. Regular information is required and provided on a regular basis and the frequency of its provision is unaffected by either external influences or the information itself. An example of this is the sales report, which is usually based on a fixed interval of a day, a week or a three-monthly trading period. This regular information, when compared with previous information with the same basis permits the calculation of trends, cycles and seasonal fluctuations. Ad hoc information, such as the result of a market research project, is provided at a time decided by those who request it. It often deals with one specific activity which may be a unique, 'one-off' job. Information provided on an exception basis is produced when 'actual' varies from 'standard'. It is usually provided for a managerial level so that a policy of management by exception can be carried out. This type of information is very useful in financial management where budgeting and cost accounting are efficient.

At whatever level and however frequently information is provided, it is valueless unless it informs the recipient. To do this, certain criteria are essential. It must be timely, accurate, relevant, accessible and well organised. As well as outlining standards for useful information, these criteria also point to the various problems associated with information retrieval and provision. Information must be timely, that is, if it is meant merely to inform the user, it must be provided as near the events described as to be meaningful; if it is meant to be acted upon, the opportunity for action must still be open. Accuracy of information is an essential quality, without which it is meaningless. This is not to say that approximation is not acceptable, but information must be accurate to the degree of detail required. Relevance and accessibility of information help to make the user and the organisation he serves work more smoothly so that expensive and unnecessary delays while the relevant information is located and extracted do not occur. Good organisation of information ensures that the user can quickly absorb and, if necessary, act upon facts provided.

The problems inherent in information retrieval and provision become apparent when all these criteria need to be satisfied simultaneously. Often, some standards are dropped so that others can be maintained. A sales report may be required immediately. It is prepared quickly by a clerk who miscalculates some figures and does not have the time to sort the data under appropriate areas. The result is that accuracy and presentation have been sacrificed for the sake of speed. Often in business if a financial report is to be accurate it can only be provided some long time after the transactions took place or else the information concerned has to be tediously extracted from a huge mass of irrelevant data. This is where the use of computers comes into its own. The ability of electronic data processing to process accurately large amounts of data at high speeds is of great importance and utility to the majority of today's organisations. Computers can be programmed to extract relevant information, according to predetermined criteria, from a whole mass of data held within the organisation. Extraction is performed on a logical basis and calculations can be carried out upon the data retrieved before the results are output, as either a printed list, a table or a file for further processing.

Chapter 2 Introduction to the package

The FIND–2 Multiple Enquiry System offers a solution to many of the problems associated with information retrieval and presentation. It is a general purpose package which can handle these problems at computer speeds without programming costs. This chapter attempts to illustrate this ability by introducing the package and summarising the facilities offered. Details of the retrieval and report facilities are covered in Chapters 6 and 7 on pages 27 and 42 respectively.

The package is designed for use on customers' own data files set up on E.D.S., twin disc, F.D.S. and magnetic tape conforming to 1900 Series housekeeping standards. Users' files are not overwritten or interfered with in any way and therefore no special security methods are necessary. Both the type of backing store on which the file is held and the record format are specified to the package at the time of the run thereby increasing the operational flexibility of the system.

The package provides facilities for interrogating these files based on criteria specified by the user through a simple enquiry language; reporting for regular or one-off requirements; and the reorganisation of files by combining and decollating information together with the reformatting of record content, for further processing by FIND–2 or the user's own programs.

It is recognised that a general purpose package cannot anticipate all eventualities in the hardware configurations to which it could be harnessed. FIND–2, however, attempts to push back these boundaries by allowing the user to alter the sizes of various areas within the program without recompiling, thus optimising use of the available core store, and by enabling the user to insert his own routines to adapt the package to meet a much wider range of requirements.

RETRIEVAL

The enquiry facilities provided by the Multiple Enquiry System are similar to those provided in the Single Enquiry System except that batches of up to 96 enquiries may be processed simultaneously with output to an intermediate file, called the hit file, instead of direct output to the line printer. The following types of enquiry can be specified, all of which may be combined within one batch.

1 STANDARD ENQUIRY. This is a straightforward type of enquiry whereby records satisfying criteria in the question are output to a hit file. This type of enquiry is used where information from the records is required to answer the following type of query:

   Have we anyone on the personnel file who speaks fluent French and German with sufficient managerial experience to take over our European Sales office?

   or

   Which parts have run below their minimum stock levels in the past 6 months?

2 COUNT ENQUIRY. This is used to count the number of records on the file which satisfy the question, when further details are not required. For example:

   How many accounts have we which are overdrawn, or overdue?

   or

   How many orders have we taken for product X since the advertising campaign was stepped up?

3 LEAD ENQUIRY. This facility is designed to decrease interrogation time when several enquiries have a condition or conditions in common. If a LEAD enquiry is a hit, the result is as if it were a STANDARD enquiry, in that a bit in the hit word is set and a hit total is accumulated. If the enquiry is not satisfied, then the system immediately branches out of the enquiry program to read the next record. That is, all subsequent enquiries in the batch are assumed to be redundant for this record. As an example, a LEAD enquiry may be used to precede enquiries on a file which contains different record types where only selected types are to be considered in depth, or to precede a set of enquiries which fall within certain dates or relate to particular projects.

Enquiry facilities

FIND–2 offers the user close control over the analysis of files owing to the many interrogation features provided.

1 ACCESSING PART OF THE FILE. It is possible to access just the relevant part of a file during interrogation with a resultant saving in processing time. This can be accomplished on sequentially indexed disc files between two record keys whilst on magnetic tape the enquiry may be directed to between two sentinels or to one specific reel of a multi-reel file.

2 ENQUIRY LANGUAGE. The enquiry language has been modified to make it more mnemonic and conform more readily to natural English. Several 'shorthand' facilities have been provided to ease the specification of enquiries by enabling a series of relationships to be generated by program, and by allowing several constants to be specified within one condition. The facilities for analysis have been enhanced providing:

   (a) A step by step search through a field when the exact position of the required information is unknown.

   (b) Selection of records satisfying a specific number of possible conditions.

   (c) Addition and subtraction of two fields before making the comparison e.g. .WAGE -TAX > £1000.

   (d) Inexact or incomplete constants to be specified by ignoring unknown or unwanted characters e.g. NAME = SM*TH

3 TYPES OF DATA. The types of data which may be interrogated range from a string of up to 72 characters down to a single bit. Numeric information held in various binary forms can be compared including integers and fractions, sterling (existing and decimal) and binary dates. Variable length fields can also be interrogated and printed.

4 PRESERVING THE ENQUIRY PROGRAM. The enquiry program generated from a batch of enquiries may be preserved on direct access or magnetic tape together with other parameters for re-use where regular jobs are being processed. This feature provides a convenient means of storing parameters and in addition enables the constants in the enquiries to be amended with the minimum amount of re-specification.

OUTPUT AND REPORT FACILITIES

The system permits reports to be produced on the line printer. Output can also be directed onto a direct access device or to magnetic tape for further processing or storage of historical data.

The modulator design of the package enables the output phase to be used completely independently. Hence output can be produced directly from a user's file or from a hit file produced during interrogation or from both simultaneously.

Format flexibility

The reports which may be produced range from a simple listing requiring minimum specification with formatting carried out by program, to a detailed tabular analysis containing row and column totals together with percentages and arithmetic calculations. Reports containing summarised information can also be created for higher levels of management by printing only totalled information in the form of lists or one page summaries. Information from one record may be organised into several lines of print as required. The spacing requirements from line to line may be individually specified for output to pre-printed stationery.

In addition up to two records may be printed per line (together with multiple lines per record) providing an efficient means of printing such items as internal telephone directories, mailing envelopes, etc.

Output facilities

The following facilities together with the above formatting possibilities, provide sophistication in presentation which up until now has only been associated with a specialised print program.

HEADINGS

Headings may be printed at the beginning of the report together with a separate page heading at the top of each page. Control totals may also be preceded by their own special annotation.

RECORD SELECTION

Record selection may be additionally carried out at the report stage based on the enquiries satisfied, the values of control fields, accessing part of the file or on the basis of a match between two files.

MERGING

Two files may be input to produce a further merged file by combining two records into one, or to produce a report. This facility adds a further dimension to the package. For example, a customer name and address file can be brought in when processing a transaction file to produce invoices, or several separate files may be combined into one integrated file or data base by repeated runs.

DECODING

Reports can be made more readable by translating coded information, which is often employed to economise on storage. For example, a personnel number may be converted to a name, or a part number to a description. A further use of this facility may be found in routine data processing operations where a coding system is being changed. An updated file could be produced substituting the new codes in place of the existing ones.

TOTALLING AND ARITHMETIC OPERATIONS

Totalling is available for producing up to eight levels of column total on twelve fields. Alternatively a total across a row or line of print may be produced. In addition a counting facility is also provided which can be used independently to print the number of occurrences of items on a file. Arithmetic operations can be performed on numeric data, for example, calculation of a price by multiplying a quantity by a rate. Finally a means of producing percentages is available with some forms of output.

EDITING

Editing of output fields is provided for the normal requirements of signing, floating £ and $ symbols and cheque protection.

DECIMAL CURRENCY CONVERSION

Decimal currency conversion is available on printed output for converting sterling to new decimal currency and vice versa. During the transition period, this facility will be particularly useful for printing an amount in one or both types of currency, on the same report.

ICL

Find-2 Multiple Enquiry System

Part 2 The System

4187(12.69)

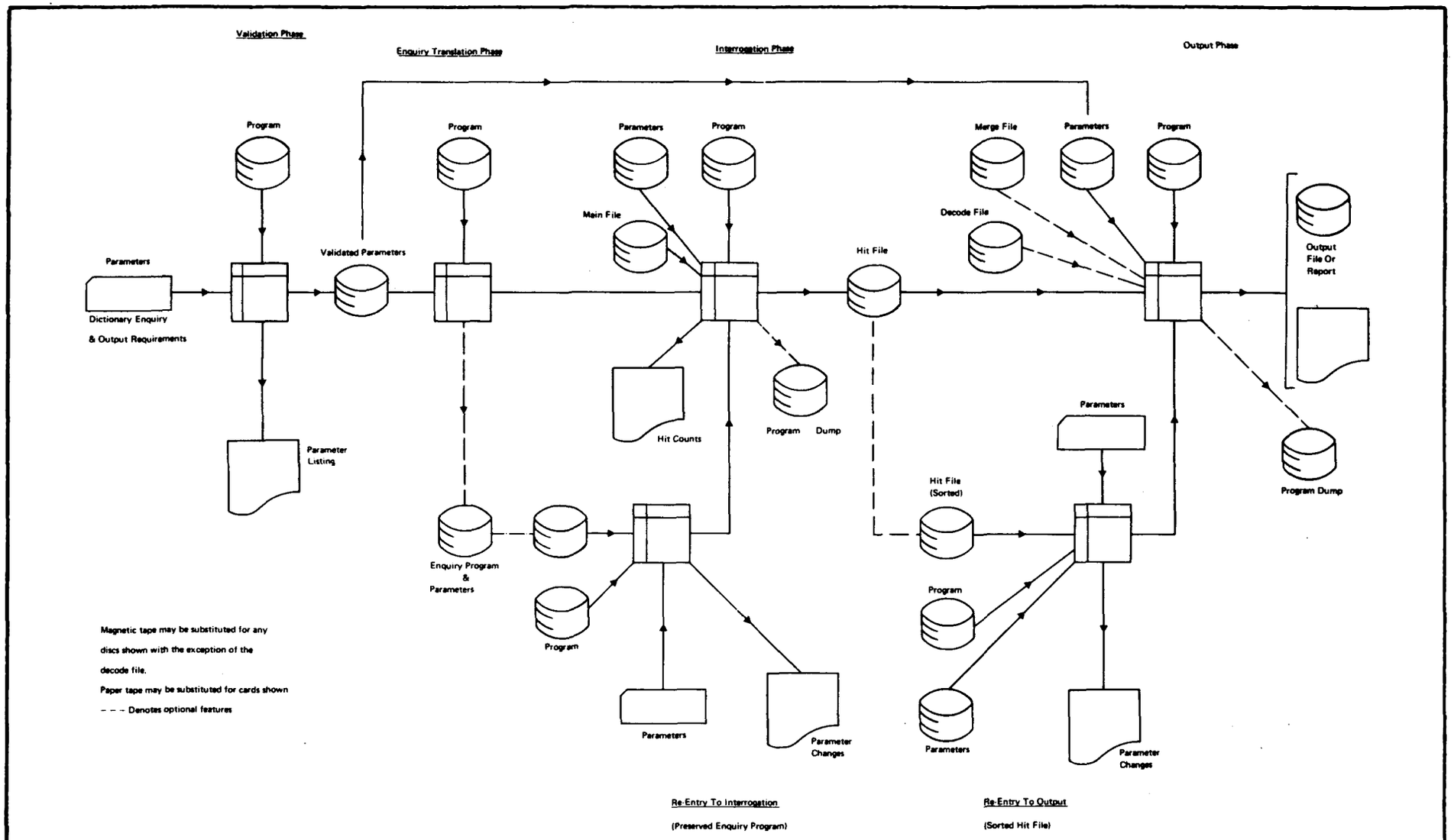


Figure1 FIND-2 Multiple Enquiry System

Chapter 3 Details of the system

There are five phases within the system as outlined below and shown in Figure 1 on page 8.

Validation	This phase checks the user's input and creates a file of validated parameters.
Enquiry translation	The file produced by the validation phase is translated into one large enquiry program containing all the enquiries.
Interrogation	The enquiry program generated in the previous phase is used to interrogate a master file. Records which satisfy any enquiry are written to a hit file.
Output	Information from the hit file or main file is output onto magnetic tape, a direct access device, or as a listing on the line printer.
Re-entry	This phase enables the user to re-enter the system. This may be required when using enquiry or output details preserved on a parameter file from a previous run, for example, after sorting a hit file to produce further output in a different sequence.

The function of each of these phases and the method of processing data within each phase is described below. At the end of this chapter there is a description of decode file updating which is a separate program available to users of the system.

VALIDATION OF PARAMETERS

The function of this initial phase is to check all parameters input by the user before commencing processing and to create a file of validated parameters on magnetic tape or a direct access device.

The user specifies the enquiries and the information to be output from records satisfying the enquiry conditions in the form of parameters punched into cards or paper tape. This input, together with details describing the file is checked at the initial validation stage so that any errors can be notified before processing begins. As a matter of course, only correct enquiries are processed in the next phase. Parameters, once checked, are written to a work file. If the user wishes to retain this parameter file, it must be named, otherwise it will be scratched at the end of the run.

At the beginning of this phase it is possible to reorganise or extend the core requirements of the program by use of the #STORAGE parameter described in Chapter 8 page 71. This means that the user can control at run time the way in which the program distributes the available store between certain major areas, for example, the input or output buffers.

At the end of the validation phase, the program halts if errors have occurred, otherwise control is passed to either the enquiry translation phase or the output phase, depending on the parameters specified. The input peripheral is then released.

ENQUIRY TRANSLATION

This phase translates the enquiry tables produced by the validation phase into one large enquiry program. If the user wishes to retain this program, it may be written onto the parameter file.

INTERROGATION

During this phase the generated enquiry program is used to interrogate the whole or part, of a master file held on a magnetic tape or direct access device. The user of the Multiple Enquiry System has a choice of the three enquiry options described in Chapter 5 page 17.

Processing within the interrogation phase is as follows. Records are input from the main file and each is

interrogated by the generated enquiry program already in store. On sequential disc files, tags are followed to overflow records to preserve file sequence. Random disc files are processed serially.

For standard enquiries, when at least one of the enquiries is a hit, a special word known as the *hit word* is added to the end of the input record and the record is output to the hit file. The hit word is used later in the system to check the enquiries satisfied by the record. One hit word is added to the record for each set of 24 enquiries. (4 hit words for 96 enquiries). Where only one enquiry is being processed, the hit word is omitted enabling the output of a hit file in identical format to the original file.

At the end of the run the line printer is allotted to print the total number of hits for each enquiry, the total number of records interrogated and percentage hit rate for each enquiry. If required, it is possible to terminate the run before reaching the end of the file, and output this control information. The program either halts at the end of the run or enters the output phase, depending on the paramters specified. If no hits are achieved the run is terminated without control passing to the output phase.

Dump and restart facilities, which are included in this phase, are described fully in Chapter 13 page 115 .

OUTPUT

The function of this phase is to organise the output of information from the hit file or main file onto magnetic tape or a direct access device, or as a listing on the line printer.

Processing is carried out as follows. A group of parameters is read into store from the parameter file, specifying requirements for producing one report or file. The parameter file may contain any number of groups of parameters relating to different output requirements. Records from the hit/main file are selected for output depending on the selection facilities specified on the parameters. Note that when merging, the hit/main file is the file defined first in the output parameters. At this point the nature of processing depends on the output option specified and the facilities being used. The four output options available to users of the Multiple Enquiry System are Print, List, Total and Table.

1 PRINT With this option, the specified fields are automatically formatted with column headings generated from their names. A grand total is automatically accumulated for each numeric field. One line of print is output for each record.

2 LIST and TOTAL In the LIST option fields selected by the user are output as lines of print, or records to magnetic tape or disc, in a format specified by the user. If output is to the line printer, details from one record may be printed over several lines, with variable spacing between the lines. It is also possible to print two records on one line and these two facilities may be combined together.

The results of arithmetic operations between fields or between fields and constants may be output. If output is to the line printer, fields and arithmetic results may be totalled. These totals may be used to provide overall totals at the end of the run (called *GRAND* totals) and sub-totals at up to seven control levels. One control field is associated with each level of sub-totalling. When the value of a control field changes, all totals associated with that level and all higher levels are printed. The control field values and their associated record counts may also be printed. After a control break the normal output listing process resumes with the appropriate record counts and sub-totals reset to zero.

In the TOTAL option output is only produced at a control field change giving details of control field values and sub-totals. This option is therefore useful for summary reports in which the details of individual records are not required. Output may be to magnetic tape, disc or the line printer.

Both the LIST and TOTAL options include the facility of merging two input files to produce one output file or report. The two input files must be sorted in ascending sequence on a common key. The facility may be used in two ways depending on whether output is required when keys in the two files *MATCH* or when the keys in the two files differ. The second case is called a *NON-MATCH.* When matching is specified fields may be output from both matching records. If non-matching is specified, output will only be from those records on the *MAIN* file for which there is no matching record on the *MERGE* file.

In the LIST option, fields held on a file in coded form may be decoded on output. This is achieved by means of referencing a disc decode file which holds all coded and decoded values of a particular field.

3 TABLE This option enables a two dimensional table to be output on the line printer. Control fields are specified on two axes, X and Y, which are used to determine the position of the output information across (X) and down (Y) the page. Unlike the previous three options which print information either record by record or when a control field changes, the Table option accumulates totals in store until they are to be printed. The

overall operation differs according to whether the file is sorted or unsorted.

(a) SORTED. When the input file is sorted, values for the Y (vertical) control fields do not have to be specified and an unlimited number of lines may be output with totals being printed at any change of value in the Y control field. This differs from output produced with the Total option in that the X control fields determine where information is to be placed along the line. The Table option prints each line of a sorted file as it is formulated.

(b) UNSORTED. With unsorted input, results are accumulated in store and printed at the end of the file. Again the values of X and Y control fields found in each record determine whether the record should be output and the position of the fields to be printed.

RE-ENTRY

The package allows the user to re-enter the system using a previously validated parameter file. Entry can be to the interrogation phase, for example to repeat a regular enquiry, or directly into the output phase, for example to print a sorted hit file.

A parameter specifying the name of the parameter file is input on cards or paper tape to enable the parameter file to be opened. As the option of naming this file is provided, on re-entry to the system the name of the file containing the enquiry program and/or the validated parameters must be given.

1 On entering to run a previously generated enquiry program, the enquiry program is reloaded. The header details of the file to be interrogated may be re-inserted to amend the file generation details. Constants which have been defined to the system in a special way as shown in Chapter 8, page 58, may also be amended. After this has been done, control is passed to the interrogation phase.

2 When re-entering the output phase it is necessary to specify which group of output parameters is to be used. The option of amending the file labels for both the hit/main file and the merge file is also available. Details of merging and other output options are given in Chapter 7. The files are opened and control is passed to the output phase for processing in the normal way.

DECODE FILE UPDATING: PROGRAM #X63E

This is not one of the five component phases of the Multiple Enquiry System but is a separate program. It is provided for updating or amending the decode file once it has been created with basic software.

The illustration, Figure 2, on page 12 outlines the system.

The program, although intended primarily for updating FIND-2 decode files will also amend any indexed sequential E.D.S. or twin E.D.S. file conforming to 1900 Housekeeping standards.

Amendments consisting of deletions and/or additions are input on cards or paper tape, the parameters conforming to FIND-2 standards. The file is updated in overlay mode and the amendments are listed on the line printer. Error flags show if any amendment is not accepted. At the end of the run a count of the records in the file is output.

The decoding facility in the Multiple Enquiry System will not handle files containing first level overflow, so if this condition occurs during amendment a message to this effect is output on the line printer. The decode file should then be reorganised using a standard reorganisation program generated by #XJEZ. Details of this program may be found in the ICL 1900 Series manual *Library Specifications*, technical publication 4011.

Although further additions can be made to a file which already contains overflow, the warning message will only be given when records are inserted into the first level overflow area. Thus if this area is already full, and records are added to the second level overflow area, the message will not be output.

To reduce the possibility of overflow it is recommended that deletions be made before additions.

#X63E can be used to create a decode file if index buckets have been set up with suitable keys. Failing this, library programs #XREB, #XREJ or the File Organisation program may be used.

A complete or selective print of the decode file can be made using the FIND system.

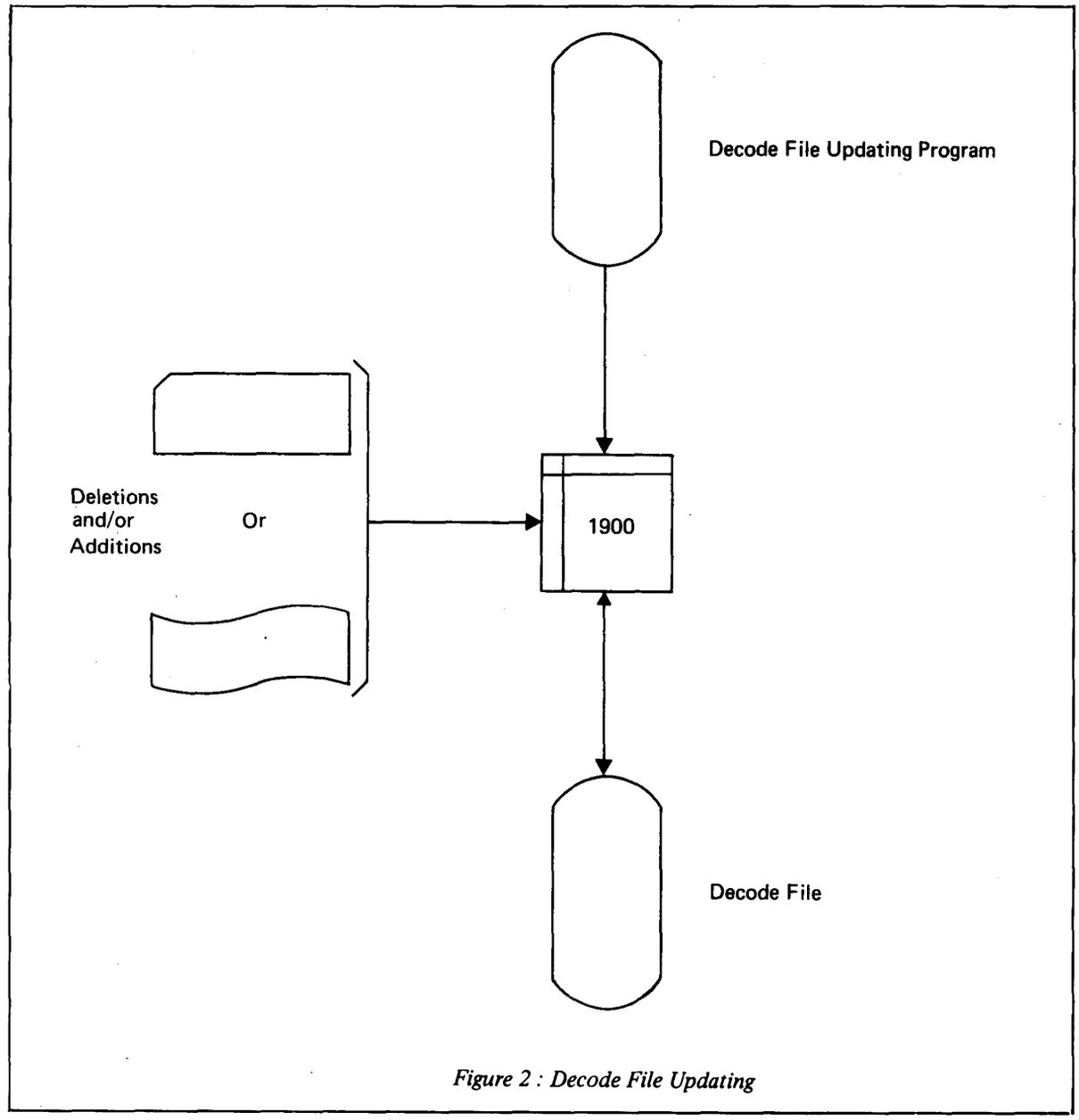


Figure 2 : Decode File Updating

Chapter 4 Input, Output and system files

There are seven files used or produced by the FIND Systems, of which four are optional. They are:

File	*Interrogation phase*	*Output phase*
Main Input	✓	✓
Merge *		✓
Output *		✓
Hit	✓	✓
Decode *		✓
Parameter	✓	✓
Dump *	✓	✓

* = optional file

The main input files may be input to the Single and Multiple Enquiry Systems. The remainder are exclusive to the latter. All files must comply with ICL 1900 Series Housekeeping Standards and may be held on either E.D.S., twin disc, F.D.S. or magnetic tape with the exception of the decode file which must be held on E.D.S. or twin disc and the dump file which may not be held on F.D.S.

The Multiple Enquiry System will provide 1024 word buffers for the main, hit and output files and the area will be used to double buffer files with blocks of 512 words or less and to single buffer block sizes in the range 513 to 1024 words. It is possible to increase the buffer sizes at run time to handle double buffering on larger block sizes.

The format and record structure of the seven files are described below.

MAIN INPUT FILE

This file, containing all the user's data, may be input to either the interrogation or output phase. The Multiple Enquiry System cannot alter or overwrite this file, thus no special security is necessary to protect the user's information.

File Format

MAGNETIC TAPE

On magnetic tape the format will be to ICL 1900 Series Housekeeping Standards, for example:

HDDR	TM	SDS	Data blocks, optional user tape sentinels and dump sentinels	TM	EOF or EOR

where

HDDR	=	Header Label
TM	=	Tape Mark
SDS	=	Start of Data Sentinel
EOF(R)	=	End of File (Reel)

Multi-reel files are handled. User tape sentinels are skipped except where the package has been directed to access a part of the file between two sentinels. The program will also skip over program dumps if they occur on the main

file. The data is processed serially on magnetic tape.

DIRECT ACCESS

Files held on this device may be serial, sequential or random. Serial and random files will be processed serially, sequential files sequentially. With the latter, the file indexes can be used to reduce the area of search during interrogation or input. Multicartridge files are acceptable where no other file required by the system is sharing the same cartridge.

Record Structure

Records may be of either fixed or variable length. The maximum record sizes are as follows:

Magnetic tape	1024 words
E.D.S.	1022 words
F.D.S.	1021 words

Variable length fields must be positioned at the end of the record following any fixed length fields. The Multiple Enquiry System will interrogate and print both types of field. Different record types may occur on the same file. Files structured on the basis of header and trailer records will not be handled directly by the package but may be manipulated through the user entries. Further details of the types of field handled by the package and the way in which the record format is defined to the system are given in Chapter 5 page 20 on File Definition.

MERGE FILE

This file is input to the output phase when using the merge facility. Details of this file are identical to the main input file except that files structured with header and trailer records cannot be handled. It is not possible to access part of the file through tape sentinels which are completely ignored or via the files index on sequential disc files.

OUTPUT FILE

This file is produced by the output phase.

File Format

MAGNETIC TAPE

On magnetic tape the format will be as follows:

HDDR	TM	SDS	Data Blocks	TM	EOF or EOR

Dump blocks containing program dumps may also occur between dump sentinels if a separate file is not provided when using the dump and restart facility. Multi-reel files may be output.

DIRECT ACCESS

On disc, multi-cartridge files may be created and will be serial. The user must allocate the file prior to the run and supply the file header details by means of a parameter, described in Chapter 8, at run time. The file area is extendable.

Record Structure

Variable length fields cannot be output. Using the merge facility,output information may be selected from the two input files. The Total option enables a summarised file of totals to be created. An output record may be up to 1024 words in size. The block size may be specified by the user and, if omitted, the main input file block size will be assumed.

HIT FILE

This file, containing the records which have satisfied one or more enquiries, is a product of the interrogation phase. If the file is output to a direct access device the user must allocate the file prior to the run. File header

details are then supplied to the package by parameter at run time. If necessary, the file area is extended during the run and it can be contracted to its original size at the end by means of a special entry point. If the file is output to magnetic tape, it can be relabelled SCRATCHVTAPE and given a zero retention period by means of a special entry point.

File Format

Format is identical to the output file. The output block size may be specified at run time but, if omitted, the input file block size will be assumed.

Record Structure

When only one enquiry is being processed the output record is identical to the input record and no hit word is added. When more than one enquiry is being processed one 24-bit hit word is added to the end of the record for each multiple of 24 enquiries in the batch. The user must specify the hit file block size if the addition of the hit words on the end of the record could cause the maximum record size to exceed the original block size of the main file.

DECODE FILE

This file, which must be held on E.D.S., holds the decode table to be used in the output phase. When the program is held on E.D.S. the decode file may be located on the same cartridge. Any number of decode files may be held on this cartridge for different FIND runs but only one file may be accessed in one run.

File Format

The file, created from cards or paper tape using the standard ICL utility routine #XJEZ, must be loaded sequentially with standard indexes. A program for amending any existing decode file is available within the FIND–2 system. After the decode file has been amended, it must be reorganised using #XJEZ if overflow has occurred. The size of the file is limited only by the available on-line E.D.S. capacity and may be set up with 1, 2, 4 or 8 block buckets.

Record Structure

The user may determine the length of records on the file. The code or record key may be up to 64 characters in length, and the decoded entry to be output which must immediately follow it in the record may be up to 127 characters in length. Records are stored in ascending sequence according to their code and must be of fixed length.

PARAMETER FILE

This file is used as a work file to hold the validated parameters and the generated enquiry program, when required. The name FINDWORKFILE is provided by the system but the user may specify a file name by parameter at run time. When the parameter file is on disc, the appropriate named file must be allocated prior to the run and for convenience may be held on the same cartridge as the program. The file area will be extended if necessary during the run and will be contracted at the end of the job. The file is given a retention period of 500 days for tape and zero for disc, unless it is specified by parameter, in which case the retention period may be supplied by the user. Unless this parameter is present, the parameter file will be created on the same medium as the program file. Blocks on a magnetic tape parameter file are 512 words long, and a parameter file on disc should be allocated with 4- or 8-block buckets.

DUMP FILE

This file contains information enabling a restart to be made in the event of a hardware breakdown or other unexpected halt. Dumps are usually made to a separate dump file. If the data is on magnetic tape, dumps may be made between dump sentinels on the hit and/or output file, if no run time parameter specifying a separate tape is provided by the user. The name ICT - DUMPFIND is provided by the system.

File Format

On magnetic tape dump blocks are up to 512 words in size. On direct access devices 128 word buckets are used. The format of dump files is described in detail in the ICL 1900 Series Manuals *Direct Access* and *Magnetic Tape.*

Chapter 5 File definition

This chapter deals with the four parameters which are used to describe or define files used by the system. They are:

#READ

#WRITE

#UPDATE (#X63E only)

#DICTIONARY

The user's data file is defined to the FIND-2 Multiple Enquiry System by the #READ and #DICTIONARY parameters. #WRITE is an optional parameter which is used to name an output file. Note that the #DICTIONARY parameter is used to describe fields in the input files; the format of any output file is described in the same way as output reports and is dealt with in Chapter 7, page 41.

This chapter deals with the format conventions for all parameters, and then with these three parameters in detail. It then gives a complete example of a file definition, including an example of a simplified Dictionary to be supplied to the user.

FORMAT CONVENTIONS

Each parameter is begun by a major directive distringuished by the hash mark (#) as the first character of the parameter. It is only necessary to specify the first four characters, including the # of the major directive, for example #DIC instead of #DICTIONARY, but it can be specified in full if required.

Within the parameters the fields are in free format. In general a comma or space may be used to terminate each field and any number of spaces can occur between fields. The #READ, #UPDATE and #WRITE parameters and the #FORM parameter dealt with in a later chapter are exceptions to this general rule. On the #READ, #UPDATE and #WRITE parameters the commas between fields are mandatory but any number of spaces may still occur between fields. For these two parameters it is unnecessary to specify trailing commas for optional fields at the end of a parameter if these fields are not being used. This will be shown later in some examples.

The last field within a parameter should be terminated by a space.

Leading zeros are not necessary in numeric fields and the positive sign is assumed if no sign is given.

Comments which are to appear in the parameter listing but are otherwise to be ignored may be inserted into the parameters in the form of cards or paper tape blocks, beginning with a hash mark (#) followed by at least three spaces.

Note: Where the word card is used in explaining the parameter formats in this chapter, paper tape blocks may be substituted.

The conventions concerning the use of brackets to explain the parameter formats in this chapter are shown below. The brackets are not punched.

1 Gothic brackets indicate that a choice should be made between the values bounded by the brackets.

Example [MT / ED / FD]

2 Round brackets indicate that the field between the brackets is mandatory in that parameter and that the value of the field should be inserted.

Example *(Filename)*

3 Square brackets indicate that the field is optional.

Example [*Reel sequence number*]

#READ

This parameter names the files to be input to the system. It also specifies the bounds of any part-file interrogation and defines the dump frequency.

Parameter format

The general format of the #READ parameter is as follows:

#READ, [*MT* / *ED* / *FD*], *(Filename)*, [*Reel sequence number*],
[*File generation number*], [*Dump frequency*],
[{*Part-file specification* / *Output group number*}]

The major directive, #READ, may appear on the same card as the remainder of the parameter or may occupy its own. Any number of cards may be used for remaining items but one item must not be split between two cards. The appropriate device type must be chosen and inserted after the #READ. ED means both exchangeable and twin disc. The file name should be twelve characters long. If less than twelve characters are punched before the comma, the rest of the file name is space filled. Any characters over these twelve are ignored. Leading zeroes may be omitted on all numeric fields.

For magnetic tape files the reel sequence number or file generation number may be omitted and zero will be assumed. On disc files the reel sequence number should be omitted and the file generation number may be specified as -1 if it is required to interrogate the highest generation of the file on-line to the system. The commas between fields are mandatory even if the fields are absent.

The frequency of dumping can be specified as follows:

1	No or field blank	No dumping will occur.
2	EOR	Dumps to occur on reaching the end of each reel of the input file except for the final reel.
3	*n*(thousand)	Dumps to occur after processing *n*thousand records on the main input file.

Dump and Restart is covered in more detail in Chapter 13, page 115. Dump frequency is specified for the main input file only.

PART FILE SPECIFICATION

The specification of part-file interrogation differs according to the type of main input file. If it is required to interrogate the complete file, this field should be left blank.

Direct access

On direct access indexed sequential files interrogation may be performed between two record keys. The format is:

n – *m*

where *n* and *m* are values of keys within the file, for example, 1024 – 3428. If *n* and *m* are intended to represent the values of keys in character form they must be bounded by apostrophes, for example: 'A271' – 'C403'. Space, comma and hyphen are therefore permitted characters, but an apostrophe is not. If interrogation is to be from the start of the file *n* should be zero, and if interrogation is to be the end of the file *m* can be replaced by EOF. The representation of these two conditions is independent of the representation of the other key, for example:

0-3007 and 20-EOF or 0-'AA17' 'X61A' - EOF

FIND–2 will allow for character keys of up to 64 characters, in accordance with ICL 1900 Series Housekeeping

Standards. Binary values may be signed, for example, −213 – −104. and keys represented by binary values must be signed single or double length binary fields. The system assumes an ascending sequence of key values in all cases.

Each key may be specified on separate cards to allow specification of long character keys. In this case, the separating hyphen may appear on either card, but it must be present.

Magnetic tape

For magnetic tape files interrogation may be performed between two user sentinels. The format is:

n – *m*

and only the part of the file between the *n*th and *m*th user sentinels will be processed. If interrogation is to be from the start of the file *n* should be zero, for example, 0–4. If interrogation is to be to the end of the file *m* should be replaced by EOF, for example, 4-EOF. Similarly, interrogation to the end of the reel can be specified by EOR instead of EOF.

Output group number

When #READ refers to the parameter file at re-entry to the output phase, this field specifies the number of the output group to which re-entry is to be made. The field must be an integer in the range 1 to 20.

Example

```
#READ,  ED,  FILEONE,  ,  −1
#READ,  FD,  TESTFILEDATA,  ,  1,  NO,  '14THA'  −  EOF
#READ,  MT,  TESTDATA,  ,  ,  ,  2–4
```

#WRITE

This is an optional parameter used to give a name to a file being created by the FIND–2 system, for example, a parameter or output file. The parameter is limited to a single card. When referring to the hit file, the parameter may only be omitted if all enquiries are COUNT type.

The format for direct access and magnetic tape files differs. These are dealt with separately below.

Direct access

The format is as follows:

#WRITE, {ED / FD} , *(Filename), (Old file generation number),*

(New file generation number), (Retention period)

The appropriate device type must be chosen and inserted after #WRITE.

The file name should be twelve characters long. If less than twelve characters are punched before the comma, the rest of the file name is space filled. Any characters over twelve are ignored. Leading zeros may be omitted on all numeric fields.

The old generation number of the file must be specified. If it is sufficient to open the highest generation of the file on-line to the system, −1 may be specified.

The new generation number of the file must be specified and will be given to the file created.

The retention period is mandatory but may be specified as zero.

The block size of the file created is specified when the file is allocated by the allocator routine and is therefore not present on this parameter.

Magnetic tape

The format is as follows:

#WRITE, MT, *(Filename),* [*Reel sequence number*] ,

[*File generation number*], *(Retention period),*

[*Block size*]

The file name should be twelve characters long. If less than twelve characters are punched before the comma, the rest of the file name is space filled. Any characters over twelve are ignored. Leading zeroes may be omitted on all numeric fields.

The reel sequence number or file generation number may be omitted, in which case zero will be assumed. However, the commas between fields are mandatory even when the fields are absent.

The retention period is mandatory.

The block size is optional and may be specified for the hit file or an output file. The parameter file always has a block size of 512 words. If block size is omitted, it will be set to 1024.

Note: If the block size is not present in the start of data sentinel of the input file, an output block size of 1024 words will be assumed if not specified.

#UPDATE(DECODE FILE UPDATE ONLY)

This parameter names the file to be amended by the decode file update program, and is limited to a single card/ paper tape block.

The format is as follows:

#UPDATE, (File name), (File generation number), [Record length]

The filename should be 12 characters long. If less than twelve characters are punched before the comma, the rest of the file name is spacefilled. Any characters over twelve are ignored.

If the generation number is given as - 1 the file with the highest number on-line will be opened.

To conform to Housekeeping standards each record must contain the record length in its first word. If the records to be added to the file are all of fixed length then this length may be specified as the last field, but this is not possible if the records are variable length. Inclusion of the field eliminates the need for the length to be specified on each #ADD parameter.

#DICTIONARY

The #DICTIONARY parameter defines the fields on the main or merge input file, giving the name by which each field is referenced in enquiry or output parameters. Definition can start on the same card as the directive #DICTIONARY, or on a new card. The Dictionary may occupy any number of cards.

There are two general formats depending on whether the field being defined is of fixed or variable length. Though it is only necessary to define those fields required for any particular run, it is suggested that the whole file is defined. Thus a standard Dictionary once specified may be input for all runs on that file, until there are format changes.

Fixed length fields

The format is as follows:

(Field name), (Field type), (Start address), (Length)

Note: Any comma could be replaced by a space. The definition of each field in the record, comprising the above items, must not be split between cards.

FIELD NAME

The field name may consist of any number of alphanumeric characters, of which the first must be alphabetic and the combination of the first seven unique within the Dictionary. The field name will be terminated by the first space or comma character and should not therefore contain these characters. If the field contains less than seven characters, the field name will be space filled up to that number. If the field name contains more than seven characters, all after the first seven will be ignored.

START ADDRESS

There are three ways in which a start address can be written:

1 Defining just the word address, if a field starts in character position 0 of a word.

2 Giving the word address and character position, if a field does not start at character position 0.

3 Giving the word address and bit position, if a field starts at a particular bit not corresponding to the start of a character.

The following general format should be used for start address specification:

[W] *(Word address)* [{C / B} *(Character or bit position)*]

Note: The preceding 'W' is an optional descriptor.

Examples

W12	Character 0 of Word 12
W47C2	Character 2 of Word 47
17B11	Bit 11 of Word 17

Note: All addresses are assumed to start at 0, the first word of a record being word 0, the most significant character 0, and the most significant bit of a word being bit 0.

LENGTH

The field length can be given in one unit as either words, characters or bits, but not as a mixture. For most field types the choice is limited to one or two of these, see the next section under the heading *Field type.*

The general format for defining the length of a field is as follows:

(Length) [W / C / B]

where W denotes length in words, C length in characters, and B length in bits. If the code letter after the length is omitted then the lengths will be assumed to be in words.

FIELD TYPE

The field type code denotes the type of information held in the field, and can be one of the following single letters:

B, C, F, M, P, W, H, D

Note: Table 1 on page 22 shows the units in which the length and address may be stated, together with the maximum field lengths.

B A binary field not held as an integral number of words. Such a field could be used for storing a bit pattern in which each bit represents a particular item, or for holding a positive integer. Integers held in an integral number of words are specified using field types W, D or H as shown on page 22 .

Examples

RESULTS	B	W16C2, 12B (a 12 bit field starting at character 2 of word 16)
BITS	B	W94B15, 3B (a 3 bit field starting at the 16th. bit of word 94)

C A character field.

Examples

INITIALS	C	12C2, 1W	(a 1 word field starting at word 12, character position 2)
TYPE	C	14 1C	(a 1 character field starting at word 14, character position 0)
DATE	C	W6C3, 6C	(a 6 character field starting at the last character of word 6)

F A binary field, 1 word long, holding a positive or negative fraction.

Examples

RATIO	F	W10,1	(a 1 word fractional field starting at word 10)
RATE	F	14,1W	(a 1 word fractional field starting at word 14)

M A binary field, holding a positive or negative mixed number, of which the fraction part is 1 word long.

The complete length is either 2 or 3 words.

Example

QTY	M	W37,	3W	(a 3 word field comprising a 2 word integer and 1 word fraction, starting at word 37)

P A binary field, holding a positive or negative fixed point number, of which the fraction occupies less than 1 word.

The length is expressed in words and bits.

The overall length of the field is always 1 or 2 words, and the bit length gives the length of the fraction part.

In this format the 'W' in the length is mandatory.

Examples

VALUE	P	19,	1W6B	(a 1 word field holding a fixed point number with the fraction held in the last 6 bits of the word)
GROSS	P	W28,	2W12B	(a fixed point number held in two words with the last 12 bits of the second word holding the fraction)

W, D or H

Binary fields used to hold positive or negative integers. When defining values not relating to currency, field type W should be used.

When defining sterling values, the codes should be used as follows:

W = binary pence (sterling)

D = binary new pence (decimal sterling)

H = binary new tenths of pence (decimal sterling)

When defining decimal currency other than decimal sterling, field type D should be used. The holding units can be $1/10^{n}$ths of the basic unit, where n ranges from 1 to 6.

Examples

GROSS	W	W43,2	(2 words containing binary pence)
GROSS	D	18, 1W	(1 word containing new binary pence)
REMAIN	H	21, 1	(1 word containing new binary tenths of pence)
DOLLAR	D	25, 2	(2 words containing binary cents)

Code	*Field type*	*Start address*			*Length*			*Maximum field length*	*Comments*
		Word only	*Word and character*	*Word and bits*	*Word*	*Char.*	*Bits*		
B	Bit patterns Integers	✓	✓	✓	–	✓	✓	23 bits	
C	Character	✓	✓	–	✓	✓	–	127 char.	
F	Binary fraction	✓	–	–	✓	–	–	1 word	Fixed length
M	Mixed number	✓	–	–	✓	–	–	3 words	The integer may be 1 or 2 words
P	Fixed point number	✓	–	–	✓	–	–	2 words	Length expressed in words and bits
W,D,H	Integers	✓	–	–	✓	–	–	2 words	

Table 1: Summary of field types

Variable length fields

Variable length fields may be of type V or type X. There is no limit to their length other than that imposed by the maximum record length.

TYPE V (TERMINATOR TYPE)

This type of field is terminated by a specified character. A single character from the ICL 64 character set denoting the end of a field can be any character except a space. The terminator need not be the same for each field. The format is as follows:

(Field name), (Field type), (Number of preceding fields/Terminating character), (Address)

Notes:

1 Any comma can be replaced by a space.

2 The format of the field name is the same as for a fixed length field.

3 The number of variable length fields preceding the field being defined must be stated, and all these preceding variable length fields must be defined in the dictionary, even if they are not actually referred to in the other parameters.

4 The variable length fields defined in the dictionary must be defined in ascending order of their positions within the record, that is: 0, 1, 2, 3 . . . . *n*.

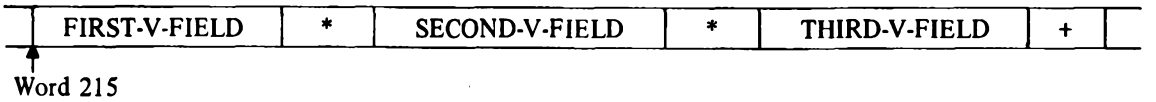
5 It is only possible to give the address of the first variable length field. The position of all other variable length fields will be found relative to this address. The format of the start address is the same as for a fixed length character field.

6 If it is required to omit a field within a record the terminator must still be present.

7 Up to 36 variable length fields may be defined. The two types of variable length field, V and X, cannot be combined in one file.

Example

The following section of file:



would be defined under #DICTIONARY as:

```
FIRST-V-FIELD   V   0/*  W215, SECOND-V-FIELD   V   1/*,
THIRD-V-FIELD   V   2/+
```

TYPE X (SUBRECORD TYPE)

Each variable length field of this type is preceded by a word containing the length of that field. Each field begins at a word boundary. The count word holds, in bits 9 to 23, the whole number of words including the count word itself, in which the field is contained. Bits 0 to 8 are not checked and therefore may hold user information.

The format is as follows:

(Field name), (Field type), (Number of preceding fields), (Address)

Notes:

1 Any comma can be replaced by a space.

2 The format of the field name is the same as for a fixed length field.

3 The number of variable length fields preceding the field being defined must be stated. These preceding fields do not have to be defined; the first one plus the one required is sufficient. They must, however, be present in the record, blank fields being indicated by a count word containing a count of one.

4 The start address is only required for the first variable length field, which must be defined. The format of the address is the same as for a fixed length field.

5 Any number of type X variable length fields may be defined. The two types of variable length field, V and X cannot be combined in one file.

Example

It is supposed that the user wishes to reference the second and fiftieth field on a file of type X variable length fields. The first field begins at word 215. The fields would be defined under #DICTIONARY as follows:

```
X-ONE,   X,   0,   215,   X-TWO,   X,   1,   X-FIFTY,   X,   49
```

EXAMPLE OF FILE DEFINITION

This file definition is considered in three phases. There is an illustration of the relevant user's input file and then this file is defined under the #DICTIONARY parameter. Finally there is an example of the simplified Dictionary that the computer department would provide for the user.

The example in this section is the same as that used in Chapter 6 page 28 ; an enquiry to find a member of staff to fill a secretarial vacancy. In Chapter 6 the example is used to show the user how to define his enquiry, and more details are given about the example. In this chapter the example is used to show how to define the file to the system. The file to be searched is a personnel file containing details of staff.

File format

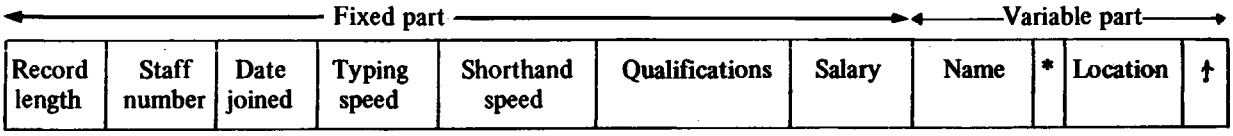
The fields on each record in which this enquiry will be interested are those containing location, typing speed, shorthand speed, a reference to the length of service and a qualifications list. The record is likely to contain other fields such as the staff number, salary and name which, while not necessary for the enquiry, would be desirable to have output with the results.

The staff number, typing speed, and shorthand speed are assumed to be held as fixed length character fields. To simplify the example the length of service condition is given as years, but in practice this information would usually be obtained by referencing a field containing the date the employee joined the company. This date could

be stored as a 6 character field, but for this example it will be assumed that it is held in binary as the number of days since 1st January 1900. FIND–2 can interrogate the date in either form.

The example enquiry includes the condition GRADUATE EQL 1, as will be shown later. This is an example of a field that can take one of only two values, either 1 or 0, to be indicated by the value of a single bit in a qualifications field, where the bit set indicates 1 and the bit clear means 0. A separate bit could be used for each qualification, for example, one bit in this field could indicate that the employee has 5 or more '0' levels, and another bit could indicate 2 or more 'A' levels. NAME and LOCATION would both be character fields but it is assumed that the length of each record has been minimised by defining them in the variable part of the file. The NAME field is always terminated by an asterisk (*) and the LOCATION field by an upward arrow (↑).

If these were the only fields in the record, the format could be as follows:



In practice a record is likely to hold many other fields, irrelevant for this example. However, it is sufficient to define in the Dictionary only those fields that are to be interrogated or output by the current enquiry. Extra definitions will be ignored.

File definition

The file would be defined first by the #READ parameter, also specifying the dump frequency and part file interrogation as required for the enquiry, as follows:

```
#READ,  MT,  PERSONFILE,  1,  1,  NO,  2–EOR
```

The #WRITE parameter can be used to name the hit file as follows:

```
#WRITE,  MT,  FILLVACANCY,  1, 1,  83,  1024
```

The fields can then be defined by the #DICTIONARY parameter as follows:

```
#DICTIONARY STAFF–NO C W1, 2W JOIN–DATE W 3, 1W

TYPING–SPD C 4 2C, SHORTHD–SPD C W4C2 3C

GRADUATE B 5C1, 1B A–LEVEL>1 B 5B7 1B,

O–LEVEL > 4 B 5B8, 1B SALARY W 6, 1,

NAME V 0/* W7, LOCATION V 1/↑
```

Note: In this example, for ease of reference the fields have been defined in the same order as they occur in the record. In practice, the fields may be defined in any order.

In order to specify an enquiry the user needs to know a certain amount about how the data is held on the file, so that the correct constants can be compared with the correct field types. For a user of the package with little or no computer experience, the computer department could present these details in the form of a table, which for this example could be as follows:

Field name (first 7 characters significant)	*Field type*	*Field length*	*Description*
STAFF_NO	C	8C	8 character code
JOIN_DATE	W	1W	Date of joining the company
TYPING_SPD	C	2C	2 digit number
SHORTHD_SPD	C	3C	3 digit number
GRADUATE	B	1B	Bit set if a graduate
A_LEVEL>1	B	1B	Bit set if more than 1 'A' level
O_LEVEL>4	B	1B	Bit set if more than 4 'O' levels
NAME	V	–	Full name in characters
LOCATION	V	–	Full name of location, for example, LONDON, BRISTOL, MANCHESTER

Table 2: Example of simplified Dictionary

Chapter 6 Enquiry language and facilities

This chapter is divided into three main sections. The first section contains the basic enquiry information, showing exactly how an enquiry can be specified. The second section deals with some useful extra facilities in the system, and is intended for the user who has gained experience of using the basic facilities and has more detailed knowledge of how data is stored and processed by the computer. The final section is a summary of facilities.

BASIC ENQUIRY INFORMATION

Introduction

In a company with access to a computer installation, many departments can make use of the FIND–2 Multiple Enquiry System, but it is quite feasible that the department wishing to do so would contain no computer experts. The system is such, however, that anyone can quite easily formulate an enquiry and pass this to the computing department for punching and processing. In fact, it is anticipated that the simplicity of the system will attract non-computer personnel, and that its flexibility will enable them to apply the package to their own particular needs. Such people will be referred to throughout this and the following chapter of the manual as *users*, and this chapter attempts to show them, step by step, how an enquiry is written down for processing by the system.

If the organisation is to be such that the user will define his problem to the system himself, the computer department would be required to perform the initial definition of the files to be interrogated.

A computer file can be organised in much the same way as a conventional manual file. Each computer file normally contains one type of information, for example, relating to personnel, and comprises a number of *records*, for example, one record for each person on a personnel file. A record is subdivided into a number of different items of data referred to as *fields*, for example, name, staff number, etc. As with a manual file, records of a different type or format may exist in the same file if required.

The format of a record is defined to the system by giving names to the fields in what is called a *Dictionary*. The computing department would normally perform this initial definition of the files to be interrogated and provide the information to the user departments in the form suggested at the end of Chapter 5.

Enquiry language

The format of an enquiry specification has been designed to be flexible, and as near the ordinary everyday language of the original problem as possible. The writing of an enquiry in the system language is described step-by-step in this section, defining the various components and building them up into the complete enquiry.

Before continuing with an example illustrating how an enquiry is specified, a summary of some of the terms used in this section is now provided. A fuller explanation is given in the text.

SUMMARY OF TERMINOLOGY

CONDITION	–	A condition can be either a simple condition of the form: (field name) (relationship) (constant) or a complex condition comprising several simple conditions linked by an operator and bounded by brackets.
BRACKETS	–	Brackets are used in the enquiry to avoid ambiguity. For example, A AND B OR C is ambiguous and must be defined as either A AND (B OR C) or (A AND B) OR C.
CONSTANT	–	An explicit value, character or numeric (binary), to be compared with a field.
ENQUIRY	–	An enquiry consists of one or more conditions linked by the logical operators, AND and OR. Ambiguity of logic is avoided by the use of brackets.
FIELD NAME	–	The name associated in the Dictionary with the field of a record containing a

particular item of information.

LOGICAL OPERATOR — The system allows for the two operators AND and OR. An operator always links the two conditions on either side of it.

RELATIONSHIP — A relationship connects a field name and a constant. The allowed relationships are written in the enquiry using either mnemonics or symbols, for example, EQL or =.

The following example is related to that introduced in Chapter 5; more details about the enquiry are given in this section.

The firm concerned wish to fill a secretarial vacancy with a member of the present staff. The personnel file would have to be searched to find a secretary with a typing speed of 70 words per minute and a shorthand speed of 120 words per minute. The successful applicant would also have either a degree or five years experience in the firm. The enquiry would have to be formulated in the enquiry language and specified correctly in order to be processed by the system.

CONDITIONS

For the purpose of the enquiry, each of the requirements in the example will be regarded as a separate *condition.* Thus, the fact that the applicant should be located in a London office is an enquiry condition. Each condition can be broken down into three components, *field name, constant* and *relationship.*

A *field* is a section of a record on an input file, containing a specific piece of information. This information can be referenced by using the *field name.* It is useful if the field name indicates clearly the information stored in that field, and this use of mnemonics helps to make the enquiry language more like everyday language. Therefore, continuing the above example, each record on the personnel file might contain the name of the town in which that person works, in a field named LOCATION.

Note: If a field name comprises more than one word, there must be no spaces; the words should be hyphenated, for example, PLACE–OF–WORK.

An enquiry involves searching a named field to see if it contains a certain value, and this required value is specified in the enquiry as the *constant.* In the example the constant would be LONDON. There is a complete list of the types of constant and the field types with which they can be compared on page 33.

The field name and constants are linked by one of six *relationships.* These are equal to, not equal to, greater than, greater than or equal to, less than, and less than or equal to. They are written in the enquiry using either the following mnemonics or associated symbols:

EQL	NEQ	GTR	GEQ	LSS	LEQ
=	/ =	>	> =	<	< =

It is now possible to build up the complete condition, in the example, as follows:

LOCATION EQL LONDON

which could also be written as:

LOCATION = LONDON

If this and all the other conditions are satisfied by a record, that record will output to a hit file.

LOGICAL OPERATORS

The conditions can be linked in different ways, using the logical operators AND and OR. It is sometimes necessary for each condition to be satisfied, and in such a case the conditions would be linked by AND. For example the first part of the above example would be written as:

TYPING–SPD GEQ 70 AND SHORTHD–SPD GEQ 120 AND LOCATION EQL LONDON

Each of these conditions would have to be satisfied.

If it were only necessary for one condition to be satisfied, for example, the applicant having a degree or five years experience with the firm, then the OR operator is used, and the second part of the enquiry would be written as:

GRADUATE EQL 1 OR SERVICE GEQ 5

The OR implies that either condition being true would satisfy the enquiry.

In the example one of the above requirements is needed in addition to all those governed by the AND operator, and therefore these two must also be included under the influence of an AND. The complete enquiry would be written as:

(TYPING–SPD GEQ 70 AND SHORTHD–SPD GEQ 120 AND LOCATION EQL LONDON AND (GRADUATE EQL 1 OR SERVICE GEQ 5))

In practice it is unnecessary to repeat the same operator between several conditions. As long as the operator is inserted once, at each depth of logic, the program will assume the same type of operator between the other conditions. The position of the operator is not important. For example, the same enquiry could be written as:

(A, B, C AND (D OR E))

or,

(A AND B, C, (D OR E))

Thus the previous example could be written as:

(TYPING–SPD GEQ 70, SHORTHD–SPD GEQ 120, LOCATION EQL LONDON, AND (GRADUATE EQL 1 OR SERVICE GEQ 5))

BRACKETS

The use of brackets when writing enquiries is logically important. They are used to group conditions linked by the same logical operator. There must be at least one pair of brackets for each depth of logic, as shown in Figure 2. For example, (A AND B OR C) would be ambiguous and must be defined as either (A AND (B OR C)) or ((A AND B) OR C). The system will indicate an error if two different operators occur at the same depth of logic.

Redundant brackets will be indicated by error procedures and then ignored wherever they occur, but for each opening bracket there must be a corresponding closing bracket, and vice-versa, even when the brackets are redundant.

For example, (A AND B AND C) could just as well be defined as (A AND (B AND C)) or (A AND (B) AND C) and the program will ignore the redundant brackets and halt in case correction is required. The enquiry (A AND (B AND C) could not be correctly interpreted, and the program would therefore halt indicating the error.

The number of operators in any enquiry is not limited but a maximum of ten depths of logic is allowed in the system. A set of brackets indicates each depth of logic. In the above examples two depths of logic have been shown. The way in which further depths are built up can be illustrated as follows:

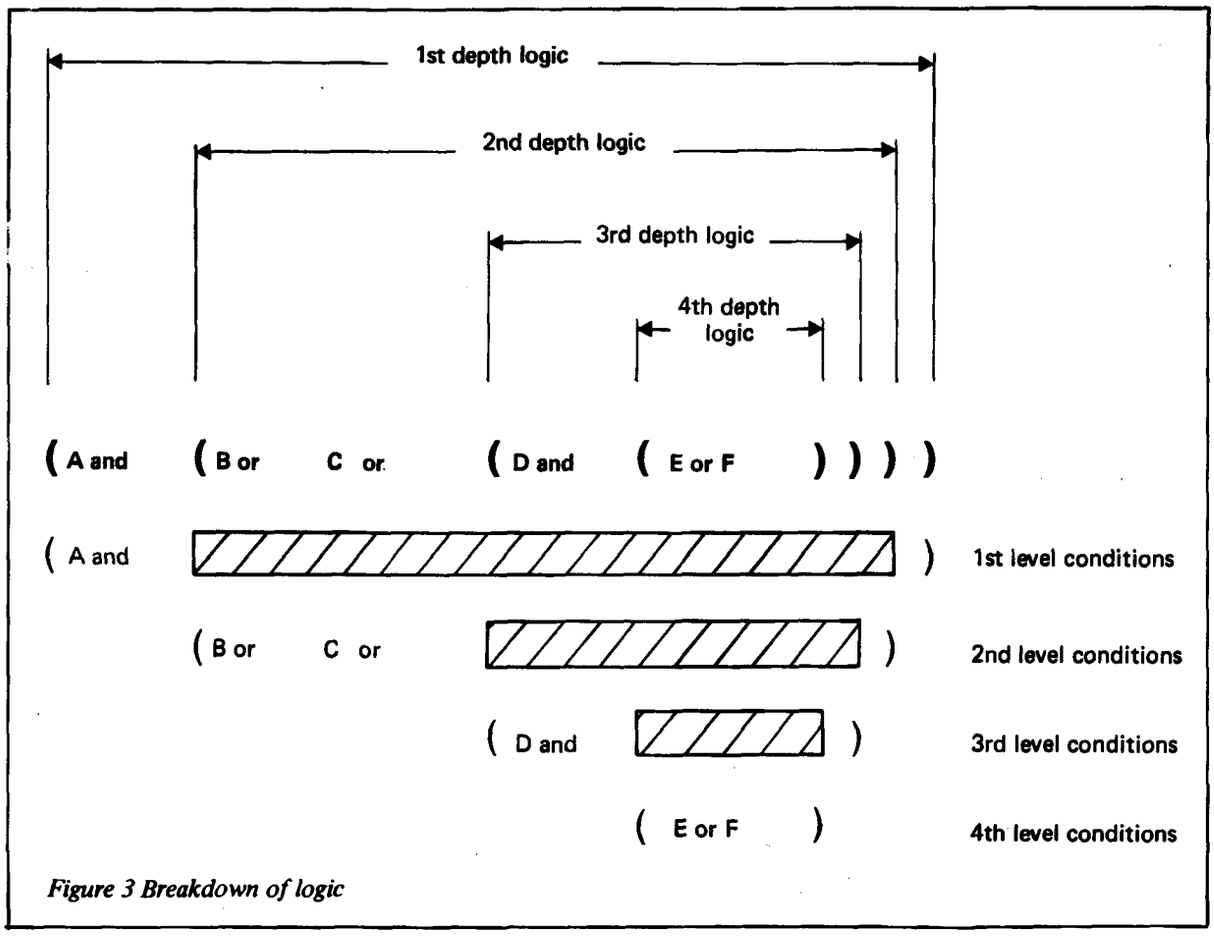


Figure 3 Breakdown of logic

The parts of the example used throughout this chapter can be broken down as follows:

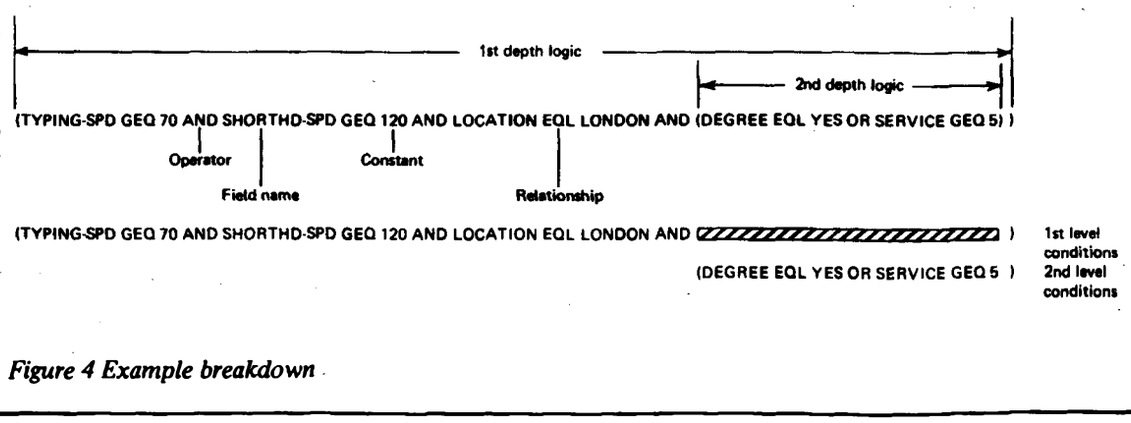


Figure 4 Example breakdown

Note: In these examples all closing brackets appear at the end of the enquiry. This is not always the case but is dependent on the logic of the enquiry.

CONSTANTS

This section described the types of constant that can be used, and indicates the field types with which each can be compared. Field types are dealt with fully in Chapter 5 page 20, and the user will have been supplied with a simplified Dictionary from the computer department, telling him under which field type his data is classified. The field types mentioned below are given for reference only. Constants can be either character or numeric as defined below. All constants are terminated by either a space or comma, and therefore these symbols cannot be used within a constant except in the case of character constants defined in a particular way.

Character (field types C, V or X)

Character constants normally consist of names or identification data, and will comprise any characters from the standard ICL set of sixty four characters.

Examples

```
NAME EQL BROWN,
PART–NO = B/1472C
```

In these examples the constants (BROWN and B/1472C) are terminated by a space or comma, and therefore a space or comma cannot be used within the constant.

Where it is necessary to use a space or comma within the constant, the constant may be written within apostrophes.

Examples

```
NAME EQL 'JOHN BROWN'
```

Spaces can be used anywhere within a constant defined in this way but apostrophes cannot.

Character constants are used with fields that have been defined in the Dictionary to be of character type. The constant begins with the first non-space character after the relationship and its maximum length is 72 characters. A constant may be shorter or equal in length to the size of the field with which it is being compared. It should never be longer and will be flagged as an error if it is. In performing the comparison, the number of characters compared is governed by the length of the constant and not the field length. It should be noted that leading zeros on character fields are not ignored. For example, if a four character field PART–NO could contain the value 0123 then the condition must be specified as PART–NO = 0123.

Numeric

The following six types of numeric constant are acceptable. Where appropriate a number may be preceded by a plus or minus sign, but if none is given then a positive value will be assumed. All constants must be terminated by a space or a comma.

1 INTEGERS (FIELD TYPES B,M,P OR W). Positive and negative integers can be specified. The constant should be a string of digits preceded by a sign or a space and terminated by a space or comma.

Example

```
QTY    GTR    150
LOSS   GEQ   –235
```

2 DECIMAL FRACTIONS (FIELD TYPES F,M OR P). Positive and negative decimal fractions can be referenced. The fraction constant should take the form of a decimal point preceding a string of digits terminated by a space or comma.

Example

```
RATION  LSS    .25
RATE    GEQ   –.763
```

3 MIXED NUMBERS (FIELD TYPES M OR P). A constant of the form *x. y*, where *x* and *y* are numbers, is acceptable. The constant can optionally be preceded by a sign, and the fraction is terminated by a space or comma.

Example

FIELD LEQ 24.3333

4 STERLING. A sterling constant may be in one of the following formats.

(a) Sterling (field type W)

± LxSyDz.n for example, L255S10D2.5

± LxSyDz for example, –L15S6D4

± Lx for example, +L5

where *x*, *y*, *z* and *n* are numbers, with *y* less than 20 and *z* less than 12.

(b) Decimal sterling (field types D or H)

± £ *x*. *y* for example £6.755, £234.5

where *x* and *y* are strings of numbers.

Note: Ordinary decimal currencies, for example dollars and cents, are specified in the same way as integers or mixed numbers.

5 DATE (FIELD TYPES W OR C)

(a) Dates relative to 1st January 1900. The field type W indicates that the date is stored on the files in a special form; as a number of days since 1st January 1900. The constant to be compared with a field of type W would have the format *xx/yy/zz*, where *xx* = day, *yy* = month, *zz* = year.

Example

DATE GEQ 01/04/62

Alternatively, dates may be held as characters in field type C, and the date constant will be expressed in characters as explained above for character constants.

Note: A special constant is available to the user; @ TODAY. The current days date is held as the number of days since 1st January 1900, and therefore this constant can only be compared with a date held in a field of type W.

Example

DATE EQL @ TODAY

(b) Dates not relative to 1st January 1900. Dates are not always held relative to 1st January 1900; a facility is provided for the interrogation of dates held relative to more recent base-dates. The constant name BASEDATE is reserved in the #DEFINE parameter to provide this option. If a value is ascribed to the name BASEDATE, for instance BASEDATE = 01/01/60, then this value will be taken as the base-date for all date constants in the enquiry parameters. That is, it is assumed that all date fields referenced hold the number of binary days from the date associated with BASE DATE, and hence all date constants are converted accordingly.

If BASEDATE is not specified on the #DEFINE parameter, then 1st January 1900 will always be assumed.

The definition of the enquiry condition is the same regardless of whether BASEDATE has been defined.

This facility is not available in the Single Enquiry System.

6 OCTAL (FIELD TYPE B OR W). This type of constant is unlikely to be used by the casual user of the system, but is included here for completeness. When a field consists merely of a number of bits, the constant to compare with this field may be expressed as an octal number.

Example

FIELD NEQ #43

7 PREDEFINED CONSTANTS. The constant in a condition can be a constant name specified in the #DEFINE parameter. To distinguish the name from other types of constant, it must be preceded by the @ character.

For example, if the name START is specified under #DEFINE to correspond to 05/07/69 (#DEFINE START = 05/07/69) this name can be referred to in any type of enquiry condition and the value will replace the name.

Example

DATE > @START

is equivalent to

DATE > 05/07/69

This facility has two advantages:

(a) If the same constant is used several times in an enquiry batch, only one entry is made in a constants table. The fact that it is defined once also simplifies modification.

(b) When re-entering the system using a previously generated enquiry program it is possible to amend the value of a predefined constant at run-time. Thus there is no need to regenerate an enquiry program just because a constant has changed.

The following table summarises the field and constant types that can be compared in a condition.

Constant type	*Field types* B C D F H M P V W X	*Examples*
Character	✓ ✓ ✓	NAME EQL BROWN NAME EQL 'JOHN BROWN'
Integer	✓ ✓ ✓	LOSS GTR -255
Fraction	✓ ✓ ✓	RATE LSS .75
Mixed number	✓ ✓	FIELD LEQ 2.63
Old sterling	✓	TAX LSS L4S10D6
Decimal sterling	✓ ✓	AMOUNT EQL £2.7
Date	✓	DATE EQL 18/02/69 DATE EQL @ TODAY
Octal	✓ ✓	FIELD EQL #7572

Table 3: Possible comparisons of field and constant types

Appendix 6 contains a glossary of the terms which have been described so far.

Enquiry specification

Once the enquiry has been written in the enquiry language as explained above, it has to be completely specified for the system. The fact that there are few format rules should minimise the number of syntax errors.

GENERAL RULES

The first three rules apply generally to the specification of an enquiry, while the others are more concerned with the punching of cards or paper tape for processing.

1 The enquiry parameters are preceded by a major directive:

#ENQUIRY

This must appear on a separate card or paper tape block.

2 Each enquiry is preceded by an *enquiry identifier*, giving an enquiry a name and identifying its type. An enquiry identifier must occupy a separate card/block. Up to 96 enquiries can be submitted in one run.

3 An enquiry, which may be punched on any number of subsequent cards, must be preceded by an open bracket and terminated by a closing bracket. A new card or paper tape block may be started at any point during the punching of the enquiry, see Note 5 below.

4 A comma or space must follow each basic element, each word of an enquiry, for example, field name or relationship, being regarded as a basic element. In addition, any number of extra spaces may be punched. Brackets need not follow this rule, but it should be noted that a space must precede a bracket when that bracket directly follows a constant in an enquiry.

5 A basic element must not be split between two cards or paper tape blocks. In particular this applies to field names and constants. The end of a card or paper tape block will be assumed to mark the end of an element.

6 Only the first 72 columns of a card/block should be used.

A complete set of enquiry parameters, including the major directive #ENQUIRY, will therefore be as follows:

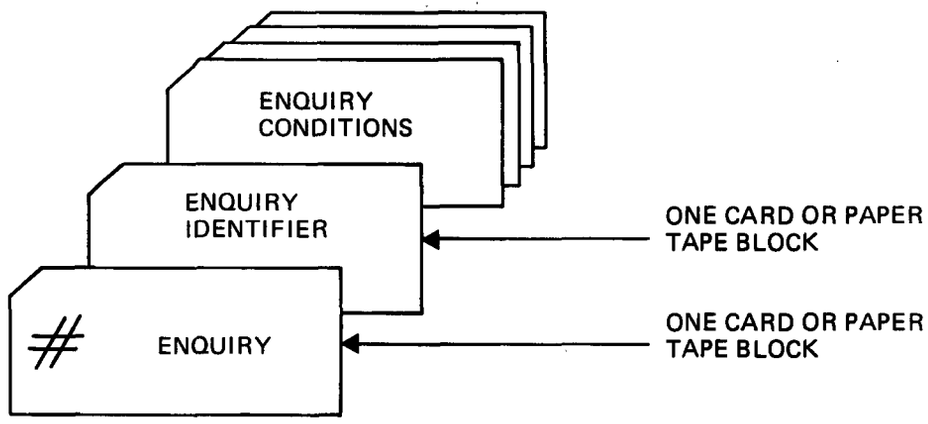


Figure 5 Enquiry parameters

ENQUIRY IDENTIFIER

This parameter occupies one card or paper tape block only. The first character of the parameter, a question mark, must appear in the first column of the card or paper tape block, and be followed immediately by the enquiry name. This name may be of any length, terminated by a space or comma, but only the first four characters will be used in the system. The enquiry type is then specified.

Comments

For all types of enquiry, the enquiry identifier may be terminated by a comment preceded by the square bracket symbol ([).

Standard enquiry

The standard enquiry can be specified by name as follows:

?ENQ1, STANDARD

If a space is left where the enquiry type should be, a standard enquiry is assumed. The above enquiry could also be written therefore simply as:

?ENQ1

Count enquiry

A count enquiry outputs only the number of records that have satisfied the enquiry. This is known as the *hit-count.* The enquiry is specified as:

?ENQ2, COUNT

Lead enquiry

A LEAD enquiry may be positioned anywhere in a batch of enquiries and affects all subsequent enquiries, so that those enquiries following a LEAD enquiry are only executed when the LEAD enquiry is a hit.

Apart from this administrative function a LEAD enquiry is executed in the same way as a STANDARD enquiry.

The enquiry identifier is specified as

?ENQ3 LEAD

Typical applications of the lead enquiry are where a batch or part of a batch of enquiries contains one or more common conditions. For example:

1 Preceding enquiries on a file containing different record types, where only selected types are to be considered in depth

2 Preceding a set of enquiries which, for example, fall within certain dates or relate to particular dates

Examples

```
?VACANCY            [VACANCY APPLICANTS
?STOCK, COUNT       [STOCK OF PART A
?JULY, LEAD         [JULY SALES FIGURES
```

In all the examples in the above section on the enquiry identifier, the commas could be replaced by spaces.

ADDITIONAL INFORMATION

The facilities dealt with in this section are intended for the more experienced user, but should be of value to the non-computer person once the basic enquiry facilities have been mastered.

Additional language facilities

The following facilities are available within each of the enquiry types and provide more flexibility in analysing a file.

PART-FILE INTERROGATION

It is possible to limit the area of search to a relevant part of a file, when this is known. This facility can greatly reduce interrogation time.

On indexed sequential disc files, the search can be specified between two keys. This applies strictly to sequential files set up with indexes in accordance with ICL 1900 Series Housekeeping Standards.

On magnetic tape, the area of search can be specified between two user tape sentinels.

A part-file interrogation is specified under the #READ directive as described fully in Chapter 5, and applies to all enquiries in the batch.

STEP SEARCH

A search can be made for a constant when its exact position in a field of a record is not known.

The general format for specifying this facility is STEP *m* BY *n* (followed by the condition) where *m* gives the total number of comparisons to be made (or fields to be examined) and *n* gives the number of characters by which the field address is to be incremented after each comparison.

The condition in the step statement is limited to one simple condition although the repeated constants facility may be used. The field type may be fixed or variable length character (C, V or X), and the relationship EQL or NEQ. The use of other relationships is not meaningful and is therefore not allowed, The step statement is treated as a single condition, and does not need to be bounded by brackets unless it is the only condition in the enquiry.

As mentioned above, the step search works in one of two ways, searching within a field or over several adjacent or equally spaced fields.

Searching within a field (usually a variable length field).

A single field could contain quite a large amount of information. For example, it might contain both the title and author of a book

Elements of Basic Chemistry by B. Burner

The search might be for all chemistry books on the file. The statement, therefore, would read as follows:

STEP 100 BY 1 TITLE = CHEMISTRY

Searching several fields (including fixed length fields).

In this case the search might be for a particular qualification on a personnel file, where the qualifications are not stored in any particular order.

Here it is assumed that each qualification field is of a fixed length of 10 characters and the step increment is therefore equal to the field length. The statement will be as follows:

STEP 4 BY 10 QUAL–1 = PH.D

Only the first field has to be defined in the Dictionary.

In both types of search, the search is terminated when either the constant is found and the condition is a hit, or when the condition is not a hit and one of the following occurs:

1 The number of steps has expired.

2 The end of the record has been reached when searching a fixed length field.

3 The end of the field has been reached when searching a variable length field.

Repeated constants as explained on page 37 may be used with this facility.

M OUT OF *N* CONDITIONS

This facility handles the type of enquiry where it is sufficient that *m* out of a possible *n* conditions are satisfied. The user has to list the possible *n* conditions, and to associate *m* with them. The format for this facility is as follows:

(ANY *m* FROM *n* conditions)

where *m* gives the number of conditions to be satisfied. The *n*-list must be terminated by a bracket and a bracket must immediately precede the ANY operator.

Example

If a job vacancy were to be filled, one qualification might be absolutely necessary but any two out of the remaining three might be sufficient. The enquiry can be written as follows:

(DEGREE EQL YES AND (ANY 2 FROM LANG EQL FRENCH OR LOCATION EQL LONDON OR PRESENT–GRADE EQL B))

The ANY operator should be thought of as a different type of operator from AND or OR, and the conditions linked by it are at a different depth from other conditions linked by an AND or OR.

Note: *m* must be greater than zero, and less than *n*.

Only simple conditions should be listed in the *n*-list. The system caters for one depth of logic within the ANY statement. The operators between conditions within the influence of the ANY operator are ignored and may be omitted. The example above would therefore read as follows:

(DEGREE EQL YES AND (ANY 2 FROM LANG EQL FRENCH LOCATION EQL LONDON PRESENT–GRADE EQL B))

PARTIAL CHARACTER CONSTANTS

When performing a character equality comparison, the option of directing the comparison routine to ignore specified character positions is available. The characters to be ignored should be replaced in the constant by

asterisks. The facility could be used as follows:

Examples

Mis-spelling or duplicate spellings can be dealt with by the system. A name might be spelt Smith or Smyth, and in this case a condition could read:

NAME EQL SM*TH

Alternatively, only partial information might be available to the user. The witness of a hit-and-run accident might remember that the car involved was black and that the registration number was A something 437. The enquiry interrogating the file of stolen cars would then be written as:

CAR-NO = A**437 AND COLOUR = BLACK

ARITHMETIC RELATIONSHIPS

Limited arithmetic facilities are available in the interrogation phase. The option is provided by replacing the field name in the condition by the sum or difference of two field names. At run time the values are retrieved from the two fields, the sum or difference taken, and the result is compared with the constant.

Examples

WAGE – TAX GTR £1000

FIELDA + EXTRA LSS 400

This facility can be used to compare two fields in the record for equality.

Example

FIELDA – FIELDB = 0

The two binary fields should be defined in the Dictionary to be of the same binary type.

Language 'short cuts'

These facilities provide a shorthand method of specifying certain conditions which occur in enquiries. They do not provide any additional logical facilities, but greatly reduce the amount of user specification.

REPEATED CONSTANTS

It is sometimes necessary to check whether a field contains one of several values. To avoid writing a sequence of similar conditions, it is possible to list all the constants after the relationship, separated by semi-colons.

Example

JOB–TITLE = DRIVER ; DVR ; CHAUFFEUR

This would generate within the system:

JOB–TITLE = DRIVER OR JOB–TITLE = DVR OR

JOB–TITLE = CHAUFFEUR

Note: The constants must still be terminated by a space or comma, and not by the semi-colon.

The facility can also be used with the NEQ relationship but in this case the generated conditions would be linked by AND. Other relationships would not be meaningful and are therefore not allowed.

This facility can be used in conjunction with arithmetic conditions and step conditions.

Brackets need not bound the condition unless it is the only condition in the enquiry.

The limit to the number of different constants is dealt with under general program limitations in Appendix 3.

REPEATED FIELD NAMES

The step search facility previously mentioned caters for a step-wise search through adjacent or equally spaced fields. The repeated field names facility is similar, but is used when a search is required of irregularly spaced

fields and reduces the amount of enquiry specification. All the fields to be searched have to be defined in the Dictionary, but it is only necessary to specify one condition in the enquiry.

The facility requires that the field names are seven characters long with the first five characters identical. The last two characters should be numeric and range from 01 to *n* where *n* is the total number of fields to be checked. The condition is written as follows:

GENERATE *n* FROM (condition)

Note: GENERATE may be abbreviated to GEN.

Example

GEN 4 FROM GRADE01 EQL B

would generate the following conditions:

(GRADE01 EQL B OR GRADE02 = B OR
GRADE03 EQL B OR GRADE04 = B)

GRADE01, GRADE02, GRADE03 and GRADE04 would have to be defined in the Dictionary.

The conditions to be generated must be simple, not arithmetic or step.

The GENERATE statement does not have to be bounded by brackets unless it is the only statement in the enquiry.

RANGE CONDITION

With this option it is possible to specify a range of values in a single condition. The relationship in the condition is replaced by BETWEEN (or BET) and this is followed by two values. The first three characters of the relationship BETWEEN are mandatory. Further characters up to the first space will be ignored. The constants can be defined in the usual way, but they must be separated by a comma. The lower value must be specified first.

Example

AGE BET 25,35

This condition would generate the following:

AGE GEQ 25 AND AGE LEQ 35

Brackets need not bound the condition unless it is the only condition in the enquiry.

SUMMARY

It is intended here to summarise the enquiry facilities available in this system, enabling users to decide how best to tackle their own particular problem. The system can be used, as its name implies, to handle mixed batches of enquiries, dealing with up to 96 enquiries at a time. The system is capable of providing a regular report or dealing with an ad hoc enquiry.

The type of enquiry to be used, standard, count or lead, will depend on the particular problem. If it is simply required to print out details from records that satisfy the conditions in the enquiry specification, then the standard enquiry will be used. The count enquiry can be used where no details from the records are required and it is sufficient to know how many records satisfy the enquiry. If the user wishes to economise on interrogation time, the lead enquiry is useful for testing whether following enquiries need be checked, or whether they do not apply to the current record.

The conditions of the enquiry must then be written in the enquiry language and linked by AND and OR operators according to the logic required. The enquiry parameters can then be specified in full. Once the user is familiar with the basic enquiry specification, he may wish to save himself time by using the language shortcuts. These allow for quick methods of specifying repeated constants or field names. By using the relationship BETWEEN (or BET) it is also possible to specify a range of values as the constant.

With greater knowledge of the system a user will also be able to employ the more advanced logical facilities. If he knows how the information is stored on his file, he might be able to limit this search to the relevant part only by specifying a part-file interrogation under the #READ directive. The step search facility may be useful

when the exact position of the information is not known, and the partial character constants facility useful if only some characters of a character constant are known. Limited addition and subtraction facilities are available, comparing the sum or difference of two fields with a constant, if the fields are of the same binary type. A further facility exists for selecting a record if only a certain number of the specified conditions are satisfied and it does not matter which conditions those are, for example, if it is only necessary for four out of a possible six conditions to be satisfied.

Chapter 7 Output options and facilities

The output phase of the FIND–2 Multiple Enquiry System can be used either on its own or in conjunction with file interrogation, that is, it can produce output direct from a user's file, or from a hit file produced by an interrogation. Output can be produced onto disc, magnetic tape or the line printer. Reports generated on the line printer can range from a simple listing requiring minimum specification to a detailed table report.

The four basic output options are Print, List, Total and Table, each producing a different type of output. Various output facilities are available and can be used with the List, Total or Table options.

The first section of this chapter described the four options and the second section gives details of the output facilities and contains a table showing the compatibility of the various options and facilities. Examples of the output produced by each option are given in Chapter 9.

OUTPUT OPTIONS

Print

This option is the same as the output from the Single Enquiry System. It produces output to the line printer only and is suitable for reports which do not require records to be output in a particular format and for which no field totals other than grand totals are required. All reports are limited to one line of output per record and one record per line.

The only parameter specification that is required is a list of the fields to be output and the text for an optional main heading to be output at the start of the report. All other format details are set up automatically by the program.

Field names are used as column headings at the top of each page and each individual field is allocated a set number of print positions based on its dictionary type. Positive fields are unsigned, negative fields are signed with a floating minus sign and sterling fields are edited with a floating pound sign. Type B fields (binary fields not held as an integral number of words) are treated as positive binary integers (i.e. they are right-justified) and given a space allocation in the output line of eight decimal characters.

Variable length fields are allocated a length of thirty characters in the print line and any overflow is printed on the following lines directly underneath within the same print positions. All splits in character text are made at word boundaries except in the cases where the text does not contain any spaces.

Conversion facilities are available for the output of sterling fields so that fields which are held in pence, new pence or 1/10ths new pence can be output either as old or decimal sterling.

The output fields contained in a single line are separated by two spaces (these spaces are not applied beyond the last field in the line). The line spacing can be set at run time to produce either a single or a double line spaced report.

Date and report page numbers are output at the top of each page but these can be suppressed at run time if required.

All numeric fields are totalled at the end of each report and are output on a separate page.

List

This is the most flexible of the output options and requires detailed specification by parameter. It produces a list of output records on the line printer and can also be used to output to disc or magnetic tape when totalling is not being performed.

The format and content of each line is completely specifiable by the user. Each record can occupy any number of lines. Either one or two records can be output on one line. The spacing between each individual line of one output record can be specified separately so that this option can be used to output to pre-printed stationery.

A paper line-up facility aids the use of pre-printed stationery in the LIST output option. If switch 2 is set ON

before an output report, the program will halt 'LOAD PAPER' immediately prior to printing the report. After loading the paper a GO message will cause a page to be thrown, four formatted dummy records to be printed, four more pages to be thrown, and a halt: 'CHECK PAPER LINE-UP'. If the paper is satisfactorily aligned then the user should either OFF Switch 2 and GO to print the report, or realign the paper and GO for more dummy records. This facility is not available if the paper has to be reloaded or reset during a report run.

In reports which have only one record per line, fields within individual records may be subtotalled within control levels and details of this facility are given in the next section.

The relevant totals are printed out when a control field changes value, and if the control field is specified in the print line, its old value will be printed out with the totals. A short heading can also be printed out on the line above the totals. A specified vertical paper movement may be associated with a control level so that each time this control changes value, the report will be continued after the appropriate spacing.

Total

This option produces output to the line printer, disc, or magnetic tape. It is very similar to the LIST option and requires the same detailed specification by parameter.

Only the accumulated totals within control levels are output and not details of the individual records. When a control field changes value, all the relevant totals are output and if the control fields are specified in the output format, the old value of the changed control and the values of all the higher level controls are output with the totals. Sub headings for totals and page throws can be printed as in the LIST option.

Table

This option produces output to the line printer only and is used to set up a two dimensional table of totals. Control fields are set up along two axes known as the X and Y axes. The X control fields define the controls to be used across the page of a report and the Y control fields define those to be used down the report. Up to three levels of X controls and up to two levels of Y controls can be used. The combination of the two controls thus defines a table of separate entities and for a record of fall into one of these entities, it must satisfy the appropriate X and Y control values. Within each entity it is possible to accumulate the totals of any number of different fields from each record and these fields will be referred to as a table group. For either axis, specified names of enquiries used in a previous interrogation run may replace specified values of control fields, thus greatly extending the selection criteria defining columns and rows of the table.

A table will hold a maximum of twenty eight fields across the page, in addition to any Y control fields which are specified in the print line. If a file is sorted on the Y controls, the length of the table is unlimited but for an unsorted file the table is limited to twenty-seven lines. This limit can be extended by use of the #STORAGE parameter explained in chapter 8.

OUTPUT FACILITIES

The flexibility of this package is enhanced by the output facilities provided:

Headings	Arithmetic operations
Record selection	Percentage calculations
Merging	Variable length field output
Decoding	Editing facilities
Totalling	Decimal currency conversion
Paper line-up by program	

The following table shows the compatibility of these output facilities with the four output options.

Facilities	*Options*			
	PRINT	LIST	TOTAL	TABLE
LP output	√	√	√	√
Print more than 1 line per record		√	√	
Print two records per line		√		
MT/Disc output		√	√	
Merging		√	√	
Decoding		√		
Variable line spacing		√	√	√
Optional headings	√	√	√	√
Automatic page headings	√			
Automatic formatting	√			
Printing variable length fields	√	√		
Arithmetic		√	√	√
Column totalling		√	√	√
Row totalling				√
Percentages				√
Paper line-up by program		√	√	

Table 4: Compatibility of output facilities with output options.

Headings

Headings are optional and may be specified for all printed output. Main headings will appear once only at the start of the tabulation; page headings will be repeated at the start of each page. Each type of heading is limited to 800 characters and so the maximum number of lines of heading which can be printed varies from 5 to 8 according to the size of the line printer.

Record selection

Various types of record selection are available so that results can be presented as efficiently as possible.

ENQUIRY SATISFACTION

Records may be selected for output on the basis of enquiries which they have satisfied. This applies only when output is from a hit file, where each record has the hit word(s) attached.

FIELD VALUES

With the TABLE option, records may be selected on the values of X and Y control fields.

MATCH CONDITIONS

When merging a main and secondary file on input, the user can specify that records be output when either a match or non match is obtained.

PART–FILE ACCESS

With this facility, the user can specify that only records from a certain part of his file are to be output.

Merging

The user may merge two files together to combine the information into a new file or report. Merging takes place at field level and it is possible to produce a new record format for an output file comprising information from two input records held on the main and merge files. These input files can be either the user's files or files produced

by the system. They must be sorted on a common key. The user specifies MATCH or NONMATCH on the #MERGE parameter. With MATCH, output is from one or both of a matching pair of records. With NONMATCH output is from records on the main file which have no match on the merge file. Several records on the main file may contain the same key whereas the merge file must have one record per key.

Decoding

This facility permits the decoding of information held on a file in coded form. For example, a geographical location stored in coded form as *26* could be output in full as *Manchester.* This facility may be used when outputting to either line printer, magnetic tape or disc, but the decode table must be held on disc since quick access to the table is required for efficient processing. Any number of fields may be decoded in an output report and up to six different decode files may be used simultaneously.

The field to be decoded may be held in character form (type C) or as a number of bits (type B). Where the latter type of field is being decoded and the field is not an integral number of characters in length, the coded field extracted from the record will be extended to occupy an integral number of characters by adding zero bits to the least significant end. The comparison with the decode file is then made on a character by character basis.

Totalling

In the LIST and TOTAL options it is possible to output sub totals at various levels within an output report. These levels for sub totals are determined by the values of control fields and up to eight levels of control including a *Grand* total can be specified for one output report. A field can be totalled when any of the seven control values change or at a *Grand* total level only, and up to twelve numeric fields may be totalled in one report, except where two records are to be printed on one line in which case totalling is not permitted. It is also possible to output the number of records totalled within each control level.

Any file which is to be used to produce a report with sub totals should be sorted on the control keys so that the control values will change the least number of times.

In the TABLE option, both horizontal and vertical control fields are specified and all output fields are in fact totals within these control values. Within each X control value, it is possible to print out any number of field totals within the overall limit of twenty-eight totals across the line, and the optional facility is provided to total the values of each of these fields across the range of X control values, thus providing a row total for each field.

In the PRINT option *Grand* totals only are produced for all numeric fields.

Arithmetic operations

A field to be printed or totalled may be the result of a calculation either between two fields or between a field and a constant. The types of calculation which can be performed are addition, subtraction, multiplication and division. Any one calculation can only involve two fields but each result may be used in a further calculation as illustrated in the next example:

Salary	Overtime	Gross	Tax	Insurance	Deductions	Net
A	B	C=A+B	D	E	F=D+E	G=C-F

In this way, complicated calculations can be built up. It is unnecessary to print fields A, B, D and E, or even to print the intermediate totals such as C and F if they are not required.

Percentage calculation

When using the Table option the percentage of a total with respect to its column total may be calculated. This percentage will be calculated to one decimal place and output in the form nn.n except in the case of 100% which will be output as 100. The percentages will be printed out to the left of the associated totals. If this facility is used the fields should either be all positive or all negative so as to give meaningful results.

Variable length field output

Printed output will cater for the listings of variable length fields. The user may specify any number of print positions, up to a maximum of 127 characters, in which the field will be printed. If the field in the record exceeds the output length allocated, continuation will overflow into the succeeding print lines in the same positions until the whole field has been printed. The overflow lines will be regarded as part of any spacing lines which have been specified, a useful facility when using pre-printed stationery. Up to six variable fields may be output for each format line specified.

Editing facilities on output fields

A full list of the types of fields which can be output is given in Chapter 8, page 59. Additional editing may be employed for:

1 Cheque protection on sterling and numeric fields.

2 Floating £ and $ symbols.

3 Suppression of leading zeroes on character fields.

4 Two-character positive and negative symbols specified by the user to be output to the right of the field. This facility is optional and if not specified, positive values will be output unsigned, and negative values preceded by a floating minus sign.

All fields which exceed their print line allocation are automatically filled with upward pointing arrows.

Decimal currency conversion

On printed output, the facility is available for converting sterling to new decimal currency and vice versa. This will be especially useful during the changeover from sterling to decimal currency as the facility enables the user to output an amount in both types of currency on the same printout. Appendix 4 gives full details of the conversions which can be made.

Paper line-up by program

In the List and Total options, a facility is incorporated to enable the user to ensure that pre-printed stationery is correctly aligned before beginning the printing run. The program will print a group of dummy records in the same format as the true output will appear, and will repeat this action as many times as is required to obtain correct paper alignment.

Chapter 8 Parameter usage and formats

This chapter is divided into two sections. The first deals with the sequence of parameters to be used in the various phases and describes the function of each parameter; the second part deals with the parameter format rules and details.

SEQUENCE OF PARAMETERS

The parameter sequence is shown for the following parts of the system:

1 Interrogation phase

2 Output phase

3 Re-entry phase

4 Decode file update program

The diagrams denote the input sequence of parameters to the program. Notes on batching output runs follow the diagram of the output phase options

1 INTERROGATION PHASE

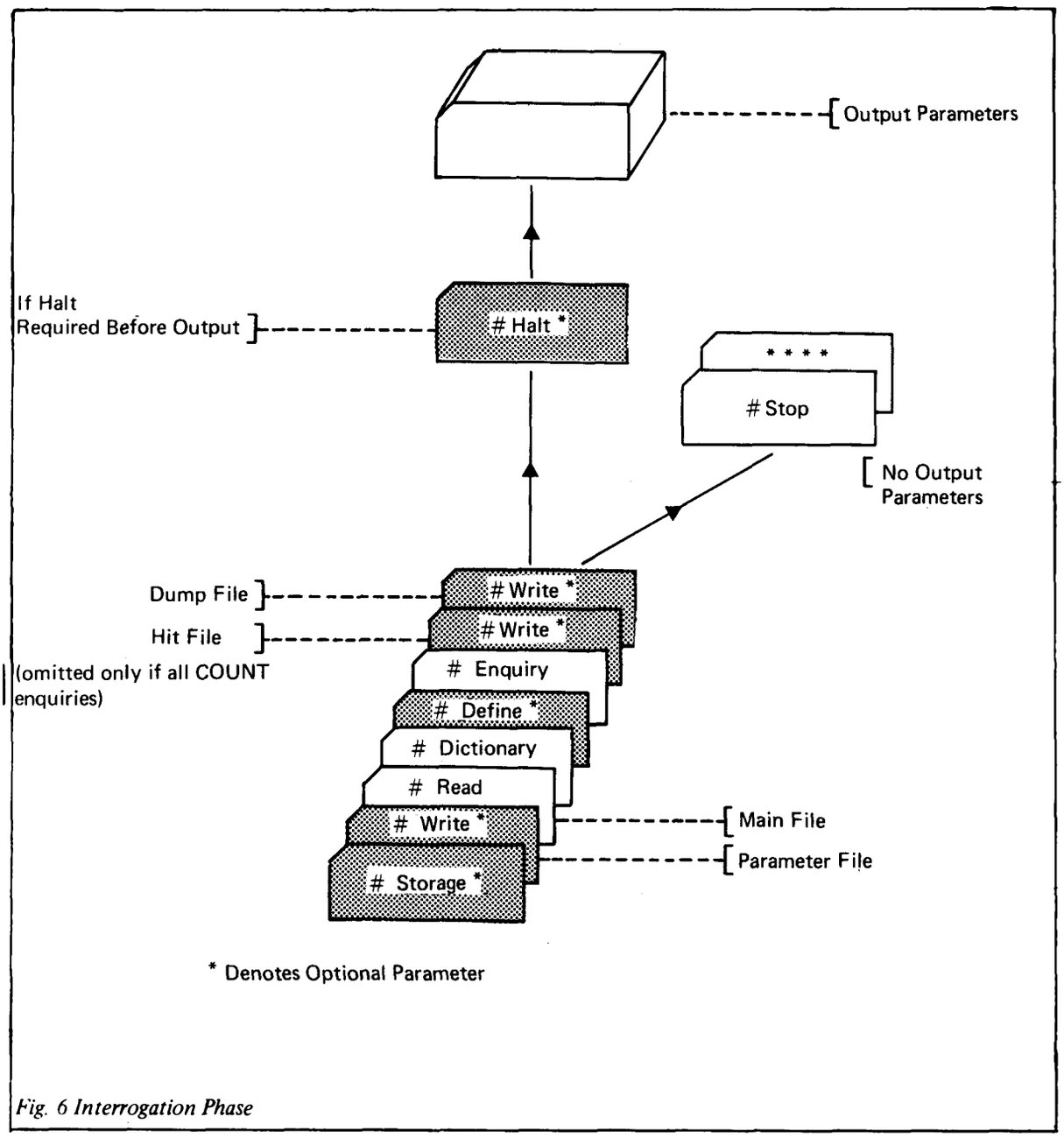


Fig. 6 Interrogation Phase

2 OUTPUT PHASE

(a) PRINT option (LP output only)

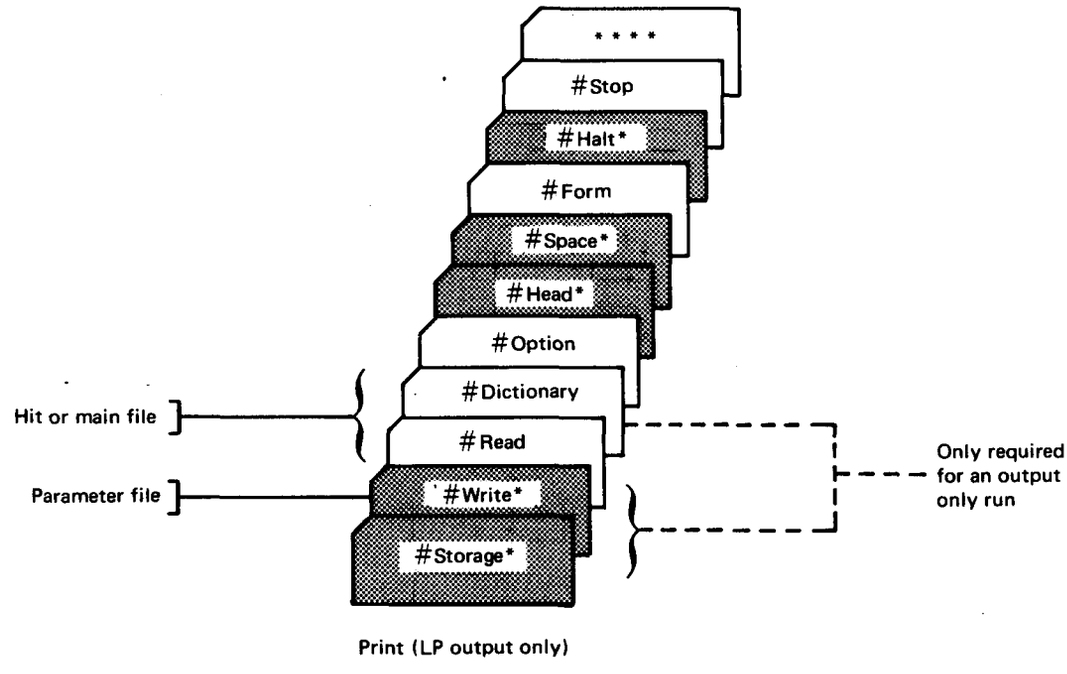


Figure 7 Output phase print (LP output only)

(b) LIST/TOTAL option (LP, MT or disc output)

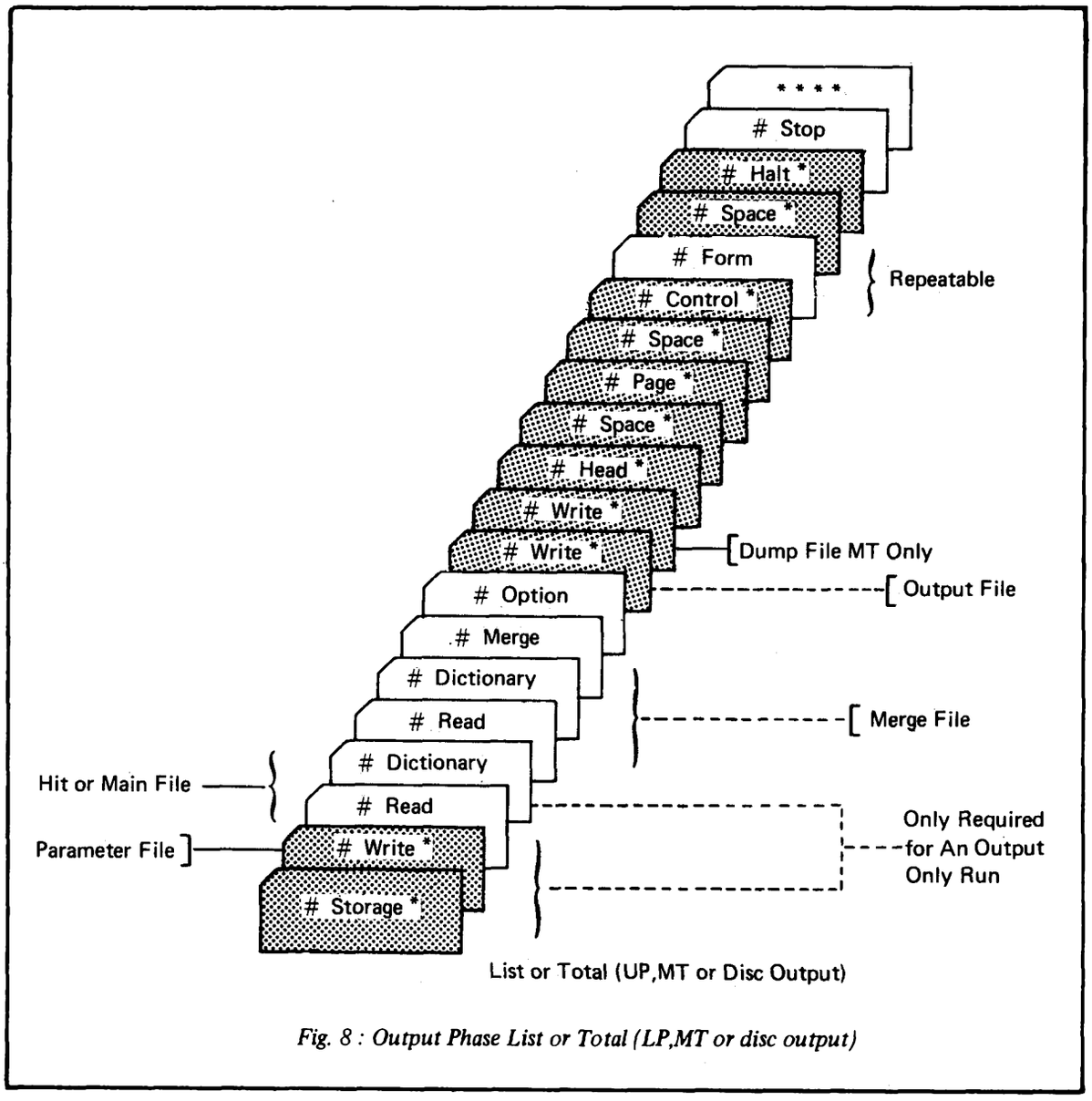


Fig. 8 : Output Phase List or Total (LP,MT or disc output)

(c) TABLE option (LP output only)

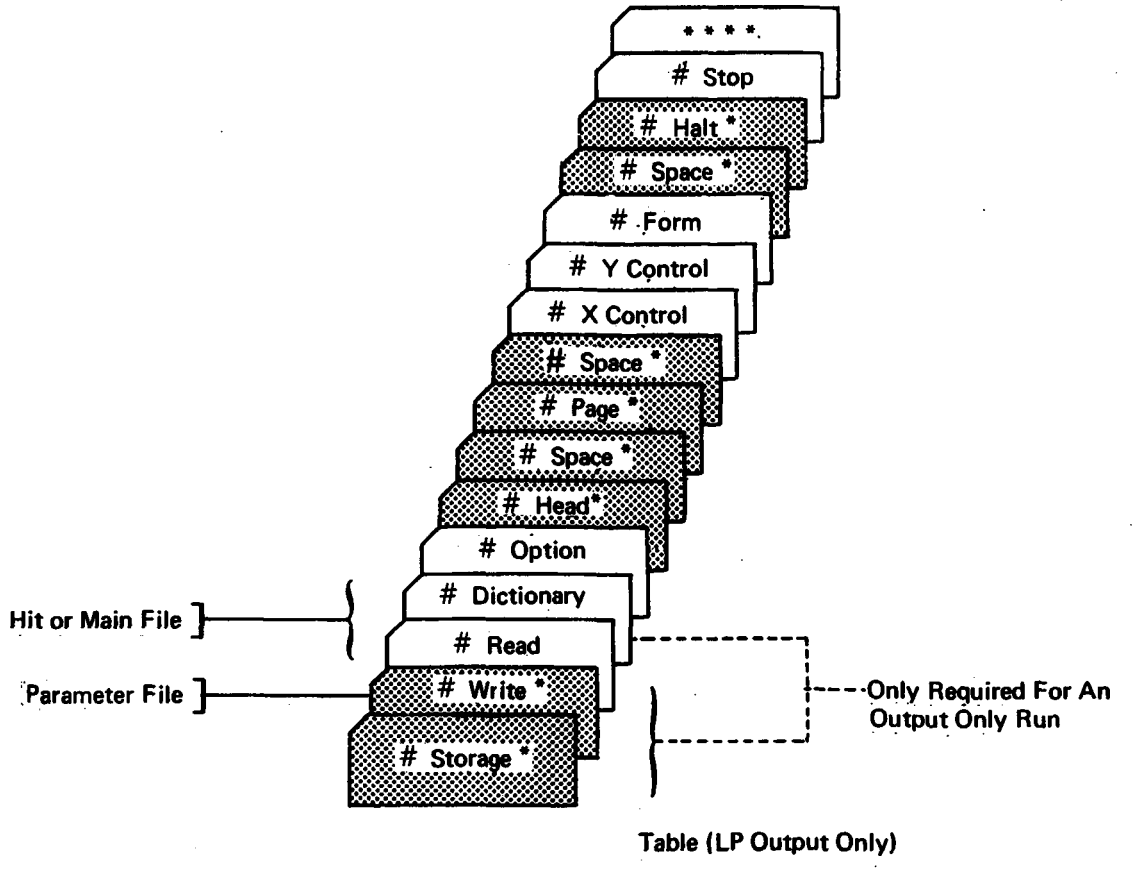


Fig. 9 : Output phase - Table (LP output only)

(d) Batching output runs

As the output phase can be used entirely on its own, it is possible to build up a batch of runs that will produce several reports from one file or a selection of reports from several files. In one output batch there is no limit to the number of reports/files that can be successively produced from one file. Up to twenty input files can be referenced in one batch.

The preceding output phase diagrams show the parameters required to produce one output report/file and these will be referred to as an 'output set'. If further reports/files are required from the same input file there is no need to re-define the input file so the next output report set can start with the #OPTION parameter. The parameters required to produce all output reports/files from one file will be referred to as an output group.

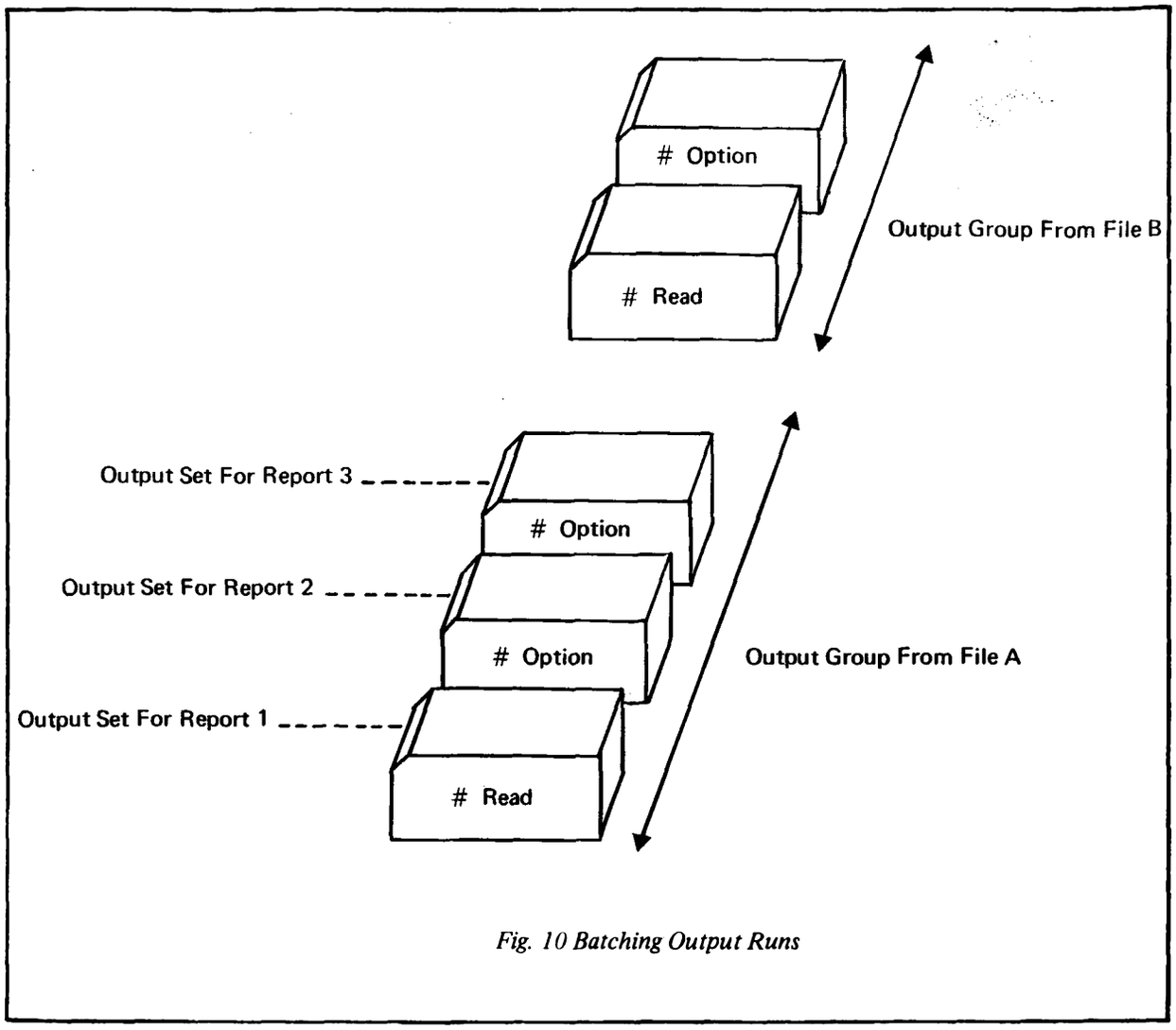


Fig. 10 Batching Output Runs

If merging is specified in the first output set this will be assumed to apply throughout the output group, that is until the next #READ is encountered.

If #HALT is used at the end of an output set, the next output set must start with #READ. This means that if a main input file is to be sorted between output runs, it has to be re-defined. The Dictionary does not have to be respecified unless a file of a different format is to be processed.

3 RE–ENTRY PHASE

The following parameters are those required to re-enter the system using a previously created file of parameters. They differ according to which phase is to be entered.

(a) Interrogation

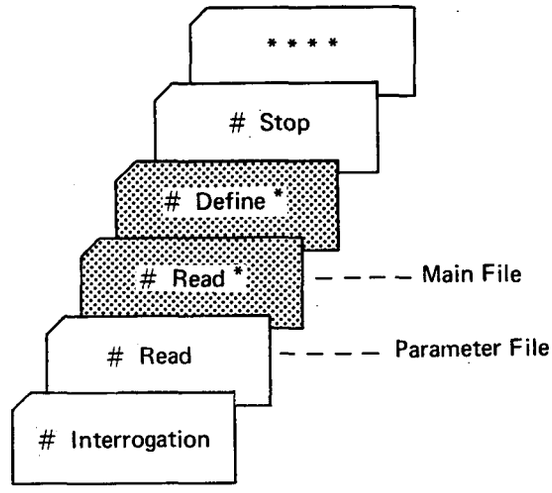


Fig. 11 : Re-starting at the Interrogation Phase

(b) Output

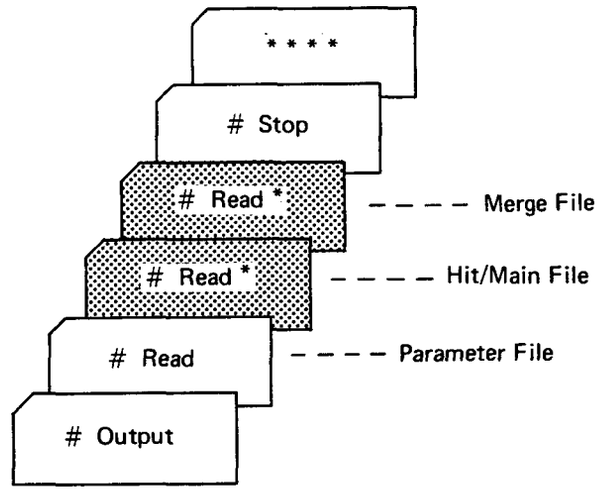


Fig. 12 : Re-starting at the Output Phase

When re-entering at the output phase reference can be made to any output group of parameters but not to individual sets within a group.

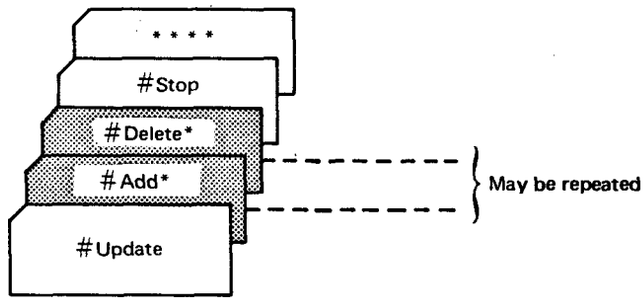


Figure 13 Decode file update parameters

Parameter functions

The function of each parameter is described below, and each parameter is dealt with in alphabetic sequence.

#ADD (#X63E only) specifies the record to be inserted or altered.

#CONTROL defines the control levels and associated field names for sub-totalling in the LIST and TOTAL options.

#DEFINE predefines constants by associating each with a name. These constants can then be referred to by name in the enquiry program. This facility has two advantages:

1 If the same constant is used several times, only one entry is made in a constants table.

2 It is possible to amend the values of predefined constants at run time, thus benefitting the use of preserved enquiry programs, as there is no need to regenerate the enquiry program. For example, an enquiry program for producing a monthly report from a file can be generated once and then used regularly, the only new parameters necessary being those for the redefinition of the date constants.

#DEL (#X63E only) specifies record to be deleted.

#DICTIONARY specifies the record format of input files and associates each field to be referenced by the enquiry program with a unique field name. If two input files are defined, that is, a main file and a merge file, the field names must not duplicate each other.

Note: If the format of a record is altered by means of own coding, the Dictionary must define the record to be processed and not necessarily the record as it appears on the file.

#ENQUIRY specifies all the enquiry details for up to 96 enquiries.

#FORM. The use of this parameter varies according to the output option specified.

1 PRINT OPTION. The parameter specifies the fields to be output and the type of output for sterling fields.

2 LIST, TOTAL AND TABLE OPTIONS. The parameter specifies the number of records to be output per field per group for the TABLE option, the spacing between output fields, the constants and fields to be output together with details of the editing required and the levels at which each numeric field is to be totalled. The parameter also specifies whether a record count is to be output for each control level, whether the % facility is to be used in the TABLE option and whether a field is to be decoded.

#HALT is used to halt the program before an output run so that the input file can be sorted before the next output run is commenced.

#HEAD specifies the main heading to be output at the beginning of an output report to the line printer.

#INTERROGATION introduces a re-entry to the interrogation phase of the program using a previously created parameter file.

#MERGE specifies the key field on which two files are to be merged and whether merging is to take place on the basis of a MATCH or a NON MATCH.

#OPTION specifies the output device type (line printer, magnetic tape or disc), the size of the line printer or output record, the output option and any record selection required from a hit file.

#OUTPUT introduces a re-entry to the output phase of the program using a previously created parameter file.

#PAGE specifies the heading to be printed at the top of each page of an output report to the line printer.

#READ defines input files. It specifies the device type, the filename, the RSN and FGN where applicable, the dumping frequency which is to be employed and any part file specification required.

#SPACE specifies the details of line spacing to be adopted on a line printer. Space lines can be inserted between headings, between records and between each line of a multi line record. Details of spacing can be linked to a paper tape loop or specified simply as a number of lines.

#STOP indicates the end of a parameter batch. This parameter is followed by the standard document terminator, that is, four asterisks (****) punched in the first four positions of a card/block.

#STORAGE enables run time alteration to be made to the sizes of certain major data areas in the program, for example, buffer sizes. This enables the scope of the package to be extended in terms of for example, buffer sizes or number of fields defined on machines with more than 16K core or re-arrangement within a 16K core environment by increasing some areas and decreasing others. The occupancy of the program is amended accordingly.

#UPDATE (#X63E only) defines the decode file to be amended and may be used to set the record length.

#WRITE defines output files. It specifies the output device, the file name, the RSN and FGN as applicable, the retention period, and the block size for MT files.

#XCONTROL specifies the range of values for the X control fields in the Table option and their associated levels of control.

#YCONTROL specifies the Y control fields for the Table option and, if the input file is unsorted, the range of values.

PARAMETER FORMATS AND CONVENTIONS

This section describes the formats of the following parameters:

#ADD	#OPTION
#CONTROL	#OUTPUT
#DEFINE	#PAGE
#DELETE	#READ
#DICTIONARY	#SPACE
#END	#STOP
#ENQUIRY	#STORAGE
#FORM	#UPDATE
#HALT	#WRITE
#HEAD	#XCONTROL
#INTERROGATION	#YCONTROL
#MERGE	

Format conventions

Each parameter is named by a major directive which is distinguished by a hash mark (#) in the first column of a parameter. It is only necessary to specify the first four characters (# inclusive) of a major directive but it can be specified in full.

Within each parameter the fields are in free format. Commas and spaces are used as separators and any number of spaces may occur between fields. The symbols comma and space (▽) indicate that either a comma or a space may be used as a separator. The comma alone (,) indicates that a comma must be used.

If optional fields are omitted, commas must be used as field separators so as to indicate clearly how many fields have been omitted. The last field should always be terminated by a space so that all trailing commas for optional fields at the ends of parameters should be omitted.

Leading zeros on numeric fields are not necessary and a positive sign is assumed if no sign is given.

The conventions concerning the use of brackets to explain the parameter formats are shown below. The brackets are not punched.

1 Gothic brackets indicate that a choice should be made between the values bounded by the brackets.

Example
⌈MT⌉
⌊ED⌋
⌊FD⌋

2 Round brackets indicate that the field between the brackets is mandatory in that parameter and that the value of the field should be inserted.

Example (Filename)

3 Square brackets indicate that the field is optional.

Example [Reel sequence number]

Punching conventions

The parameters may be punched in standard ICL 80 column cards or 8 track paper tape.

CARDS

In cards the parameter details must be punched between colums 1 and 72 inclusive. Columns 75 to 80 inclusive are reserved for an optional sequence number which can run through the complete parameter pack. If these columns are left unpunched on the first card, then no sequence check will be made, otherwise a strictly ascending sequence check will be made on all cards. Columns 73 and 74 are not used by the system and may contain user information.

PAPER TAPE

The newline character terminating a paper table block may be punched anywhere after the last non-space character in a block. For compatability between cards and paper tape, the maximum block size is limited to 72 character positions, excluding the newline.

GENERAL

If a parameter requires more than one card/paper tape block the split must be made between two fields and not in the middle of a field.

All card/paper tape input documents must be terminated by the standard document terminator. That is, four asterisks (****) must be punched in the first four positions of a card/paper tape block.

Parameter formats

#ADD (#X63E ONLY)

This parameter causes a record to be inserted or replaced. If the record key is not already present on the file, then the new record will be inserted, but if the key already exists the old record will be automatically replaced by the new one. Thus it is not necessary to delete a record before replacing it.

#ADD has two formats. If the record length has already been specified under #UPDATE then the format is as follows:

		Column 72	*Columns 76 to 80*
#ADD			
(	Record	)	[sequence number]
(	Record	)	[sequence number]
	etc.		

#ADD appears on a separate card. The record, *excluding* word 0 (the record length), is punched as characters starting in column 1, continuing up to column 72 if necessary and may overflow on to subsequent cards.

If the record length has not been given in the #UPDATE parameter then the format of #ADD is as follows:

		Column 72	*Columns 76 to 80*
#ADD			
(Record length), (	Record	)	[sequence number]
(	Record	)	[sequence number]
	etc.		

#ADD appears on a separate card. Word 0, the record length, is punched as an integer starting in column 1 with leading zeros if preferred. The integer is terminated by a comma and the record in characters must follow immediately after the comma. If the record overflows on to subsequent cards then the characters must start in column 1.

In both types of #ADD the sequence numbers are checked only within each record. There is no overall sequence check for parameters in #X63E.

The #ADD directive applies to all subsequent cards until another directive is encountered, so all insertions and replacements may appear under one #ADD. Each record must start on a new card/block.

#CONTROL

This parameter defines the control values for subtotalling in the LIST and TOTAL options

#CONTROL ▽, (Level) ▽, (Field Name) ▽, {n / PAGE / CHANNELn} ▽, [Comment]

The major directive is specified once only at the beginning of the set and may appear on the same card as the first control. The information for each control must start on a new card and no overflow is allowed. There may be up to seven levels of control, level 1 being the control that changes least often and is referred to as the highest control. The control numbers used should always range from 1 to n and be input in descending sequence. For example, if three levels of control are employed they should be numbered 3, 2, 1 and input in that sequence.

The control field is referred to by its dictionary name. A control field must be no more than 12 characters long and may not be variable length (type V or X). A control field cannot be decoded.

If CHANNEL n or n are specified, they have exactly the same form and meaning as they do when used under the #SPACE directive (see page 69). The appropriate vertical spacing is applied following the output of the lines of totals resulting from a control change at the particular level.

Example

```
#CONTROL 2, AREA 3
1, REGION , PAGE
```

This specification would cause the lines of totals resulting from a change of area to be separated in the List option from succeeding listing lines by three space lines. A throw to head of form would follow the output of totals lines resulting from a change of region.

If PAGE is specified, a page throw will occur every time the associated control changes.

Any comment may occupy up to twenty characters and must be preceded by an open square bracket. This comment will be used as a heading before the printing of the totals associated with the control change.

Details of all control fields must be specified and input in reverse order, that is from the lowest to the highest control.

Example

```
#CONTROL  3, WEEK,  [WEEKLY TOTAL
2, MONTH,  PAGE,  [MONTHLY TOTAL
1, YEAR  [YEARLY TOTAL
```

The information for each new month or year will start on a new page.

#DEFINE

This parameter predefines constants for use in the enquiry language by associating each with a name and comprises a dictionary of constants.

#DEFINE ▽ , (Name) ▽ , (EQL) (=) ▽ , (Constant)

Constant definitions can start on the same card/block as the major directive. Several definitions may be specified on the same card/block but a single definition must not be split between cards/blocks. All constants names must be of the same format as dictionary field names, that is, any number of alphanumeric characters of which the first must be alphabetic and the combination of the first seven unique within this parameter but they may duplicate field names.

Constants may be any of the following types:

1	Character (bounded by apostrophes)	for example	'BROWN'
			'JOHN BROWN'
	partially defined		'SM*TH'
2	Integer & Fractions	for example	+19, .591, −27.02
3	Sterling	for example	±L5,±L5S12D7,±L5S12D7.5
4	Decimal Sterling	for example	±£8.4
5	Date		DD/MM/YY
6	Octal	for example	#76342

Details of these constants and their limitations are given in Chapter 6.

Numeric constants under this directive are normally set up in a form suitable for comparison with fields having single word integer parts, for example single word type W fields, or two-word type M fields. If a numeric constant is to be set up in a form suitable for comparison with fields having two-word integer parts, the value of the constant should be immediately followed by (2). This requirement also applies to date and currency constants. Where the field has a single word integer, the constant may be followed by (1), but this is not essential.

A decimal sterling constant with three digits to the right of the decimal point, for example £266.305 or £471.120, will be converted to 1/1000's of the base unit (that is 1/10 pence); a form suitable for comparison with a type M field. A decimal sterling constant with two digits to the right of the decimal point, for example £305.79, will be converted to 1/100's of the base unit (pence); a form suitable for comparison with a type D field.

Example

```
#DEFINE DATE = 02/02/69 AREA = 'LONDON'
PRICE = L0S18D9          VALUE = 26290000 (2)
```

Reserved names

Two constant names are reserved to have special meanings and must not be used outside their special context.

1 TODAY. On any entry to the program, the date, as the number of days from 1st January 1900, is automatically associated with the constant name TODAY. Thus any reference to the constant @TODAY in an enquiry condition is equivalent to a reference to 'today's date'.

2 BASEDATE. The constant name BASEDATE is used if it is required to refer to a date field which has a basedate of other than 1st January 1900. If all dates are relative to 1st January 1900, BASEDATE must not be specified in the #DEFINE parameters.

If a value is associated with BASEDATE, for example BASEDATE = 01/01/60, then all subsequent dates in #DEFINE definitions and all date constants in the enquiry parameters will be converted so that they represent the number of days from the specified basedate, which in this case is 1st January 1960.

Note:

1 All dates specified under the #DEFINE directive *before* BASEDATE is defined will be converted according to the standard base-date. Thus it is possible to interrogate dates held relative to 1st January 1900 in the same run as dates relative to a more recent base-date.

2 If BASEDATE is re-defined on a re-entry run, then any dates already defined and any dates preceding the BASEDATE on the new #DEFINE parameter will be assumed to be relative to the old value of BASEDATE (or 1st January 1900 if BASEDATE has not been defined previously). Only dates which follow BASEDATE will be converted relative to the new value.

#DELETE (#X63E ONLY)

This parameter causes records to be deleted from the file, either singly or in a batch. The format is

#DELETE (Key 1) [- (Key 2)]

#DELETE may appear on a separate card. The keys may be expressed in character or binary form. In the former case they must be bounded by apostrophes, and hence may not contain apostrophes.

If a single record is to be erased then the key of that record is specified (Key 1). If two keys are specified, separated by a hyphen, then records with these and all intermediate keys will be deleted.

A key may not be split over two cards but if two keys are specified then each may appear on a separate card, in which case the hyphen must appear on the same card as the first key.

Binary keys may be single or double length integers but must be positive to conform to Housekeeping Standards.

#DELETE applies to all subsequent cards/blocks until another directive is encountered. Hence all deletions may appear under one #DELETE parameter. Each deletion or batch of deletions must start on a new card.

#DICTIONARY

This parameter defines the fields on the main or merge input file, giving the name by which each field is referenced in enquiry or output parameters. Definition can start on the same card as the directive #DICTIONARY, or on a new card: the Dictionary may occupy any number of cards.

There are two general formats depending on whether the field being defined is of fixed or variable length.Variable length fields may be subdivided further into type V (Terminator) or type X (Subrecord).

Fixed length fields

The format is as follows.

#DICTIONARY (Field name)∇(Field type)∇[W](Word address)[$\begin{Bmatrix}C\\B\end{Bmatrix}$(Character or bit position)]∇(Length)$\begin{bmatrix}W\\C\\B\end{bmatrix}$

The following points should be noted when defining fields in the record:

1 The field name may consist of any number of alphanumeric characters, of which the first must be alphabetic and the combination of the first seven unique within the Dictionary. Only the first seven characters of the field name are considered: if the field contains less than seven characters, the field name will be space filled up to that number. The field name should not contain either space or comma characters inserted by the user.

2 The character 'W' preceding the start address is an optional descriptor. If a field does not start at character position 0, the character position (C) of the start address must be given. If the field does not start at a bit corresponding to the start of a character, the bit position (B) must be given.

3 The field length can be given in one unit as either words (W) characters (C) or bits (B), but not as a mixture. If the code letter after the length is omitted, the length will be assumed to be in words.

Note: For further information on the format of the #DICTIONARY parameter when used to define fixed length fields, see Chapter 5, page 20.

Variable length fields

The format of the parameter is as follows:

#DICTIONARY (Field name)∇(Field type)∇ $\left\{\begin{array}{l}\text{Number of preceding fields/Terminating character}\\ \text{Number of preceding fields}\end{array}\right\}$ ∇(Address)

The format of the field name for variable length fields is the same as for a fixed length field.

Note: The two types of variable length field, V (Terminator type) and X (Subrecord type) cannot be combined in one file.

The following points should be noted when defining fields of type V and X.

1 TYPE V (TERMINATOR TYPE) This type of field is terminated by a specified by a specified character. Any character may be used, and a different character can be used for each field if required.

(a) The number of variable length fields preceding the field being defined must be stated. All these preceding variable length fields must be defined in the Dictionary.

(b) The variable length fields defined in the Dictionary must be defined in ascending order of their positions within the record, that is: 0, 1, 2, 3 . . . . n.

(c) It is only possible to give the address of the first variable length field. The format of the start address is the same as for a fixed length character field.

(d) If it is required to omit a field within a record the terminator must still be present.

(e) Up to 36 variable length fields may be defined.

2 TYPE X (SUBRECORD TYPE) Each variable length field of this type is preceded by a word containing the length of that field. Each field begins at a word boundary. This count word holds the whole number of words in the field, including the count word, in bits 9 to 23. Bits 0 to 8 are not checked and therefore may hold user information.

(a) The number of variable length fields preceding the field defined must be stated. These preceding fields do not have to be defined; the first one plus the one required is sufficient. They must, however, be present in the record, blank fields being indicated by a count word containing a count of one.

(b) The start address is only required for the first variable length field which must be defined, and has the same format as for a fixed length field.

(c) Any number of type X variable length fields may be defined.

Field type

The field type code denotes the type of information held in the field. The range of acceptable codes is shown below, with a short description of each field type.

Note: For information on maximum field lengths, the user is referred to Chapter 5, page 22.

B A binary field not held as an integral number of words, used for example, to store a bit pattern in which each bit represents a particular item, or to hold a positive integer. Integers held in an integral number of words are specified using field type W, D or H.

C A character field.

F A binary field, one word in length, holding a positive or negative fraction.

M A binary field, holding a positive or negative mixed number, of which the fraction part is one word long. The complete length is either 2 or 3 words.

P A binary field, holding a positive or negative fixed point number, of which the fraction occupies less than one word. The length is expressed in words and bits; the overall length of the field is always 1 or 2 words, and the bit length gives the length of the fraction part. In this format, the 'W' in the length is mandatory.

W, D or H Binary fields used to hold positive or negative integers. When defining values not relating to currency, field type W should be used.

When defining Sterling values, the codes should be used as follows:

W = binary pence (sterling)

D = binary newpence (decimal sterling)

H = binary new tenths of pence (decimal sterling)

When defining decimal currency other than decimal sterling, field type D should be used. The holding units can be $1/10^n$ths of the basic unit, where n ranges from 1 to 6.

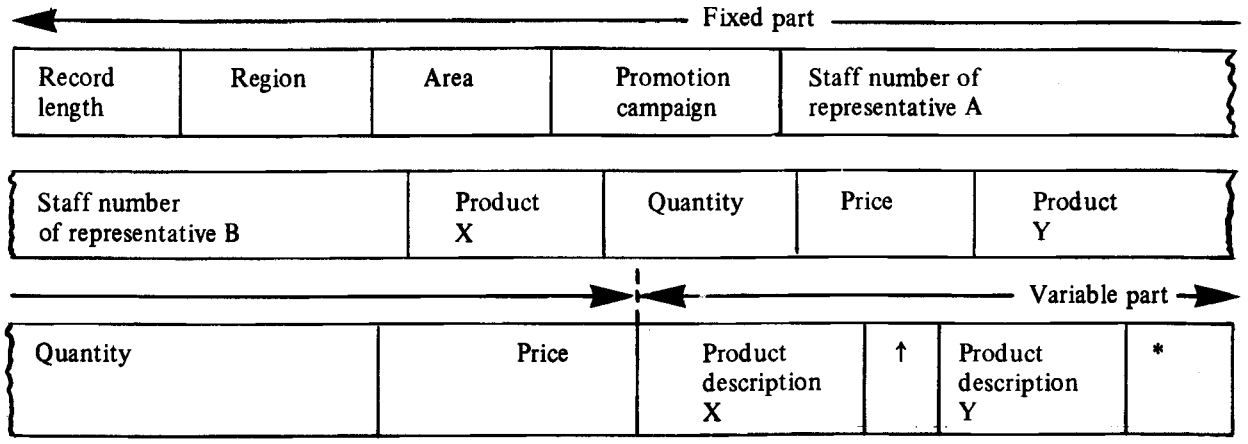
An example of a file definition, together with a Simplified Dictionary, is given below. For further information on the #DICTIONARY parameter, see Chapter 5, page 20.

Example of file definition and a simplified Dictionary

The file definition is considered in three phases. The relevant users' input file is illustrated, and the file is then defined under the #DICTIONARY parameter. Finally, an example is shown of the Dictionary that the computer department would provide for the user.

The example in this section consists of a Sales analysis of the products of a company. The file under scrutiny contains sales details such as selling price and quantity sold, together with the staff numbers of company representatives in a particular area. One field in the record (PROMOTE-CPN) can take one of only two values, either 1 or 0, to be indicated by the value of a single bit. The setting of this bit indicates whether a product promotion campaign has been launched in the area. The last two fields in the record contain product descriptions: it is assumed that the length of each record has been minimised by defining them in the variable part of the file. The first product description is always terminated by an upward arrow (↑) and the second by an asterisk (*).

The record format is as follows:



The fields can be defined by the #DICTIONARY parameter as follows:

#DICTIONARY REGION C W1,1W AREA W2,1W
PROMOTE-CPN B 3,1B A-STAFF-NO C 3C1,2W
B-STAFF-NO C 5C1,2W X-PRODUCT C 7C1,3C
X-QUANTITY M W8, 3W X-PRICE P W11,1W6B
Y-PRODUCT C W12,3C Y-QUANTITY M W13,3W
Y-PRICE P W16,1W6B
X-DESCRIPT V O/↑,W17 Y-DESCRIPT V 1/*

Field name (first 7 characters significant)	*Field type*	*Field length*	*Description*
REGION	C	4C	4 character code
AREA	W	1W	Full name of location
PROMOTE-CPN	B	1B	Bit set if promotion campaign launched
A-STAFF-NO	C	8C	8 character code
B-STAFF-NO	C	8C	8 character code
X-PRODUCT	C	3C	3 character code
X-QUANTITY	M	3W	Quantity held as 2-word integer and 1 word fraction.
X-PRICE	P	1W6B	1 word field holding a fixed point number with the fraction held in the last 6 bits of the word
Y-PRODUCT	C	3C	3 character code
Y-QUANTITY	M	3W	Quantity held as 2-word integer and 1 word fraction
Y-PRICE	P	1W6B	1 word field holding a fixed point number with the fraction held in the last 6 bits of the word
X-DESCRIPT-TYPE	V	–	Full product description
Y-DESCRIPT-TYPE	V	–	Full product description

Table 5: Example of a dictionary

#ENQUIRY

This parameter specifies the requirements which give rise to a file enquiry: the requirements are regarded as conditions which are broken down into the three components: field name, constant and relationship.

The set of enquiry parameters needed to specify an enquiry are preceded by the major directive, #ENQUIRY.

This must appear on a separate card or paper tape block.

Each enquiry is preceded by an enquiry identifier, giving an enquiry a name and identifying its type. An enquiry identifier must occupy a separate card/block. Up to 96 enquiries can be submitted in one run.

The format of the parameter is as follows:

? (Enquiry name) ∇ ; { STANDARD / COUNT / LEAD } ([) (Comment)

The first character of the parameter, a question mark, must appear in the first column of the card or paper tape block, and be followed immediately by the enquiry name. This name may be of any length, but only the first four characters will be used in the system.

Enquiry type

1 STANDARD ENQUIRY The STANDARD enquiry can be specified by name; alternatively, if no enquiry type is specified, a STANDARD enquiry is assumed.

2 COUNT ENQUIRY The COUNT enquiry outputs only the number of records that have satisfied the enquiry (the hit-count).

3 LEAD ENQUIRY The LEAD enquiry may be positioned anywhere in a batch of enquiries and affects all subsequent entries: those enquiries which follow a LEAD enquiry are only executed when the LEAD enquiry is a hit.

Note: For a full description of the Enquiry Language, and enquiry facilities, see Chapter 6 page 27.

The following simple example demonstrates the manner in which a complete enquiry may be written.

The firm concerned in this example wish to fill a vacancy for a systems analyst with a member of the present staff. The personnel file would have to be searched to find a programmer with two years experience in programming. The successful applicant would be aged 24 or over, be situated in Manchester, and have either a degree of six years experience in the firm.

The enquiry identifier would be specified as follows:

? VACANCY [VACANCY APPLICANTS

The complete enquiry would then be written as

(PROGRAMMING-EXP GEQ 2, AGE GEQ 24,LOCATION EQL MANCHESTER, AND (GRADUATE EQL1 OR SERVICE GEQ 6))

#FORM

This parameter varies according to the output option it is used with.

1 PRINT OPTION. The parameter specifies the fields to be output and the type of output for sterling fields.

#FORM

(Field name) [Edit symbol]

The major directive appears on a separate card/block. A list of the dictionary names of the fields to be printed follows on subsequent cards/blocks. Fields will be output in the order in which they are specified.

Fields may be of any length, but the first seven characters must be unique as defined in the dictionary. The first ten characters are used as column headings. Fields defined as type B in the dictionary may not be output with the PRINT option. At least one space must be left between field names.

Edit symbols are used to specify the type of output required for monetary fields as follows:

L = sterling

£ = new decimal sterling

C = decimal currency

E = binary dates

These symbols are enclosed in round brackets which must follow immediately after the field name. Full details of the scope of these editing symbols are given in Appendix 4. Details of the limits on the size of the fields which can be output are given in Appendix 3.

The number of fields which can be output on one line is governed by the field lengths and the size of the line printer used. If too many fields are specified the overflow fields will be ignored. An error will be indicated on the parameter listing and the option to continue the run is given.

Examples

#FORM

NAME ADDRESS DEPARTMENT SALARY (L)

#FORM

PART–NO, AMOUNT, COST(£), DELIVERY

2 LIST, TOTAL AND TABLE OPTIONS. The parameter specifies the record format and details of all fields to be output.

#FORM [{Number of fields per group / Number of records per line}]

Format line

Field definitions

The major directive appears on a separate card/block together with the number of records per line (1 or 2) for the List and Total options or the number of fields per group for the Table option. If the number of records per line is omitted, 1 is assumed (LIST and TOTAL).

The number of fields per group should only be specified when the fields in a group (i.e. corresponding to one X control value) are not to be repeated across the table (TABLE).

For line printer output the major directive must be repeated for each line of print specified. If #FORM is omitted the format is assumed to be the same as that of the last output report; in this case however the option must not have changed.

For tape and disc output several #FORMs may be used to produce one output record, as the maximum number of fields which may be specified under one #FORM is 40. If the format of the output record is to be identical to that of the main input file the #FORM directive should be omitted.

Format line

The output format of each field and the spacing between fields is specified by means of coded letters.

A Alphanumeric

B Spaces

Z Zero, used to zeroise fields as opposed to space filling, when output records are being written to a file on disc or magnetic tape

X Integers

F Fractional digits

L Pound sterling digits

S Shilling digits

D Pence digits

£ Decimal sterling digits

C Decimal currency digits

E Binary dates

These code letters are preceded by a number indicating the size of field required.

The following table shows the limits of the field sizes which can be specified. The Maximum limit for binary fields is arranged to coincide with double length working.

Field type	*Minimum size*	*Maximum size*
Alphanumeric	0A	127A
Spaces	1B	127B
Zero field	1Z	127Z
Integers	0X	15X
Fractions	3F	7F
Mixed	2X.1F	15X.6F
Pounds, shillings and pence	1L/2S/2D	12L/2S/2D
Shillings and pence only	2S/2D	14S/2D
Pounds only	1L	15L
Decimal sterling	1£	12£.3D
Decimal currency	1C	15C
Binary date	8E	8E

Table 6: Limits of field sizes

When specifying fields lengths for numeric fields, all floating and trailing symbols or percentage fields must be allowed for within the size of integer specified (pound or shilling specification for sterling fields). When outputting to magnetic tape or disc four zeroes or spaces should be allocated to the first word in the record, as this word will Subsequently be overwritten by the record length. At least one space should be left on the parameter between the specification of one field and the next, as commas must *not* occur in the format line.

The following minor conventions should be noted regarding the use of the various types of output field.

1 CONSTANTS IN THE FORMAT LINE. For the LIST and TOTAL options, constants may be inserted in the format line. Each constant must start with a non-numeric character, be terminated by a space and must not contain a comma.

2 MIXED NUMBER. A mixed number (integer followed by a fraction) should be separated by a decimal point.

Example

6X.3F will result in 999999.999

or –99999.999

3 SOLITARY FRACTION. When specifying the printing of a negative solitary fraction it should be noted that two print positions will be used for the sign and decimal point.

Example

4F will result in .999.

A fractional field which contains a mixture of both positive and negative values should be specified in a mixed number format.

Example

2X.4F will result in 0.9999 or –0.9999

4 STERLING. Sterling fields may take one of three forms:

Pounds shillings and pence	for example	4L 2S 2D
Shillings and pence	for example	3S 2D
Pounds only	for example	7L

In the first two forms the space may be replaced by an oblique, and each may be followed by a fractional field.

Example

3S/2D.3F resulting in 999/11.999

Decimal sterling is specified using the symbols £ and D separated by a decimal point.

Example

4£.2D resulting in 9999.99 or –999.99

In both types of sterling the £ sign is not printed unless specified on the field cards, as explained later. Alternatively, it may be specified as a constant in the format line. When post signing is used, two additional positions should be specified for the pound digits.

Example

9999/19/11CR should be specified as 6L 2S 2D

5 DATES. Dates held in character form will be output using the normal code letter for all alpha numeric output, 'A'. Obliques may be inserted between day and month etc. by specifying the day and month as separate fields and inserting an oblique symbol as a constant in the print line between the two fields.

A binary date must always be expressed as 8E and will be output in the form DD/MM/YY including the obliques.

Example

25/11/69.

6 BIT FIELDS. Fields which have been defined as type B in the Dictionary may be output as either alphanumeric, code A, or as an integer code X.

When using the code letter A, the field is processed left justified and rounded up to the next multiple of 6 bits. That is, the least significant end is zero filled.

When outputting with code letter X, the field is processed right justified in one word and output as an ordinary integer. If it is required to output single bit fields as ones or zeros code letter X should therefore be used.

7 DECIMAL CURRENCY. Decimal currency fields, other than decimal sterling, should be specified as xC.yC where y indicates the value of the holding units.

e.g. 4C.2C indicates holding unit = $\frac{1}{100}$ basic unit.

A field value of 10000 described as

(a) 4C.2C will be output as 100.00

(b) 4C.4C will be output as 1.0000

8 ALPHANUMERIC. The output code A may be used to transfer data as character strings to a print line or output record, beginning at the start address of the field given in the field definition (see below). If the field in question is of type W, M, F, P, H or D, it may be transferred in binary form to an output record by the use of 4A, 8A or 12A. For example, 8A would be used to transfer a two word M or W field, since four characters are contained in a 1900 word. 4A would be used to transfer a single integer field of type W in binary form to an output record.

Output code A may also be used to write the results of arithmetic operations, or totals (see below under *Field definitions*) to output records in binary form. These arithmetic values are held internally in three words, comprising two integer words and one word containing the fractional part of the value. The user need not concern himself with the format of these values if he is simply printing the results or totals. If the user wishes to write the results to output records, however, the use of 12A or 8A will transfer the whole mixed number or just the two word integer part in binary form.

If the file containing these results or totals is to be re-input to FIND-2, the fields may be defined in the Dictionary as three word type M and two word type W fields.

Field definitions

For each field specified in the format line, there must be an associated field definition. The field definitions must be input immediately after the format line specification and in the same order as the fields to which they correspond.

Each field definition must start on a new card/paper tape block and may occupy an unlimited number of cards/blocks. If a definition overflows, the comma terminating each item must appear on the same card/block as the item which it terminates (note that commas are mandatory as item separators in the field definition). Thus a trailing comma will be taken to indicate that the next card/block continues the current field definition, and trailing commas should therefore be omitted from the end of complete field definitions.

In the TABLE option, if the fields in a group are to be repeated across the table for each X control value (i.e. the number of fields per group has not been specified on the #FORM parameter):

(a) The Y control(s) must be specified as the first field(s) in the print line and in the order Y1, Y2

(b) The field definitions of the first group only should be specified

There are four types of field card.

1 GENERAL PURPOSE FIELD CARD

The general form is:

[Field tag], (Field name), [Positive symbol], [Negative symbol], [Edit symbols], [Totalling requirement],

(a) Field tag

This may be up to seven characters long, the first of which must be numeric. It is an optional field and is used to name results of calculations for reference in other calculations.

(b) Field name

The output field is usually a straight extract from the record and is therefore referred to by its dictionary field name. The special reserved field name TODAY may also be specified in conjunction with the format 8E to output today's date. An arithmetic function between two field names, field tags or constants may be specified in place of a single field name.

$$\left\{\begin{array}{l}\text{Field name}\\ \text{Field tag}\\ \text{'Constant'}\end{array}\right\} \nabla \left\{\begin{array}{l}+\\ -\\ *\\ /\end{array}\right\} \nabla \left\{\begin{array}{l}\text{Field name}\\ \text{Field tag}\\ \text{'Constant'}\end{array}\right\}$$

A calculation may comprise any combination of the above, the results of previous calculations being referenced via the use of field tags.

The form of the constants may be any of the valid forms for numeric constants used in the enquiry language (see Chapter 6, page 27). Constants should be bounded by apostrophes to distinguish them from field names, for example:

,AMOUNT + '236' or ,PRICE + '£25.42'

The arithmetic symbols used are:

+ Addition

– Subtraction

* Multiplication

/ Division

When arithmetic results are being totalled in the TABLE option, and the fields being totalled in each group are different, arithmetic should not be specified between fields in different groups.

A field tag identifying an arithmetic result may not be used in an arithmetic expression on a subsequent line when records are output two to a line.

Where arithmetic results are to be referenced in subsequent calculations the output of such intermediate results in print lines or records can be avoided by the use of coded letters OX or OA in the format line.

(c) Positive symbol

An optional two character symbol of the user's choice for example, CR, which will be output to the right of all positive field values.

(d) Negative symbol

An optional two character symbol for example, DB, which will be output to the right of all negative field values. This replaces the otherwise automatic floating minus sign.

(e) Edit symbols

Any one of the following four symbols may be optionally specified.

(i) £ indicating a floating pound sign.

(ii) $ indicating a floating dollar sign.

(iii) * for cheque protection on numeric and sterling fields all leading zeroes being replaced by asterisks.

(iv) Z indicating zero suppression on character fields (zero suppression is automatic on numeric fields).

(f) Totalling requirement

(i) LIST and TOTAL options.

When using control totalling with these options it is necessary to specify each of the levels at which the field is to be totalled. Up to twelve fields may be totalled at seven levels, plus a grand total. Hence the numbers 1 to 7 and/or GRAND may be specified. The level numbers should be separated by a comma and can occur in any order.

For example, a field to be totalled at levels 1, 3, 4 and GRAND would be input as 1, 3, 4, GRAND.

(ii) TABLE option. With this option, this field is used to indicate whether a percentage and/or a row total is to be produced. It therefore takes the form

[COLUMN%] [ROW]

An allowance of 5 extra print positions must be made in specifying the length of the field in the format line when requesting a column %.

With the ROW total facility, a separate field size must be specified at the end of the format line, but no field definition is required.

ROW totalling is not permitted when the number of fields per group has been specified after the #FORM directive, that is, when field cards for the entire line are present.

COLUMN% is allowed only when the file is unsorted on the Y control field.

Example

An example of a complete field definition could be

1VALUE,PRICE * QTY,,,£,2,3

The contents output would be the product of the two fields PRICE and QTY. It would be preceded by a floating £ symbol and totals output when control levels 2 or 3 change value.

More complete examples may be found in Chapter 9.

2 FOR FIELDS BEING DECODED. The format for this type of field definition is as follows:

[Field tag], (Field name), (DECODE), (File name), [Buffer number]

This type of definition can be used only with the LIST option. The field being decoded must be type B or C and may not be a control field.

Example

,AREA,DECODE,DECODEFILE

A maximum of six decode files may be used during a run. Decoding will operate satisfactorily if the user simply supplies the appropriate decode file names alongside the field names under #FORM (as indicated above) and sets the decode file input buffer (code number 13, see page 94) to an appropriate size. Under these circumstances, however, records from all the decode files used will be read into the same storage area. The repeated over-writing of data in core and the consequent repeated disc accesses which this entails can be avoided by allocating an individual buffer to each decode file. This is achieved by use of the #STORAGE codes 14 to 18 in addition to code 13, enabling an additional five buffers to be expanded (their pre-set size is zero) for use by up to five additional decode files. These numbers are then supplied as additional items on the field definitions under #FORM, immediately after the decode file names. For example, a run using four different decode files, each of which is to use an individual buffer, might include under #STORAGE:

13/128, 14/256, 15/128, 16/128

and under #FORM:

, FIELDA, DECODE, DECODEFILEA, 13

, FIELDB, DECODE, DECODEFILEB, 14

, FIELDC, DECODE, DECODEFILEC, 15

, FIELDD, DECODE, DECODEFILED, 16

, FIELDE, DECODE, DECODEFILEA, 13

It should be noted that the file numbers of the decode files used in error messages are not connected with the above code numbers of the input buffers, since more than one file can share a buffer if required. The file numbers 6 to 11 are in fact allotted to the different files in the order in which they are initially encountered in the field definition. In the above example DECODEFILEA would be given the number 6, DECODEFILEB would be 7, and so on.

CONTROL FIELD RECORD COUNTS. For use with the List and Total Options, the format for this type of field definition is as follows:

[Field tag], (COUNT), { GRAND / Control level }

This specified that a record count is to be accumulated within the associated control level and output on the totals line when the control value changes. Only one control level may be specified.

Example

,COUNT,2

4 TABLE OPTION COUNT. The number of records satisfying particular X and Y control values may be accumulated in the table. The field card is as follows:

[Field tag] ,(COUNT),[COLUMN],[ROW]

The restrictions on the use of COLUMN and ROW are the same as for the general field card.

#HALT

This parameter is used to halt the program immediately before an output run. The major directive constitutes the whole parameter.

#HEAD

This parameter specifies the main heading.

#HEAD	
(Up to 60 characters of text)	[Line number]
Columns 1-60	Columns 61-72

The major directive appears on its own card. The heading text is input on the following cards/blocks in the first 60 columns of each.

Headings are limited to a total of 800 characters; thus the limit on the number of lines of heading depends on the line printer size, as follows:

Number of print positions	*Line limit*	*Maximum number of cards per line*
96	8	2 (Using first 36 columns of second card).
120	6	2
160	5	3 (Using first 40 columns of third card).

An optional line number may appear anywhere in columns 61-72. Line numbers must be in ascending sequence and must not exceed the maximum value for the appropriate line printer size.

The use of line numbers eliminates the need to provide blank cards in the following cases:

1 When the final card (or cards) of a line is blank.

2 When a complete line is blank.

If line numbers are not used the maximum number of cards per line must always be provided. If the maximum number of lines is not used the final blank lines need not be specified.

If the optional sequencing (in columns 75-80) is used on card parameters, it should continue throughout the cards forming the #HEAD parameter.

Example

#HEAD	
PERIOD REPORT	2
SALES ANALYSIS	3

#INTERROGATION

This parameter initiates a re-entry to interrogation. The directive constitutes the whole parameter.

#MERGE

This parameter specifies the merge key and the basis for merging

#MERGE (Key 1) ∇ , (Key 2) ∇ , {MATCH / NON MATCH}

The parameter must occupy only one card/block.

Key 1 Field name defined in the main dictionary

Key 2 Field name defined in the merge dictionary

The merge file field names must not duplicate the main file field names.

The file which is defined first in a parameter set is always taken to be the main file.

The main file may contain several records with the same key, whereas the merge file must only have 1 record for each key.

The last field specifies whether output is required when a match occurs or when no match is found.

Example

#MERGE PARTNO, ITEMNO, MATCH

#OPTION

This parameter specifies the output device type, the output record size and any record selection required from a hit file on the basis of enquiries satisfied.

#OPTION ▽, (Output device type) ▽, (Record size) ▽, { PRINT / LIST / TOTAL / TABLE } ▽, [[Hitn ▽,] (Record selection criteria)]

The major directive and mandatory fields must appear on the same card/block but the record selection may overflow onto further cards/blocks.

The output device type is specified as follows:

LP	Line Printer
MT	Magnetic Tape
ED	Exchangeable Disc/Twin Disc
FD	Fixed Disc

For the line printer, the size of the line must be specified as the user's output *line*-size rather than *line-printer* size

Example

```
#OPTION    LP    136    LIST
```

The #OPTION directive shown above will enable two records to be printed on each line, on stationery of width 68 characters. It should be noted that the half page size must be an integral number of words.

No value greater than 160 may be quoted.

For magnetic output device types, the size of the output record must be specified in words. The maximum size is 1024 words.

PRINT, LIST, TOTAL or TABLE specifies the type of option required and must be compatible with the device type specified. Details of compatibility between the device type and option selected are given in table 4 in Chapter 7, page 42.

Record selection criteria

Record selection in the basis of enquiries satisfied only applies if output is to be from a hit file.

Those enquiries which must all have been satisfied for the record to be output must be preceded by ALL. Those enquiries of which at least one must have been satisfied for the record to be output must be preceded by ANY. The enquiries are referred to by the names specified in the enquiry parameters.

Example

```
#OPTION    LP    120    LIST ALL,    ENQ1,    ENQ2
ANY,    ENQ3,    ENQ4
```

If there was only one enquiry at interrogation, record selection criteria must not be specified.

Output only runs may be performed using a previously created hit file as input, but with a different hit selection from the one originally specified (if any). To do this, in place of enquiry names after ALL and ANY, the user must insert items with the form ZZn: n is a number of 1 or more digits defining the position, starting at 1, of the desired enquiry in the original correct enquiry list.

To indicate to the system the number of hit words, the user should specify HITn, where n is the number of hit words, as an extra item immediately after the option type item on the parameter. If this is omitted, it will be assumed that n = 1.

The number of hit words is obtained by dividing the number of correct enquiries in the original list in the interrogation run by 24 and rounding up.

Example

An interrogation has been done using 40 enquiries EN01, EN02, . . . EN40.

A separate output only run to select from the hit file those records satisfying EN02, EN04 and either of EN05 or EN06 would require a parameter for the form:

```
#OPTION LP 120 LIST HIT2 ALL ZZ2 ZZ4 ANY ZZ5 ZZ6
```

#OUTPUT

This parameter initiates a re-entry to output. The directive constitutes the whole parameter.

#PAGE

This parameter specifies the heading to be printed at the top of each page of an output report to the line printer.

#PAGE	
(Up to 60 characters of text)	[Line number]
Columns 1-60	Columns 61-72

Rules for this parameter are exactly the same as those for #HEAD.

#READ

This parameter names the files to be input to the system. It also specifies the bounds of any part-file interrogation and defines the dump frequency.

#READ, {MT / ED / FD}, (Filename), [Reel sequence number],

[File generation number], [Dump frequency], [{Part-file specification. / Ouput group number.}]

The major directive, #READ, may appear on the same card as the remainder of the parameter. Any number of cards may be used for remaining items but one item must not be split between two cards. The appropriate device type (MT, FD or ED) must be inserted after the #READ directive. If the file name contains more than twelve characters, only the first twelve are considered; if less than twelve characters, the rest of the field is space filled. Leading zeros may be omitted on all numeric fields. The commas between fields are mandatory even if the fields are absent.

Reel sequence number and file generation number

For magnetic tape files the reel sequence number or file generation number may be omitted and zero will be assumed. On disc files the reel sequence number should be omitted and the file generation number may be specified as −1 if it is required to interrogate the highest generation of the file on-line to the system.

Dump frequency

The frequency of dumping from the main input file can be specified in one of three ways:

1 NO or field left blank	No dumping will occur.
2 EOR	Dumps occur on reaching the end of each reel of the input file, except for the final reel.
3 n (thousand)	Dumps occur after processing n thousand records on the main input file.

Part-file specification and output group number

If the user wishes to interrogate a complete main input file, the part-file specification field should be left blank.

On direct access indexed sequential files interrogation may be performed between two record keys (*n* and *m*); on magnetic tape files interrogation may be performed between two user sentinels (*n* and *m*).

The format for both types of part-file specification is:

n – *m*

where *n* and *m* are values of record keys within a direct access indexed sequential file, or user sentinels within a file on magnetic tape.

The system assumes an ascending sequence of key values in all cases.

If interrogation is to be from the start of the file, *n* should be zero, and if interrogation is to be to the end of the file *m* should be replaced by EOF.

Note:

1 Magnetic tape files: interrogation to the end of a reel can be specified by EOR, for example, 4-EOR.

2 Direct access indexed sequential files:
if *n* and *m* represent the values of keys in character form they must be bounded by apostrophes, for example: 'A196'-'C307'. Apostrophes are therefore not permitted as key characters. Binary values may be signed, for example, -213--104. and keys represented by binary values must be signed single or double length binary fields.

Each key may be specified on separate cards to allow specification of long character keys. In this case, the separating hyphen may appear on either card, but it must be present.

Output group number

When #READ refers to the parameter file at re-entry to the output phase, this field specifies the number of the output group to which re-entry is to be made. The field must be an integer in the range 1 to 20.

Example

```
#READ, MT, TAPEFLE, , , ,2-4
#READ, ED, DISCFLE, , - 1
#READ, FD, TESTFILEDATA, , 1 ,NO, '14THA'-EOF
```

OPTION

This parameter specifies the output device type, the output record size and any record selection required from a hit file on the basis of enquiries satisfied.

#OPTION ∇, (Output device type) ∇, (Record size) ∇ { PRINT / LIST / TOTAL / TABLE } ∇ [[Hitn ∇,] (Record selection criteria)]

The major directive and mandatory fields must appear on the same card/block but the record selection may overflow onto further cards/blocks.

The output device type is specified as follows:

LP	Line Printer
MT	Magnetic Tape
ED	Exchangeable Disc/Twin Disc
FD	Fixed Disc

For the line printer, the size of the line must be specified as the number of print positions available on the printer to be used, that is 96, 120 or 160.

For magnetic output device types, the size of the output record must be specified in words. The maximum size is 1024 words.

PRINT, LIST, TOTAL or TABLE specifies the type of option required and must be compatible with the device type specified. Details of compatibility between the device type and option selected are given in table 4 in Chapter 7, page 42 .

Record selection criteria

Record selection in the basis of enquiries satisfied only applies if output is to be from a hit file.

Those enquiries which must all have been satisfied for the record to be a hit must be preceded by ALL. Those enquiries of which at least one must have been satisfied for the record to be a hit must be preceded by ANY. The enquiries are referred to by the names specified in the enquiry parameters.

Example

```
#OPTION    LP    120    LIST ALL,  ENQ1,  ENQ2
ANY,  ENQ3,  ENQ4
```

If there was only one enquiry at interrogation, record selection criteria must not be specified.

Output only runs may be performed using a previously created hit file as input, but with a different hit selection from the one originally specified (if any). To do this, in place of enquiry names after ALL and ANY, the user must insert items with the form ZZn. N is a number of 1 or more digits defining the position, starting at 1, of the desired enquiry in the original correct enquiry list.

To indicate to the system the number of list words, the user should specify HITn, where n is the number of hit words, as an extra item immediately after the option type item on the parameter. If this is omitted, it will be assumed that n = 1.

The number of list words is obtained by dividing the number of correct enquiries in the original list in the interrogation run by 24 and rounding up.

Example

An interrogation has been done using 40 enquiries EN01, EN02, . . .EN40

A separate output only run to select from the hit file those records satisfying EN02, EN04 and either of EN05 or EN06 would require a parameter of the form:

```
#OPTION LP 120 LIST HIT2 ALL ZZ2 ZZ4 ANY ZZ5 ZZ6
```

#OUTPUT

This parameter initiates a re-entry to output. The directive constitutes the whole parameter.

#PAGE

This parameter specifies the heading to be printed at the top of each page of an output report to the line printer.

#PAGE	
(Up to 60 characters of text)	[Line number]
Columns 1-60	Columns 61-72

Rules for this parameter are exactly the same as those for #HEAD.

#SPACE

This parameter specifies the details of line spacing.

$$\text{\#SPACE} \quad \nabla \quad \left\{ \begin{matrix} n \\ \text{PAGE} \\ \text{CHANNEL 1-7} \end{matrix} \right\}$$

The parameter occupies one card/block.

The number of lines to be spaced (the maximum is ten) is signified by 'n' but if a new page is detected before the spacing is complete, the extra lines of spacing will be ignored.

PAGE specifies a throw to a new page.

CHANNEL 1-7 specifies a throw to Channel 1-7 in a paper tape loop.

#STOP

This parameter indicates the end of a parameter batch.

The major directive constitutes the whole parameter.

#STORAGE

After the major directive, definitions of altered areas may appear beginning on the same or a subsequent card/block. These definitions take the form x/y where x is the code number of the area whose size is being altered, and y is its new size either as a new number of entries for a table or as a new number of words of store (see Chapter 10). The new sizes may be specified in any order of code number, and if any code number is specified more than once, the last size specified is the one which will apply. Use of ALL/0 reduces all variable areas to zero, regardless of what has been specified previously. Subsequent size specifications then modify this overall reduction. The fields may occupy any number of cards or paper tape blocks, including the one containing the major directive. A space or comma should appear between each definition.

Example

```
#STORAGE  ALL/O,   1/512  8/50,   5/50
```

#UPDATE

This parameter names the file to be amended by the decode file update program, and is limited to a single card/ paper tape block.

#UPDATE, (File name), (File generation number), [Record length]

The filename should be 12 characters long; if more than 12 characters are present, the extra characters are ignored, if less than twelve, the rest of the file name is space filled.

If the generation number is given as -1 the file with the highest number on-line will be opened.

To conform to Housekeeping standards each record must contain the record length in its first word. If the records to be added to the file are all of fixed length then this length may be specified as the last field but this is not possible if the records are of variable length. Inclusion of the field eliminates the need for the length to be specified on each #ADD parameter.

#WRITE

This is an optional parameter used to give a name to a file being created by the FIND-2 system, for example, a parameter or output file.

$$\#WRITE,\ \left\{\begin{matrix}ED\\FD\\MT\end{matrix}\right\},\ (Filename),\ \left\{\begin{matrix}\text{Old file generation number}\\\text{Reel sequence number}\end{matrix}\right\},$$

$$\left\{\begin{matrix}\text{New file generation number}\\\text{File generation number}\end{matrix}\right\},\ (Retention\ period),\ [Block\ size]$$

The parameter is limited to a single card. When referring to the hit file, the parameter may only be omitted if all enquiries are COUNT type.

The appropriate device type must be chosen and inserted after #WRITE.

If the file name contains more than twelve characters, only the first twelve are considered; if less than twelve characters, the rest of the field is space filled. Leading zeros may be omitted on all numeric fields.

The requirements for the parameter format depend upon whether the file concerned is held on magnetic tape or a direct access device.

Magnetic tape

1 The reel sequence number or file generation number may be omitted, in which case zero will be assumed. However, the commas between fields are mandatory even when the fields are absent.

2 The retention period is mandatory.

3 The block size is optional and may be specified for the hit file or an output file. The parameter file always has a block size of 512 words. If block size is omitted, it will be set to 1024.

Direct access

1 The old generation number of the file must be specified. If it is sufficient to open the highest generation of the file on-line to the system, -1 may be specified.

2 The new generation number of file must be specified and will be given to the file created.

3 The retention period is mandatory but may be specified as zero.

4 The block size of the file created is specified when the file is allocated by the allocator routine and is therefore not present on this parameter.

#XCONTROL

This parameter specifies the range of values for Xcontrol fields and their associated control levels.

$$\#XCONTROL,\nabla\ (Xlevel),\nabla\ \left\{\begin{matrix}(Field\ name)\\(?)\end{matrix}\right\}\ \nabla\ ,\ \nabla\ \left\{\begin{matrix}(List\ of\ values)\\(List\ of\ enquiries)\end{matrix}\right\}$$

The major directive is specified once only at the start of the control definitions and may appear on a separate card/block.

Each control level definition must start on a new Card/block and may occupy as many cards/blocks as necessary.

Up to three levels of control are allowed and must be specified as X1, X2 or X3. Level 1 represents the control which changes value least often. Control must be input in decreasing numeric sequence so that X1 is input last.

At any one level, each group of values which lie within one value of the next higher control level must be terminated by a semicolon, except the last group for which the semicolon is optional. If a group of values is repeated it is only necessary to specify the semicolon designating the end of the group.

If a group is to remain unspecified (so that all values are accepted) an upward pointing arrow ↑ is inserted in place of the string of values. This method of indicating an unspecified group may be taken to the logical limit of a single unspecified X control group, that is, no X control at all, with the parameter appearing as shown in the example below:

```
# XCONTROL, X1, WD15, ↑
```

The X control field is a dummy, and the whole line width is available to hold fields to be totalled in a single column. This facility offers a simple form of control totalling at one or two levels (the Y control) in which the file may be unsorted at one level.

The parameter can operate in two modes: control by field value and control by enquiry.

1 CONTROL BY FIELD VALUE. The control field is indicated by its dictionary name. Values of this field for which output is required are specified in the list with semicolons as appropriate.

  Non numeric field values are limited to twelve characters and may not be variable length. Character fields may not be bounded by apostrophes. Numeric fields may be of any type.

2 CONTROL BY ENQUIRY. If ? is specified in place of the field name, record selection is performed on the hit word. Hence this mode is possible only when a hit file is used as input to the current output group. The list of enquiry names corresponds to the list of values in the control field mode, and a record will be output if it satisfies a particular enquiry. In this way selection may be achieved on a range of field values, for example. Other restrictions on control field values may be overcome in this way. The enquiries used should be mutually exclusive, as fields are totalled into one column only.

  If an output only run is to be performed using a hit file, the ZZn standard must be used in place of enquiry names as described for the #OPTION parameter. The number of hit words is not specified under #XCONTROL, but must be stated under #OPTION, using HITn, which need not be followed by any enquiry selection specification.

The number of X values which can be specified at each control level is limited as follows:

1 For the lowest X control level the limit is 24 values.

2 For the second lowest level the limit is 16.

3 For the highest of three levels the limit is 8.

Examples

1

1966	1967				1968				1969	
	QTR1	QTR2	QTR3	QTR4	QTR1	QTR2	QTR3	QTR4	QTR1	QTR2

```
#XCONTROL  X2  QTR  ↑ ; 1,  2,  3,  4  ; ; 1,  2
X1  YEAR,  1966  1967  1968  1969
```

2
```
#XCONTROL  X3  TYPE  A,  B  ;;;;
X2 ? ENQ1  ENQ2  ; ENQ3  ENQ4
X1  AREA  NORTH,  SOUTH
```

#YCONTROL

This parameter specifies the Y control level and for an unsorted file, its range of values.

#YCONTROL ∇, (Ylevel) ∇, (Field name) ∇, [{PAGE / List of values}]

The major directive need appear once only at the start of the control definitions and may appear on a separate card/block.

Each control except the first, must start on a new card/block and may occupy as many cards/blocks as necessary.

Up to two controls are allowed and the level should be specified as Y1 or Y2.

Level 1 represents the most significant control which changes value least often.

Where two levels of control are specified, the file must be sorted on the most significant control.

The parameter can operate in two modes: control by field value and control by enquiry.

1 CONTROL BY FIELD VALUE. The control field is identified by its dictionary name. Values of this field must be specified in the following cases:

   (a) One Y control: if file is unsorted

   (b) Two Y controls: for the second level only if the file is unsorted on this field

2 CONTROL BY ENQUIRY. For selection by enquiries satisfied on the Y control axis, the Y control parameters are specified in a way exactly analogous to the X control parameters, for example:

```
# YCONTROL Y2 ? ENQ1 ENQ2 ENQ3 ENQ4
Y1 DEPT
```

   Under the #FORM directive, a dummy Y control field must be specified for the appropriate level (any field will do) and if the format line includes 4A for this dummy field then the enquiry name will be printed in the appropriate place.

PAGE can be used if two Y controls are used and must be specified with Y1. A page throw will then be given each time the value of this control changes.

Up to 27 values may be specified for an unsorted file and the rules for specification are the same as those for the #XCONTROL parameter.

Example

```
#YCONTROL   Y2 AREA
Y1 REGION PAGE                          File sorted by region and area.
```

Example

```
#YCONTROL   Y1 TOWN BIRM MANC LIVR      File sorted or unsorted.
NEWC GLAS EDIN DUND
```

Chapter 9 Examples

The purpose of this chapter is to illustrate the use of some of the parameters by means of detailed examples.

Examples 1 to 5 concentrate on the enquiry specification. Each example describes an information retrieval problem and gives details of the type of information held on the appropriate master file. Suitable enquiry parameters which could be used to interrogate the file are then illustrated.

Examples 6 to 10 illustrate five different types of reports and the associated output parameters which could be used to produce them.

Example 11 shows how the Decode File Update program may be used.

EXAMPLE 1

This example illustrates the basic enquiry facilities and Print output option.

From a list compiled by agents, a user wishes to select, for purchase or rental, a warehouse which meets his requirements. Details of suitable buildings are to be printed out. Eligible buildings must:

1 have at least 50,000 sq. feet of floor space

2 be located in London

In addition, the user will definitely rent the building if the rental is £1,500 p.a. or less, but in the event of rent being higher than this sum, he will buy, provided the price is not greater than £20,000 and he can obtain a mortgage for at least half this sum.

The file would consist of records each containing at least the following information.

1 Floor area in square feet. Probably a type W field, 1 word long

2 Location. Probably a type V or X variable length field.

3 Annual rental in pounds. Probably a type W field, 1 word long.

4 Purchase price in pounds. The same field type as 3.

5 Minimum acceptable deposit in pounds. The same field type as 3.

The simplified Dictionary (see Table 2 on page 25 of Chapter 5) would probably include:

Field name	*Field type*	*Field length*
FLOOR	W	1W
LOCATION	V	–
RENTAL	W	1W
PRICE	W	1W
DEPOSIT	W	1W

The enquiry would then be:

```
#ENQUIRY
?WAREHOUSES    STANDARD    [SUITABLE WAREHOUSES
(FLOOR GEQ 50000 AND LOCATION EQL LONDON AND
(RENTAL LEQ 1500 OR ( PRICE LEQ 20000 AND DEPOSIT LEQ 10000 ) ) )
```

Note: If the monetary fields were actually held on the file in pennies instead of pounds, as is frequently the case, the user simply has to put the constant values in sterling form and need not try to work out the number of pennies in twenty thousand pounds. Thus the monetary values in the above example would appear respectively as:

L1500, L20000 and L10000.

To achieve a simple listing of certain fields from each record which is a hit, the user can specify PRINT in the details of the #OPTION parameter and simply list the names of the fields required under #FORM:

```
#FORM
OWNER  ADDRESS  FLOOR  RENTAL  PRICE  DEPOSIT
```

where OWNER and ADDRESS are defined as, say, type V fields in the dictionary.

Note: If the monetary fields were held on the file as binary pence, sterling output could be obtained by writing the appropriate output fields as:

RENTAL(L) PRICE(L) DEPOSIT(L)

As page of output might then appear as:

19/08/69 PAGE NO

OWNER	ADDRESS	FLOOR	RENTAL	PRICE	DEPOSIT
J.R. DAVIES & CO LTD	5 HARDY ST EC2	78000	1350	18150	2000
B.B. ARKWRIGHT (TANNERS) LTD	MOOR END RD SE1	60000	945	22000	8000
INTERNATIONAL DYES & STAINS LTD	MILL LANE NEW CROSS	110000	1670	15000	8000

EXAMPLE 2

This example illustrates short cuts in the enquiry language.

The user wishes to search personnel files to count the number of employees at certain job locations who will be affected by a proposed extension of the pension fund to come into operation on the 20th August 1969. Those concerned will be employees aged between 20 and 25 on that date who have been with the company for at least 1 year and are located in Sheffield, Walsall or London.

The file might contain two date fields, (held as binary days since 1900), and a variable-length field for location. The simplified Dictionary might thus include:

Field name	*Field type*	*Field length*
DBIRTH	W	1W
DJOIN	W	1W
LOCATION	X	

The enquiry could then be phrased:

```
#ENQUIRY
?PERSONNEL COUNT
(DBIRTH BET 19/08/44, 19/08/49 AND DJOIN < = 19/08/68
AND LOCATION EQL SHEFFIELD ; WALSALL ; LONDON )
```

EXAMPLE 3

This example illustrates the arithmetic and step facilities. A user's file contains, in no particular order, details of company shares, including fields giving the dividends paid in each year and a variable length field containing descriptions of the relevant company's spheres of activity, for example, chemicals or shipping. The user wishes to retrieve details of shares of companies active in the electronics field whose dividend in 1969 was less than that in 1968. The simplified Dictionary might include:

Field name	*Field type*	*Field length*	*Description*
DIV68	W	1W	(as a binary %)
DIV69	W	1W	(as a binary %)
ACTIVITIES	X	–	

The enquiry could then be phrased:

```
#ENQUIRY
?ELEC          [SLIPPING COMPANIES
( DIV69 – DIV68 LSS 0 AND STEP 1023 BY
  1 ACTIVITIES EQL ELECTRONIC )
```

The step search has the effect of searching the entire variable length field advancing a character at a time, up to a limit of 1023 steps, searching for the key-word ELECTRONIC.

EXAMPLE 4

This example illustrates the facility of selecting any *m* from *n* records. A company wishes to select one of its employees for a particular post. The management wish to select the successful candidate on the basis of a number of factors, not necessarily related, which are qualifications, age, present grade and length of experience with the company. They eventually decide that candidates must satisfy any two of the following four conditions:

1 They must have a first class degree

2 They must be over 25

3 They must be at Grade D or above (E, F etc)

4 They must have more than 5 years service with the company.

Thus by one enquiry they can select such diverse types of person as

1 A fairly inexperienced but highly qualified and ranked graduate

2 A senior man, not academically qualified, but more experienced

3 An experienced man, not academically qualified, who has recently joined the company

and so on.

The file might contain a field holding the class of a person's degree as a number 1, 2, 3, with zero indicating no degree, P indicating a pass degree. Two other numeric fields could hold date of birth and date of joining the company with a fourth (character) field holding the person's grade.

These terms might appear in a simplified Dictionary as:

Field name	*Field type*	*Field length*
DBIRTH	W	1W
DJOIN	W	1W
GRADE	C	1C
CLASS	C	1C

The enquiry could then be phrased:

```
#ENQUIRY
?ENQ1          [SUITABLE CANDIDATES
(ANY 2 FROM  DBIRTH LEQ 19/08/44 OR DJOIN LEQ 19/08/64 OR
CLASS EQL 1 OR GRADE GEQ D )
```

EXAMPLE 5

This example illustrates the use of the Lead enquiry type within a batch of enquiries. The example shows how the Lead enquiry can be used to reduce interrogation time and also, how several reports can be produced from one hit file according to the selection of satisfied enquiries.

A personnel manager wishes to know details of all staff within the following categories:

1 An experienced programmer earning over £2000 who speaks French.

2 A programmer aged 20-25 years based in Reading or London.

3 Staff aged 20-25 years who have worked with the company since January 1st 1968 and who are not in the pension scheme.

Dictionary details of the personnel file are as follows:

Field name	*Field type*	*Field length*	*Description*
NAME	V	–	
ADDRESS	V	1	
AREA	C	4C	First 4 characters of town or city.
JOINDATE	W	1W	Date first employed
PENSCH	B	1B	0 = Not a member of pension scheme 1 = Member of pension scheme.
LANG	C	16C	Eight 2 character fields of first 2 letters of languages spoken.
TITLE	D	1W	4 character code
SALARY	D	1W	Decimal sterling length.

Enquiry parameters

```
#ENQUIRY
?ENQ1                    [ PROGRAMMERS
(TITLE = PROG )
?ENQ2                    [ FRENCH SPEAKING STAFF
(STEP 7 BY 2 LANG = FR )
?ENQ3                    [ STAFF EARNING >£2000
(SALARY > £2000 )
?ENQL LEAD
(AGE BET 20, 25 )
?ENQ4                    [ STAFF IN READING
(AREA = READ )
?ENQ5                    [ STAFF IN LONDON
(AREA = LOND )
?ENQ6                    [ WITH COMPANY JAN 68
(JOINDATE LEQ 01/01/68 )
?ENQ7                    [ NOT IN PENSION SCHEME
(PENSCH NEQ1 )
```

The position of the Lead enquiry ensures that only the records concerning staff in the required age group of 20-25 years are interrogated further.

The three reports required by the personnel manager can be printed from the hit file by selecting records according to the enquiries satisfied.

Output parameters

#OPTION LP 120 PRINT ALL, ENQ1, ENQ2, ENQ3

#FORM

NAME, ADDRESS, DEPT, AGE, SALARY(L)

#OPTION LP 120 PRINT ALL, ENQ1, ANY, ENQ4, ENQ5

#FORM

NAME, DEPT, AGE, SALARY(£)

#OPTION LP 120 PRINT ALL, ENQ6, ENQ7

#FORM

NAME, ADDRESS

#STOP

EXAMPLE 6

This example shows output parameters in use and illustrates a Print option report. A company requires to know the name, address, employee number, department and salary of all its employees. All the relevant information is held on the personnel file and details of part of the file dictionary are given as follows:

Field name	*Field type*	*Field length*	*Description*
NAME	C	15C	Employee's name
ADDRESS	V	–	Employee's address
NUMBER	W	2W	Employee's number
DEPARTMENT	C	4C	Department initials
SALARY	W	1W	Salary in sterling pence

The output parameters required to produce the report are as follows:

```
#OPTION, LP, 96, PRINT
#HEAD
(--- 30 spaces ---) PERSONNEL DETAILS
#FORM
NAME, ADDRESS, NUMBER, DEPARTMENT, SALARY(L)
#STOP
```

The output report produced by these parameters would be as follows:

06/08/69 PAGE NO 1

PERSONNEL DETAILS

NAME	ADDRESS	NUMBER	DEPARTMENT	SALARY
DICKENS C	61 ASH RD TILEHURST	10034007	ISD	£2010.15.0
WELLS H G	18 LINDEN AVE MAIDENHEAD	9447301	ISD	£1800. 0.0
CHESTERTON G K	36 TROTSWORTH AVE VIRGINIA WATER	8884052	MSD	£1750.10.0
THACKERAY W M	18 BUXTON AVE CAVERSHAM	11044670	MSD	£3035. 0.0
TROLLOPE A	147 BEECHAM RD READING	9960533	MSD	£1502. 0.0
BENNETT A	132 BURNHAM LANE SLOUGH	8941062	OSD	£2040. 0.0
HARDY T	29 CHURCH RD COOKHAM DEAN	12420036	OSD	£ 945. 0.0
MEREDITH G	171 WOODSIDE MILL CHALFONT ST PETER	9637712	OSD	£4400. 0.0
SHAW G B	16 ALDERBURY RD LANGLEY	10426750	OSD	£1100.10.0
BUTLER S	15 BETCHWORTH AVE EARLEY	10107440	PDL	£1010. 0.0
AUSTEN J	138 DEE RD TILEHURST	9733014	STD	£1717. 0.0
SCOTT W	1 FARNHAM DRIVE CAVERSHAM	11475722	STD	£ 906. 0.0

EXAMPLE 7

This example shows output parameters in use and illustrates a List option report from single line records. A large company which is divided into departments within branches within regions requires a detailed report on its expenditure on monthly salaries.

The information required is as follows:

1 The name and job title of each member of staff together with (a) salary (b) overtime payment and (c) expenses

2 The sum of (a) and (b) and the overall sum of (a) (b) and (c).

3 The totals of each of these items by department, branch and region.

File dictionary details are given as follows:

File name	*Field type*	*Field length*	*Description*
REGION	C	2C	
BRANCH	C	12C	
DEPT	C	12C	
NAME	C	15C	
JOB-TITLE	C	2C	2 character code
SALARY	W	1W	
O/TIME	W	1W	
EXPENS	W	1W	

The file is sorted on department within branch within region.

The two character code for JOB-TITLE is to be decoded using the decode file JOBTITLEFILE.

The output parameters required to produce the report are as follows:

#OPTION, LP, 160, LIST

#HEAD

REGION/BRANCH/DEPT BREAKDOWN OF EXPENDITURE ON SALARIES,

OVERTIME AND EXPENSES

#SPACE 4

#PAGE

(--- 17 spaces ---) BRANCH DETAILS (--- 46 spaces ---)

SALARY (--- 7 spaces ---) O/TIME (--- 11 spaces ---) SUM (--- 7 spaces ---)

EXPENS (--- 9 spaces ---) TOTAL

SPACE 2

#CONTROL 3, DEPT, [DEPARTMENTAL TOTAL

2, BRANCH, PAGE, [BRANCH TOTAL

1, REGION, [REGIONAL TOTAL

#FORM

15A 2B 20A 2B 2A 2B 12A 2B 12A 2B 6L 2S 2D

2B 5L 2S 2D 2B 6L 2S 2D 2B 5L 2S 2D 2B

6L 2S 2D 2B 2X 2B 3X 2B 3X 2B 4X

, NAME
, JOB-TITLE, DECODE, JOBTITLEFILE
, REGION
, BRANCH
, DEPT
, SALARY ,,, £, 3, 2, 1, GRAND
, O/TIME ,,, £, 3, 2, 1, GRAND
1SUM, SALARY + O/TIME ,,, £, 3, 2, 1, GRAND
, EXPENS ,,, £, 3, 2, 1, GRAND
, EXPENS + 1SUM ,,, £, 3, 2, 1, GRAND
, COUNT , 3
, COUNT , 2
, COUNT , 1
, COUNT , GRAND
#STOP

The output report produces by these parameters would be as follows:

EXAMPLE 7 - Illustration of List Option Report showing totalling and arithmetic capabilities

01/07/68 PAGE NO

REGION/BRANCH/DEPT BREAKDOWN OF EXPENDITURE ON SALARIES, OVERTIME AND EXPENSES

	BRANCH DETAILS				SALARY	O/TIME	SUM	EXPENS	TOTAL	
COLLINS W	CLERICAL ASSISTANT	SW	BATH	ACCOUNTS	£70 5 0	£1 10 3	£71 15 3	£2 0 11	£73 16 2	
FIELDING H	ACCOUNTANT	SW	BATH	ACCOUNTS	£102 10 0	£4 15 6	£107 5 6	£2 0 3	£109 5 9	
PEACOCK T L	SENIOR ACCOUNTANT	SW	BATH	ACCOUNTS	£130 0 0	£0 0 0	£130 0 0	£3 15 7	£133 15 7	
RICHARDSON S	ACCOUNTANT	SW	BATH	ACCOUNTS	£91 0 0	£0 0 0	£91 0 0	£1 10 0	£92 10 0	
SMOLLETT T	ACCOUNTANT	SW	BATH	ACCOUNTS	£91 0 0	£3 10 0	£94 10 0	£0 0 0	£94 10 0	
DEPARTMENTAL TOTAL										
				ACCOUNTS	£484 15 0	£9 15 9	£494 10 9	£9 6 9	£503 17 6	5
JOHNSON S	SALESMAN	SW	BATH	SALES	£85 10 0	£5 10 0	£91 0 0	£5 17 1	£96 17 1	
etc	etc	etc	etc	etc	etc	etc	etc	etc	etc	
SHELLEY P B	SALES MANAGER	SW	BATH	SALES	£150 0 0	£0 0 0	£150 0 0	£15 0 0	£165 4 0	
DEPARTMENTAL TOTAL										
				SALES	£683 5 0	£30 11 7	£713 16 7	£34 15 0	£748 11 7	6
COLERIDGE T S	TECHNICAL ASSISTANT	SW	BATH	PRODUCTION	£72 10 0	£6 10 4	£79 0 4	£1 0 6	£80 0 10	
etc	etc	etc	etc	etc	etc	etc	etc	etc	etc	
WORDSWORTH W	ENGINEER	SW	BATH	TECH SERVICE	£136 10 0	£10 15 6	£147 5 6	£2 10 0	£149 15 6	
DEPARTMENTAL TOTAL										
				TECH SERVICE	£487 10 0	£27 16 2	£515 6 2	£8 10 2	£523 16 4	5
BRANCH TOTAL										
			BATH		£2185 0 0	£95 18 2	£2280 18 2	£70 14 11	£2351 13 1	21

A Series of pages providing 'branch reports' for the S W Region will follow

EXAMPLE 7 - Illustration of List Option Report showing totalling and arithmetic capabilities

01/07/68	BRANCH DETAILS				SALARY	O/TIME	SUM	EXPENS	TOTAL		
ORWELL G	CLERICAL ASSISTANT	SW	BRIDGWATER	ACCOUNTS	£83 10 0	£2 11 4	£86 1 4	£6 15 5	£92 16 9		
etc	etc	etc	etc	etc	etc	etc	etc	etc	etc		
LAWRENCE D H	SENIOR TECHNICIAN	SW	BRIDGWATER	TECH SERVICE	£117 0 0	£0 0 0	£117 0 0	£15 15 8	£132 15 8		
DEPARTMENTAL TOTAL				TECH SERVICE	£1217 0 0	£48 11 0	£1266 1 0	£41 10 2	£1307 11 2	7	
BRANCH TOTAL			BRIDGWATER		£6200 0 0	£227 16 2	£6427 16 2	£203 7 4	£6631 3 6		35
REGIONAL TOTAL		SW			£25424 0 0	£1580 10 6	£27004 10 6	£2010 8 8	£29014 19 2		206

This is the last 'branch report' for the S W Region. Reports follow for branches throughout all regions

EXAMPLE 7 - Illustration of List Option Report showing totalling and arithmetic capabilities

01/07/68	BRANCH DETAILS				SALARY	O/TIME	SUM	EXPENS	TOTAL		
KEATS J	ACCOUNTANT	SE	MAIDSTONE	ACCOUNTS	£83 10 0	£4 15 0	£87 15 0	£2 13 0	£90 8 0		
etc	etc	etc	etc	etc	etc	etc	etc	etc	etc		
LAMB C	TECHNICIAN	SE	MAIDSTONE	TECH SERVICE	£81 10 0	£4 12 5	£86 2 5	£2 5 0	£88 7 5		
DEPARTMENTAL TOTAL				TECH SERVICE	£311 10 0	£23 16 0	£335 16 0	£27 11 1	£362 17 1	3	
BRANCH TOTAL			MAIDSTONE		£1508 15 0	£91 19 2	£1600 14 2	£6 17 7	£1662 11 9		16
REGIONAL TOTAL		SE			£8301 10 0	£464 15 7	£8765 15 7	£328 19 11	£9094 15 6		92
GRAND TOTALS					£71166 0 0	£2888 17 0	£74054 17 0	£3001 10 6	£77056 7 6		716

This is the final 'branch report' for the final region

EXAMPLE 8

This example shows output parameters in use and illustrates a List option report from multi line records. The example shows how the output facilities available with the List option can be used to output reports to pre-printed stationery.

The dictionary details of an invoice file are given as follows:

Field name	*Field type*	*Field length*	*Description*
NAME	V	–	
REF	C	8C	
ORDNO	C	6C	Order number
ACCNO	C	6C	Account number
DATE	W	1W	Binary date
ADDRESS	V	–	
QTY	W	1W	
CATNO	C	6C	
DESCR	C	30C	
PRICE	D	1	Decimal sterling pence
DESP	C	2C	Two character despatch code

The output parameters required to give the illustrated report are as follows:

```
#OPTION, LP, 120, LIST
#FORM
4B 35A 11B 8A 11B 6A 11B 6A 11B 8E
, NAME
, REF
, ORDNO
, ACCNO
, DATE
#SPACE 2
#FORM
4B  35A
, ADDRESS
#SPACE 4
#FORM
4B 6X 15B 6A 15B 30A 15B 5L 2S 2D
, QTY
, CATNO
, DESCR
, PRICE ,,,£
#SPACE 2
#FORM
```

```
90B 6L 2S 2D
, QTY * PRICE ,,,£
#SPACE 3
#FORM
6B 15A
, DESP, DECODE, DESPCENSFILE
#SPACE PAGE
#STOP
```

The output report produced by these parameters would be as follows:

EXAMPLE 8 - Illustration of LIST Option Report showing Flexibility of Output Format

CUSTOMER	REF	ORDER NO	ACCOUNT NO	DATE
J T WILLIAMS & CO LTD	04172/KG	227171	100313	01/03/68

ADDRESS
15,WARKWORTH AVENUE,HULL,YORKSHIRE

QTY	CAT NO	DESCRIPTION	LIST PRICE
27	202755	RING BINDER	£0.75

CHARGE
£20.25

DESPATCH CENTRE
WREXHAM

CUSTOMER'S SIGNATURE

CUSTOMER	REF	ORDER NO	ACCOUNT NO	DATE
G B DOWSETT (ELECTRONIC EQUIPMENT) LTD	00177/HH	304460	87115	01/03/68

ADDRESS
18,PHILADELPHIA ST,ST PAULS,BRISTOL 7

QTY	CAT NO	DESCRIPTION	LIST PRICE
15	990516	FILING CABINET	£10.38

CHARGE
£155.70

DESPATCH CENTRE
LIVERPOOL

CUSTOMER'S SIGNATURE

CUSTOMER	REF	ORDER NO	ACCOUNT NO	DATE
MARKHAM INDUSTRIAL CHEMICALS (RESEARCH DIVISION) LTD	21954/AL	284563	114078	02/03/68

ADDRESS
206,LONDON RD,BASILDON,ESSEX

QTY	CAT NO	DESCRIPTION	LIST PRICE
560	337642	FOOLSCAP PAD	£0.10

CHARGE
£56.00

DESPATCH CENTRE
LIVERPOOL

CUSTOMER'S SIGNATURE

EXAMPLE 9

This example illustrates the Total option, using a bulk delivery firm which keeps a file of records on each of its drivers and requires a weekly delivery report.

Dictionary information

Field name	*Field type*	*Field length*	*Description*
REGION	C	2C	Regional code
AREA	C	15C	Area code
DEPOT	C	15C	Depot
MILES	W	1W	Mileage covered
HRS	W	1W	Hours worked
TONN	W	1W	Tonnage delivered
PAY	W	1W	Gross pay in sterling pence.

Output parameters

```
#OPTION, LP, 96, TOTAL
#HEAD (---- 30 spaces ----) WEEKLY DELIVERY ANALYSIS
#SPACE      5
#PAGE
REGION    AREA    DEPOT      MILEAGE      HOURS      TONNAGE
GROSS-PAY      DRIVERS.
#SPACE      2
#CONTROL    3    DEPOT
            2    AREA
            1    REGION
#FORM
4B 5A 2B 15A 2B 15A 2B 7X 3B 5X 3B 5X 3B 7L 2S 2D 1B 3X 1B 3X
, REGION
, AREA
, DEPOT
, MILES , , , , 3
, HRS , , , , 3, 2
, TONN , , , , 3, 2, 1
, PAY , , , £, 2, 1
, COUNT , 3
, COUNT , 2
#SPACE 1
#STOP
```

The output report produced by these parameters would be as follows:

EXAMPLE 9 - Illustration of TOTAL Option Report

01/07/68 PAGE NO 1

WEEKLY DELIVERY ANALYSIS

REGION	AREA	DEPOT	MILEAGE	HOURS	TONNAGE	GROSS PAY	DRIVERS
SW	CORNWALL	PENZANCE	31210	563	615		11
SW	CORNWALL	TRURO	21877	487	602		8
SW	CORNWALL	FALMOUTH	22273	461	597		9
SW	CORNWALL			1511	1814	£861 16 8	28
SW	DEVON	PLYMOUTH	62749	994	1311		26
SW	DEVON	EXETER	41922	703	1044		19
SW	DEVON	BARNSTAPLE	9461	205	409		4
SW	DEVON	TORQUAY	15228	381	500		7
SW	DEVON			2283	3264	£1588 11 0	56
SW	SOM & WILTS	BRIDGWATER	20074	551	732		9
SW	SOM & WILTS	TAUNTON	19942	530	607		9
SW	SOM & WILTS	BATH	11165	394	390		6
SW	SOM & WILTS	CHIPPENHAM	36001	658	803		13
SW	SOM & WILTS	DEVIZES	9227	261	251		3
SW	SOM & WILTS			2394	2783	£1114 10 1	40
SW	BRISTOL & GLOS	BRISTOL	77453	1288	1421		21
SW	BRISTOL & GLOS	CHELTENHAM	20064	511	644		10
SW	BRISTOL & GLOS	TETBURY	15502	426	511		7
SW	BRISTOL & GLOS	GLOUCESTER	16165	431	561		9
SW	BRISTOL & GLOS			2656	3137	£1340 17 3	47
SW					10998	£3905 15 0	
S	HANTS & DORSET	WIMBORNE	18990	540	591		8
etc	etc	etc	etc	etc	etc	etc	etc

EXAMPLE 10

This example illustrates the parameters and output of the Table output option. The form taken is that of a sales analysis report. A company has sales offices in sixteen towns and cities throughout Great Britain, there being four sales areas in each of four regions, Northern England, Southern England, Scotland and Wales. The report gives the size of and income from the sales of a certain product for each department in both halves of the year 1968. The price of the product is £200. The row totalling facility is used to give the total sales and income for each department for the whole year.

The file has been sorted on both Y control fields, and sub totals are printed each time the Y1 control value changes. The column totals at this level give the total sales and income for the whole region for both halves of the year, as well as the totals for the region for the whole year. The old value of the control field is printed out with this line of sub totals.

The grand totals at the end are printed out automatically and give the total sales and income for the whole of Great Britain for both halves of the year, and finally the total sales and income for Great Britain for the whole year.

The parameters to specify this output are as follows:

```
#READ HITYFILENAME,  0, 0,  NO
#OPTION  LP 160,  TABLE, ALL, ENQ1, ENQ2, ENQ3, ANY, ENQ4, ENQ5
#HEAD
                                                                    SALES
ANALYSIS OF ITEM NO. 50 FOR THE YEAR 1968
#SPACE 2
#PAGE

REGION        AREA                                                  JAN-JUNE
                                                                JULY-DEC
                                                        TOTALS FOR YEAR
#XCONTROL
X2,  HALF-YEAR, 1, 2
X1,  YEAR,  1968
#YCONTROL
Y2, AREA,
Y1 REGION
#FORM
1B 5A 2B 4A 8B 8X 18B 10L 16B 8X 18B 10L 13B 8X 18B 10L
REGION
AREA
QTY , , , , , ROW
VALUE , , , £, , ROW
#STOP
```

EXAMPLE 10 - ILLUSTRATION OF TABLE OPTION REPORT

SALES ANALYSIS OF ITEM NO. 50 FOR THE YEAR 1968

REGION	AREA	JAN–JUNE		JULY–DEC		TOTALS FOR YEAR	
NORTH	LIVR	391	£78200	420	£84000	811	£162200
NORTH	NEWC	365	£73000	376	£75200	741	£148200
NORTH	MANC	376	£75200	398	£79600	774	£154800
NORTH	BIRM	363	£72600	380	£76000	743	£148600
NORTH		1495	£299000	1574	£314800	3069	£613800
SCOTL	EDIN	475	£95000	592	£118400	1067	£213400
SCOTL	GLAS	362	£72400	405	£81000	767	£153400
SCOTL	ABER	384	£76800	396	£79200	780	£156000
SCOTL	INVE	379	£75800	403	£80600	782	£156400
SCOTL		1600	£320000	1796	£359200	3396	£679200
SOUTH	LOND	360	£72000	372	£74400	732	£146400
SOUTH	SOUT	369	£73800	396	£79200	765	£153000
SOUTH	BRIS	321	£64200	320	£64000	641	£128200
SOUTH	OXFO	294	£58800	280	£56000	574	£114800
SOUTH		1344	£268800	1368	£273600	2712	£542400
WALES	CARD	310	£62000	330	£66000	640	£128000
WALES	GLAM	245	£49000	247	£49400	492	£98400
WALES	BANG	210	£42000	201	£40200	411	£82200
WALES	SWAN	262	£52400	251	£50200	513	£102600
WALES		1027	£205400	1029	£205800	2056	£411200
		GRAND TOTALS					
		5466	£1093200	5767	£1153400	11233	£2246600

EXAMPLE 11

This example illustrates the Decode File Update program. A decode file called LOCATIONCODE, file generation number 2, contains variable length records as follows:

Word 0 : Record length in binary

Word 1 : Code. 4 numeric characters (file key)

Word 2 et seq. : Alphabetic decode

Thus, part of the file might look like this:

Word 0	Word 1	Word 2 etc.
4	0011	ABERDEEN
5	0012	BIRMINGHAM
5	0015	LONDON (N)
5	0016	LONDON (S)
5	0017	LONDON (W)
5	0018	LONDON (E)
4	0019	BRISTOL
5	0020	MANCHESTER
4	0025	CARDIFF
4	0026	EXETER
5	0032	NEWCASTLE
5	0033	LIVERPOOL

It is required to amend the file in the following way:

1 Delete the BIRMINGHAM entry

2 Replace the four London districts with EPPING, TOOTING, EALING and WAPPING

3 Delete CARDIFF, EXETER and NEWCASTLE

4 Insert 0022, 0023 for READING and NOTTINGHAM

The parameter set might be as follows:

```
#UPDATE, LOCATIONCODE, 2
#DELETE
'0012'
'0025' – '0032'
#ADD
4,0015EPPING
4,0016TOOTING
4,0017EALING
4,0018WAPPING
4,0022READING
5,0023NOTTINGHAM
#STOP
****
```

Chapter 10 Modifying the package

This chapter describes in detail two techniques available for modifying the package. The first may be used at run time to alter the sizes of the program's various data areas. The second technique is concerned with introducing the user's own routines into the program. This may be done when consolidating the semi-compiled version of the package.

RUN-TIME ALTERATION OF SIZES OF PROGRAM'S DATA AREAS

Description of the technique

The program may be modified to suit a particular requirement by altering the sizes of certain of the program's main tables and data areas by means of the #STORAGE parameter. This enables the user, for example, to take advantage of a core store configuration of up to 32K by processing more than 96 enquiries at a time, or outputting more than 100 fields from each record. Alternatively the program may be tailored more closely to particular needs within 16K of core store by reducing the size of certain areas of which little or no use is to be made, and expanding others.

Each area is referred to by means of a code number. The Dictionary area, for example, has the code number 5, and its pre-set size is enough for 500 field definitions. If this area is to be increased in size to accomodate 600 field definitions, the system is provided with the code number and the new number of definitions thus:

5/600

following the #STORAGE directive.

Certain variable areas, for example, buffers, do not take the form of tables, and here the system must be supplied with the new size required in words. The main input buffer, for example, has the code number 1 and is preset to 1024 words. This could be reduced to 512 to save core store by specifying,

1/512

Areas capable of modification

Details of all the variable areas are supplied in the following table. Where an area has an entry in the 'Preset No. of Entries' column, that area should be varied by supplying the new number of entries required. In other cases the new size should be supplied in words.

Description	*Code number*	*Preset number of entries*	*Preset size in words*
Magnetic tape/Disc buffer * used for (a) Input from and Output to the parameter file (b) Input from the main file during Interrogation (c) Input from the main/hit file during the Output phase	1	–	1024
Magnetic tape/Disc buffer for output to the hit file	6	–	1024
Magnetic tape/Disc buffer for input from a merge file during an output run	11	–	512
Magnetic tape/disc buffer for output to an output file	10	–	1024
Magnetic tape/Disc buffer for input ** from a decode file	13	–	0
Dictionary area including both the main and merge file dictionaries if both are specified. Note that if the main/hit file dictionary is re-defined for output it overwrites the previous one. One entry is used for each field definition.	5	500	1500
Enquiry name index/hit counts area: one entry for each enquiry.	2	96	193
Enquiry constants table for holding the constants specified in the enquiries.	3	–	400
Enquiry constants dictionary: one entry for each enquiry constant whose value is to be supplied under #DEFINE	4	100	300
Area for holding the generated enquiry program in core; approximately 5 words per condition	7	–	1600
Table holding output field descriptions for the Print, List and Total options: one entry for each output field.	8	100	600
Table for holding constants for insertion in the print line(s) in the List option.	9	–	200
Y control field table one entry for each specified Y control field value in the Table option.	19	27	86
Totals area for the Table option: one entry for each specified value of the Y control field.	20	27	2268
Area in which 'Bucket + indexes' are held when decoding.	12	–	50

Table 7: Variable areas in the program

* This area should not be reduced below 512 words

** This buffer will be used to hold as many buckets as its size allows. It is recommended that its size is adjusted to allow the whole of the decode file to be held in core at once if possible. For large decode files on the other hand, a size permitting one decode file bucket to be held at a time represents the optimum. Its desired size must be set by the user at run time.

\+ The size of this area in characters is given by the sum of the sizes of all decoding buffers used (individual buffer sizes are given by the formula (1 + A) X B, where A = the maximum number of buckets which will fit into the buffer, B = the key size in characters). This should be rounded up to a whole number of words.

This area should clearly not be reduced below a size permitting two keys to be held, that is when only one bucket is held in core at a time. It may of course be reduced to zero if decoding is not being carried out.

Parameters and processing

The general format of the parameters is:

#STORAGE Code number/number of entries or new size (repeatable item)

The directive must appear as the first in a parameter list. It must not be used in parameters for re-entering the program. Rules for the punching of the parameter details with regard to sequence numbers, separating characters, etc., are identical to those for other parameters as described in Chapter 8, page 55 . The items re-defining sizes may appear in any order on any number of cards or paper tape blocks including the one containing the directive. The same code number may be specified more than once, in which case the last size definition will always apply. Use of:

ALL/O

has the effect of reducing all variable areas to zero length, and subsequent size definitions may be used to qualify this.

Errors in the #STORAGE parameter details are indicated in exactly the same way as for other parameters, see Chapter 12, page 111. Faulty size definitions are indicated and rejected, while correct ones are applied for the remainder of the validation phase. At the end of validation of all parameters, the program will halt (See Chapter 12) if faulty items have appeared under #STORAGE.

When all the #STORAGE details have been read and validated, the program attempts to make any necessary change in its size. If this is successful, a console message gives the new program size. If it is not, a console message gives the amount of extra core store required by the program before it can continue. For the format of these messages see Chapter 11, page 108. After successful size adjustment, the program outputs a summary of the sizes of all the variable areas on the line-printer and resumes validation of subsequent parameters (Note that the sizes summary will not be output unless the #STORAGE parameter is used).

Example

It is required to double-buffer input from a disc file which has 8-block (1024 - word) buckets. Therefore the size of the main input buffer must be increased to 2048 words. It is also required to process approximately 700 basic conditions instead of the 320 (approximately) which the system is pre-set to handle. The area which holds the enquiry program will therefore require enlarging to at least 3500 words, and an increase in the size of the enquiry constants table will probably be required as well. If, additionally, output listings containing up to 200 fields from each record are required, the parameter could consist of:

#STORAGE 1/2048, 7/3600, 3/800, 8/200

There is now no need for the user to do anything further. The system will automatically double-buffer now that it has room to do so, and the new interrogation and output limits are applied throughout the remainder of the validation phase and all subsequent processing.

Assessment of new size of program

The information given above is sufficient to enable desired changes to be made in the sizes of the program's data areas. However, an accurate idea of the size of the program which will result from proposed alterations may be required. This would be particularly important when altering the relative sizes of areas within an overall 16K

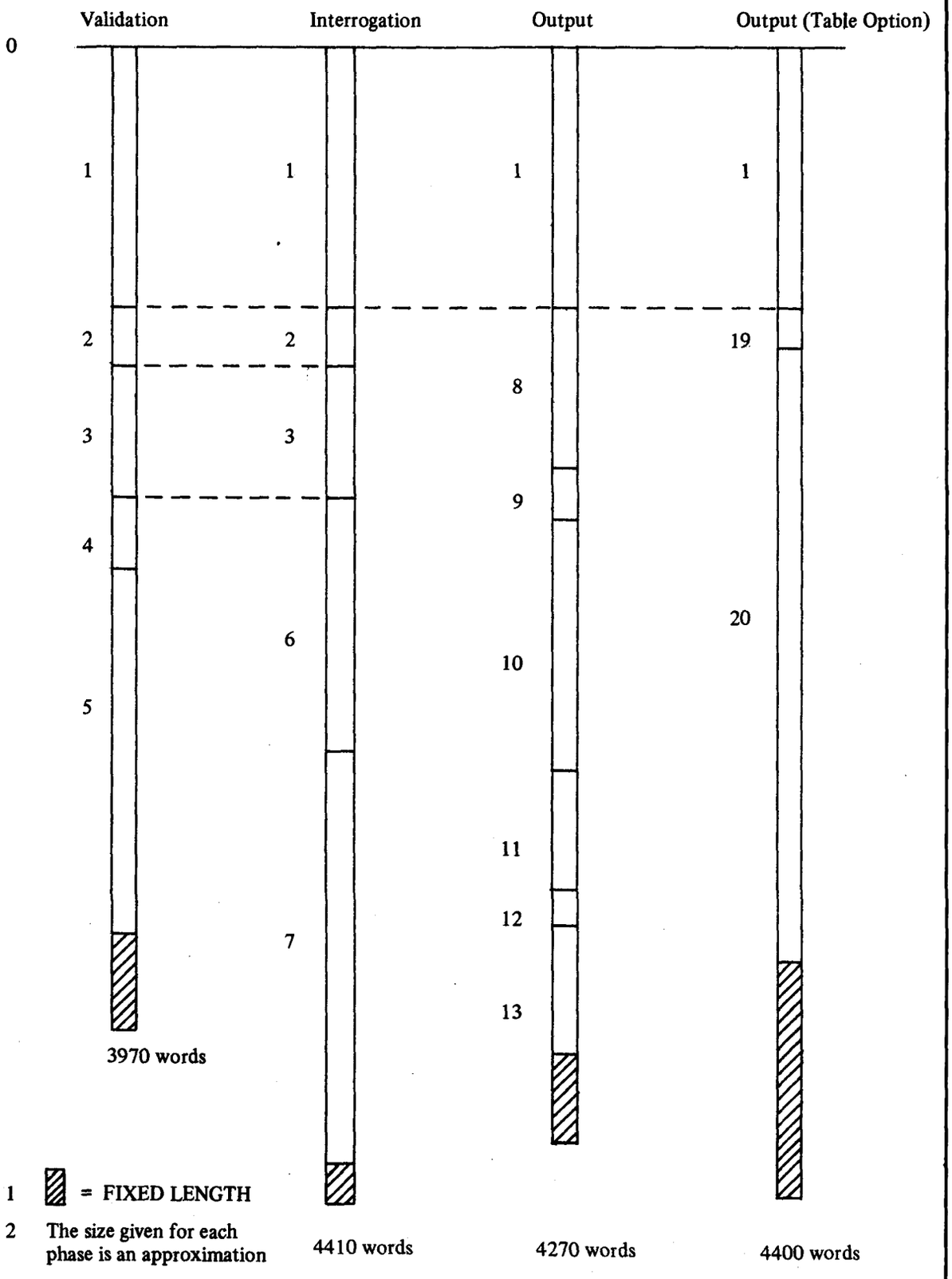


Fig 14 : Data Area Usage

limit. To help in this respect, Fig.13, page 96, shows how the program's data area is divided up between the various variable areas, and between four basic program overlays: Validation, Interrogation, Output and Output (Table option).

The code numbers are used to identify the individual areas and the sizes illustrated are the preset ones.

It can now be seen from this diagram that, for example, an alteration in the size of the main file input buffer (code number 1) will inevitably alter the overall size of the program by the same amount. Altering the size of the dictionary area (code number 5) alone, however, will not affect the overall size of the program unless its size is increased by more than about 840 words, and the validation phase becomes the largest. However, an increase in the size of the dictionary area, would almost certainly be associated with increases in the interrogation or output phases.

It is also clear that a reduction in the program's overall size will not necessarily occur if the area reduced is not used in all four phases. A reduction in the size of the dictionary area, for example, will not compensate for an increase in the size of the main input file buffer unless other areas in the other three phases are reduced at the same time. The ALL/O parameter item may be found particularly useful here if, for example, the validation and interrogation phases only are to be used. For example, the parameters:

#STORAGE, ALL/O, 1/2048, 2/24, 3/300, 5/200, 6/512, 7/1000

would enable validation and interrogation runs to be performed, double-buffering input from a file with 1024 - word blocks/buckets, batching up to 24 enquiries with a total of approximately 200 conditions, and with 200 fields defined in the input record still within the 16K limit. The following table gives the number of words per entry for these areas whose new sizes are supplied as numbers of entries:

Code number	*Number of words per entry*
2	2
4	3
5	3
8	6
19	3
20	84

From these figures it is possible to calculate that the above parameters would in fact result in an overall reduction in the program's size of approximately 500 words. It is also possible to calculate that in the previous example given above, where substantial increases to various areas in the program were made, the overall increase in program size would be approximately 3400 words.

To assist the user further, a summary of the overall sizes of the four basic overlays is output on the line printer after the summary of sizes of the variable areas. The largest of these will of course be the overall program size and will be the size given in the console typewriter message for the information of the operator.

Notes:

1 If #STORAGE is used in a parameter set, and the validated parameter file produced is subsequently used for re-entering the system, the program will again adjust its size if necessary to that which the original #STORAGE details made necessary. Identical console typewriter messages will inform the operator of any extra core store required, and of the final program size, but no summaries will appear on the line printer.

2 In the program X63F, issued in semi-compiled form to enable own coding to be used, all the variable areas described in Table 6 will have a pre-set length of zero words, except the main input file buffer (Code number 1), which will be preset at 512 words: and the Dictionary area which will be preset at 150 words. This makes use of #STORAGE essential in running this program, but permits own coding segments to use some of the space made available and thus provides a program which will still load into the core available in a 16K processor. Reduction of core store used in data tables *at run time* would not enable a user to expand the program area and still load into 12K.

USER OWN CODING

Introduction

If the FIND–2 Multiple Enquiry System as described does not completely cover every detail of a particular application, it can however be modified. There are various points within the sytem at which the user can have access to some of the system data. This data can then be modified before processing is continued. Modification of this kind is achieved by the incorporation of users own coding segments which are compiled into the system thus increasing its overall scope.

The power of this type of modification depends on the position of the user entry points, the data which is accessible and the point in the system at which the user can instruct processing to be continued.

The positions of own coding entries are as follows:

INTERROGATION PHASE

After record input

After interrogation

At the end-of-run condition

OUTPUT PHASE

In the List and Total options after record input when the selection condition has been satisfied.

In the Table option after record input when the enquiry selection condition has been satisfied.

In the List option after assembly of output in the output buffer.

In the Table option after the assembly of output in the output buffer.

At the end of a List option output run.

In each case the data which is accessable consists of the record buffers and the records may be examined and altered in situ or moved to new buffers. New input and output files may be introduced as long as care is taken not to duplicate the system file numbers as given in the operating instructions in Chapter 11, page 105.

After a record has been examined, control is passed back to the main system with the instruction either to continue normal processing or to ignore the current record and start processing the next. In addition in the case of the first user entry routine where the record is examined before interrogation, the user segment may instruct the system to treat the record as a hit and to skip over the interrogation phase, but this is only possible when there is only one enquiry specified.

Suggested uses

The following suggestions are made concerning the uses of own coding at each of the entry points.

USER ENTRY 1 After record input (interrogation phase)

1 To perform an initial check on records to decide whether they should be interrogated or not.

For example for a file containing two types of records such as master and detail records.

2 To reorganise the format of the input record.

(a) To convert a field format not accepted by FIND–2 to acceptable formats.

For example floating point to fixed point. Types of variable length fields to V or X types.

(b) To perform editing.

For example arithmetic between fields other than simple addition and subtraction.

USER ENTRY 2 After interrogation (interrogation phase)

1 To perform further interrogation

For example interrogation of fields in non-standard format.

2 To perform further editing.

For example reconversion of fields which were altered in the first own coding segment.
General format changes. Reduction of record size before output to hit file.

USER ENTRIES 3 and 4. After record input and selection (output phase)

1 To perform further selection

2 To perform further editing

For example conversion of fields which cannot be output by FIND–2 to character format. Expansion of non-standard variable length fields to type V or X.

USER ENTRIES 5 and 6. After assembly of an output record (LIST option) or a line of print, (output phase)

1 To perform editing on output records.

Note: In the List option, if the facility to subtotal fields at control changes is used, control is not passed to the user segment immediately before the printing of a totals line.

USER ENTRIES 7 and 8. At the end of a run (interrogation phase or List option output run)

1 To print end of run totals or arithmetic results accumulated by the user.

2 During the interrogation phase, to process a final composite record formed from a group of records on the input file.

3 In the List output option run, to change the own coding routines applied to successive output reports.

Program linkage of user routines

At each of the user entry points a call is given to an associated cue name FINDUE1, FINDUE2 ..... These cue names are found in the source segment OWNCODING and can be accessed by the user. Under each cue name is a dummy routine which returns control immediately to the main system. If a user wishes to incorporate his own coding segment he can alter the instructions under the appropriate cue name to call his own routine.

A calling routine in the main program takes the form

```
LDN   3   CRPA      X3  =  Start address of parameter area
CALL  0   FINDUE1   X0  =  Link address
```

The source steering segment contains

```
#CUE          FINDUE1
    LDN   3   0
    EXIT  0   0
#CUE          FINDUE2
    LDN   3   0
    EXIT  0   0          etc.
```

Information is passed to and from the user segments in the following ways:

A list of accessable record areas is passed to the user segments by parameter.

If records are moved by the user segments, the new record addresses are passed back to the system in accumulators.

The point at which processing is to be continued is determined by the value in the link accumulator on exit.

Details of the entry and exit conditions are given as follows:

FINDUE1 After record input

Parameter : Start address of record

Exit conditions :

```
EXIT  0   0   Continue with interrogation
EXIT  0   1   Ignore this record and read next
EXIT  0   2   Treat record as a hit. (One enquiry only)
X3    =   0   or New address of record
```

FINDUE2 After interrogation

Parameters : (1) Start address of record

(2) 0 = MISS for all enquiries

1 = HIT for at least one enquiry.

Exit conditions:

EXIT	0	0	Output record to hit file
EXIT	0	1	Ignore record and read next
X3	=	0	or New address of record

FINDUE 3/4 After record input and selection

Parameters: (1) Start address of main record (hit words attached).

(2) Start address of merge record (FINDUE3).

Exit conditions:

EXIT	0	0	Continue processing
EXIT	0	1	Ignore record and read next
X3	=	0	or New address of main record
X4	=	0	or New address of merge record (FINDUE3)

FINDUE5 After assembly of an output record or print line

Parameters: (1) Start address of main record (hit words attached)

(2) Start address of merge record

(3) Start address of output record or line

(4) Line number of multi line output record

Exit conditions:

EXIT	0	0	Output this record/line
EXIT	0	1	Omit this line/record from the output
X3	=	0	or New address of output record/line

FINDUE6 After assembly of a line of totals for printing

Parameters: (1) Start address of current input record

(2) A value 0 = normal totals line

1 = a Y1 sub-totals line

where Y1 and Y2 levels are used

2 = a GRAND TOTALS line

(3) Start address of output line in core

Exit conditions:

EXIT	0	0	Print this line
EXIT	0	1	Omit this line
X3	=	0	or New address of output line.

FINDUE7 At the end of an interrogation phase run, before the enquiry analysis is printed.

Parameter: (1) Start address of trailer label (magnetic tape) or end-of-file bucket (disc) of main file. Start address of sentinel where a run ends on a user tape sentinel used as a part-file key.

Exit conditions:

EXIT 0 0 Run terminates normally.

X3 = 0

If X3 is positive, its contents are taken as the start address of an area to be treated as a record, for a final pass through the enquiries phase before terminating.

FINDUE8 At the end of a List option output phase run, after the printing of end-of-run totals, if any.

Parameter: As for FINDUE7

Exit conditions:

EXIT 0 0 Run terminates normally. The contents of X3 are immaterial.

Suggested programming techniques (PLAN)

1 All user segment names should start with the letter Z so that they do not duplicate any of the names used by the FIND–2 system.

2 The master link address in XO must be preserved and restored to XO before exiting.

3 X3 should be used to pass parameter information to the user segments. This is simply achieved by leaving X3 unchanged when inserting the calls to the user routines, so that on entry to these routines X3 still contains the start address of a parameter area whose format for each entry is described above.

4 Work areas can be defined under #LOWER and made common to all user segments.

5 User segments can be overlayed if required. The FIND–2 system occupies overlay areas 1–5 so that further overlay units to hold user segments can be defined in area 6, and CALL instructions should be replaced by ENTER.

6 Switches 12 to 23 in word 30 are reserved for use by the user segments and could be used in the following way:

Several segments may be associated with one user entry point and the selection of those which are to be included could be made according to whether a switch is on or off. This method requires only one version of the program to be compiled whether or not user segments are to be used.

Suggested programming techniques (COBOL)

1 Points 1, 2, 5 and 6 above (PLAN programming) also apply to COBOL.

2 In order to reference in a COBOL subroutine the areas whose addresses are supplied, the calling sequence written into OWNCODING by the user must be the standard calling sequence generated by a COBOL compiler as a result of a COBOL CALL or ENTER instruction.

This interface requires a series of instructions to store the addresses of parameters in X3, following the call. A sample set of instructions is shown below:

```
#CUE        FINDUE2
    LDX  4  0(3)
    STO  4  PARAM
    LDX  4  1(3)
    STO  4  PARAM+1
    CALL 1  ZCOBROUT
    LDX  3  PARAM      address of RECORD
    LDX  3  PARAM+1    address of HIT/MISS SWITCH
```

where PARAM, PARAM+1 are defined under the #LOWER in OWNCODING. In COBOL terms, the RECORD and the HIT/MISS SWITCH are themselves the parameters.

3 The own coding for each entry must be in the form of a COBOL subroutine and must comply with all the normal rules for COBOL subroutines. The LINKAGE section must define the data areas (parameters) the addresses of which are being supplied. In the above example, the LINKAGE section of ZCOBROUT might contain, for example:

```
01  REC
 02 LENGTH     PIC 1(23)
 .  .          .
 .  .          .
 .  .          .
 77 HIT        PIC 9(4)
```

4 Areas of storage cannot be created in a subroutine for use outside it. For example if a COBOL programmer wishes to substitute a new record area for the record area given, he must define the new area in the OWNCODING segment, cause the area address to be passed to the COBOL routine as above, and supply its description in the LINKAGE section. The address of the area must then be passed back to FIND–2 in X3 as usual after exit from the COBOL routine.

5 Where a choice of exit points is provided, the choice must be handled in the COBOL routine by the setting of a switch word (not word 30), which is defined in both OWNCODING and the COBOL LINKAGE section as described above. The switch word is examined on exit from the COBOL routine to determine which exit point to apply:

```
EXIT  0  0  or  EXIT  0  1.
```

In order to make it possible in some cases to include own coding within 16K store, all variable table areas which can be altered by #STORAGE are set initially to zero. These areas can then be expanded as required so that the maximum space can be allocated to user segments.

Procedure for compiling and consolidating user own coding segments

One version of the FIND–2 Multiple Enquiry System, program #X63F, is issued in semi compiled form with its steering segment and the segment OWNCODING in source. This version is in the form of a COSY file, named X63FSOURCE, with the program in subfile FIND-2SOURCE, and incorporates the tape overlay package %TROL. As described in the operating instructions in Chapter 11, page 105 the COSY file can be set up on tape or if required on disc; it may be updated to include the disc overlay package %EROL. The overall steering line in the issued program is preset to LIST, OBJECT so that the steering segment OVERLAYM, and the segment OWNCODING will be automatically listed on compilation. The program name may be changed if required.

User own coding segments can be written in any language which will run on 1900 computers and which can use the program interface described in the previous section. The following section describes the procedure whereby these segments can be compiled and consolidated into the FIND–2 system and gives the name of the program which can be used for each individual operation. Details of these programs can be found in the ICL 1900 Series *Library Specifications* Manual.

USER SEGMENTS IN PLAN

1 Source program on magnetic tape

(a) The FIND source segments must be amended to incorporate the user segment interface and the tape COSY file must be updated to include the source user segments (Program #XPMR).

(b) The FIND source segments and the source user segments must be compiled and then consolidated with all other segments on the tape COSY file (Program #XPLV, #XPLW, #XPLY, #XPLN, or #XPLT with #XPCL or #XPCK).

2 Source program on E.D.S.

(a) The FIND source segments must be amended to incorporate the user segment interface and the disc COSY file must be updated to include the source, user segments (Program #XPMZ).

(b) The FIND source segments and the source user segments must be compiled (Program #XPLZ).

(c) The semi-compiled segments on the disc COSY file must all be consolidated (Program #XPCL).

USER SEGMENTS IN A HIGH LEVEL LANGUAGE

1 Source program on magnetic tape

(a) The source own coding segments must be compiled using the appropriate compiler to produce semi-compiled output on cards, paper tape or magnetic tape.

(b) For card or paper tape semi-compiled output (for example FORTRAN and ALGOL segments) the semi-compiled segments must then be added to the tape COSY file (Program #XPMR).

For magnetic tape semi-compiled output (for example COBOL segments) the semi-compiled segments must be converted to sub file format (Program #XPMV) and then either added to the tape COSY file (Program #XPMR) or inserted into the subroutine group (Program #XPMV).

(c) The FIND source segment should be amended to incorporate the user segment interface (Program #XPMR), together with an appropriate #LIBRARY directive if any COBOL subroutines are to be available in a subroutine group on the compiler tape. Alternatively, the required COBOL subroutines may be added to the FIND source tape (Program #XPMV) before compilation (Program #XPLV).

2 Source program on E.D.S.

(a) The source own coding segments must be compiled using the appropriate compiler.

(b) The FIND source segment must be amended to incorporate the user segment interface and any magnetic tape, card or paper tape semi-compiled segments must be added to the disc COSY file (Program #XPMZ).

(c) The FIND source segment must be compiled. (Program #XPLZ).

(d) The semi-compiled segments on the disc COSY file must all be consolidated, any COBOL subroutines required being consolidated by means of an appropriate *LIB parameter (Program #XPCL or Program #XPCK).

ICL

Find-2 Multiple Enquiry System

Part 3 Operating The System.

4187(12.69)

Chapter 11 Operating environment and instructions

ENVIRONMENT

The package operates on a wide range of 1900 configurations, decisions being specified at run time as to which form of backing store is used to hold a file. The purpose of the initial part of this chapter is to outline which peripherals can be used to hold the various inputs and outputs of the systems and to show the varying requirements when running the system on disc only, magnetic tape only, or mixed disc/tape configuration.

Uses of backing store

1 The program file may be held on magnetic tape, E.D.S. or twin disc. If dumping is required, it will be made onto the same medium as that holding the program. If the program is on magnetic tape a second unit must be available for a work tape, but if the program is on disc it is recommended that the dump file is allocated on the cartridge holding the program.

2 All other files except the decode file can be held on any of the following devices:

MT, E.D.S., twin disc or F.D.S. The decode file must be held on disc (E.D.S. or twin disc).

Choice of media

E.D.S./TWIN DISC ONLY

It is possible to run the system on only 1 E.D.S. but this imposes the following restrictions.

1 The parameter file, input file and hit/output files have to be held on one cartridge and the main file is thus limited to less than one cartridge.

2 Timings are adversely affected when the input and output files share one cartridge, owing to the head movements involved.

3 Merging and decoding facilities together with dump and restart can be used but further restrictions on file size are imposed and timings are adversely affected.

To avoid restrictions on file size, the input and hit/output files should be on separate cartridges and the program file, parameter file, decode file and dump file can all share one cartridge without loss of efficiency. Hence three disc drives would be required. If the merge facility is required when outputting to disc, a further drive would be required, making a total of four, to avoid any restrictions on file size.

MAGNETIC TAPE ONLY

The system may require between three and six magnetic tape decks. Only three decks are required when output is made to the line printer and no dumping or merging is required. With decks, no limitations are imposed on file size and it is assumed that program dumps are to be made to either the hit or output file. If this is unacceptable a fifth deck can be employed to hold the program dumps. When dumping to a separate tape a sixth deck is necessary if it is required to use the merge facility when producing an output file.

MIXED DISC/MAGNETIC TAPE CONFIGURATION

It is recommended that the program file, parameter file, decode file and dump file share one disc cartridge as no loss of efficiency is involved and the program naturally operates more efficiently when overlaid from disc. The remaining three files, that is, the main input file, secondary input file and hit or output file may then be held on E.D.S., twin disc, F.D.S. or magnetic tape.

OPERATING INSTRUCTIONS

Two sets of operating instructions are provided for:

1 The three versions of the Multiple Enquiry System programs

(a) Program #X63C (Object program overlaid from E.D.S.)

(b) Program #X63D (Object program overlaid from MT)

(c) Program #X63F (Semicompiled program overlaid from either E.D.S. or MT).

#X63F, the semicompiled version of the Multiple Enquiry System with a source steering segment is in the form of a tape COSY file. It can therefore be updated to include own coding segments and to replace the tape overlay routine %TROL with the disc overlay routine %EROL if required. The program name may also be changed. The COSY file can then be compiled to produce the equivalent of #X63C on disc or #X63D on tape.

Details of compiling the semicompiled version are given in Chapter 10.

2 The decode file updating program.

Program #X63E (Object program non-overlaid)

1. PROGRAMS #X63C, #X63D, #X63F

Executive priority

The program as issued has an executive priority of 80.

Use of peripherals

Device	*Type of data*	*Allotment and release*
LP0	Parameter listings and results	Used throughout run
TR0/CR0	Parameter input	Allotted at beginning of run and released at end of parameter input.
MT1/ED1/FD1	Input file (Main/Hit)	Used throughout run
MT2/ED2/FD2	Parameter file	Used throughout run
MT3/ED3/FD3	Output/Hit file	Optional use throughout run
MT4/ED4/FD4	Merge file	Optional use in output phase
MT5/ED5/FD5	Dump file	Optional use throughout run
ED6 to ED11	Decode files	Optional use in output phase

Running under GEORGE

GEORGE 1

There are no special restrictions on running the Multiple Enquiry System under GEORGE 1.

GEORGE 2

When off-lining line-printer output under GEORGE 2 page overflow replies are never generated. This means that page headings and numbers (after the first page) will not occur, since FIND-2 monitors the page overflow condition.

GEORGE 3

1 Page overflow replies are not given, as in GEORGE 2

2 If the #STORAGE facility is used to increase the program's running size, the maximum amount of core store that can be given to the program is governed by an installation parameter. This value may only be increased by operator action, and only then if the facility of operator amendment in this way is in fact followed.

3 Dump and restart cannot be used with GEORGE 3 Mk. 4.

4 The maximum permitted size of a direct access file under GEORGE 3 Mk. 4, is 245K words.

5 When creating a filestore file for the parameter file, the qualifier SERIAL must be used.

For example

CE FINDWORKFILE (*ED,BUCK4,KWORDS10,SERIAL)

If SERIAL is omitted the default value SEQUENTIAL is taken which may cause the error halt

PARAMETER FILE ERROR

HALTED ZZ

6 If the #WRITE parameter for the parameter file is omitted the program assumes the standard file name FINDWORKFILE. The program attempts to open a file of this name, with the highest generation number, in input mode. Having determined the exact generation number it closes the file and reopens it in output mode. The original ASSIGN command for the parameter file is cancelled when the file is closed; the second attempt to open the file will cause the program to halt ED FINDWORKFILE. The user must either prevent this halt by using a #WRITE parameter, or give a second ASSIGN command once the halt has occurred.

7 It should be noted that the line printer is released after validation. At the end of the interrogation phase, the program halts LP, indicating that a line printer is required for the enquiry analysis. The printer file must be reassigned, possibly with the APPEND qualifier.

8 The most convenient way to reassign the hit file as the input file for the output phase is to monitor the display message INTERROGATION COMPLETE. At this stage, the hit file which has previously been closed can be assigned as file 1.

A comprehensive example of a GEORGE 3 job description for a FIND-2 Multiple Enquiry System appears at the end of this chapter.

Running under Executive alone

The following steps cover the three different type of run which can be made with the system:

Interrogation followed by output

Interrogation only

Output only

These steps also apply when re-entering the system. They are very simple and only differ from each other in the files which have to be loaded. These files can be loaded in advance or as requested by the program. For example, files for the output phase may be loaded after the message 'Interrogation Completed' following an interrogation. The appropriate name should be substituted for #PROG.

Narrative — *Console message*

1 If twin disc/E.D.S. files are used, load the cartridge or cartridges containing:

(a) The program #X63C

(b) The parameter file

(c) The main input file and optionally

(d) The dump file

(e) The hit file - for interrogation run

(f) The secondary input file

(g) The decode file } for output run

(h) The output file

2 If magnetic tape files are used load tapes for the following files:

(a) The program #X63D

(b) The main input file

(c) Scratch tape 1 - for parameter file. (or on re-entry, previously created parameter file). and optionally } interrogation runs

Narrative			*Console message*
	(d) Scratch tape 2 - for hit file	} interrogation runs	
	(e) Scratch tape 3 - for dump file or optionally		
	(f) Secondary input file (merging)	} output runs	
	(g) Scratch tape 2 - for output file		
	(h) Scratch tape 3 - for dump file		
3	If both twin disc/E.D.S. and magnetic tape files are used load the appropriate cartridge(s) and magnetic tape(s) for the above files.		
4	Load the program into store from either twin disc/E.D.S. or magnetic tape		FI #X63C#DISC FI #X63D#TAPE
5	Load the parameters in the paper tape or card reader.		
6	For double line spacing on reports		ON #PROG 0
7	For suppression of date and page numbers on reports		ON #PROG 1
8	To cause the program to halt before printing for paper loading and alignment		ON #PROG 2
9	To halt at end of validation (or end of translation and preservation of enquiry program if switch 7 is on)		ON #PROG 6
10	To preserve enquiry program		ON #PROG 7
11	Set on switches (12-23) as required. For example Own coding.		
12	If parameters are on paper tape If parameters are on cards		GO #PROG 20 GO #PROG 21
13	To end an interrogation early		GO #PROG 28
14	To contract the hit file (disc only) after an output run has finished		GO #PROG 29
15	To scratch the hit file (magnetic tape only) after an output run has finished		GO #PROG 29
16	To temporarily suspend the program during interrogation		ON #PROG 3

The following messages indicate the progress of the program:

Message	*Reason*	*Action*
DISPLAY:-INTERROGATION COMPLETE	The end of interrogation is monitored in combined interrogation and output runs.	Load files for output runs.
DISPLAY:-INTERROGATION INTERRUPTED	The interrogation has been temporarily suspended to obtain an intermediate enquiry analysis on the line-printer.	GO to resume interrogation
DISPLAY:-LOAD PAPER	The program has been halted prior to printing an output report to allow paper to be loaded.	Load paper and GO
DISPLAY:-CHECK PAPER LINE-UP	A dummy output report has been printed to allow paper alignment to be checked.	If the paper is satisfactorily aligned OFF switch 2 and GO to print the report. Otherwise, realign the paper and GO for more dummy records.
DISPLAY:-HH	An output report has finished and another is to follow.	
DISPLAY:-ZZ	An output report has been abandoned due to an error and the next report is being automatically started.	

Message	*Reason*	*Action*
HALTED: HH	1 An interrogation only run is complete. 2 A validation only run is complete. 3 The last output report is complete. 4 An interrogation run has been temporarily suspended.	

Exception conditions

Exception conditions arising from unavailability of peripherals, core or cartridge space:

Message	*Reasons*	*Action*
0#PROG HALTED:- TR	No tape reader has been allocated	Allocate PTR and GO
0#PROG HALTED:- CR	No card reader has been allocated	Allocate CR and GO
0#PROG HALTED:- LP	No line printer has been allocated	Allocate LP and GO
0#PROG HALTED:- PL	Paper low on line printer	Load paper and GO
0#PROG HALTED:- ED/FD (File Name)	The program has failed to open a file on the appropriate device because 1 The correct generation of a file is not on-line 2 A utility program has the system control area open. 3 Executive has too many files open.	Check that correct generation of file is on-line and GO If message is repeated, abandon run and repeat at a later date.
DISPLAY:#PROG NEEDS NNNNN WORDS MORE followed by: 0#PROG HALTED: ST	As a result of using STORAGE, the program needs NNNNN extra words of store	On a single programming machine – abandon run. Otherwise make more core available and GO or abandon run and repeat at a later date.
DISPLAY:PARAM FILE LESS THAN 4 BLOCK BUCKETS followed by 0#PROG HALTED:-P2	The disc parameter file used has 1 or 2 block buckets	Abandon run, reallocate file with 4 or 8 block buckets
DISPLAY:#PROG uses NNNNN WORDS OF STORE	As a result of using STORAGE, the program is now running in NNNNN words of store	
0#PROG HALTED: MO FILE N*	The program cannot extend File N, when writing to a disc output file	
DISPLAY: FILE N* EXTENDED	File N has been successfully extended	

Exception conditions

Exception conditions arising from parameter or data errors

Message	*Reason*	*Action*
0#PROG HALTED:-P1	Parameters errors	GO to continue run or correct errors, reload program and GO 20/21
0#PROG HALTED:-P2	Serious parameter errors	Correct errors reload program and GO 20/21
0#PROG HALTED:-LB FILE N*	Long block on file N	GO to read next block OR abandon run.
0#PROG HALTED XX FILE N*	Magnetic tape parity failure on file N	Abandon run

Message	*Reason*	*Action*
0#PROG HALTED:-IR FILE N*	Illegal record length on writing to file N (zero or greater than block or bucket size)	Abandon run, examine input data, output file
0#PROG HALTED:-Z1	Recoverable input file errors	GO to continue run
0#PROG HALTED:-ZZ	Irrecoverable input file error	Abandon run

Notes:

1 N *represents the program file number, details on page 106.

2 EXTENSION OF DISC FILES. When writing to an output file on disc, the program will extend the file as necessary. Extension will initially be on the same cartridge or the last cartridge of a multicartridge file. If the file cannot be extended the message

0 PROG HALTED: MO FILE N

will occur.

If it is possible to make space available on the cartridge (for example, by deleting a program using scratch files) then after doing so, typing

GO #PROG

will cause another attempt to extend the file.

Alternatively, typing

AL #PROG 7 *nnnnnn

(where nnnnnn is the serial number of an alternative cartridge or unit) followed by typing

GO #PROG

will cause the file to be extended onto the second cartridge, if space is available. If unsuccessful, the program will again halt as described above.

3 If there are no hits at all at the end of interrogation the program displays

DISPLAY: NO HITS

and halts

HALTED: HH

even if there are output reports to follow. Processing may be resumed if required by means of

GO #PROG

4 Full details of error messages are given in Chapter 12.

2. PROGRAM #X63E (DECODE FILE UPDATING PROGRAM)

Executive priority

The program as issued has an executive priority of 80.

Use of peripherals

Device	*Type of data*	*Allotment and release*
LPO	Parameter list (Amendment record)	Used throughout run
TRO/CRO	Parameter input	Used throughout run
ED1	Decode file	Used throughout run

Running under GEORGE

No special considerations.

Running under Executive alone

Narrative	*Console message*
1 Load the cartridge or cartridges containing:	
(a) Program #X63E	
(b) The decode file	
2 If the program is held on magnetic tape load the program tape containing #X63E.	
3 Load the program into store from either:	
Disc	FI #X63E DISC
Magnetic tape	FI #X63E TAPE
4 Load the parameters in the paper tape reader or card reader	
5 If parameters are on paper tape	GO #X63E 20
If parameters are on cards	GO #X63E 21
6 The run has terminated successfully	Halted HH

Exception conditions

Message	*Reason*	*Action*
0 #X63E HALTED:-TR	No tape reader has been allocated	Allocate PTR and GO
0 #X63E HALTED:-CR	No card reader has been allocated	Allocate CR and GO
0 #X63E HALTED:-LP	No line printer has been allocated	Allocate LP and GO
0 #X63E HALTED:-PL	Paper low on line printer	Load paper and GO
0 #X63E HALTED:-ED	The program has failed to open a file on the appropriate device because	
	1 A utility program has the system control area open.	If message is repeated, abandon run and repeat at a later date.
	2 Executive has too many files open.	
	3 Integrity code failure.	
0#X63E DISPLAY:- ED (File name) 0#X63E DISPLAY:- PURGE DATE NOT EXCEEDED 0#X63E HALTED:- ZZ	The data retention period of the file has not expired	Zeroise the retention period using #XJEX, or wait until expiry. Then reload program.
0#X63E DISPLAY:- ED (File name) 0#X63E HALTED:- FILE NOT IN SYSTEM	File not available to program	Allocate the appropriate cartridge to the program, then GO #X63E
0#X63E HALTED:- P1	Errors in #ADD or #DEL parameters; the flagged amendments only will have been ignored	Correct parameters and rerun program (for flagged amendments only)
0#X63E HALTED:- P2	Fatal parameter errors, i.e. not in #ADD or #DEL. No updating will have been performed although the parameters are listed in full	Correct parameters and rerun program
0#X63E HALTED:- O1	Run successful but first level overflow records have been written	If the file is to be used for the decoding facility then it must first be reorganised
0#X63E HALTED:- DA ERRORS X ABANDON	Certain file format errors will cause this message. See the *Direct Access* manual (TP4107), chapter 17, for further information.	

All error messages are explained in detail in Chapter 12.

EXAMPLE OF A GEORGE 3 JOB DESCRIPTION

The job description shown below will perform an interrogation, and will output any number of reports from the hit file to the line printer.

The following assumptions have been made:

1 The program #X63C has been loaded into PROGRAM X63C

2 FIND-2 parameters have been loaded into MESPARAMS

3 The input file is a disc-like file, FINDINPUT

4 The disc-like files FINDWORKFILE and FINDHITFILE have been created

```
JOB FIND, :USERNAME
LOAD PROGRAM X63C
ASSIGN *CRO, MESPARAMS
ASSIGN *ED1, FINDINPUT
ASSIGN *ED2, FINDWORKFILE(WRITE)
ASSIGN *ED3, FINDHITFILE (WRITE)
ASSIGN *LP0, MESOUTPUT
MONITOR ON, DISPLAY
ENTER 1
IF HALTED (P1), RESUME
IF HALTED (P2) OR HALTED (ZZ), GOTO 1LABEL
IF HALTED (Z1), RESUME
IF NOT HALTED (LP), GOTO 2LABEL
ASSIGN *LP0, MESOUTPUT (APPEND)
RESUME
IF MONITOR (DISPLAY) AND NOT DISPLAY (INTERROGATION COMPLETE), GO TO 2LABEL
MONITOR OFF, DISPLAY
ASSIGN *ED1, FINDHITFILE
3LABEL RESUME
IF HALTED (Z1), GOTO 3LABEL
IF HALTED (ZZ) OR HALTED (HH), GOTO 1LABEL
IF NOT HALTED (LP), GOTO 2LABEL
ASSIGN *LP0, MESOUTPUT (APPEND)
GOTO 3LABEL
2LABEL PRINT 0(2000), 8000(4000)
1LABEL DELETE
LISTFILE MESOUTPUT, *LP
ENDJOB ALL
```

It should be noted that to read parameters on-line from a paper tape job description file, the job description must be input in GRAPHIC MODE, since this is the mode used by the program. This prevents the use of JOB; INPUT must be used to load the job description. To run the job either use RUNJOB or issue the file name as a macro, with the required macro parameters.

This job description can easily be enhanced to include the use of merge, decode or output files or to cater for re-entry. The user may also find that a macro can be devised to suit his particular requirements. An example of a macro to run the program #X63A can be found in the ICL 1900 Series manual *FIND-2 Single Enquiry System*, (Edition 1, TP 4155).

Chapter 12 Error procedures

There are two basic types of error notification; console messages and error flags. The validation phase of the Multiple Enquiry System checks and prints all the parameters for a given run. Errors are flagged on this printout, and full details of these error flags are given in the second main section of this chapter.

According to the nature of the error found and where it occurs, the program may omit the faulty field or parameter, or make an assumption as to what was intended. The program flags assumptions on the parameter listing in the same way as errors. The operator is informed of these events via the console typewriter, and these and all other error messages are dealt with in the following section.

CONSOLE MESSAGES

Details are given in the following tables of the specific action to be taken in the event of errors arising either in the parameter or on the input file. A summary of exception conditions is given in Chapter 11.

Parameter errors

Console message	*Remarks*	*Action*
HALTED P1	Errors have been found in the parameters, but the program has made assumptions as to what was intended.	Examine parameter listing and continue with one of the following: 1 Accept program assumptions and GO 2 Abandon run, correct parameters, reload program and GO 20/21.
HALTED P2	Serious errors have been found in the parameters	Examine parameter listing, abandon run, correct parameters, reload program and GO 20/21.
ENQUIRY PROGRAM FULL HALTED: P2	The enquiry specified is too large for the program.	Abandon run, and rephrase the enquiry in question, reducing the number of relationships or operators.

Table 8: Console messages concerning parameter errors

Errors relating to magnetic tape files only

Console message	*Remarks*	*Action*
HALTED LB FILE N	The program has encountered a block on File N larger than the program buffer.	Either: 1 GO, causing next block to be read, 2 Abandon run, and examine file format.
HALTED XX FILE N	A magnetic tape parity failure on File N.	1 Abandon run, restart from beginning or last program dump.

Note 'N' represents the programs file number as specified on page 106.

Table 9: Console messages concerning magnetic tape files

Errors in input data

These errors occur while processing the user's file and have been divided into errors from which a recovery can be made (HALTED Z1 messages), and errors from which a recovery cannot be made (HALTED ZZ messages).

RECOVERABLE ERRORS IN INPUT DATA

Console message	*Remarks*	*Action*
VARIABLE FIELD MISSING (INTERROGATION) HALTED Z1	The end of the record has been encountered before reaching the required variable length field (type X) or the field length word has been found to contain zero in bits 9 to 23, during interrogation.	1 GO to miss the record and continue the program. Repeated occurrence of the message suggests faulty Dictionary definition of type X fields and in such a case: 2 Abandon run and examine Dictionary definitions.
VARIABLE FIELD MISSING (OUTPUT) HALTED Z1	The same faults as above have been detected but during output of a type X field.	1 GO at this point and whenever the condition recurs, the field in question and any subsequent fields in the record to be output will be filled with upward arrows. 2 Abandon run and examine Dictionary definitions.
FILE N BLOCK COUNT ERROR HALTED Z1	The data block count in the trailer label on magnetic tape (of file N) does not contain the total number of data blocks read and skipped over by FIND–2.	1 GO, the program will continue to end the run or open the next reel. 2 Abandon run and examine file.
DECODE KEY NOT FOUND HALTED Z1	The field to be decoded cannot be found on the decode file.	1 GO to continue: upward arrows will be inserted in the output buffer for this and any subsequent field whose key cannot be found. 2 Abandon run and examine files.

Table 10: Recoverable errors in input data

The following messages, terminated by HALTED ZZ, indicate errors which are irrecoverable during the run. If an irrecoverable error is encountered during the processing of an output report when further reports are to follow, the HALTED ZZ message is replaced by DISPLAY ZZ and the program proceeds automatically with the processing of the next report.

IRRECOVERABLE ERRORS IN INPUT DATA

Console message	*Remarks*	*Action*
*n*th TERMINATOR MISSING HALTED ZZ	The end of record has been encountered before reaching the *n*th variable length field (type V) during interrogation or output.	Abandon run and examine Dictionary definition of type V fields and/or file format.
FILE N BLOCK SIZE >INPUT BUFFER HALTED ZZ	The maximum data block size on magnetic tape (file N), as stored in the start of data sentinel, is greater than the input buffer.	Abandon run and examine file.

Console message	*Remarks*	*Action*
FILE N BUCKET SIZE >INPUT BUFFER HALTED ZZ	The input buffer has been reduced by #STORAGE below the value of the file bucket size.	Abandon run and reorganise #STORAGE
FILE N FILE INDEX ERROR HALTED ZZ	Refers to indexed sequential files only. An error in the file N indexes has led the program to a bucket which: 1 Is out of range or 2 Does not contain the required record.	Abandon run and examine indexes of file N.
FILE N FILE OVERFLOW ERROR HALTED ZZ	Refers to sequential files only. An error in the overflow system in file N has lead the program to a bucket that: 1 Is out of range, or 2 Does not contain the required record.	Abandon run and examine file N.
FILE N PART-FILE KEY NOT FOUND HALTED ZZ	With an indexed sequential disc file (N), the range of record keys specified on the #READ parameter cannot be found in the file indexes, or part-file keys have been specified for a file that is random, serial or non-indexed sequential. If the file is on magnetic tape, the EOF has been reached before encountering the first required user tape sentinel. Note: This message may also occur if there are no records on file N within the range specified.	Abandon run and examine parameter and/or file N.
FILE N UNACCEPTABLE SENTINEL HALTED ZZ	The magnetic tape file N has been found to have a sentinel which is not a start of data sentinel, start of dump or end of dump sentinel.	Abandon run and examine file N.
PARAMETER FILE ERROR HALTED ZZ	Oddities or errors found in the parameter file during re-entry or output phase. For example, a required set of output report parameters is not on the file.	Abandon run.
DECODE INDEX TOO SMALL HALTED ZZ	The area allotted to the decode index is not large enough to hold the necessary number of keys.	Abandon run. Adjust this area by means of #STORAGE parameters in the next run.

Table 11: Irrecoverable errors in input data

In the event of any other error, abandon and take core store print of 2000 words from Location 0, and 4000 words from Location 8000.

Errors in running

For #X63F, errors in running, which are not due to faulty own coding on the part of the user, should be reported via the normal software error reporting procedure. In addition, the following items of documentation should be supplied to enable the error to be traced:

1 A source listing of the own coding incorporated.

2 A source listing of the FIND–2 segments OWNCODING and OVERLAYM.

3 A consolidation listing for the program.

4 A core print of the program's lower variable, common lower variable, upper variable and common upper variable store, according to the addresses for these obtained from the consolidation listing.

ERROR FLAGS

In the Multiple Enquiry System, the parameters are checked and printed out during the validation phase. In the Decode Update program, parameters are validated at the same time as amendments are made to the file. An error in a parameter will be indicated in two ways:

1 By a single character error flag printed to the left of the parameter, indicating the type of error, and

2 By an upward arrow (↑) on the line below the parameter, indicating the faulty field or character.

Up to four errors will be flagged and arrowed on one card or paper tape block, and then the checking of that parameter is discontinued. The flags will be left justified and will appear in print positions 1 and 3 for errors in major directives, or print positions 9, 11, 13 and 15 for errors in parameter fields.

Single character flags will be used to indicate the type of error found. The precise meaning of a flag alongside a particular parameter will be quite evident.

Multiple Enquiry System flags

A = Invalid name, that is, field or enquiry mispunched or not present in the Dictionary or list of allowed enquiry types.

B = Wrong Dictionary field type.

C = Enquiry too large.

D = Parameter field types incompatible.

E = General format error.

F = Wrong parameter type.

G = Sequence number error.

H = Field in parameter missing; parameter incomplete.

J = Parameter or field not accepted; area full, or too many specified.

K = Logic error in enquiries.

L = Redundant brackets or wrong operator in enquiries.

M = Major directive out of sequence.

Decode File Update program flags

D = Key 1 > Key2 in #DEL parameter.

F = Directive error.

G = Sequence number error.

H = Incomplete parameter.

J = Key out of range of indexes, or index error.

N = Single #DEL key not found.

O = Record nof inserted owing to lack of overflow space.

P = Key incorrect length.

Chapter 13 Dump and restart

Dump and restart procedures are included in the FIND–2 Multiple Enquiry System. They are available in the interrogation and output phases as a security measure when processing large files.

DUMP

Dumping options

Flexibility is provided by allowing the user to specify whether dumping is required and, if so, at what frequency. This specification is made by means of the #READ parameter for the main input file. The format of the #READ parameter is described in Chapter 5 and this shows the position of the dump frequency specification.

The specification can be one of the following:

1 NO or field left blank	No dumping will occur.
2 EOR	Dumping will occur at the end of each reel of the main input file, except for the final reel.
3 *n* (thousand)	Dumping will occur after processing *n* thousand records from the main input file.

Dump media

Dumping will normally be made to a separate file. The dumping frequency is specified on the #READ parameter for the main input file, and the dump file medium and file name are specified on a #WRITE parameter. The format of the #WRITE parameter is given in Chapter 5, page 19; for the sequence of parameters the user should refer to Chapter 8 page 47. When dumping to a separate disc file, the restart routine requires a file name ICT-DUMPxxxx, where xxxx are four characters of the users' choice.

When a line printer output is involved, the dump will not occur until the end of a page is reached.

When dumping to direct access, it is the users' responsibility to allocate a file on the appropriate cartridge. A file of 80 blocks should be sufficient to accommodate one program dump but the file may be larger. It will not be extended automatically by the program.

When the hit file or output file is magnetic tape, dumps may be made to this tape by omitting the #WRITE for the dump tape.

Batched runs

Dumps may be made during both interrogation and output. In this case a separate file will be used for each phase. Once a dump file is opened during the output phase it is left open until the end of the run. A single #WRITE in the first parameter set therefore suffices for all output phase dumping. The dumping frequency may, however be changed on successive #READ parameters so that, for example, there will be no dumping during certain reports.

RESTART

The dump parameters are displayed on the operator's console during dumping and should be punched on cards or paper tape to effect the restart. It should be remembered that the program is overlaid and a parameter specifying the program file is required for direct access. The routines to be used are:

#XJRT (To restart from magnetic tape dumps)

#XJRE (To restart from direct access dumps)

Full details of these routines and dump and restart procedures can be found in the ICL 1900 Series manuals *Magnetic Tape* (Edition 1, TP 4091) and *Direct Access* (Edition 1, TP 4107).

Appendix 1

Comparison of FIND-2 Multiple Enquiry System with FIND-2 Single Enquiry System

For those users who are familiar with the FIND–2 Single Enquiry System, this appendix summarizes the differences and added facilities in the Multiple Enquiry version of FIND–2.

1 A larger configuration is required, in particular 16K of core store instead of 8K.

2 Up to 96 enquiries may be batched for processing in one pass of the file to be interrogated.

3 The Multiple Enquiry System is a 'two-run' system with intermediate output of records satisfying the enquiries to a hit file, and a subsequent output or reporting run, selecting records from the hit file on the basis of enquiries satisfied.

4 The interrogation and output runs may be used separately if desired. The file used as input to the output run may be a normal user's file instead of a previously created FIND–2 hit file.

5 Validated parameters in the Multiple Enquiry System are written to a parameter file. The user may take advantage of this by using the validation phase of the program separately to produce validated parameter files (optionally with generated enquiry programs) for repeated use, re-entering the system at the interrogation or output phases.

6 The output run of the program provides:

(a) Output to line printer, disc or magnetic tape.

(b) A user-formatted output line or record.

(c) Control totalling

(d) Output of two-dimensional tables of accumulated totals.

(e) Merging of records from two input files.

(f) Arithmetic between output fields.

(g) Much more extensive editting than in the Single Enquiry System.

(h) Output to the line-printer of up to 100 fields from each record instead of 30, with multi-lines per record and/or two records per line.

(i) Variable spacing between lines of print as specified by the user.

(j) User-specified main and page headings.

(k) Decoding of fields.

7 Own coding entries in both the interrogation and output phases of the program.

8 The facility for altering the sizes of the program's main tables and storage areas at run-time.

9 No *max* or *min* type enquiries.

10 A Lead type enquiry is provided which any record must satisfy before subsequent enquiries in a batch are applied to it.

11 A constants dictionary may be used to make the values of constants in enquiries variable when re-running validated/generated enquiry programs.

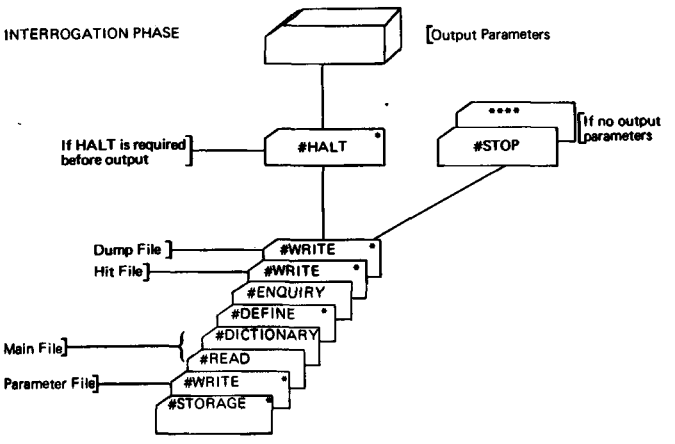
12 The dictionary of field definitions is pre-set to 500 definitions instead of 100.

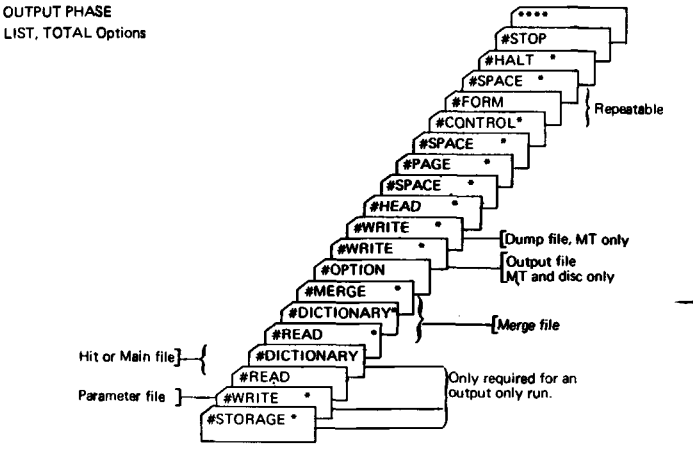
13 Because up to 96 enquiries may be applied to a file in one pass, there is no facility as in the Single Enquiry System for batching together separate passes of the file to be interrogated. Separate output runs from the same file or different files may, however, be batched.

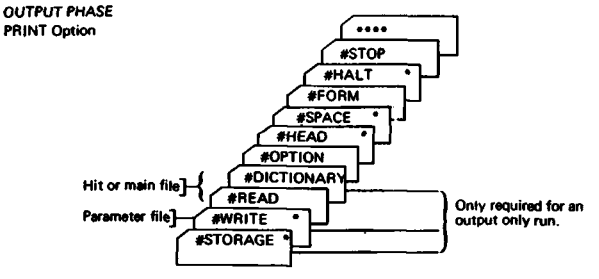
Appendix 2
Parameter format reference card for FIND-2

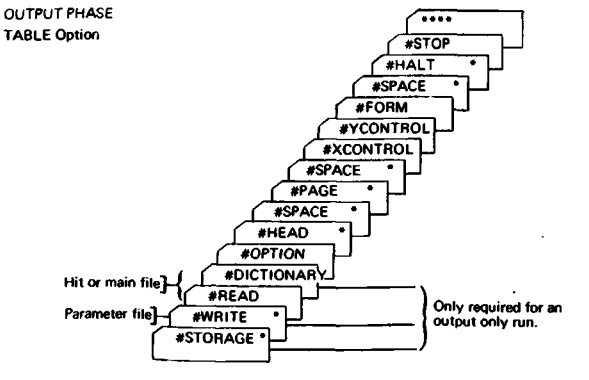
The information contained in this Appendix is also published as a parameter format reference card, Form No 14/106.

M.E.S. Parameter sequences









Parameter formats

#STORAGE , (Field Code)/(Area Size)

#WRITE , {MT / ED / FD} , (Filename), [{Old F.G.N. / R.S.N.}] , [F.G.N.] , (Retention Period) [,block size]
- Omission implies block size of input file
- Omission implies zero
- Up to 12 chars

#READ , {MT / ED / FD} , (Filename) , [R.S.N.] , [F.G.N.] , [Dump Frequency] , [Part File Specification]
- m-n, m-EOR, m-EOF
- No, ∇, EOR, n thousand
- Omission implies zero
- Up to 12 chars

#DICTIONARY
Fixed Length Fields
(Field Name) ∇ (Field Type) ∇ (Start Address) ∇ (Length)
- B, C, F, M, P, W, D or H
- Up to 7 chars, 1st alpha

Variable Length Type V Fields
(Field Name) ∇ (V) ∇ (No of preceding var. flds/Term char) ∇ [Start Address]
- Any char except ∇
- Range 0-35

Variable Length Type X Fields
(Field Name) ∇ (X) ∇ (No of preceding var. flds.) ∇ [Start Address]
- No limit

#DEFINE ∇ (Name) ∇ {= / EQL} ∇ (constant)
- Up to 7 chars, 1st alpha (can duplicate field names)

#ENQUIRY
Enquiry Identifier
(?) (Enquiry Name) ∇ [{STANDARD / COUNT / MAX / MIN / LEAD}] ∇ [{Hit Count / (Max/Min Key) ∇ (ncount)}] ∇ [Comment]
- Range 1-30
- 4 chars

#OPTION ∇ {LP ∇ {96 / 120 / 160} / MT, ED, (Size of Output Record) / FD} ∇ {PRINT / LIST / TOTAL / TABLE} ∇ [HIT n ∇] [ALL, (Identifiers)] ∇ [ANY, (Identifiers)]

#MERGE ∇ (Key 1) ∇ (Key 2) ∇ {MATCH / NON MATCH}
- Merge Dictionary Field Name
- Main Dictionary Field Name

#CONTROL ∇ (Level) ∇ (Field Name) ∇ [PAGE] ∇ [(Comment)]
- Range 1-7

#XCONTROL ∇ (X Level) ∇ (Field Name) ∇ (Value 1) ∇ [Value 2] ∇ . . . . [Value n]
- Range X1-X3

#YCONTROL , (Y Level) ∇ (Field Name) ∇ [{PAGE / (Value 1) ∇ . . . . (Value n)}]
- Range Y1-Y2

#HEAD
[Cols 1-60] (up to 60 chars of text) [Cols 61-72] [line no]

#PAGE
[Cols 1-60] (up to 60 chars of text) [Cols 61-72] [line no]

#SPACE ∇ {n / PAGE / CHANNEL 1-7}

#FORM ∇ [No of recs/line] ∇
LIST,TOTAL,TABLE options PRINT option

Format Line) (Field Definitions) (Field Names) [(Edit Symbol)]

Field Definitions

(i) [Field Tag], {Field Name / Field Tag / 'Constant'} {+ / - / * / /} {Field Name / Field Tag / 'Constant'} , [Sign if Positive], [Sign if Negative], [{£ / $ / * / Z}] {[Totalling] / [Column %], Row}
- Up to 7 chars, 1st numeric

(ii) [Field Tag] , (Field Name) , (DECODE) , (Filename)

(iii) [Field Tag] , (COUNT) , {GRAND / Control Level}

#END

#HALT

#STOP

Enquiry facilities

Simple Conditional Statement (Fieldname Relation Constant)

Fieldname = Const1; Const2; etc. (OR implied)

Fieldname BETWEEN Const1 , Const2

Fieldname ± Fieldname Relation Constant

ANY m FROM (n Conditional Statements)

STEP p BY q (Conditional Statement with = or /=)

GENERATE m FROM (Conditional Statement)

CONSTANTS

Octal	eg #76342	
Binary	eg +19 , .591 , −27.02	
Sterling	±Lx , ± Lx/y/z , ± Lx/y/z.n	
Decimal Sterling	£x.y	
Date	DD/MM/YY	
Character	(i) with a space terminator	eg BROWN
	(ii) with a significant space	eg 'JOHN BROWN'
	(iii) partially defined	eg SM*TH
Predefined	@ DATE	(#DEFINE DATE = 02/08/69)
	@ TODAY (current date from executive)	

RELATIONSHIPS				OPERATORS
EQL	=	NEQ	/=	AND
LSS	<	LEQ	<=	OR
GTR	>	GEQ	>=	

POSSIBLE COMPARISONS OF FIELD AND CONSTANT TYPES

Constant types	Field types B	C	D	F	H	M	P	V	W	X	Examples
Character		✓						✓		✓	NAME EQL BROWN NAME EQL 'JOHN BROWN'
Integer	✓		✓		✓	✓	✓		✓		LOSS GTR -225
Fraction				✓		✓	✓				RATE LSS.75
Mixed number						✓	✓				FIELD LEQ. 2.63
Old sterling									✓		TAX LSS L4S10D6
Decimal sterling			✓		✓						AMOUNT EQL £2.7
Date									✓		DATE EQL 18/02/69 DATE EQL @TODAY
Octal	✓								✓		FIELD EQL #7572

Appendix 3 Program limitations

GENERAL LIMITATIONS

1 The minimum configuration is either:

(a) 16K, 1CR/PTR, 1LP, 3MT, or

(b) 16K, 1CR/PTR, 1LP, 1E.D.S./T.E.D.S.

2 Random disc files are processed serially.

Note: Entries marked with an asterisk may be amended by means of the #STORAGE parameter.

INPUT LIMITATIONS

1 Input files must conform to ICL 1900 Series Housekeeping Standards.

2 Binary keys on direct access files are limited to signed single or double length integers.

*3 The maximum block/bucket size is 1024 words.

4 The maximum record size is 1024 words.

5 It is not possible to reference data held in sentinels.

*6 The total number of entries in the file dictionaries (main and merge file dictionaries) is preset to 500.

7 Field names may be of any number of alphanumeric characters of which the first must be alphabetic. The system uses the first seven characters only and these therefore must be unique (within both dictionaries if both are present).

8 The following reserved system words must not be used as field names:

OR and COUNT.

No field name may start with any of the following: AND, ANY, STE and GEN.

9 Only one type of variable length field can be used on any one file.

10 A maximum of 36 type V variable length fields are allowed in a record. There is no limit to the number of type X fields.

11 A space cannot be used as a terminator for a type V field.

12 If it is required to omit a variable length field in a record, the terminator or word count must still be present.

13 Floating point fields are not processed by the system.

14 Maximum field sizes permitted on the input file are given in Chapter 5.

15 A file to be used for decoding must be an indexed sequential E.D.S. file.

16 The encoded form of the field in the decode file must be specified as the file key and the decoded form of the field must immediately follow the key in the record. Records must be of fixed length.

17 First level overflow must not be present in a decode file.

18 Only two files may be merged together in the output part of the program.

19 All parameter input must be on one device.

20 Paper tape blocks for input parameters are limited to 72 characters.

ENQUIRY LIMITATIONS

*1 Up to 96 enquiries can be dealt with in one pass of the file being interrogated.

2 Three types of enquiry are available: Standard, Count and Lead.

3 Ten depths of logic may be specified.

*4 A total of approximately 300 relations may be specified.

5 The size of a constant in an enquiry is limited as follows:

(a) Character : 72 characters

(b) Octal : #37777777

(c) Monetary/date/decimal values : by limits of fields

6 Limitations of the comparisons of fields and constants are shown in Table 3, page 33.

7 Enquiry names may be of any length. The first four characters are used by the System and must be unique.

8 Enquiry names must not begin with the reserved system words: ALL, ANY.

9 The hit count specified in the enquiry identifier is limited to a maximum of 131071.

10 ANY *m* FROM *n* (conditions).

(a) *m* is greater than 0 and less than *n*.

(b) All conditions must be simple, that is, single criteria conditions. (No arithmetic relationships are allowed).

(c) Only one level of logic is allowed within the ANY statement.

(d) The ANY statement must be bounded by brackets.

11 STEP *m* BY *n* (condition).

(a) *m* is greater than 1 and less than 1024.

(b) *n* is greater than or equal to 1 and less than 4096 characters.

(c) The condition must not include arithmetic and must specify a fixed or variable length character field.

(d) The relationship must be EQL or NEQ.

12 GEN *n* FROM (condition).

(a) *n* is greater than 1 and less than 100.

(b) The condition must be as in 10 (b) above.

(c) The field name must be at least seven characters long, with the sixth and seventh characters numeric in the range 00 to 99.

13 ARITHMETIC CONDITION

(a) The arithmetic must be addition or subtraction, and only two fields may be computed.

(b) The fields must be of the same type but not necessarily the same length.

(c) The field type must be D, F, H, M, or W.

14 REPEATED CONSTANTS

(a) Any simple condition, including arithmetic, is allowed.

(b) Constants are separated by semi-colons but each constant must be terminated by a space or comma.

(c) This facility may be used with a STEP condition.

15 In a range condition the lower constant must be specified first.

16 With the exception of brackets, each item must be terminated by a space or comma or the end of the parameter.

17 A parameter break may occur between any two items but must not occur in the middle of an item.

18 The enquiry identifier is distinguished by the ? character in column 1, and therefore this character must not appear in the first column of an enquiry set parameter.

19 Each enquiry must be bounded by brackets.

20 It is not possible to reference the count word of a type X field.

OUTPUT LIMITATIONS

1 A table showing the limitations of the four main output options in terms of facilities provided is given in Chapter 7. Further limitations are listed below.

2 Generally:

(a) The maximum number of lines for main and page headings depends on the line printer size:

160 p.p. : 5

120 p.p. : 6

96 p.p. : 8

(b) The highest page number output is 9999.

3 In the Print option:

(a) A maximum of 30 fields from each record may be output.

(b) Automatic grand totalling of all binary fields output is performed.

(c) The number of variable-length fields which may be output is restricted by the line printer size, being a maximum of:

160 p.p. : 5

120 p.p. : 3

96 p.p. : 3

(d) Only the first ten characters of a field name are used as a column heading.

(e) The size of output fields is limited as follows:

Field type	*Maximum size*	*Maximum positive value*	*Maximum negative value*
B	8 decimal characters per line		
C	127 characters		
V or X	30 characters per line		
W (Numeric)	An 8 digit number	99999999	–9999999
W (old sterling)	8 places of pounds, 2 of shillings, and 2 of pence.	£9999999 19 11	–£999999 19 11
D (decimal sterling)	8 places of pounds and 2 of pence.	£9999999.99	–£999999.99
D (decimal currency)	8 places of basic unit and 2 decimal places	99999999.99	–9999999.99
H (decimal sterling)	8 places of pounds and 3 of pence	£9999999.995	–£999999.995
M	An 8 digit number, plus 4 decimal places.	99999999.9999	–9999999.9999

Field type	*Maximum size*	*Maximum positive value*	*Maximum negative value*
F	A 4 digit number plus 4 decimal places	9999.9999	–999.9999
P	An 8 digit number, plus 4 decimal places	99999999.9999	–9999999.9999

Table 12: Limits to size of output fields.

4 In the List and Total options:

*(a) A maximum of 100 fields from each record or pair of merged records may be output (unless output is of unedited records to magnetic tape or disc).

*(b) The maximum output block size on magnetic tape is 1024 words.

(c) The maximum output record size on magnetic tape or disc is 1024 words.

(d) Merging and the two-records-per-line facility may not be used together.

(e) Totalling and the two-records-per-line facility may not be used together.

(f) Up to 7 levels of control totalling may be used, as well as grand totalling.

(g) A maximum of 12 fields may be totalled from each record or pair of merged records, and each may be totalled at any selection of control levels and/or Grand.

(h) Listing with control totals and/or grand Totals may only be output to the line printer. A listing only or totalling only run may also be output to magnetic tape or disc.

(i) Variable length fields may not be output to magnetic tape or disc.

(j) A maximum of six variable length fields may be output in one print line.

(k) When merging, variable length fields may only be output from both records provided restriction (j) is observed and both files have the same type of variable length field.

(l) When merging, any part-file specification or selection of records by enquiries satisfied will only be applied to the Main File.

(m) Control fields may not be decoded.

(n) A maximum of 16 arithmetic operations may be specified.

(o) In arithmetic and totalling, a standard triple-length form is used to hold values, comprising 2 words for the integer part, and 1 for the fractional part.

*(p) Store space is available for between 400 and 700 characters of constant information to be put in the output print lines from each record.

(q) Control fields may not be variable length and must be 12 characters or less.

5 In the Table option:

(a) A maximum of three X Control levels may be specified.

(b) A maximum of two Y Control levels may be specified.

(c) A maximum of 24 values for the lowest X Control field may be specified.

(d) X and Y control values must be less than 12 characters long and may not be bounded by apostrophes.

(e) A maximum of 28 field totals (including row totals) may be accumulated for outputting on each line.

(f) A file for which 2 Y Control levels are specified must be sorted on the higher Y Control field.

*(g) A maximum of 27 different Y Control field values may be specified.

Appendix 4 Decimal currency conversion

The FIND–2 package incorporates the facility for converting old sterling to new and new sterling to old on printed output. This appendix shows how the conversions can be made in each of the four output options and gives details about the degrees of accuracy involved.

Sterling can be held on files in one of three forms:

1 As old pence (field type W)

2 As new pence (field type D)

3 As tenths of new pence (field type H)

PRINT OPTION

Field output forms are determined by use of the codes L, £ and C. The following table lists the various possibilities for this option.

Field type	*Output code*	*Output form*	*Floating £ symbol*	*Rounding*
W	–	Integer	–	–
	L	Old sterling (£ s d)	✓	–
	£	Decimal sterling (£n.nn	✓	To 1 new penny*
	C	Decimal sterling (£n.nn)	✓	To nearest new penny
D	–	Decimal sterling (£n.nn)	✓	–
	L	Old sterling	✓	To nearest old penny
	£	Decimal sterling (£n.nn)	✓	–
	C	Decimal currency (Cn.nn) (for example, dollars and cents held in cents)	–	–
H	–	Decimal sterling (£n.nn5)	✓	To nearest new halfpenny
	L	Old sterling (£ s d)	✓	To nearest old penny
	£	Decimal sterling (£n.nn5)	✓	To nearest new halfpenny
	C	Decimal sterling (£n.nnn)	–	–

* indicates that the conversion is carried out by reference to the Board of Trade banking conversion table.

Table 13: Currency output for Print option

In all cases except that indicated by an asterisk above, conversion is performed by straight calculation and the results are rounded to the nearest significant digit of the output form. 1 to 4 are rounded down and 5 to 9 rounded up.

LIST, TOTAL and TABLE OPTIONS

Field output forms are determined by the format line specification and the following table lists all the various possibilities for these options.

Field type	*Output format*	*Output form*	*Rounding*
W	xX	Integer	
	xL	Sterling pounds	To nearest pound
	xL 2S 2D) xS 2D)	Old sterling	
	xL 2S 2D.yF) xS 2D.yF)	Old sterling	Truncated fraction of old penny
	x£.2D	New sterling	To 1 new penny*
	xC.2C	New sterling	To nearest new penny
D	xX	Integer	
	xL	Sterling pounds	
	xL 2S 2D) xS 2D)	Old sterling	To nearest old penny
	xL 2S 2D.yF) xS 2D.yF)	Old sterling	Truncated fraction of old penny
	x£.2D	New sterling	
	xC.yC	Decimal currency (basic unit determined from value of y)	
H	xX	Integer	
	xL	Sterling pounds	
	xL 2S 2D) xS 2D)	Old sterling	To nearest old penny
	xL 2S 2D.yF) xS 2D.yF)	Old sterling	Truncated fraction of old penny
	x£.3D	New sterling	To nearest new halfpenny
	xC.3C	Decimal currency	

* indicates that the conversion is carried out by reference to the Board of Trade banking conversion table.

Table 14: Currency output for List, Total and Table options.

In all cases except that indicated by an asterisk above, conversion is performed by straight calculation and the results are rounded to the nearest significant digit of the output form. 1 to 4 are rounded down and 5 to 9 rounded up.

Glossary

Batching run	:	A facility for producing several reports from one file or a selection of reports from several files.
Bit	:	A bit is a method of holding information within the computer and takes the form of a single digit having the value 0 or 1.
Condition	:	Can be either a single condition of the form: (fieldname) (relationship) (constant) or a complex condition comprising several simple conditions linked by an operator and bounded by brackets.
Constant	:	An explicit value, character or numeric (binary), to be compared with the contents of a field.
Decoding	:	The facility of translating coded information, used to economise on storage, into a more readable form for report preparation.
Dictionary	:	Detailed definition of the fields comprising each record on the user's files.
Dump and Restart	:	A security measure which preserves information about the state of a run at specific intervals. The information preserved is such as to eliminate the need to restart the run from the beginning in the event of a breakdown.
Enquiry	:	Consists of one or more conditions linked by the logical operators, AND and OR. Ambiguity of logic is avoided by the user of brackets.
Enquiry program	:	A unique set of coding formed from the enquiry parameters to maximise the efficiency of the interrogation.
Enquiry selection	:	Selection of records for output on the basis of the enquiry satisfied by each record.
Field	:	A defined section of the record holding a certain item of information.
Field name	:	The name associated in the Dictionary with a particular field.
Hit	:	The satisfaction of an enquiry by a record. The *hit file*, containing all such successful records is produced during the interrogation phase.
Hit word	:	Added to the end of an input record when one of the enquiries is a hit. One hit word is added for each set of 24 enquiries (4 words for 96 enquiries). The hit word is used later in the system to check the enquiries satisfied by the record.
Logical operator	:	Always links the two conditions on either side of it, by AND or OR.
Match	:	The process carried out during merging when records from both the main and merge files are found to have the same key value.
Merge	:	The combination of information from two files into one new file or report.
Output group	:	The parameters required to produce all output reports/files from one file.
Output set	:	The collection of parameters required to produce one output report/file.
Own coding	:	Coding written by the user, to be incorporated into the program at specific points, in order to modify the action of the package.
Parameter	:	Definitions of file, enquiry, etc. that determine the use to be made of the system on a certain run.
Part-file interrogation	:	Limiting the area of search to a specific part of a file.

Partial character-constant : A facility which provides the option of ignoring certain character positions in a constant when performing a character equality comparison.

Range condition : A facility which specifies that the contents of a field lie between two constant values.

Re-entry : The use of previously validated parameters; for example, to repeat a regular enquiry, or to produce output from a sorted hit file.

Relationship : Connects a field name and a constant. The allowed relationships are written in the enquiry in either the following mnemonics or symbols:

EQL, NEQ, GTR, GEQ, LSS, LEQ

=, /=, >, >=, <, <=

Row total : A facility enabling a horizontal total to be produced, of a field appearing in a series of Table option groups.

Step search : A search for a constant, when its position in a field of a record is not known.

Table option group : A number of fields totalled in one column of a Table, that is, under a single value of the lowest X-control field used (or a single enquiry name at the lowest X-control level). Each group may consist of totals of the same, or different fields.

Index