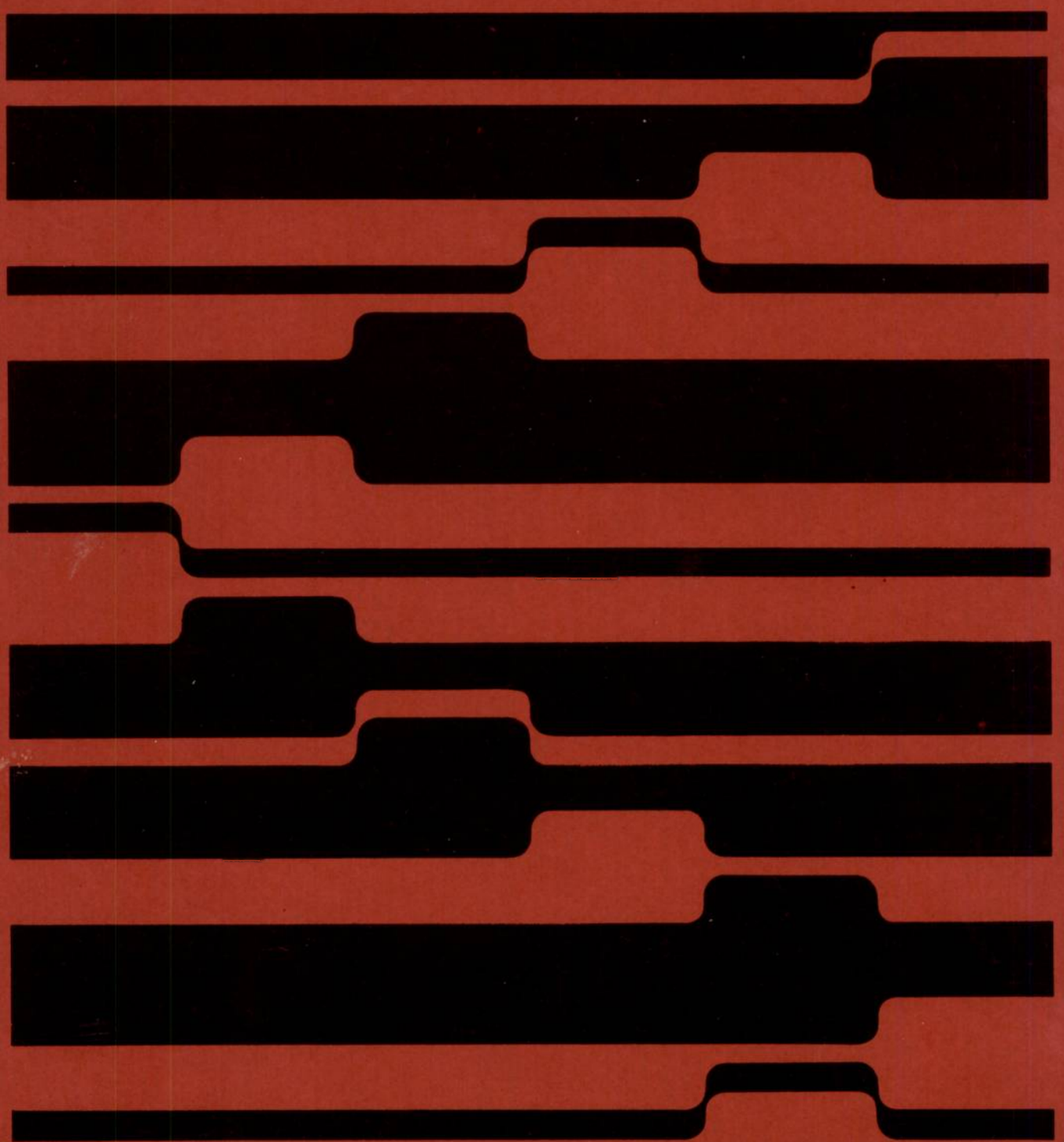
ICL

Data Management Software Framework Programs

1900 Series



ICL

Data Management Software Framework Programs

1900 Series

A

Technical Publication 4146

First Edition May 1970

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

Preface

This manual describes the two Data Management Software framework programs. The function of the first program is to create printed output from records held on magnetic tape files. The second program edits records held on magnetic tape files. The framework programs, using parameters in a given format, provide the basic program structure within which the user's processing can be performed.

The manual is one of a series describing ICL Data Management Software, and should be read in conjunction with the manual *Data Management Software Introduction,* which explains the philosophy behind ICL's approach to data management.

The manual is intended to be read by those already using or considering the use of the Data Management Software programs described. The other programs in the Data Management Software suite may be used either by programmers or by people with no experience of programming. The programs described in this manual, however, are designed primarily for programmers who have been assigned the task of coding for their installation's specific requirements. The manual will also be of use to systems analysts and designers in so far as it explains what functions the programs are capable of performing.

There are two parts to the manual, each dealing with a single program as follows:

Part 1 describes PRINTMAKER which when combined with the user's own coding forms a complete print program.

Part 2 describes SUBEDIT which when combined with the users own coding forms a complete editing program.

Within each part there are chapters giving a general introduction to the program, a description of file input and output, and a detailed explanation of the user coding and parameter set needed for the program. There is an example of how the program could be used in an actual application and a chapter giving operating procedures and exception conditions.

It should be noted that a general description of all the Data Management Software programs will be found in the introductory manual. This manual also describes parameter standards and programming conventions which are not fully dealt with in the individual manuals. A typical example of the applications of the software is given.

The minimum configuration for each program is given in the introductory chapter to that program. No program requires more than 16K of store.

Contents

Chapter 1 Introduction to PRINTMAKER 1
PROCESSING SUMMARY 2
Consolidation phase 2
Parameter processing 2
Main processing 3
PAGE-CHANGING 3
MINIMUM CONFIGURATION 4

Chapter 2 User coding 5
FORM OF THE USER SEGMENT 5
Data areas 5
COMMON/HEADINGS/ 5
COMMON/LINES/ 5
COMMON/WORKING/ 6
COMMON/BLOCK 1/ 6
User subroutines 7
START 7
RECRD 7
LEVEL SUBROUTINES 7
PHEAD 8
PFOOT 8
USENT 9
CLOSE 9
Extra subroutines available 9
BLAH 9
SHUNT 10
PRINT 10
Use of Switches 10
Totalling 10

Chapter 3 Input and Output 11
INPUT 11
Input file 11
Parameters 11
OUTPUT 11
Line printer output 11
PARAMETER LIST 11
THE REPORT 11
Console output 11

Chapter 4 The parameter set 13
READ TAPE LABEL PARAMETER (#READ) 13

KEYS PARAMETER (#KEYS) 14
PAGE PARAMETER (#PAGE) 14
HEADINGS PARAMETER (#HEAD) 15
END OF PARAMETERS MARKER 15
RESTART PARAMETER (#RESTART) 15

Chapter 5 Example 19

Chapter 6 Operating instructions and exception conditions 25
OPERATING INSTRUCTIONS 25
Additional facilities 25
TO RE-ALIGN THE PAPER OR LOAD MORE PAPER 25
TO STOP IN THE MIDDLE OF A RUN 25
RESTARTS 26
EXCEPTION CONDITIONS 26

Chapter 7 Introduction to SUBEDIT 29
PROCESSING SUMMARY 29
Consolidation phase 29
Parameter processing 30
Main processing 30
MINIMUM CONFIGURATION 30

Chapter 8 User coding 31
FORM OF THE USER SEGMENT 31
Data areas 31
COMMON/STORE 31
COMMON/WORKING/ 31
User subroutines 32
START 32
RECRD 32
CLOSE 32
Extra facilities 33
THE SUBROUTINE SHUNT 33
THE CUE NEXT 33

Chapter 9 Input and output 35
INPUT 35
Input file 35
Parameters 35
OUTPUT 35
Output file 35
Line printer output 35
Console output 35

Chapter 10 The parameter set 37
READ TAPE LABEL PARAMETER (#READ) 37
WRITE TAPE LABELS PARAMETER (#WRITE) 38

END OF PARAMETERS MARKER 38

Chapter 11 EXAMPLE 41

Chapter 12 Operating instructions and exception conditions 43
OPERATING INSTRUCTIONS 43
EXCEPTION CONDITIONS 43

Illustrations

Figure 1	PRINTMAKER outline flowchart	facing page 1
Figure 2	Layout of common areas in PRINTMAKER	4
Figure 3	PRINTMAKER example: file formats	18
Figure 4	PRINTMAKER example: parameters	18
Figure 5	PRINTMAKER example: specimen printout	19
Figure 6	PRINTMAKER example: compilation listing	21 to 23
Figure 7	SUBEDIT outline flowchart	28
Figure 8	Layout of common areas in SUBEDIT	30
Figure 9	SUBEDIT example: file formats and control parameters	40
Figure 10	SUBEDIT example: parameters	40
Figure 11	SUBEDIT example: compilation listing	42

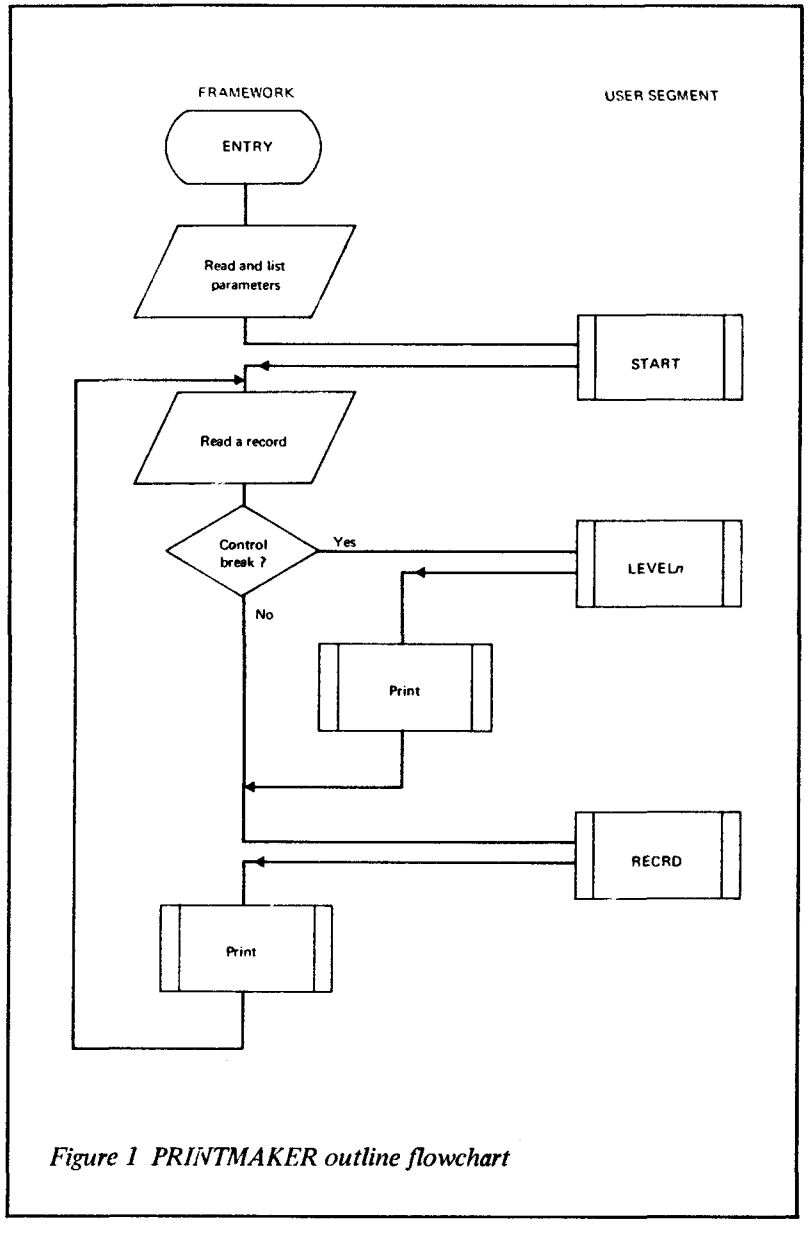
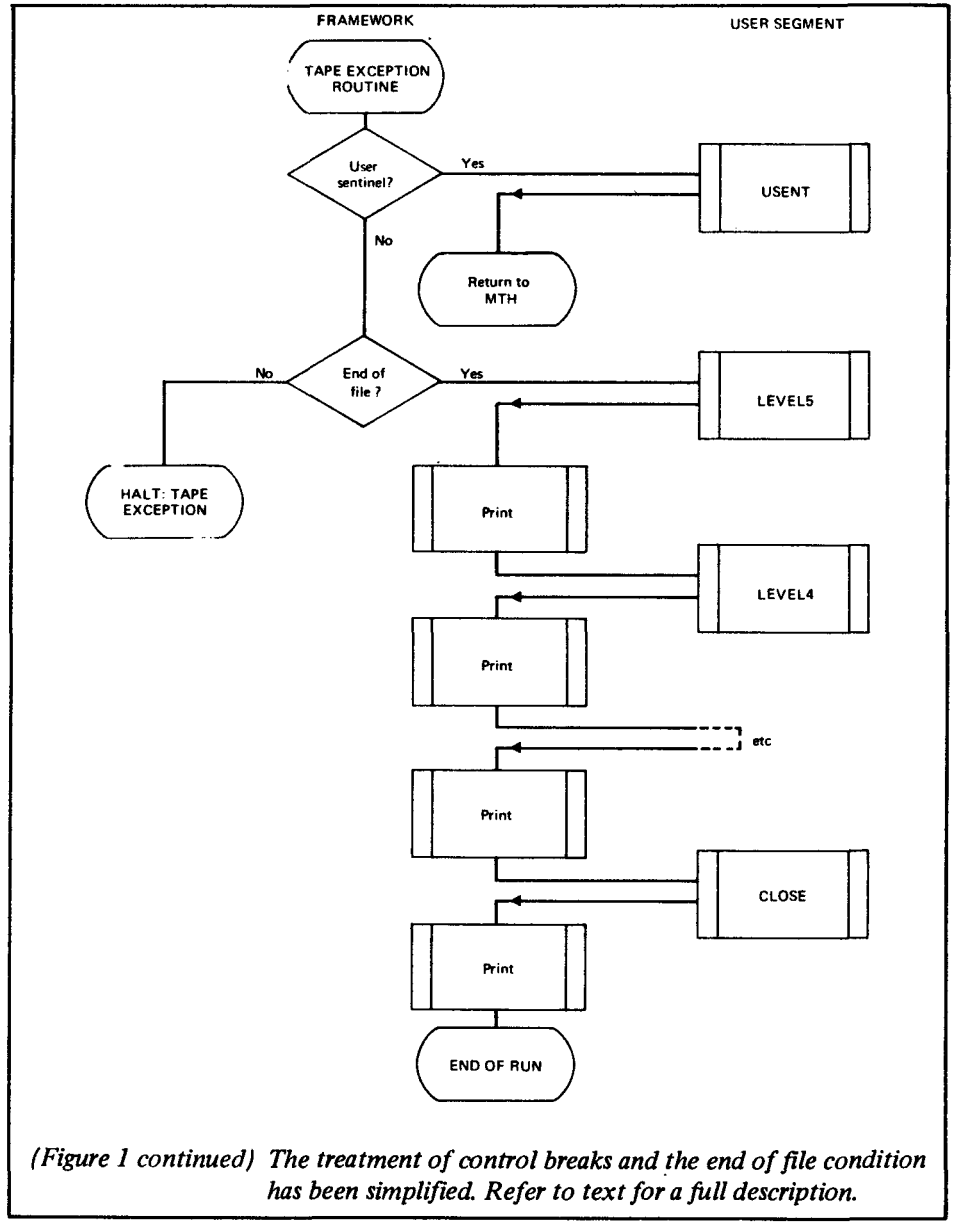


Figure 1 PRINTMAKER outline flowchart



(Figure 1 continued) The treatment of control breaks and the end of file condition has been simplified. Refer to text for a full description.

Chapter 1 Introduction to PRINTMAKER

PRINTMAKER consists of a group of routines in the form of a skeleton program which the user combines with a segment, or segments, of his own coding to form a complete print program.

The complete print program reads records from an input file, processes the data and prints the required output. While the user's coding is concerned mainly with the processing of the data, the PRINTMAKER routines deal with the standard functions of opening the input tape and reading the records. The PRINTMAKER routines also enable the user to control the printing of the output.

The user has the entire responsibility for the content of each line that is printed and for defining his spacing requirements.

The user's subroutines prepare the print lines by setting up each line of 120 characters for printing, together with the appropriate print and feed control character. Up to ten lines may be set up by each user subroutine. When the lines have been set up, PRINTMAKER takes over and ensures that they are correctly printed.

The user's subroutines perform the following operations:

1 Start of run processing

2 Main processing of each record

3 The provision of totals when a control break occurs (whenever there is a change of key)

4 The insertion of information at the head of each page

5 The insertion of information at the foot of each page

6 The processing of user sentinels

7 End of run processing

PRINTMAKER performs all checks for control breaks and page changes and passes control to the user subroutine for the processing required at such points. Information on keys, headings and page changes is given by parameters input on cards or paper tape. A standard Data Management Software parameter, #READ is used to open the input tape.

The following table provides a summary of the functions performed by the respective routines.

FRAMEWORK routines	*USER routines*
1 Read run time parameters	
	A Start of run procedure
2 Open input file	
3 Read next record from file and check for control breaks, calling up relevant user routines if break is detected.	
	B Up to six routines to handle control breaks at various levels

FRAMEWORK routines	*USER routines*
4 Pass record to USER	
	C Process record as required
5 Print data on return from *each* USER routine checking also for:	
6 page changing	
	D Set up page heading etc.
7 foot of page	
	E Set up data for foot of page
On reading file sentinels:	
user sentinel	F Process sentinel as required
8 file end sentinel, call up USER routines for final control break	
	G End of run procedures
9 Close file etc., end of run.	

PROCESSING SUMMARY

Consolidation phase

The PRINTMAKER Framework is held in the program library as a free-standing segment. The user forms his own subroutines into a segment or segments and includes a reference to PRINTMAKER. The user's segment is then submitted to an appropriate PLAN3 compiler preceded by a steering line defining the user's steering requirements in the usual way. The reference to PRINTMAKER is treated as a call to a library subroutine and ensures that the PRINTMAKER framework is extracted from the library for consolidation with the user's segment or segments. PRINTMAKER is thus consolidated in the same way as a library subroutine. PRINTMAKER can therefore be supplied in semicompiled form on the library tape and need not be recompiled each time it is needed.

The output from this stage is an object version of the PRINTMAKER program, consisting of the user's segment or segments, the PRINTMAKER framework and any other library routines. This program has the name that was given to the user's segment. The object program will be output on the normal output medium of the compiler that has been used. If the user's steering line has included LIST, SHORTLIST or FULLIST, a list of the appropriate form will also be output by the compiler. This will show only the user's segment. Listing of the PRINTMAKER framework is suppressed, as with any other library routine.

Parameter processing

At the start of the run the PRINTMAKER program reads and lists the parameters.

It then calls the user subroutine START for any preliminary processing that may be required. It may be necessary, for example, if the headings are permanently held in user subroutines and not input by parameter (see page 5) to insert at this stage, information, such as the date or the generation number of the input file, which is relevant only for the present run.

The START subroutine may also be used for any other start of run processing. PRINTMAKER initializes its own working areas; totalling areas must be set and initialized by the user.

The user may provide a PRINTMAKER program with additional parameters containing other control factors of his own choice. He may design such parameters in any convenient form. When the START subroutine is called,

PRINTMAKER has not yet released the peripheral from which the standard parameters were read. The user may therefore add his own parameters to the back of the standard set and read them in this subroutine with any suitable PERI or I/O Generator statement.

Main processing

PRINTMAKER reads the input file record by record. Each record except the first is tested for control breaks, that is, a change in any of the fields named as key fields by the #KEYS parameter.

If no control break is detected, PRINTMAKER calls the user's main subroutine to perform any processing that may be necessary on the current record. Where there is only one type of input record, this subroutine will normally contain only one processing routine. This processing will normally include calculations, and the preparation of lines for printing.

Where there are several different types of input record, however, the subroutine will normally contain several parallel processing routines. In this case the user coding should include some means of selection so that each record is processed according to its type.

Similarly, records of a particular type can be ignored by by-passing all calculation and print distribution routines for such records, with the result that information from these records does not appear in any of the output printed by the program. The user should note, however, that PRINTMAKER tests all records for control breaks in the same way and this may result in records that the user wishes to ignore giving rise to unwanted control breaks. A GAMP filtering run (see *Data Management Software: Updating*) will remove such records more satisfactorily.

When a control break is detected, PRINTMAKER transfers control to the appropriate user subroutine. The user must provide the following subroutines (described fully in Chapter 2):

1 START to prepare headings and start of run processing.

2 RECRD to process the current record, that is, to set up print lines and totalling areas for the record.

3 LEVEL to take control break totals: six of these subroutines should be provided, one for each possible level of control.

4 PHEAD to process page headings.

5 PFOOT to process page overflows.

6 USENT to process user sentinels

7 CLOSE to implement end of run processing.

All the subroutines listed *must* be provided; if the user does not wish to use a particular facility he should substitute a *dummy*, that is, a subroutine consisting of one instruction: an EXIT to the main PRINTMAKER program.

It is not necessary, however, to use the subroutines for the purposes listed above. Although they will most often have these functions, the user may use a call to his own subroutine for any convenient purpose.

PRINTMAKER observes the rule that a break at any level of control implies a break at all lower levels. When a control break is detected at any level, the user subroutines for any lower levels (starting with the lowest level) are called before the RECRD subroutine to process the current level is called. When the control break has been processed at all relevant levels, PRINTMAKER calls the RECRD subroutine to process the first record of the next group.

PAGE-CHANGING

Throughout the run PRINTMAKER deals with all aspects of page-changing. If a user uses only head-of-form and single or double line spacing PRINTMAKER will maintain a correct count of lines left in the page and will call the end of page routines (PFOOT) and page heading routines (PHEAD) when the line-count is exhausted. Under such conditions the user can also force a page change by setting the line-count to zero or by setting a page overflow signal, a hole punched in channel eight of the printer control loop which will be encountered before the line-count is exhausted.

Note: under the conditions mentioned above, head-of-form and single or double line spacing, all lines, except those set up in PFOOT, will lead to an adjustment of the line-count; head-of -form will reset it to the maximum value while single or double line spacing will reduce it by one or by two. The user should take this into consideration if he wishes to change the line count from the standard 56 lines incorporated in the program, and should ensure that the new count, together with any lines printed after PFOOT, does not accidentally exceed one page of print.

If, however, the user incorporates throws to any of channels 2 to 7, PRINTMAKER will not adjust the line-count

for such lines although page-heading and single or double line spacing will continue to do so. Unless the user can define some suitable line-count therefore relying on only single or double line spacing he will have to force page changes by either of the methods mentioned above, setting the line-count at zero at some point, as calculated in his own routines, or relying on a page overflow signal.

At the end of run, the last record on the input file is followed by a control break at all levels of control. All the LEVEL subroutines are therefore called, starting with the lowest, LEVEL5. PRINTMAKER then calls a CLOSE subroutine for any additional end of run processing, such as the preparation of print lines showing final totals. PRINTMAKER then throws to a new page and prints a single line, giving a count of the records read during the run.

MINIMUM CONFIGURATION

The minimum computer configuration required for PRINTMAKER is:

1 1900 Series central processor with 8K words of core store

1 card or paper tape reader

1 line printer

1 magnetic tape deck

Note: the core size required by the final program will depend on the size of the user's segment. Unusually complex PRINTMAKER programs may therefore require more than 8K.

```
0001    #PROGRAM        /PRINTMAKER
0002    #CMODE          MT(0)NULL(1-15)
0003    #LOWER          COMMON/HEADINGS/
0004                    HEAD0(30),HEAD1(30),HEAD2(30),HEAD3(30),HEAD4(30)
0005                    HEAD5(30),HEAD6(30),HEAD7(30),HEAD8(30),HEAD9(30)
0006    #LOWER          COMMON/LINES/
0007                    SPAC0,LINE0(30),SPAC1,LINE1(30),SPAC2,LINE2(30)
0008                    SPAC3,LINE3(30),SPAC4,LINE4(30),SPAC5,LINE5(30)
0009                    SPAC6,LINE6(30),SPAC7,LINE7(30),SPAC8,LINE8(30)
0010                    SPAC9,LINE9(30)
0011    #LOWER          COMMON/WORKING/
0012                    POINT,ENTRIES,LINECOUNT,PREV(30),COUNT,SPAC,PARAM(30)
                                                        R2= WORKING
0013     CNTRL          0,0,60,0/PARAM                          65  #00000000
                                                                66  #00000000
                                                                67  #00000120
                                                                68  #00000043   R2
0014     PRIN2          2/0,0,121,0/SPAC.3                      69  #00200000
                                                                70  #00000000
                                                                71  #00000171
                                                                72  #60000042   R2
```

Figure 2 Layout of common areas in PRINTMAKER

Chapter 2 User coding

FORM OF THE USER SEGMENT

The user's segment contains a number of subroutines each introduced by a #CUE line. The segment starts with a #PROGRAM directive, containing a program name and priority number. The program name supplied by the user is used as the name for the complete print program.

The segment should include data statements, see below, under #LOWER COMMON for any standard locations, which the user requires to access.

In addition, the segment may contain data statements to set up any other store locations which the user's subroutines may require. The layout and naming of these locations is entirely under the user's control.

The segment must contain at some point a call to PRINTMAKER. This may take the form of a BRN instruction as follows:

```
BRN          PRINTMAKER
```

Alternatively, a CALL instruction may be used:

```
CALL    0    PRINTMAKER
```

Where a CALL is used for this purpose the link accumulator specified is immaterial. Note: this CALL or BRN must *not* be obeyed. Its sole function is to instruct the compiler to bring in the PRINTMAKER framework. The cue name PRINTMAKER must not be used in the user's segment. The compiler will regard any branch to it as a branch unsatisfied within that segment, in the same way as other branches to library software. Similarly, the names of the PRINTMAKER framework subroutines, SHUNT, PRINT and BLAH, see page 9, must not appear as cue names in the user segment.

Data areas

The following names are used for areas in LOWER COMMON storage, provided by the PRINTMAKER framework, and used by the framework. Since the user will also use all of these areas, he should include suitable data statements in his own segment, using the same common area names and observing the same layout for fields within the areas. Figure 2 illustrates the layout of the common areas on a printout.

COMMON/HEADINGS/

PRINTMAKER has two methods of obtaining heading information:

1 If the headings are likely to be changed from run to run it is advisable to read them from heading parameters, see Chapter 4 page 15, at the time of the run.

2 If the headings are likely to remain constant, they can be compiled with the program as preset data.

The COMMON/HEADINGS/ area is 300 words long. If parameters are being used the first 30 words, labelled HEAD0 contain information from the first pair of headings parameters; the next 30 words, HEAD1, contain information from the second pair of headings parameters, and so on up to HEAD9.

If the headings have been preset, the user may insert the preset heading information in this area in any convenient form.

COMMON/LINES/

This area of 310 words is used to set up lines for printing. In the START, RECRD, LEVEL, PFOOT and CLOSE subroutines, print-lines may be formed in this area; in the PHEAD subroutine, heading lines should be moved to this area from COMMON/HEADINGS/. All printing is performed by PRINTMAKER from this area. The area must have the following layout:

SPAC0	contains the print feed control character for the first line to be printed.
LINE 0 (30)	contains the first line to be printed.

SPAC1	contains the print feed control character for the second line to be printed.
LINE 1 (30)	contains the second line to be printed.

and so on, ending with:

SPAC9	contains the print feed control character for the tenth line to be printed.
LINE9 (30)	contains the tenth line to be printed.

When a user subroutine is called, the lines in this area have not been spacefilled by PRINTMAKER. The previous contents of the area, however, may be overwritten with safety. If PRINTMAKER has had to call another user subroutine before all the lines provided by the previous user subroutine have been printed, any such lines are preserved in temporary store. This can happen when a page change occurs and PHEAD is called, before all lines created by the previous entry to RECRD have been printed.

COMMON/WORKING/

This area of 72 words contains information for certain control functions. The layout of this area is as follows:

POINT	contains the address of the current input record in the MTH input buffer.
ENTRIES	contains the number of entries per page; this constant is used to reset the line count at every page change. Unless it is altered by the user in the START subroutine, it has the value 56.
LINECOUNT	contains the number of lines left on the current page; a page change occurs when this reaches zero.
COUNT	contains the number of input records read.
SPAC	contains the paper feed control character for parameter printing (when the subroutine START is called, this word contains #42).
PARAM (30)	contains the input and print buffer for parameters; parameters are read into the first 20 words (when the START subroutine is called this area is spacefilled).
CNTRL (4)	contains the parameter read control area; the type/mode word is set for either cards of paper tape depending on the original entry point.
PRIN2 (4)	contains the parameter print control area.

COMMON/BLOCK1/

This area contains information on keys. The layout of this area is as follows:

KENUM	contains the number of keys, maximum 6.
KEAREA	contains the numbers of words used by the keys.
TYPSEQ (6)	contains the type and sequencing of each key.
KEYSTO (6)	contains the length and start of each key when stored, in character/ counter-modifier form.
KEYA (6)	contains the length and start of each key in File A, in character/ counter-modifier form.
KEYB (6)	contains the length and start of each key in File B, in character/ counter-modifier form.
SWICHC	contains the file-open identifier: single = 7/0 double = 7/7
SWICHA	contains the File A identifier.
SWICHB	contains the File B identifier.
SWITB	is set to non-zero to indicate that binary keys require conversion on input.

User subroutines

START

This subroutine is called at the beginning of the run, before the program has opened the input file. At this point PRINTMAKER has read and stored the standard parameters, but has not released the peripheral from which they were read. This subroutine may be used to read any non-standard parameters which the user's program requires; this should be done by giving a PERI instruction or an I/O Generator statement referring to CR0 or TR0, according to whether the standard parameters were read from cards or paper tape. If the user gives a PERI instruction he can use a peripheral control area preset in the framework (see *Data areas*).

The entry line for this subroutine is:

```
CALL    0    START
```

On entry Accumulator 2 is set to zero by the framework. When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed.

The exit line is:

```
EXIT    0    0
```

This subroutine may be in dummy form, in which case the format is simply:

```
#CUE                START
          EXIT 0    0
```

RECRD

This subroutine is called after a record has been read from the input file, and after any necessary control breaks have been processed. The subroutine is used to process the current record.

On entry, Accumulator 1 and the area called POINT contain the address of Word 0 of the current record. Accumulator 2 contains zero.

The entry line for this subroutine is:

```
CALL    0    RECRD
```

The lines which are to be printed for every record should be set up here. The area of store available is COMMON/LINES/. The first line is stored in the 30 word area beginning at LINE0, preceded by its required print and space control character at SPAC0. The second is stored in SPAC1 and LINE1 and so on to SPAC9 and LINE9.

If records are being totalled, the appropriate fields from the record should be added to a totals area. The user may set up an area for totalling organised in any convenient manner.

When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed.

The exit line is:

```
EXIT    0    0
```

Some processing of records will always be required. This subroutine cannot therefore be a dummy.

LEVEL SUBROUTINES

These subroutines are used principally to process totals; the lowest level of total is formed by accumulation at record level performed by RECRD. LEVEL subroutines are written to accumulate totals to higher levels. There are six of these subroutines available, one for each of the six levels at which a control break can occur. They are called by the program when a control break is found. The lowest level is dealt with by LEVEL5 and the highest by LEVEL0. These subroutines will normally form the print lines required at a control break. The same entry conditions and exit conditions apply to all LEVEL subroutines;

The entry line is:

```
CALL    0    LEVELn
```

where *n* is a value in the range 0 to 5.

On entry, Accumulator 2 is set to zero by the framework.

Lines to be printed should be set up in COMMON/LINES/ as described under RECRD. Totals should be added to

the next highest level at the end of a subroutine, and the area of store used for the totals at the current level should be zeroized. If this is not done, the program will form running accumulative totals.

When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed.

The exit line is:

```
EXIT    0    0
```

Where fewer than six keys are in use, some of the LEVEL subroutines may be dummies. In such a case, PRINTMAKER considers that the lowest levels are omitted. If only four keys are being used, for example, the subroutines LEVEL4 and LEVEL5 may be written in dummy form. In the extreme case, where the program is being used to list records without any intermediate control breaks, all six LEVEL subroutines may be in dummy form.

PHEAD

This subroutine is called when the line-count is exhausted and a page change is needed. It can be used to throw to a new page and to provide headings for the new page.

The entry line is:

```
CALL    0    PHEAD
```

On entry, Accumulator 2 is set to zero by the framework and Accumulator 4 contains four characters giving the current page count, right justified and zero-suppressed.

Heading lines are stored in an area beginning at HEAD0; to be printed the lines have to be moved to the area beginning at LINE0. The first heading line printed must be given a print feed control character of # 51.

When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed, but in this case there will be no throw to head of form, so that printing will be continuous. Lines to be printed should be set up in COMMON/LINES/ as described under RECRD.

The exit line is:

```
EXIT    0    0
```

If no page headings are required the subroutine may be in dummy form. However, it must normally still cause a throw to the head of the next form. This can be done by setting a print feed control character of #11, to cause a throw to channel one without printing. The dummy form of this subroutine must therefore be:

```
#CUE          PHEAD
 LDN    4     #11
 STO    4     SPAC0
 LDN    2     1
 EXIT   0     0
```

PFOOT

This subroutine is called when the framework program has detected a page overflow condition or when the linecount is exhausted. It can be used to take any action which the user requires for printing carried-forward totals and so on.

The entry line is:

```
CALL    0    PFOOT
```

On entry, Accumulator 2 is set to zero by the framework.

When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed. Lines to be printed should be set up in COMMON/LINES/ as described under RECRD.

The exit line is:

```
EXIT    0    0
```

If page-foot processing is not required, the user may provide this subroutine in dummy form:

```
#CUE        PFOOT
    EXIT  0 0
```

USENT

This subroutine is called when the framework program detects a user tape sentinel, and may be employed for whatever action the user requires. No printing will be carried out, however, on return from this routine.

On entry, Words 10 and 11 of the program contain the normal MTH exception code and addresses.

On exit from the subroutine, the framework returns to MTH to read the next record.

The entry line is:

```
CALL    0    USENT
```

The exit line is:

```
EXIT    0    0
```

The dummy form is:

```
#CUE             USENT
    EXIT    0    0
```

CLOSE

This subroutine is called at the end of the run. It will normally be used to form print lines for any final totals required, and for any other end of run processing the user may need.

The entry line is:

```
CALL    0    CLOSE
```

On entry, Accumulator 2 is set to zero by the framework. When control is returned to the framework program, Accumulator 2 must contain the number of lines to be printed: this number must not be greater than 10. If it is zero no lines are printed. Lines to be printed should be set up in COMMON/LINES/ as described under RECRD.

The exit line is:

```
EXIT    0    0
```

This subroutine may be in dummy form:

```
#CUE             CLOSE
    EXIT    0    0
```

Extra subroutines available

The subroutines BLAH, SHUNT, and PRINT may be called from within a user subroutine to perform certain standard functions.

BLAH

BLAH can be used to spacefill a 30 word area. The call takes the form:

```
CALL    1    BLAH
             0/ADDRESS
```

where ADDRESS is the start address of the area to be spacefilled.

For example, BLAH could be used to spacefill the first print line thus:

```
CALL    1    BLAH
             0/LINE0
```

The program return location is CALL + 2. All accumulators are stored on entry and reinstated before exit. V is unchanged.

SHUNT

SHUNT can be used to move character strings from the current record area to the print line area. LINE0. The call takes the form:

```
CALL    1    SHUNT
             n1, n2, n3, n4, n5
```

where

n_1 = the number of characters to be moved (to a maximum of 127; if $n_1 = 0$, 128 characters will be moved)

n_2 = the word address of the source field
n_3 = the character address of the source field
} relative to the start of the record

n_4 = the word address of the destination field
n_5 = the character address of the destination field
} relative to the start of LINE0

For example, to move a string of 17 characters from a field that starts at word 163.2 of the input record, so that it will print in print positions 14 to 30:

```
CALL    1    SHUNT
             17,163,2,3,1
```

The program return location is CALL + 6. All accumulators are stored on entry and reinstated before exit. V is unchanged.

PRINT

PRINT is the routine used by the framework for all printing, and incorporates checks on the line count, page changing, and so on. It can also be called by user coding. The call takes the form:

```
LDN     2    A
CALL    0    PRINT
```

where A is the number of lines to be printed, and must not be greater than 10. The program produces an error halt if A exceeds this figure.

The program return location is CALL + 1. All accumulators are used; their original contents will be overwritten and not reinstated. V is unchanged.

Use of Switches

PRINTMAKER uses switches 10, 11 and 12. The user subroutines should therefore make no reference to these switches.

Totalling

PRINTMAKER does not provide automatically for adding the results of one series of totalling to the next total level on the occurrence of a control change, as the layout of the totalling area is controlled by the user. The user should code this himself. Similarly, the totalling area cannot be zeroized by the framework; the user should therefore provide his own coding for this function at start of run, and as required during the run.

Chapter 3 Input and output

INPUT

Input to the complete PRINTMAKER program consists of a magnetic tape file containing the data to be printed and parameters, punched on cards or paper tape.

Input file

The input file must observe the conventions of 1900 Series Magnetic Tape Housekeeping. It may be a single or multi-reel file.

For processing of control breaks, the file should be sorted according to the sequence of keys defined in the #KEYS parameter.

Parameters

The parameters are described in detail in Chapter 4.

OUTPUT

The output from the complete PRINTMAKER program consists of line printer output and console output.

Line printer output

PARAMETER LIST

The program will print out each parameter in full exactly as it appears in the card or on paper tape. The word WRONG is printed out against the listing of any parameter found to be incorrect.

If an error message nas appeared against any parameter in the listing the message at the end of listing is:

PARAMETERS INCORRECT

In addition, the printer will output:

PARAMETERS INCOMPLETE

if the parameter set is incomplete.

THE REPORT

The main output of the program is the report. The format of the report is prepared by the user.

At the end of the run the program prints a single line giving a record count.

Console output

See under *EXCEPTION CONDITIONS*, Chapter 6.

Chapter 4 The parameter set

Parameters may be input to PRINTMAKER punched either in cards or on paper tape. Although the following descriptions refer to cards, the format for paper tape is identical except that each parameter must be followed by a Newline character.

PRINTMAKER uses the following parameters:

- a #READ parameter
- a #KEYS parameter
- a #PAGE parameter
- a #HEAD parameter
- a #END parameter

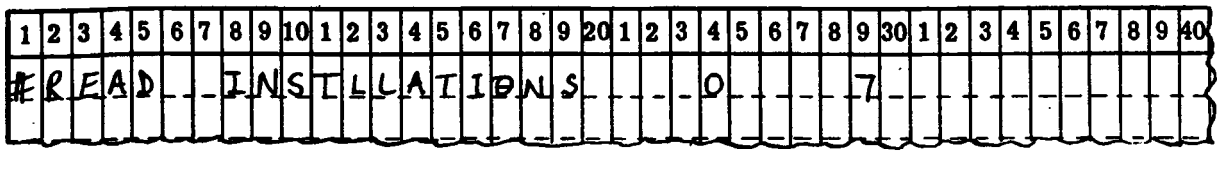
In addition a #RESTART parameter may be input to restart the program and complete a partly finished report.

READ TAPE LABEL PARAMETER (#READ)

This parameter is in standard form. Its function is to define the label on the input tape file, and the mode in which it is to be opened.

Field	*Columns*	*Contents*
A	1 to 5	The directive #READ
	6	Opening mode: 1 character as follows: 1 = do not check for absence or presence of write permit ring. 2 = check for absence of write permit ring. 3 = check for presence of write permit ring. ∇ (space) in this position is interpreted as 2. Note: These are equivalent to the standard modes of opening tapes, that is #100, #200 and #300. Details of additional checks by Executive on the 1904 and above will be found in the ICL manual *Magnetic Tape.*
B	8 to 19	12 character file name.
C	21 to 24	Reel sequence number: four digits.
D	26 to 29	File generation number: four digits.
E	31	Designation: one alphabetic character (not needed for this program).

Example



KEYS PARAMETER (#KEYS)

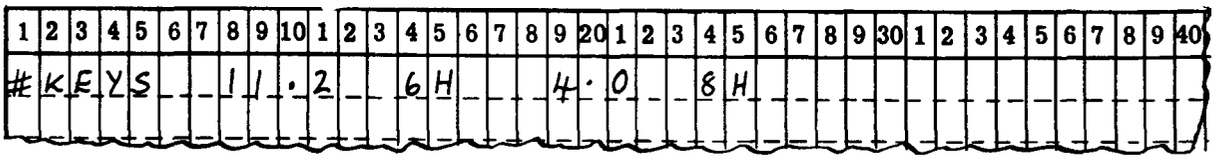
This parameter is in standard single key form. It is used to identify control breaks by indicating to the program where the keys are to be found on the record and may be omitted if no control breaks are required.

Field	*Columns*	*Contents*
A	1 to 5	The directive #KEYS
B	7 to 11	The start address of a key expressed in words and characters relative to the start of the tape record, in the form *NNN.N*
C	13 to 15	Field size: columns 13 and 14 contain a two digit number giving the length of the key. Column 15 contains a single character indicating the type of field: H = character ascending % = binary ascending D = character descending * = binary descending V = words The field size is expressed as a number of characters if Column 15 contains H or D and a number of words if it contains %, * or V.

Fields B and C may be repeated to give a maximum of six keys: the first key is the most significant

D	67	Designation: File identifier, a single alphabetic character (not needed for this program).

Example



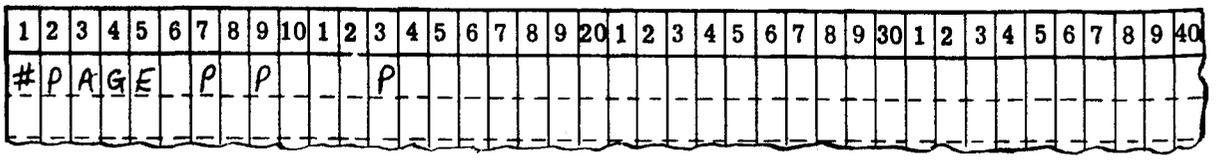
PAGE PARAMETER (#PAGE)

This parameter governs page changing on key breaks.

Field	*Columns*	*Contents*
A	1 to 5	The directive #PAGE
B	7	Page change control character: 0 or ∇ = no page change on this key; P = page change on this key

There may be up to six page changes, corresponding to the permissible maximum of six keys, and they will be indicated in columns 7, 9, 11, 13, 15 and 17.

Example



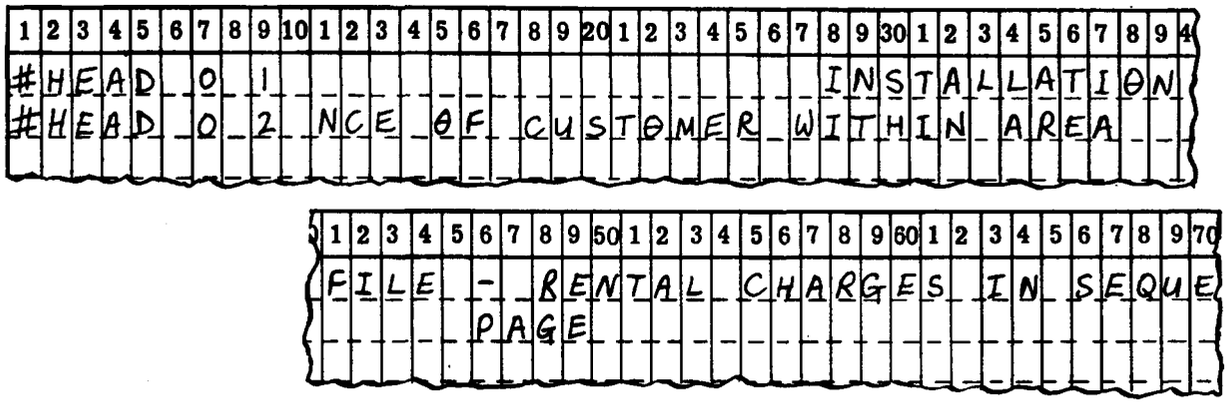
HEADINGS PARAMETERS (#HEAD)

These parameters are used if information about headings is to be given at run time. This will usually involve information that varies from run to run. Headings parameters can also be used to overwrite a pre-stored heading.

Headings parameters must always be used in pairs; the first parameter of a pair provides information for the left-hand half of a heading line, and the second provides information for the right-hand half of the line. A single half line may not be specified on its own.

Field	*Columns*	*Contents*
A	1 to 5	The directive #HEAD
B	7	Line number: a single integer in the range 0 to 9.
C	9	Sequence number: a single integer, either 1 or 2, indicating whether the card is the first or second for the line. The first card provides 60 characters of information for print positions 1 to 60 and the second card provides 60 characters for print positions 61 to 120.
D	11 to 70	Heading information: 60 characters of information which will be stored by the program in the appropriate line.

Example

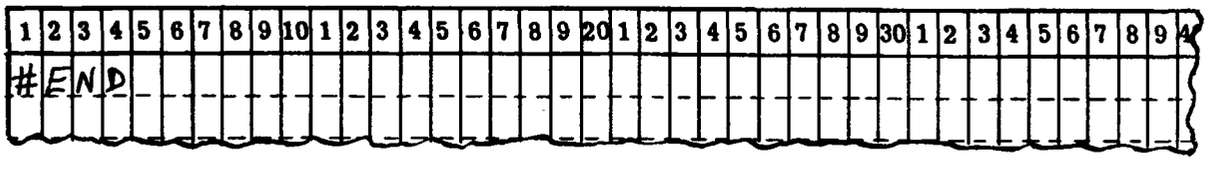


END OF PARAMETERS MARKER

This parameter is in standard form. It marks the end of the parameter set.

Field	*Columns*	*Contents*
A	1 to 4	The directive #END.

Example



RESTART PARAMETER (#RESTART)

Two classes of restart are available with PRINTMAKER, simple restarts and complex restarts.

For a *simple restart* the program reads the tape up to the specified restart point, but does not process or print the records read up to this point: it then processes and prints records from the restart point in the normal way. Totals are therefore counted only from the restart point.

For a *complex restart* the program reads and processes, but does not print, all records up to the restart point: it then processes and prints records from the restart point in the normal way. Totals are therefore counted from the beginning of the file.

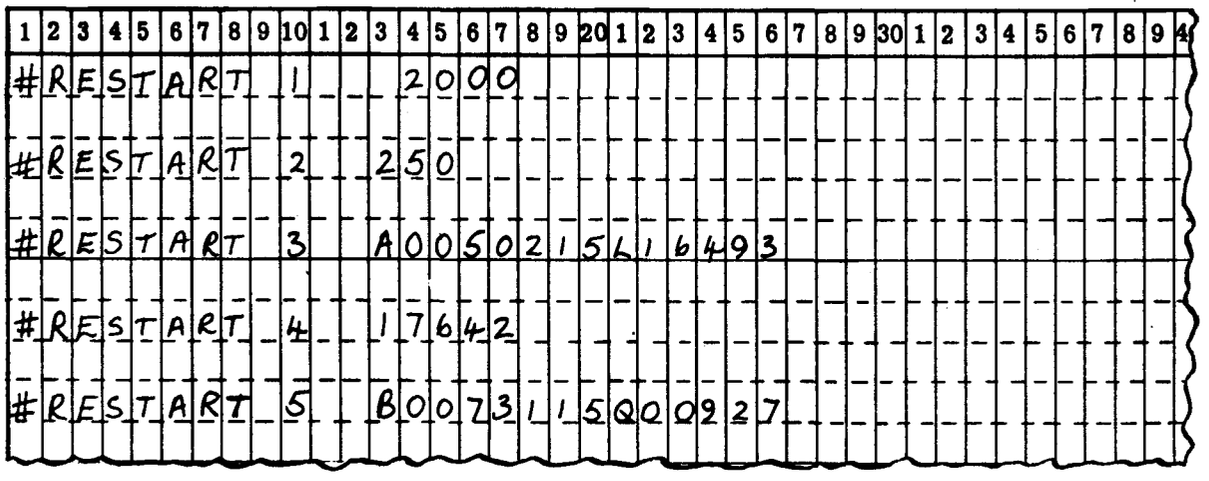
For all restarts, printing always begins at the top of a new page.

Field	*Columns*	*Contents*
A	1 to 8	The directive #RESTART.
B	10	Restart type: 1 character as follows:

Type	*Class*	*Restart point*
1	complex	Record number.
2	complex	Page number.
3	complex	Record with specified keys.
4	simple	Record number.
5	simple	Record with specified keys.

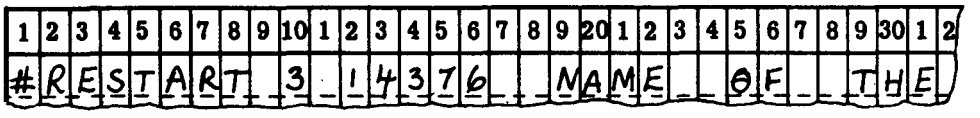
C	12 to 76	Restart point:	
		Restart types 1 & 4 :	record number in decimal starting in Column 12.
		Restart type 2 :	page number in decimal starting in Column 12.
		Restart types 3 & 5 :	Key in characters and/or decimal (see #KEYS parameter page 14).

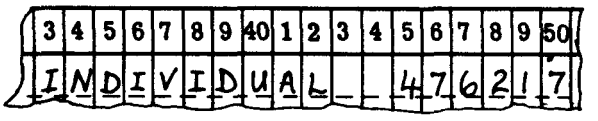
Examples



Note: if B is 3 or 5, C is the specified key in the sequence defined in the #KEYS parameter; character keys expressed in the number of characters specified in #KEYS, binary keys expressed in their decimal form and keys separated by one space from each other. If 65 characters are not sufficient to define the key a continuation parameter is used.

Example


1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1 2
#RESTART 3 14376 NAME OF THE


3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50
INDIVIDUAL 47621


-RESTART 3 1 17 674

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
WORD COUNT 15	MAKE	SERIAL		CUSTOMER					MNTHLY RENTAL	DATE INSTALLED		AREA		

MAKE, SERIAL, CUSTOMER and AREA are character fields
MONTHLY RENTAL is held in binary pence. Maximum value is £999-19s-11d
DATE INSTALLED is held in binary days

Figure 3 PRINTMAKER example: file formats

```
#READ INSTLLATIONS 0000 0007
#KEYS 011.2 06H 004.0 08H
#PAGE P
#HEAD 0 1 INSTALLATION FILE - RENTAL CHARGES IN SEQUE
#HEAD 0 2 NCE OF CUSTOMER WITHIN AREA PAGE
#END
```

Figure 4 PRINTMAKER example: parameters

Chapter 5 Example

In this example PRINTMAKER is used to create a general purpose listing and totalling program associated with a particular file. The program can be used to produce a list with totals from either the whole of a file or a selected subset. The program can do this with several versions of the file sorted into different orders.

The file in this example contains short records, all of the same format. Each record holds details of one machine installed at a customer's home or office. The machines are rented to customers on monthly terms. The record contains a make code and serial number to identify each machine. It also contains fields for the customer number, sales area, monthly rental and date installed to identify the current contract.

Certain control records may also be present; these are ignored by the printing and totalling functions of the program. The file formats are shown in Figure 3, page 18.

The #KEYS parameter specifies to the program the file sequence to be used for any particular run, and hence the points at which control breaks are to be used for intermediate totalling.

This example program is written to take two levels of total as well as final totals. Page changes can be made on each control break at the higher level by the insertion at run time of the #PAGE parameter, see Figure 4. If this parameter were omitted, printing would be continuous, completely filling the printing area on each page.

The program produces a standard printout, see Figure 5, with fields appearing in fixed columns. Each page is headed with three standard lines: the first of these consists of the date of run, title of run, and page number; the second and third contain column headings. As the column headings remain fixed for all possible runs of the program, these are preset by means of suitable data statements. The first line is likely to change for every run, and is therefore supplied to the program at run time by #HEAD parameters.

```
   28/02/69    INSTALLATION FILE - RENTAL CHARGES IN SEQUENCE OF CUSTOMER WITHIN AREA        PAGE   35

                                                  DATE          MONTHLY
 AREA       MAKE     SERIAL      CUSTOMER      INSTALLED        RENTAL

A00502      M196     00002497    15L16493      15/12/66          5 14  0
A00502      M213     00160042    15L16493      09/04/65         14  6  6
A00502      M213     00160103    15L16493      09/04/65         14  6  6
A00502      M567     00002548    15L16493      22/10/64        110 10  0

                                              MINOR TOTAL       144 17  0        COUNT      4

A00502      M196     00000382    15L00826      07/06/66          5 14  0
A00502      R729     06938801    15L00826      14/09/68         53 17  6

                                              MINOR TOTAL        59.11  6        COUNT      2

A00502      M213     00058629    15L00901      21/05/66         14 10  0
A00502      M213     00058630    15L00901      21/05/66         14 10  0
A00502      M213     00060012    15L00901      21/05/66         14 10  0
A00502      M213     00075328    15L00901      07/08/66         14 10  0
A00502      M213     00078211    15L00901      01/10/60         14 10  0
A00502      R646     00100546    15L00901      27/01/65        202 15  9
A00502      R646     00124102    15L00901      27/01/65        202 15  9
A00502      R646     00126026    15L00901      27/01/65        202 15  9

                                              MINOR TOTAL       680 17  3        COUNT      8

                                              MAJOR TOTAL       885  5  9        COUNT     14

                                               FILE TOTAL     54563 14 10        COUNT    952
```

Figure 5 PRINTMAKER example: specimen printout

A description of the coding follows. Where particular lines are referred to in these notes they are identified by the line-numbers printed by the compiler at the extreme left of the source-listing, see Figure 6, pages 21 to 23.

Line 1	The program will be called #MPTF.
Lines 2 and 3	These define for the user segment the structure of the relevant part of the framework's print buffer.
Lines 4 to 15	These define for the user segment the structure of the relevant part of the framework's headings storage area, and preset the details of the fixed column headings.
Lines 16 and 17	The user segment makes use of the common locations POINT and ENTRIES.
Lines 18 to 27	These define the user's own storage. Note that the user can arrange totalling areas in any convenient way.
Line 28	This instruction is never obeyed, but serves to bring in the framework for consolidation with the user segment.
Lines 30 to 38	The START of run subroutine. This initializes the user's totalling area, obtains the date of run in character form from Executive, and adjusts the printing depth so that only 40 lines will be printed on each page, instead of the standard 56.
Lines 39 to 59	The user's PAGE HEADing subroutine. This uses the page-number provided by the framework. It transfers the three heading lines to the print buffer, setting appropriate print feed control characters. A fourth print feed control character is also set up, to cause a blank line to be spaced after the second line of column headings. On exit the user instructs the framework to print four lines.
Lines 60 to 95	The user's RECRD subroutine to process each record. This starts with an escape procedure, which is used if the record is a control record; since Accumulator 2 contains zero on entry, this exit causes no printing to follow from the control record. Note particularly throughout this subroutine the use of Accumulator 1 to address the record. Each time that this accumulator has been used for any other purpose the address of the record is re-set in it from the standard location POINT. The framework's standard subroutine SHUNT is used to transfer character fields to the print-line. The two binary fields are converted to characters by the standard utility subroutines BDCN and CBINSTERDL1. The rental for the current record is added to the total for the current batch of records, and 1 is added to the count of machines in the current batch. The framework is instructed to print one line, with single spacing.
Lines 96 to 110	The user's LEVEL subroutine for the lowest level of totalling used. The totals formed from the previous batch of records are added to the next level, converted for printing, and then zeroized. A slave suroutine, LINE, is used to perform the detail of print distribution. Parameters inform this subroutine of the addresses of the totals to be converted and distributed, and of the appropriate caption for the current level of total.
Lines 111 to 125	Similar to the previous subroutine, but operating at the next level.
Lines 126 to 133	The user's CLOSE of run subroutine. This is similar to the LEVEL subroutines, but does not need to include any zeroizing, as no further accumulation can take place.
Lines 134 to 161	The user's slave subroutine to perform the detail of print distribution for total lines, at all levels. Note the use of the framework's standard subroutine BLAH, to spacefill the print line before print assembly.
Lines 162 to 168	This is a convenient way of writing a composite dummy subroutine for all those functions which the user does not require.

```
0000    #STEER          LIST

         18/47/45        26/02/69        COMPILED BY XPLG 25F


0001    #PROGRAM        MPTF80/USERSEGMENT  [MJK 11URR
0002    #LOWER          COMMON/LINES/
0003                    PFCC0,LINE0(30),PFCC1,LINE1(30),PFCC2,LINE2(30),PFCC3
0004    #LOWER          COMMON/HEADINGS/
0005    #
0006    #     FIRST HEADING LINE IS TO BE INPUT VIA RUN-TIME PARAMETERS
0007    #      REST ARE PRESET
0008    #
0009                    HEAD0(30)
                                                                R2= HEADINGS
0010    HEAD1           40H                                                 30  #20202020
                                                                            31  #20202020
                                                                            32  #20202020
                                                                            33  #20202020
                                                                            34  #20202020
                                                                            35  #20202020
                                                                            36  #20202020
                                                                            37  #20202020
                                                                            38  #20202020
                                                                            39  #20202020
0011                    40H        DATE          MONTHLY                    40  #20202020
                                                                            41  #20202020
                                                                            42  #20444164
                                                                            43  #45202020
                                                                            44  #20202020
                                                                            45  #20202020
                                                                            46  #20555756
                                                                            47  #64505471
                                                                            48  #20202020
                                                                            49  #20202020
0012                    40H                                                 50  #20202020
                                                                            51  #20202020
                                                                            52  #20202020
                                                                            53  #20202020
                                                                            54  #20202020
                                                                            55  #20202020
                                                                            56  #20202020
                                                                            57  #20202020
                                                                            58  #20202020
                                                                            59  #20202020
0013    HEAD2           40H  AREA     MAKE     SERIAL     CUSTOMER          60  #20204162
                                                                            61  #45412020
                                                                            62  #20202020
                                                                            63  #20554153
                                                                            64  #45202020
                                                                            65  #20206345
                                                                            66  #62514154
                                                                            67  #20202020
                                                                            68  #20436563
                                                                            69  #64575545
0014                    40H     INSTALLED        RENTAL                     70  #62202020
                                                                            71  #20202051
                                                                            72  #56636441
                                                                            73  #54544544
                                                                            74  #20202020
                                                                            75  #20202020
                                                                            76  #20206245
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                                                            77  #56644154
                                                                            78  #20202020
                                                                            79  #20202020
0015                    40H                                                 80  #20202020
                                                                            81  #20202020
                                                                            82  #20202020
                                                                            83  #20202020
                                                                            84  #20202020
                                                                            85  #20202020
                                                                            86  #20202020
                                                                            87  #20202020
                                                                            88  #20202020
                                                                            89  #20202020
0016    #LOWER          COMMON/WORKING/
0017                    POINT,ENTRIES
0018    #LOWER
0019                    FLTOT(3)          [TOTAL AREA FOR WHOLE FILE
0020                    L0TOT(3)          [TOTAL AREA FOR LEVEL 0
0021                    L1TOT(3)          [TOTAL AREA FOR LEVEL 1
0022                    LINK,HANDY,LNLINK [WORKING STORAGE AREAS
0023    #LOWER
0024    CAP1            12HMINOR TOTAL                                       0  #55515657
                                                                             1  #62206457
                                                                             2  #64415420
0025    CAP2            12HMAJOR TOTAL                                       3  #55415257
                                                                             4  #62206457
                                                                             5  #64415420
0026    CAP3            12H FILE TOTAL                                       6  #20465154
                                                                             7  #45206457
                                                                             8  #64415420
0027    CAP4            8HCOUNT                                              9  #43576556
                                                                            10  #64202020
0028           CALL  0  PRINTMAKER          [DUMMY TO BRING IN FRAMEWORK    11  070 0        0 BA
0029    #PROGRAM
0030    #CUE            START               [START OF RUN ROUTINE
0031           LDCT  2  9                                                    0  124 2 0      9
0032    ST1    STOZ     FLTOT(2)            [ZEROISE TOTALLING AREA          1  033 0 2      0 LV
0033           BUX   2  ST1                                                  2  060 2        1 PR
0034           GIVE  4  1                   [GET DATE FROM EXEC...           3  165 4 0      1
                                                                             4  010 5 0      2 R2
                                                                             5  010 4 0      1 R2
0035           STO  45  HEAD0+1             [...AND STORE IN TOP HEADING LINE
0036           LDN   4  40                                                   6  100 4 0     40
                                                                R2= WORKING
```

Figure 6 PRINTMAKER example: compilation listing

```
0037          STO   4  ENTRIES          [SET COUNT TO 40 LINES PER PAGE                    7  010 4 0     1 R2
0038          EXIT  0  0                [RETURN TO FRAMEWORK                               8  072 0       0
0039    #CUE           PHEAD            [ROUTINE TO SET UP PAGE HEADINGS
0040          STO   0  LINK             [PRESERVE LINK                                     9  010 0 0     9 LV
                                                                        R2= HEADINGS
0041          STO   4  HEAD0+25         [INSERT PAGE NO. IN TOP HEADING                   10  010 4 0    25 R2
0042          LDCT  7  3                                                                  11  124 7 0     3
0043          LDN   4  HEAD0                                                              12  100 4 0     0 R2
                                                                        R2= LINES
0044          LDN   5  LINE0                                                              13  100 5 0     1 R2
0045    PH1   MOVE  4  30               [MOVE 3 HEADINGS INTO PRINT LINES                 14  126 4 0    30
0046          ADN   4  30                                                                 15  101 4 0    30
0047          ADN   5  31                                                                 16  101 5 0    31
0048          BUX   7  PH1                                                                17  060 7      14 PR
0049          LDN   6  #51                                                                18  100 6 0    41
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0050          STO   6  PFCC0            [SET UP SPACING CONTROL FOR ALL...                19  010 6 0     0 R2
0051          LDN   6  #42              [...HEADING LINES                                 20  100 6 0    34
0052          STO   6  PFCC1                                                              21  010 6 0    31 R2
0053          LDN   6  #41                                                                22  100 6 0    33
0054          STO   6  PFCC2                                                              23  010 6 0    62 R2
0055          LDN   6  #01              [SPACE WITHOUT PRINTING - TO ...                  24  100 6 0     1
0056          STO   6  PFCC3            [..SEPERATE HEADINGS FROM LISTING                 25  010 6 0    93 R2
0057          LDN   2  4                [LOAD NUMBER OF PRINT LINES                       26  100 2 0     4
0058          LDX   0  LINK                                                               27  000 0 0     9 LV
0059          EXIT  0  0                [RETURN TO PRINT 4 LINES                          28  072 0       0
0060    #CUE           RECRD            [ROUTINE TO PROCESS EACH RECORD
0061          STO   0  LINK                                                               29  010 0 0     9 LV
0062          LDX   4  6(1)             [X1=ADDRESS OF 1ST WD OF RECORD                   30  000 4 1     6
                                                                                          31  026 4 0     0 LT
                                                                                          32  074 5       0 BA
0063          BXU   4  '4H****',RE1
0064          EXIT  0  0                [IGNORE CONTROL RECORDS  WD6=****)                33  072 0       0
0065    RE1   CALL  1  BLAH             [SPACEFILL 1ST PRINT LINE                         34  070 1       0 BA
0066                 0/LINE0                                                              35  #00000001     R2
0067          CALL  1  SHUNT            [MOVE AREA NO. TO PRINT LINE...                   36  070 1       0 BA
0068                 6,11,2,0,1         [..16CHS FROM 11.2 TO LINE0+0.1 )                 37  #00000006
                                                                                          38  #00000013
                                                                                          39  #00000002
                                                                                          40  #00000000
                                                                                          41  #00000001
0069          CALL  1  SHUNT            [MAKE CODE TO PRINT LINE                          42  070 1       0 BA
0070                 4,1,0,3,1                                                            43  #00000004
                                                                                          44  #00000001
                                                                                          45  #00000000
                                                                                          46  #00000003
                                                                                          47  #00000001
0071          CALL  1  SHUNT            [SERIAL NO. TO PRINT LINE                         48  070 1       0 BA
0072                 8,2,0,5,1                                                            49  #00000010
                                                                                          50  #00000002
                                                                                          51  #00000000
                                                                                          52  #00000005
                                                                                          53  #00000001
0073          CALL  1  SHUNT            [CUSTOMER NO. TO PRINT LINE                       54  070 1       0 BA
0074                 8,4,0,8,1                                                            55  #00000010
                                                                                          56  #00000004
                                                                                          57  #00000000
                                                                                          58  #00000010
                                                                                          59  #00000001
                                                                        R2= WORKING
0075          LDX   1  POINT            [REINSTATE POINTER                                60  000 1 0     0 R2
0076          LDN   4  10 (1)           [GET ADDRESS OF INSTALLED DATE...                 61  100 4 1    10
0077          STO   4  HANDY            [...(WD10 OF REC )                                62  010 4 0    10 LV
0078          CALL  1  BDCN             [CONVERT INSTALLED DATE TO CHARS..                63  070 1       0 BA
0079          LDX   3  HANDY            [...AND PUT INTO PRINT LINE                       64  000 3 0    10 LV
0080          LDX   3  '0/LINE0+11.3'                                                     65  000 3 0     1 LT
0081          LDX   1  POINT                                                              66  000 1 0     0 R2
0082          LDX   7  9 (1 )           [GET RENTAL (WD9 OF RECORD)                       67  000 7 1     9
0083          LDN   6  0                                                                  68  100 6 0     0
                                                                                          69  015 7 0     8 LV
                                                                                          70  011 6 0     7 LV
0084          ADS   67 L1TOT+1          [TOTAL RENTAL FOR 1ST LEVEL
0085          LDX   2  '7/LINE0+15.3'                                                     71  000 2 0     3 LT
0086          CALL  1  CBINSTERDL1+1    [CONVERT RENT TO CHARS & PUT IN...                72  070 1       1 BA
0087                 #20                [..PRINT LINE                                     73  #00000020
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0088          CALL  0  LSDSPACE                                                           74  070 0       0 BA
0089          LDN   5  1                                                                  75  100 5 0     1
0090          ADS   5  L1TOT            [COUNT NO. OF RECORDS                             76  011 5 0     6 LV
0091          LDN   4  #41                                                                77  100 4 0    33
                                                                        R2= LINES
0092          STO   4  PFCC0            [SET UP PAPER FEED CONTROL                        78  010 4 0     0 R2
0093          LDX   0  LINK                                                               79  000 0 0     9 LV
0094          LDN   2  1                                                                  80  100 2 0     1
0095          EXIT  0  0                [RETURN TO PRINT 1 LINE                           81  072 0       0
0096    #CUE           LEVEL1           [LOW LEVEL TOTALS ROUTINE
0097          STO   0  LINK             [PRESERVE LINK ADDRESS                            82  010 0 0     0 LV
0098          LDX   4  L1TOT                                                              83  000 4 0     6 LV
0099          ADS   4  L0TOT            [ADD TOTALS INTO NEXT LEVEL AREA                  84  011 4 0     3 LV
                                                                                          85  000 5 0     8 LV
                                                                                          86  000 4 0     7 LV
0100          LDX   45 L1TOT+1
                                                                                          87  015 5 0     5 LV
                                                                                          88  011 4 0     4 LV
0101          ADS   45 L0TOT+1
0102          CALL  3  LINE             [SET UP TOTALS LINE                               89  070 3       0 BA
0103                 0/CAP1                                                               90  #00000000     LP
0104                 0/L1TOT                                                              91  #00000006     LV
0105          STOZ     L1TOT            [ZEROISE TOTALS FOR THIS LEVEL                    92  033 0 0     6 LV
0106          STOZ     L1TOT+1                                                           93  033 0 0     7 LV
0107          STOZ     L1TOT+2                                                           94  033 0 0     8 LV
0108          LDN   2  1                                                                  95  100 2 0     1
0109          LDX   0  LINK                                                               96  000 0 0     9 LV
0110          EXIT  0  0                [EXIT TO PRINT ONE LINE                           97  072 0       0
0111    #CUE           LEVEL0           [HIGHEST LEVEL TOTALS ROUTINE
0112          STO   0  LINK                                                               98  010 0 0     9 LV
```

Figure 6 continued

```
0113          LDX   4  LOTOT              [ADD UP OVERALL TOTALS                  99  000 4 0     3 LV
0114          ADS   4  FLTOT                                                      100  011 4 0     0 LV
                                                                                  101  000 5 0     5 LV
                                                                                  102  000 4 0     4 LV
0115          LDX   45 LOTOT+1
                                                                                  103  015 5 0     2 LV
                                                                                  104  011 4 0     1 LV
0116          ADS   45 FLTOT+1
0117          CALL  3  LINE               [SET UP TOTAL PRINT LINE                105  070 3       0 BA
0118                   0/CAP2                                                     106  #00000003      LP
0119                   0/LOTOT                                                    107  #00000003      LV
0120          STOZ     LOTOT              [ZEROISE TOTALS FOR THIS LEVEL          108  033 0 0     3 LV
0121          STOZ     LOTOT+1                                                    109  033 0 0     4 LV
0122          STOZ     LOTOT+2                                                    110  033 0 0     5 LV
0123          LDN   2  1                                                         111  100 2 0     1
0124          LDX   0  LINK                                                      112  000 0 0     9 LV
0125          EXIT  0  0                  [RETURN TO PRINT ONE LINE              113  072 0       0
0126    #CUE            CLOSE             [END OF FILE ROUTINE
0127          STO   0  LINK                                                      114  010 0 0     9 LV
0128          CALL  3  LINE               [SET UP FINAL TOTAL LINE               115  070 3       0 BA
0129                   0/CAP3                                                     116  #00000006      LP
0130                   0/FLTOT                                                    117  #00000000      LV
0131          LDN   2  1                                                         118  100 2 0     1
0132          LDX   0  LINK                                                      119  000 0 0     9 LV
0133          EXIT  0  0                  [RETURN TO PRINT ONE LINE              120  072 0       0
0134    LINE  STO   3  LNLINK             [ROUTINE TO SET UP TOTAL LINE          121  010 3 0    11 LV
0135          CALL  1  BLAH               [SPACEFILL 1ST PRINT LINE              122  070 1       0 BA
0136                   0/LINE0                                                    123  #00000001      R2
0137          LDX   3  LNLINK                                                    124  000 3 0    11 LV
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0138          LDX   4  0(3)               [GET ADDRESS OF TOTAL CAPTION          125  000 4 3    .0
0139          LDN   5  LINE0+11                                                  126  100 5 0    12 R2
0140          MOVE  4  3                  [MOVE CAPTION TO PRINT LINE            127  126 4 0     3
0141          LDCT  2  5                                                         128  124 2 0     5
0142    LN1   LDCH  4  CAP4(2)                                                   129  024 4 2     9 LP
0143          DCH   4  LINE0+20(2)        [MOVE 2ND CAPTION TO PRINT LINE        130  034 4 2    21 R2
0144          BCHX  2  LN1                                                       131  064 2      129 PR
0145          LDX   2  1(3)               [GET ADDRESS OF TOTALS AREA            132  000 2 3     1
0146          LDX   6  0(2)                                                      133  000 6 2     0
0147          LDX      '4/LINE0+21.2'                                          134  000 2 0     5 LT
0148          CALL  0  CBINDECSL          [RECORD COUNT TO CHARS                 135  070 0       0 BA
0149                   #20                                                       136  #00000020
0150          LDX   3  LNLINK                                                    137  000 3 0    11 LV
0151          LDX   2  1(3)                                                      138  000 2 3     1
0152          ADN   2  1                                                         139  101 2 0     1
                                                                                 140  000 7 2     1
                                                                                 141  000 6 2     0
0153          LDX   67 0(2)
0154          LDX   2  '11/LINE0+14.3'                                         142  000 2 0     7 LT
0155          CALL  1  CBINSTERDL1+1      [RENTAL TOTAL TO CHARS                 143  070 1       1 BA
0156                   #20                                                       144  #00000020
0157          CALL  0  LSDSPACE                                                  145  070 0       0 BA
0158          LDN   4  #42                                                       146  100 4 0    34
0159          STO   4  PFCC0                                                     147  010 4 0     0 R2
0160          LDX   3  LNLINK                                                    148  000 3 0    11 LV
0161          EXIT  3  2                                                         149  072 3       2
0162    #CUE            LEVEL5            [NONE OF THESE ROUTINES ARE
0163    #CUE            LEVEL4            [ REQUIRED ...
0164    #CUE            LEVEL3            [ DUMMY EXIT FOR ALL OF THEM
0165    #CUE            LEVEL2
0166    #CUE            PFOOT
0167    #CUE            USENT
0168          EXIT  0  0                                                         150  072 0       0
0169    #END
```

Figure 6 continued

Chapter 6 Operating instructions and exception conditions

The operating instructions for the run in which the PRINTMAKER program is compiled will depend on the compiler being used, and will be standard for that compiler; see *PLAN Reference Manual.*

The following operating instructions are for running a compiled PRINTMAKER program.

OPERATING INSTRUCTIONS

	Action	*Message*
1	Load the input file without a write permit ring.	
2	Load the compiled program by inputting:	FI*#name* #TAPE
3	Load parameters in appropriate reader	
4	To read in the parameters either	
	(a) if the parameters are on paper tape input:	GO *#name* 20
	or	
	(b) if the parameters are on cards input:	GO *#name* 21
	The program reads the parameters, opens the tape, reads tape and prints as required by the user subroutines. At the end of the run the program outputs the message:	HALTED:– END OF RUN

Additional facilities

TO RE-ALIGN THE PAPER OR LOAD MORE PAPER

	Action	*Message*
1	Suspend the program by inputting:	SU *#name*
2	Set switch 11 by inputting:	ON *#name* 11
3	Start the program by inputting:	GO *#name*
	The program will stop at the foot of the current page and output the message:	HALTED:–RE-POSITION PAPER
4	The paper can then be re-aligned. Restart the program by inputting:	GO *#name*
	The program will halt again at the foot of the next page.	
5	Once the paper is correctly aligned input:	OFF *#name* 11

TO STOP IN THE MIDDLE OF A RUN

	Action	*Message*
1	Suspend the program by inputting:	SU *#name*
2	Set switch 11 by inputting:	ON *#name* 11
3	Start the program by inputting:	GO *#name*
	The program will halt at the foot of the current page and output the message:	HALTED:– RE-POSITION PAPER

	Action	*Message*
4	Restart the program by inputting:	GO *#name* 26
	The program rewinds and closes the input tape, prints all current totals, and outputs the message:	DISPLAY:–CURRENT PAGE NUMBER *nnnn* DISPLAY:– COUNT OF TAPE RECORDS READ *nnnn* HALTED:– RUN POSTPONED

RESTARTS

	Action	*Message*
1	Reload the compiled program by inputting:	FI *#name* #TAPE
2	Load the input tape without a write permit ring.	
3	Load a restart parameter (see page 15) followed by normal parameters on a card reader. *All parameters must be punched on cards.*	
4	Start the program by inputting:	GO *#name* 28

EXCEPTION CONDITIONS

Message	*Reason*	*Action*
HALTED:–PARAMETERS INCORRECT	Parameters are incorrect.	Correct the parameters and GO *#name* 20 or 21.
HALTED:–INCORRECT RESTART PARAMETER	Restart parameter incorrect or missing when run started with GO *#name* 28	Either (a) Correct or insert RESTART parameter and GO *#name* 28 or (b) restart with GO *#name* 20 or 21.
HALTED:–PARAMETERS INCOMPLETE	Parameter set is incomplete	Complete parameter set and GO *#name* 20 or 21.
HALTED:–INPUT TAPE EXCEPTION CONDITION 01	A long block has been found on the input file.	Abandon the run.
HALTED:–INPUT TAPE EXCEPTION CONDITION 04	A parity failure has occurred on the input file.	Abandon the run.
HALTED:–RE-POSITION PAPER	Operator intervention has suspended the program, because the paper is out of alignment (see above).	Re-align the paper and GO *#name*
HALTED:– UE	A user subroutine has attempted to print more than 10 lines at once.	This is a final halt.
HALTED:–LOAD MORE PAPER	More paper is needed.	Load more paper and GO *#name*

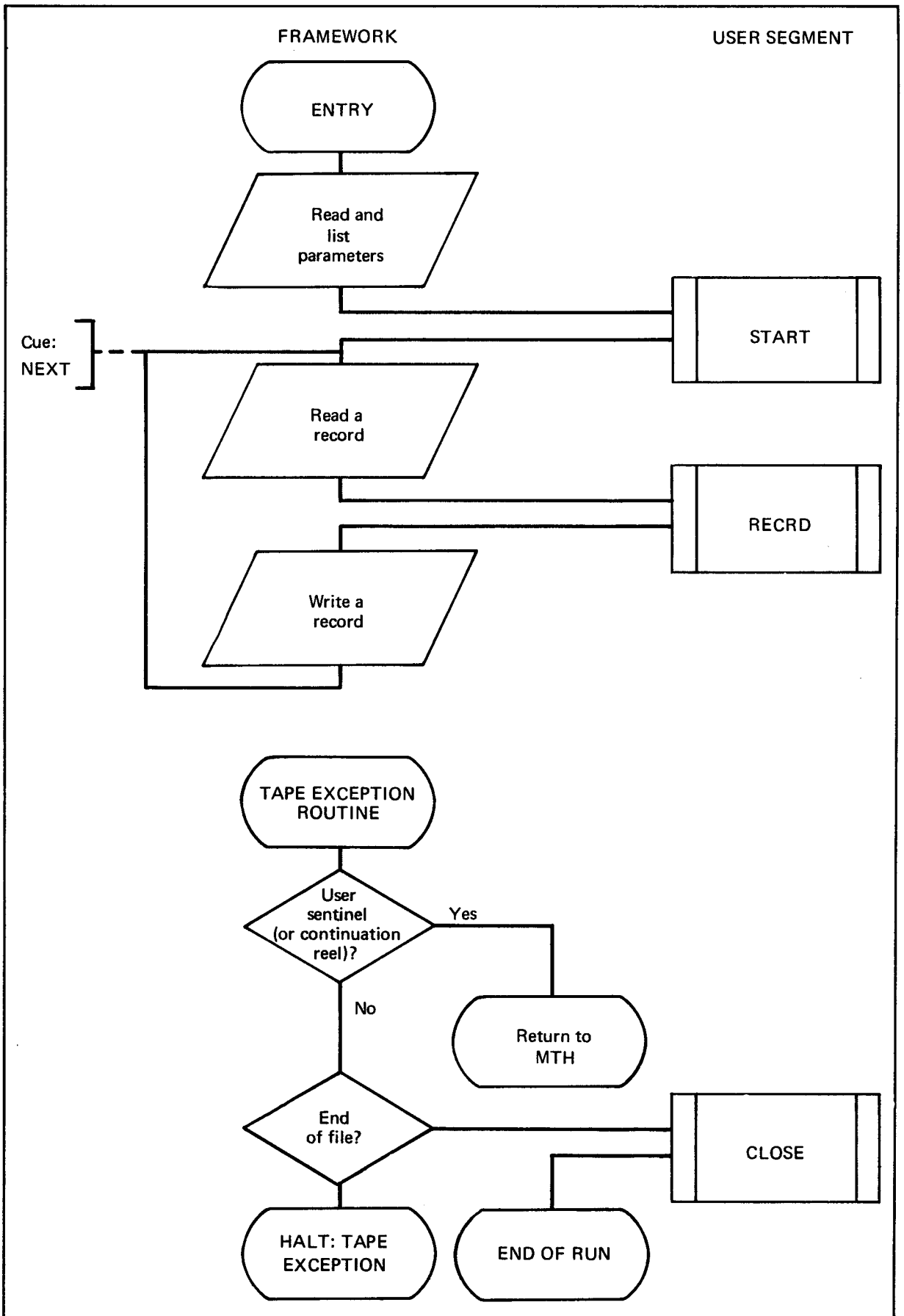


Figure 7 SUBEDIT outline flowchart

Chapter 7 Introduction to SUBEDIT

SUBEDIT consists of a group of routines in the form of a skeleton program which the user combines with a segment, or segments of his own coding to form a complete editing program.

The complete editing program can be used to edit and adjust the format of records, held on a magnetic tape file. It can also be used to filter off a subset of records. While the user's coding is concerned mainly with the processing of the data, the SUBEDIT routines deal with the standard functions of opening the input and output tapes and reading and writing the records.

The following table provides a summary of the functions performed by the respective routines.

	Framework routines	*User routines*
1	Read run time parameters	
		A Start of run procedures
2	Release slow peripherals and open input file.	
3	Read next record and pass to user to:	
		B Process record as required and then go on to step 4 (to output the record) or directly back to step 3 (to bypass the record's output).
4	Write the edited record to the output file and return to read the next record.	
5	At the end of the file:	
		C End of run procedures
6	Close input and output files. Output counts of records read and written.	

PROCESSING SUMMARY

Consolidation phase

The SUBEDIT framework is held in the program library as a free-standing segment. The user forms his own subroutines into a segment or segments and includes a reference to SUBEDIT. The user's segment is then submitted to an appropriate PLAN 3 compiler preceded by a steering line defining the user's steering requirements in the usual way. The reference to SUBEDIT is treated as a call to a library subroutine and ensures that the SUBEDIT framework is extracted from the library for consolidation with the user's segment or segments. SUBEDIT is thus consolidated in the same way as a library subroutine. SUBEDIT can therefore be supplied in semicompiled form on the library tape and need not be re-compiled each time it is needed.

The output from this stage is an object version of the SUBEDIT program, consisting of the user's segment or segments, the SUBEDIT framework and any other library routines. This program has the name that was given to the user's segment. The object program will be output on the normal output medium of the compiler that has been used. If the user's steering line has included LIST, SHORTLIST or FULLIST, a list of the appropriate form will also be output by the compiler. This will show only the user's segment. Listing of the SUBEDIT framework is suppressed as with any other library routine.

Parameter processing

At the start of the run, the SUBEDIT program reads and lists the parameters.

It then calls the user subroutine START for any preliminary processing that may be required. SUBEDIT initializes its own working areas; totalling areas must be set and initialized by the user.

The user may provide a SUBEDIT program with additional parameters containing other control factors of his own choice. He may design such parameters in any convenient form. When the START subroutine is called SUBEDIT has not yet released the peripheral from which the standard parameters were read. The user may therefore add his own parameters to the back of the standard set and read them in this subroutine with any suitable PERI or I/O Generator statement.

Main processing

Each record is read in turn from the input file and the user's subroutine RECRD is called to perform any processing required on the record.

The user must provide the following subroutines (described fully in Chapter 2):

1 START to perform start of run processing.

2 RECRD to process the current record.

3 CLOSE to implement end of run processing.

All the subroutines listed *must* be provided; if the user does not wish to use a particular facility he should substitute a *dummy*, that is, a subroutine consisting of one instruction: an EXIT to the main SUBEDIT program.

When the current record has been edited by RECRD it may be written to the output file and the program returns to read the next record.

At the end of the input file the user subroutine CLOSE is called. This subroutine performs any end of file processing required by the user. On return from this subroutine the output file is closed and counts of the number of records read and processed are printed.

MINIMUM CONFIGURATION

The minimum computer configuration required for SUBEDIT is:

1 central processor with 8K words of core store

1 card or paper tape reader

1 line printer

2 magnetic tape decks

Note: The core size required by the final program will depend on the size of the user's segment. Unusually complex SUBEDIT programs may therefore require more than 8K.

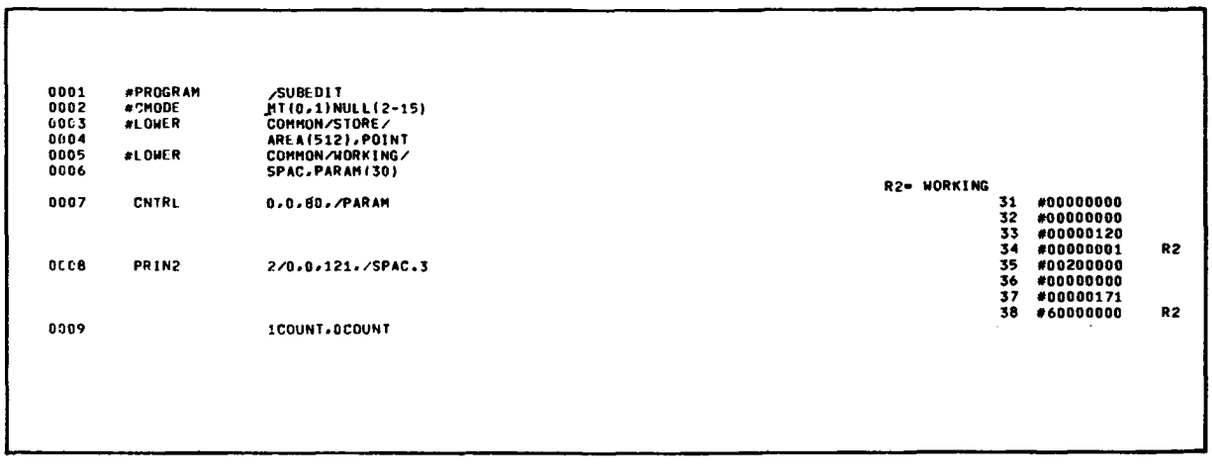


Figure 8 Layout of common areas in SUBEDIT

Chapter 8 User coding

FORM OF THE USER SEGMENT

The user's segment contains a number of subroutines each introduced by a #CUE line. The segment starts with a #PROGRAM directive, containing a program name and priority number. The program name supplied by the user is used as the name for the complete edit program.

The segment should include data statements, see below, under #LOWER COMMON for any standard locations which the user requires to access.

In addition, the segment may contain data statements to set up any other store locations which the user's subroutines may require. The layout and naming of these locations is entirely under the user's control.

The segment must contain at some point a call to SUBEDIT. This may take the form of a BRN instruction as follows:

```
BRN        SUBEDIT
```

Alternatively, a CALL instruction may be used:

```
CALL   0   SUBEDIT
```

Where a CALL is used for this purpose the link accumulator specified is immaterial. Note: this CALL or BRN must *not* be obeyed. Its sole function is to instruct the compiler to bring in the SUBEDIT framework. The cue name SUBEDIT must not be used in the user's segment. The compiler will regard any branch to it as a branch unsatisfied within that segment, in the same way as other branches to library software. Similarly, the name of the SUBEDIT framework subroutine SHUNT see page 33, must not appear as a cue name in the user segment.

Data areas

The following names are used for areas in LOWER COMMON storage, provided by the SUBEDIT framework, and used by the framework. Since the user will also use these areas he should include suitable data statements in his own segment, using the same common area names and observing the same layout for fields within the areas.

COMMON/STORE

The user must define under COMMON/STORE/ as many of the areas as he requires but the following two areas *must* be defined:

AREA (512) is the area from which the framework writes completed output records. The content of this area is entirely controlled by the user. The first word of the area must be set by the user to the number of words required in the output record.

POINT contains the address in the input buffer of the current input record. It is set by the framework before entry to RECRD. As the framework makes no further use of this location, it may be left unchanged by the user or altered in any way required. Any alteration to the content of this location will not affect the way in which the framework reads the next record.

COMMON/WORKING/

Certain of the framework's buffer and peripheral control areas are available to the user under COMMON/WORKING/. The layout of this area is as follows:

SPAC contains the line printer paper feed control character (when the subroutine START is called, this word contains #42).

PARAM (30) contains the input and print buffer for parameters; parameters are read into the first 20 words (when the subroutine START is called, this area is space filled).

CNTRL (4) contains the parameter read control area; the type/mode word is set for either cards or paper tape depending on the original entry point.

PRIN2 (4) contains the parameter print control area.

ICOUNT contains a count of input records.

OCOUNT contains a count of output records.

This information is listed in Figure 8.

User subroutines

START

This subroutine is called at the beginning of the run, as soon as the program has opened the input file but before the first record is read. At this point SUBEDIT has read and stored the standard parameters, but has not released the peripheral from which they were read. This subroutine may be used to read any non-standard parameters which the user's program requires; this should be done by giving a PERI instruction or an I/O Generator statement referring to CR0 or TR0, according to whether the standard parameters were read from cards or paper tape. If the user gives a PERI instruction, he can use a peripheral control area preset in the framework (see *Data areas*).

The entry line for this subroutine is:

```
CALL    0    START
```

The exit line is:

```
EXIT    0    0
```

This subroutine may be in dummy form, in which case the format is simply:

```
#CUE          START
    EXIT 0    0
```

RECRD

This subroutine is called after a record has been read from the input file and is used to process the current record.

The entry line for this subroutine is:

```
CALL    0    RCRD
```

On entry, Accumulator 0 contains the link for return to the main program. If the contents of this accumulator are likely to be lost by the entry of any user programming, or by any library subroutine called during that programming, they should be stored.

Accumulator 1 contains the address of Word 0 of the current input record. The address of Word 0 is also stored in a location called POINT, defined as Word 512 of COMMON/STORE/, from which it may be retrieved if the contents of Accumulator 1 are lost.

There are two ways in which the user may return control to the framework. If he uses an exit, the framework writes a record from the first 512 (0-511) words of COMMON/STORE/ which have the data name AREA. The first word of AREA must contain the word count indicating the length of the output record. The exit line for this form of return is:

```
EXIT    0    0
```

Alternatively, the user may avoid writing a record and obtain the next input record by branching to NEXT (see page 33).

Some processing of records will always be required. This subroutine cannot therefore be a dummy.

CLOSE

This subroutine is called at the end of the run. It will normally be used to output any final totals required, and for any other end of run processing that the user may need.

The entry line is:

```
CALL    0    CLOSE
```

Note: The output file has not yet been closed or rewound at this stage, so the user may write any control records that he may need. He must however give his own write instructions for these records. These should have the form:

```
SDWR    1    place
```

where *place* is the name of the area which contains the user's output record.

Note: If such an SD macro is used, a #CMODE directive must be provided in the user segment.

The exit line is:

```
EXIT    0    0
```

This subroutine may be in dummy form:

```
#CUE          CLOSE
    EXIT 0    0
```

Extra facilities

THE SUBROUTINE SHUNT

In forming a new record from an existing tape record, there is often a need to move information without altering it. Where the data fills words exactly, the instruction MOVE can be used. But where the data begins or ends in the middle of a word, the subroutine SHUNT will be found convenient.

The call takes the form:

```
CALL    1    SHUNT
             n1, n2, n3, n4, n5
```

where

n_1 = the number of characters to be moved (to a maximum of 127; if N, = 0, 128 characters are moved)

n_2 = the source word address
n_3 = the source character address
} relative to the start if the input record

n_4 = the destination word address
n_5 = the destination character address
} relative to the start of the output record area

All accumulators are stored on entry and reinstated before exit.

THE CUE NEXT

When the user wishes to ignore certain input records, or create an output record containing data from several successive input records, the cue NEXT may be used. Instead of employing the exit from RECRD the user should code:

```
BRN          NEXT
```

which will cause the next input record to be processed by RECRD. The previous record will then not be written to the output file.

Chapter 9 Input and output

INPUT

Input to the complete SUBEDIT program consists of a magnetic tape file containing the records to be edited and parameters, punched on cards or paper tape.

Input file

The input file must observe the conventions of 1900 Series Magnetic Tape Housekeeping. It may be a single or multireel file.

Parameters

The parameters are described in detail in Chapter

OUTPUT

The output from the complete SUBEDIT program consists of a magnetic tape file, line printer output and console output.

Output file

The output file contains the edited records.

Line printer output

PARAMETER LIST

The program will print out each parameter in full exactly as it appears in the card or on paper tape. The word WRONG is printed out against the listing of any parameter found to be incorrect.

If an error message has appeared against any parameter in the listing the message at the end of the listing is:

PARAMETERS INCORRECT

In addition, the printer will output:

PARAMETERS INCOMPLETE

if the parameter set is incomplete.

Console output

See under *EXCEPTION CONDITIONS*, Chapter 12.

Chapter 10 The parameter set

Parameters may be input to SUBEDIT punched either in cards or paper tape. Although the following descriptions refer to cards, the format for paper tape is identical except that each parameter must be followed by a Newline character.

SUBEDIT uses the following standard parameters:

a #READ parameter

a #WRITE parameter

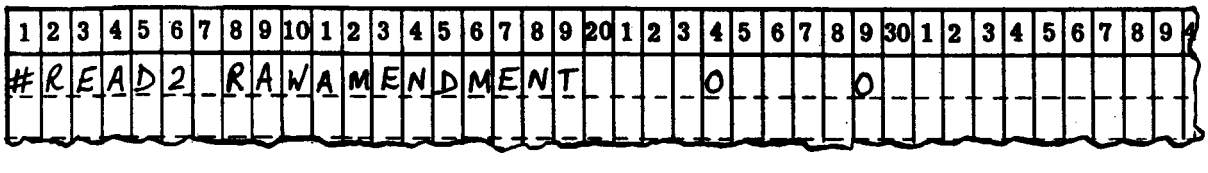
a #END parameter

READ TAPE LABEL PARAMETER (#READ)

This parameter is in standard form. Its function is to define the label on the input tape file, and the mode in which it is to be opened.

Field	*Columns*	*Contents*
A	1 to 5	The directive #READ
	6	Opening mode: 1 character as follows 1 = do not check for absence or presence of write permit ring. 2 = check for absence of write permit ring 3 = check for presence of write permit ring. ∇(space) in this position interpreted as 2. Note: These are equivalent to the standard modes of opening tapes, that is #100, #200 and #300. Details of additional checks by Executive on the 1904 and above will be found in the ICL manual *Magnetic Tape.*
B	8 to 19	12 character file name.
C	21 to 24	Reel sequence number; four digits.
D	26 to 29	File generation number: four digits.
E	31	Designation: one alphabetic character (not needed for this program).

Example



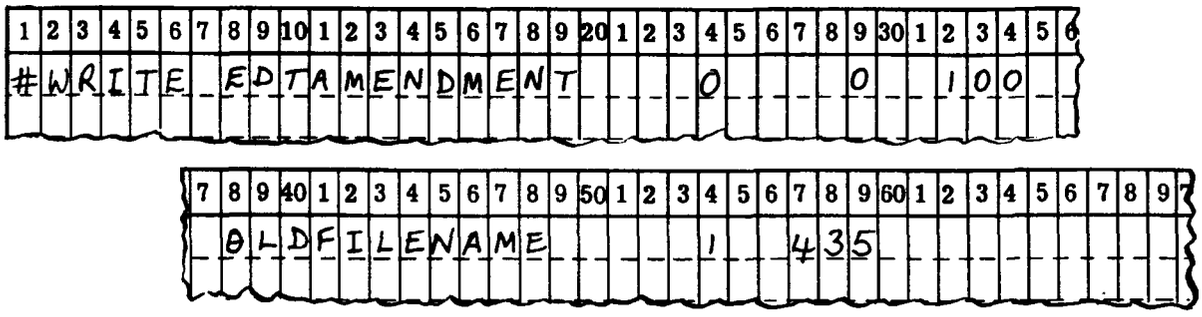
WRITE TAPE LABELS PARAMETERS (#WRITE)

This parameter is in standard form. Its function is to define the labels on the output file.

Fields G, H and I are optional. If they are punched the program will search for a tape with the specified label and open it for output; if these fields are blank, the program will open any available scratch tape.

Field	*Columns*	*Contents*
A	1 to 6	The directive #WRITE
B	8 to 19	12 character file name
C	21 to 24	Reel sequence number: four digits
D	26 to 29	File generation number: four digits
E	31 to 34	Retention period: four digits
F	36	Designation: one alphabetic character (not needed for this program)
G	38 to 49	Old file name
H	51 to 54	Old reel sequence number
I	56 to 59	Old file generation number

Example

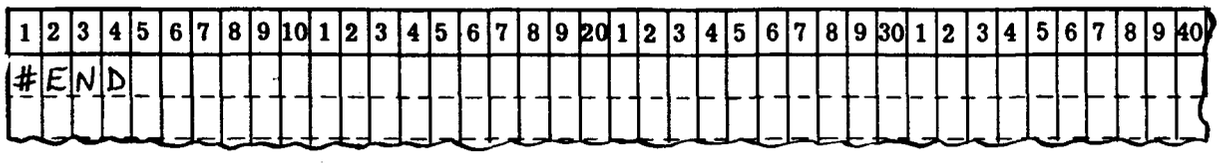


END OF PARAMETERS MARKER

This parameter is in standard form. It marks the end of the parameter set.

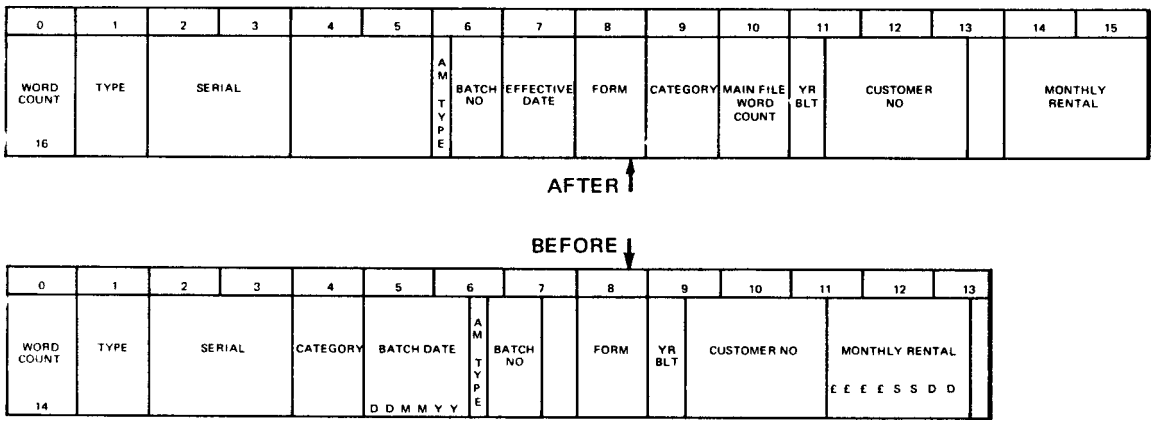
Field	*Columns*	*Contents*
A	1 to 4	The directive #END

Example



G

File formats



Control parameter

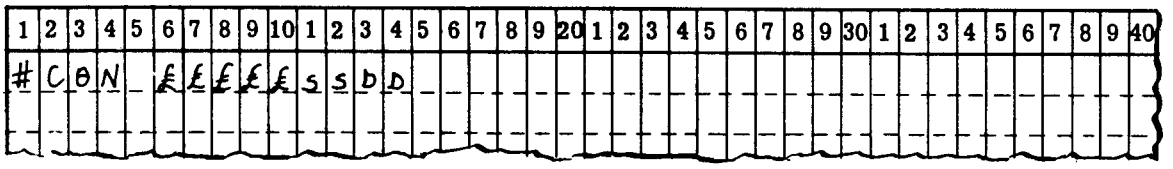


Figure 9 SUBEDIT example: file formats and control parameter

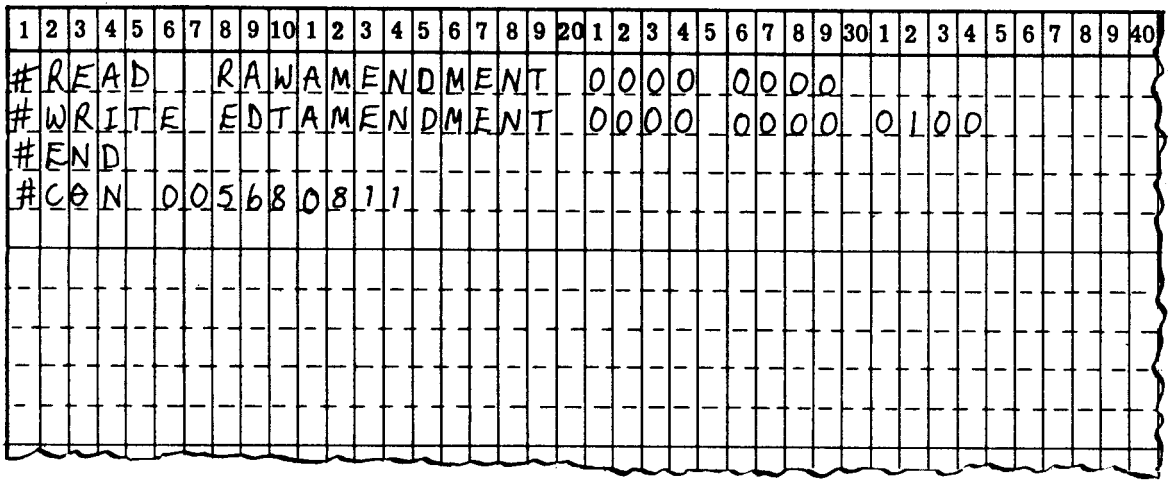


Figure 10 SUBEDIT example: parameters

Chapter 11 Example

This example, written in PLAN 3, is a simple tape editing program designed to operate on a file of amendments for a customer installation master file. At start of run the program reads a control total parameter containing the expected total of monthly rental on the amendment file. During the run the program totals the monthly rental and at end of run compares the actual acumulated total with that input by parameter; if the totals disagree the program displays a message on the console typewriter indicating that there is disagreement and showing the actual file total.

The following is a step by step explanation of the coding in the example, see Figure 11.

Line 1	The user segment must be written as an acceptable PLAN program. The user must provide a program name in the normal way.
Lines 2 to 6	The user specifies the common areas that he wishes to use.
Lines 7 to 12	The user sets up his own variable and preset data areas.
Line 14	This branch to the framework segment will never be obeyed. Its function is to instruct the compiler to bring in the framework.
Lines 15 to 30	The start of run routine. The control total parameter, see Figure 9, is read and basic validation is performed. Note that, although the framework can accept parameters on cards or paper tape, the user segment assumes cards will always be used (SUSBY instruction Line 18). Paper tape could be catered for by using the type/mode word of the parameter reading control area, thus: PERI 0 CR LDX 1 CR SRL 1 15 SUSBY 0 0(1) Finally, totalling areas are initialised.
Lines 31 to 61	The user's record processing routine. Control records (indicated by WD 4 =****) are ignored and will not appear in the program's output file. All other records are edited using library subroutines and the framework subroutine SHUNT, and placed in the output area. The monthly rental figures are totalled.
Lines 62 to 74	The user's end of file routine. The actual file total, accumulated through the run, is compared with the expected figure input on parameter at start of run. If they do not match an error message is output on the console typewriter giving the accumulated file total. If the totals do match, a message is displayed on the console typewriter indicating that the controls are in agreement.

```
0000    #STEER          LIST

        18/47/22         26/02/69        COMPILED BY XPLG 25F

0001    #PROGRAM        AMED50/USERSEGMENT  [MJK 113URR
0002    #LOWER          COMMON/STORE/
0003                    A(512),POINT
0004    #LOWER          COMMON/WORKING/
0005                    CHAR,PARAM(30)
0006                    CR(4)               [CR CNTRL AREA-PRESET BY FRAMEWORK
0007    #LOWER
0008                    TOTAL(2),CONTROL(2),LINK,X4
0009    #LOWER
0010    MSG1            28HTOT.ER.-FILE TOT:                                    0  #64576436
                                                                                1  #45623635
                                                                                2  #46515445
                                                                                3  #20645764
                                                                                4  #12202020
                                                                                5  #20202020
                                                                                6  #20202020
0011    CON             4H#CON                                                  7  #23435756
0012    MSG2            10HCONTROL OK                                           8  #43575664
                                                                                9  #62575420
                                                                               10  #57532020
0013    #PROGRAM
0014            BRN     SUBEDIT             [DUMMY TO BRING IN FRAMEWORK        0  074 0       0 BA
0015    #CUE            START               [START OF RUN ROUTINE
0016            STO  0  LINK                                                    1  010 0 0     4 LV
                                                                    R2= WORKING
0017            PERI 0  CR                  [READ CONTROL TOTAL PARAMETER       2  157 0 0    31 R2
0018            SUSBY 0 3                                                       3  150 0 0     3
0019            LDX  4  PARAM                                                   4  000 4 0     1 R2
                                                                                5  026 4 0     7 LP
                                                                                6  074 6       0 BA
0020            BXE  4  CON,ST2             [VALIDATE PARAMETER
0021    ST1     SUSWT   2HEP                                                    7  161 0 0 2416
0022    ST2     BVSR    *+1                                                     8  074 2       9 PR
0023            LDX  2  '9/PARAM+1.1'                                           9  000 2 0     0 LT
0024            CALL 0  CSTERBINF2          [CONVERT CONTROL TOTAL TO BINARY   10  070 0       0 BA
0025            BVS     ST1                                                    11  074 1       7 PR
                                                                               12  010 7 0     3 LV
                                                                               13  010 6 0     2 LV
0026            STO  67 CONTROL
0027            STOZ    TOTAL               [INITIALISE FILE TOTALLING AREA    14  033 0 0     0 LV
0028            STOZ    TOTAL+1                                                15  033 0 0     1 LV
0029            LDX  0  LINK                                                   16  000 0 0     4 LV
0030            EXIT 0  0                   [RETURN TO FRAMEWORK               17  072 0       0
0031    #CUE            RECRD               [ROUTINE TO PROCESS EACH RECORD
0032            STO  0  LINK                                                   18  010 0 0     4 LV
0033            LDX  4  4(1)                                                   19  000 4 1     4
                                                                               20  026 4 0     2 LT
                                                                               21  074 6       0 BA
0034            BXE  4  '4H****',NEXT       [IGNORE CONTROL RECORDS
                                                                    R2= STORE
0035            LDN  2  A                                                      22  100 2 0     0 R2
0036            MOVE 1  4                   [MOVE 1ST PART UNCHANGED TO O/P    23  126 1 0     4
0037            LDN  4  16                  [INSERT O/P WORD COUNT             24  100 4 0    16
0038            STO  4  A                                                      25  010 4 0     0 R2
0039            LDN  3  5(1)                                                   26  100 4 1     5
0040            STO  4  X4                                                     27  010 4 0     5 LV
0041            CALL 1  GDATEBIN            [CONVERT BATCH DATE TO BINARY...   28  070 1       0 BA
0042            LDX  3  X4                                                     29  000 3 0     5 LV
0043            LDN  3  A+7                 [...AND INSERT IN OUTPUT AREA      30  100 3 0     7 R2
0044            CALL 1  SHUNT               [MOVE AMEND TYPE & BATCH TO O/P    31  070 1       0 BA
0045                    4,6,2,6,0           [(4CHS FROM 6.2 TO A+6.0)          32  #00000004
                                                                               33  #00000006
                                                                               34  #00000002
                                                                               35  #00000006
                                                                               36  #00000000
0046            LDX  1  POINT               [REINSTATE POINTER                 37  000 1 0   512 R2
0047            LDX  4  8(1)                                                   38  000 4 1     8
0048            STO  4  A+8                 [REFORMAT REST OF RECORD           39  010 4 0     8 R2
0049            LDX  4  4(1)                                                   40  000 4 1     4
0050            STO  4  A+9                                                    41  010 4 0     9 R2
0051            LDN  5  30                                                     42  100 5 0    30
0052            STO  5  A+10                                                   43  010 5 0    10 R2
0053            CALL 1  SHUNT                                                  44  070 1       0 BA
0054                    10,9,0,11,0         [ (YR BLT & CUSTOMER NO. )         45  #00000012
                                                                               46  #00000011
                                                                               47  #00000000
                                                                               48  #00000013
                                                                               49  #00000000
0055            LDX  2  '9/11.2'                                               50  000 2 0     3 LT
0056            ADX  2  POINT                                                  51  001 2 0   512 R2
0057            CALL 0  CSTERBINF2          [MONTHLY RENTAL TO BINARY          52  070 0       0 BA
                                                                               53  015 7 0     1 LV
                                                                               54  011 6 0     0 LV
0058            ADS  67 TOTAL               [TOTAL MONTHLY RENTAL
                                                                               55  010 7 0    15 R2
                                                                               56  010 6 0    14 R2
0059            STO  67 A+14                [AND INSERT IN O/P AREA
0060            LDX  0  LINK                                                   57  000 0 0     4 LV
0061            EXIT 0  0                   [RETURN TO FRAMEWORK               58  072 0       0
0062    #CUE            CLOSE               [END OF FILE ROUTINE
0063            STO  0  LINK                                                   59  010 0 0     4 LV
                                                                               60  000 7 0     1 LV
                                                                               61  000 6 0     0 LV
0064            LDX  67 TOTAL
                                                                               62  026 7 0     3 LV
                                                                               63  026 6 0     2 LV
                                                                               64  074 6       0 BA
0065            BXE  67 CONTROL,CL1         [COMPARE ACTUAL TOTAL WITH CONTROL
0066            LDX  2  '9/MSG1+4.1'        [FIGURE - IF DIFFERENT SET UP ...  65  000 2 0     4 LT
0067            CALL 1  CBINSTERDL1+1       [...ERROR MESSAGE                  66  070 1       1 BA
0068                    #20                                                    67  #00000020
0069            CALL 0  LSDSPACE                                               68  070 0       0 BA
0070            DISTY   '28/MSG1'                                              69  160 1 0     5 LT
0071            BRN     CL2                                                    70  074 0       0 BA
0072    CL1     DISTY   '10/MSG2'           [FIGURES TALLY - OK                71  160 1 0     6 LT
0073    CL2     LDX  0  LINK                                                   72  000 0 0     4 LV
0074            EXIT 0  0                   [RETURN TO FRAMEWORK               73  072 0       0
0075    #END
```

Figure 11 SUBEDIT example: compilation listing

Chapter 12 Operating instructions and exception conditions

The operating instructions for the run in which the SUBEDIT program is compiled will depend on the compiler being used, and will be standard for that compiler; see *PLAN Reference Manual.*

The following operating instructions are for running a compiled SUBEDIT program.

OPERATING INSTRUCTIONS

	Action	*Message*
1	Load the input file without a write permit ring.	
2	Load the output file with write permit ring and if necessary give it a scratch label.	
3	Load the compiled program by inputting:	FI *#name* #TAPE
4	Load parameters in appropriate reader.	
5	To read in the parameters either	
	(a) if the parameters are on paper tape input:	GO *#name* 20
	or	
	(b) if the parameters are on cards input:	GO *#name* 21
	The program reads the parameters, opens the tape and reads the tape. At the end of the run the program outputs the message:	DISPLAY:–COUNT OF RECORDS READ *nnnn* DISPLAY:–COUNT OF RECORDS WRITTEN *nnnn* HALTED:–END OF PROGRAM
6	To suspend the program in the middle of a run input:	SU *#name* GO *#name* 26
	The program rewinds and closes the input and output tapes, displays records counts and outputs the message:	HALTED:–RUN POSTPONED

EXCEPTION CONDITIONS

Message	*Reason*	*Action*
HALTED:–PARAMETERS INCORRECT	Parameters are incorrect	Correct the parameters and GO *#name* 20 or 21
HALTED:–INPUT TAPE EXCEPTION CONDITION 01	A long block has been found on the input file.	Abandon the run.
HALTED:–INPUT TAPE EXCEPTION CONDITION 04	A parity failure has occurred on the input file.	Abandon the run.
HALTED:–OUTPUT TAPE EXCEPTION CONDITION 01	A long block has been found on the output file.	Abandon the run.
HALTED:–OUTPUT TAPE EXCEPTION CONDITION 04	A parity failure has occurred on the output file.	Abandon the run.

Amendment record

Should any additions or amendments to this Manual be necessary, they will be issued in the form of Amendment Lists.
When such changes have been incorporated, an appropriate entry should be made in the table below and the pink instruction sheet(s) should be filed behind this page.

List number	Item number
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to