technical memorandum Daresbury Laboratory

DL/CSE/TM 20

FAMULUS USER GUIDE

by

G.D. FIRTH, Daresbury Laboratory

MAY, 1982

Science & Engineering Research Council

Daresbury Laboratory

Daresbury, Warrington WA4 4AD

FAMULUS USER GUIDE

by

G.D. FIRTH

PREFACE

FAMULUS is an information storage and retrieval system which was originally developed at the Pacific South West Forest and Range Experimental Station (California) of the U.S. Department of Agriculture in 1969. It was adapted and further developed at University College, London, and then implemented on the Rutherford Laboratory IBM 360/195's from where the version implemented on the NAS AS/7000 at Daresbury was obtained. The FAMULUS User Guide is based on the FAMULUS User Manual issued by the University of Manchester Regional Computer Centre.

FAMULUS is widely used for a number of different applications, as it has the facility to print out selected information in a neat, tabulated form, and has very convenient means for maintaining data bases.

CONTENTS

1. GENERAL DESCRIPTION

FAMULUS is a computerised system for filing and retrieving information. It is intended to serve the needs of the individual research worker, research group, or a small library. It can take the place of the traditional 'shoe box' full of filing slips. There is always a considerable effort involved in conversion from a manual to an automatic system, mainly in the preparation of the data for the computer, but also in designing a system appropriate to the application. The initial effort, however, is repaid by improved retrieval and display facilities, and often by a reduction in the amount of subsequent manual data handling. FAMULUS offers various information retrieval facilities, such as automatic sorting of files into alphabetical order, indexing, searching in response to specific requests, and catalogue production. Because of its ability to collate and quantify information, it can also be applied as a tool for research on the data which it stores.

FAMULUS was designed, in the first instance, with a bibliographic application in mind. It is ideal for keeping private bibliographies or catalogues. In a small library environment FAMULUS has been used for many purposes, from keeping the main catalogue to maintaining a 'wants' list of books to be obtained. It is also being used for keeping and searching museum catalogues of specimens. The system is sufficiently flexible in design, however, to permit various other applications, for example, a linguist might use it to maintain a dictionary or grammar file, or an administrative department could keep personnel records.

FAMULUS is not restricted, therefore, to cataloguing books. It can be used to document any type of object. It is structure, not content, which matters, as it cannot do arithmetic with any numbers included in the texts, because they are considered as strings of characters. FAMULUS is not suitable for handling tabular data, or equations with subscripts, etc. displayed on the adjacent lines, or large texts with little or no regular internal structure (such as this manual).

2. FILE ORGANIZATION

A complete collection of information is held by FAMULUS on magnetic disk or tape and can be referred to as a file, or a data set. A file consists of an unlimited number of records.

A record is the basic unit of information, and the first step in designing any system using FAMULUS is to decide what logical entity in the data a record should represent. For instance, in a bibliographic file it is natural to consider one citation or reference to be a record. In complex applications it may not always be obvious, but it is of first importance to establish clearly from the outset what constitutes a record.

The record itself is broken down into fields. Each field is given a name or label appropriate to its content. Up to ten fields are permitted at Daresbury. In other implementations twenty or even sixty fields have been made available. The amount of information in a field may vary from record to record or from field to field, subject only to the restriction that records may not exceed 4000 characters in length. Every record in a file must have some or all of the same fields, in the same order.

A field is the minimum unit of information which is used to sort the file into alphabetical order. The field divisions, therefore, constitute one of the principal methods of access to records in the file. Consequently the number of fields and the division of the information in the record into fields is dependent upon the avenues of access which are required to the records. See the SORT program description for further discussion of this point.

One field may be designated as a descriptor field. In ordinary fields words are identified as such by being separated by blanks and punctuation. In the descriptor field whole phrases may be treated as units by the use of a specified delimiting character. This facility caters for index terms consisting of more than one word. Index terms may have a maximum length of 40 characters. Terms which exceed this length will be truncated by some programs, although the full terms always remain in the data base.

3. BASIC OPERATIONS

FAMULUS consists of a number of separate programs which perform the basic operations of the system. They fall into four main categories - data management, file manipulation, information retrieval, and thesaurus construction. Note that throughout this manual all reference to punched cards should be taken as meaning a card image file on disk or tape, for instance a TSO DATA type file.

3.1 Data Management

Data are punched onto cards as described in the section on data preparation. The EDIT program accepts data cards and creates a file on disk. None of the other programs accept card data; they operate only upon the files already created by the EDIT program.

It is important to distinguish carefully between data in external form (on cards) and its representation within the system as an internal file. The point here is not strictly the storage medium as such, for a magnetic disk is no less an external medium than cards from the machine's point of view; from the point of view of the FAMULUS system, however, internal files are the form in which data are generally stored and processed.

FAMULUS internal files are not readable by other programs (because of their internal format), and are not suitable for transferring data out of the system. The OSSIFY program is useful for this purpose. It performs the converse of the EDIT operation, creating a 'card deck' in the format required for EDIT input out of an internal file. This 'card deck' may be as punched cards or on other media, such as magnetic tape.

A second function of the EDIT program is to make additions, deletions, and amendments to internal files. This is accomplished by making a new copy of the file with the required changes. It is good practice to use the grandfather, father, son principle to build up files. For example, three magnetic tapes could be used in rotation, so that there is always a copy of the file in reserve in case a rare catastrophic failure destroys the current input and output tape.

3.2 File Manipulation

If the records in a file are in random order, the SORT program is used to create a new file in proper alphabetical sequence of any given field. Several copies of the file may be made in different orders to facilitate retrieval of records.

The MERGE program combines two compatible files which have been previously sorted on the same field into a single file in correct alphabetical order.

The multiplying program, MULTI, multiplies a file by generating multiple copies of a record depending upon the entries in a given field, for example, to produce an author catalogue with entries under each author (even where books have joint authors). This is one of the more sophisticated facilities of the system and may not be needed in straightforward applications.

3.3 Information Retrieval

Once the file of data has been created, FAMULUS provides various methods of retrieving information from it. The unit of information that is sought is the record in all cases, but the methods differ in requiring either a direct or indirect look-up, and also in taking either a comprehensive or selective approach to retrieval.

The GALLEY program produces a comprehensive listing of all the records in a file. Various formats are possible, and every field need not be printed, nor need they be printed in the original order. Used in conjunction with the SORT program mentioned above, GALLEY can print separate author, title, classified catalogues of bibliographic files, or language dictionaries sorted two-two ways, to quote only a few examples.

When records are qualified by subject headings, keywords, or descriptors indirect retrieval is often required. The INDEX program prints an alphabetical list of such items in a designated descriptor field. With each descriptor is a list of the record numbers in which it occurs. The record numbers correspond to the position of the record in the file and are printed to the left of each record in a GALLEY listing. Information is retrieved by consulting the index to find the relevant records and then look-

ing up each record by its number in the complete listing of the file.

Both GALLEY and INDEX are comprehensive in the sense that all the records in a file are represented in their respective printouts. When the file is large and very specific information is to be located, a selective printout from the file can reduce the amount of irrelevant material that needs to be scanned. This is the function of the SEARCH program.

SEARCH operates by scanning any fields to see whether each record satisfies a search request. The request employs Boolean logic and is capable of expressing any degree of specificity. Abbreviated terms may be used to retrieve records containing words with different endings. The records which satisfy the request are printed out in full. If an indirect look-up is wanted, the record numbers alone can be printed.

3.4 Thesaurus Construction

The usefulness of an index of descriptors depends on how representative they are of the subjects covered by the file. Standard word lists and thesauri are available on various topics. Even assuming an adequate thesaurus of descriptors, quite a lot of exact and exacting work may be required to assign the correct descriptors to each record. This does not need to be undertaken during the early stages of file construction, in fact it may not be possible until it is seen how the data base develops. Descriptors can be added to a file at any time using the powerful and convenient editing facilities of FAMULUS.

One advantage of deferring the assignment of descriptors is that a study of the vocabulary in other fields may help in choosing them. The VOCAB program prints an alphabetical list of vocabulary in any field. The COUNT program does the same and in addition counts the frequency of each word in the file. Stop lists of trivial words can be created, which can be used by KEY. This program provides an automatic keywording facility. It transfers non-trivial vocabulary from any field to a specified key field (overwriting or adding the new keywords) for use as descriptors.

Indexing experience demonstrates that in choosing descriptors it is best to use the shortest version of a keyword or phrase which gives explicit expression to your concept. This will usually be a compound noun, for

example, MERISTEM REGENERATION rather than an adjectival phrase MERISTEMIC REGENERATION or a noun-phrase like REGENERATION OF MERISTEMS. Users of libraries and standard bibliographies may have an impulse to invert compound terms, for example, THINNING, NATURAL as against NATURAL THINNING. Bibliographically, this is a complex issue and should be discussed with a librarian. We advise, in general, against inversion. Inversion is necessary and useful in long, visually-displayed files like library catalogues and printed indexes. The search efficiency of FAMULUS is not in any way increased by inversion of terms. If however, you want your printed VOCAB lists to reflect standard subject heading lists, then you may certainly use inversion, and it is necessary if you wish to sort on the keyword field after a MULTI run, to produce a subject index.

3.5 Data Preparation

Machine-readable files of data can often be converted by a special-purpose user written program to FAMULUS input format. Otherwise data must be punched on cards.

Field labels are punched in Columns 1-4 of the first card in a field. Continuation cards for the field do not carry labels. Actual text of the record begins in Column 6 and continues to Column 80. If it is necessary to continue onto another card, punching begins in either Column 6 or Column 7. If the last word of the preceding card had ended in Column 80, then punching begins in Column 7, as a space will be required in Column 6. If the word was broken in the middle, or if Column 80 was a blank separating two words, begin in Column 6. Do not use a hyphen in Column 80 to break a word if it is not complete; simply punch up to Column 80 and place the remainder of the word on the next card beginning in Column 6. In other words, Columns 6-80 of successive cards for a field are treated as a continuous string of characters which are sometimes redistributed between different lines by FAMULUS, for example, if printing with a different page width.

The fields in a record must be punched in the correct order, otherwise the record will be rejected. Fields may be omitted however, since not every record in a file will require all the allowable fields. In this case a card containing the field label is not required.

Each record must be less than 4000 characters altogether.

Each record is followed by a blank card which separates it from the next record. If the blank card is left out, the second record will be run on to the first and both will be rejected. The last record in the input deck should also be followed by a blank card.

4. JOB CONTROL CARDS

In order to use the computer, it is necessary to submit a job. A job is defined by a set of control cards which are used to:

(a) Request certain amounts of computer resources (such as processing time, etc.).

(b) Invoke the FAMULUS system and specify which FAMULUS programs are to be used.

(c) Define the various data files and the devices upon which they reside.

4.1 FAMULUS Procedures

This section is concerned with details of how to run the FAMULUS programs on the Daresbury AS/7000 computer. Catalogued procedures have been set up to execute each of the FAMULUS programs, which reside in the XRY.LOAD library. The procedures are listed below together with the program they execute:-

FAMCOUNT	COUNT
FAMEDIT	EDIT
FAMGALLY	GALLEY
FAMINDX	INDEX
FAMKEY	KEY
FAMKWIC	KWIC
FAMMERGE	MERGE
FAMMULT	MULTI
FAMOSSY	OSSIFY
FAMSORT	SORT
FAMSRCH	SEARCH

One other procedure has been set up,

FAMSOGAL

which executes the SORT and GALLEY programs as it has been anticipated that the combination of sorting information in a database and then printing it will be commonly used.

The basic principle on which the FAMULUS procedures work is to take the FAMULUS master file specified by the FILEIN parameter and carry out on it the processing indicated by the set of control cards supplied. This processing will involve producing the required print and/or new file (specified by the FILEOUT parameter). For example the EDIT program is used to produce an updated master file, the SORT program is used to produce a master file sorted in a new order and the KWIC and GALLEY programs are used to generate prints.

The control cards supplied may be either a data set or in-stream data and are identified to the procedures by means of an override card of the form:-

```
//FAMPROG.SYSIN DD -------
```

where FAMPROG is the name of the FAMULUS program being executed (see list above). This override must come after the procedure EXEC statement e.g.

```
//S1 EXEC FAMKWIC,FILEIN='JOES.MASTER.FILE'
//KWIC.SYSIN DD DSN=JOES.CONTROL.CARDS,DISP=SHR
```

note that for the FAMSOGAL procedure two overrides are necessary, one for the SORT program and one for the GALLEY program.

The parameters used with the procedures are given below, together with their default values and notes to indicate when other than the standard default value has been used. Unless otherwise stated the parameters are used by all the procedures.

DISPIN=SHR	The disposition parameter of the input FAMULUS master file, the FAMEDIT procedure has a default of OLD.
DISPOUT='(NEW,PASS)'	The disposition parameter of the output FAMULUS master file. Except for the FAMOSSY procedure where the output file is a 'card deck' of 'ossified' records and the default is '(NEW,CATLG)'.
FILEIN='&&TEMP'	The file name of the input FAMULUS master file.
FILEIN2='&&TEMP'	Used only by the FAMMERGE procedure, this is the name of the second input FAMULUS master file.

FILEOUT='&&TEMP'	The name of the output FAMULUS master file.
SPACOUT='(CYL,(5,5))'	The amount of disk space allocated to the output FAMULUS master file. The default value for the FAMEDIT procedure is '(CYL,(1,1))'.
SPACWRK='(CYL,(5,5))'	The amount of disk space allocated to temporary work files used in the procedure, used by the procedures FAMKWIC, FAMSOGAL and FAMSORT.
UNITOUT=3330	The unit type of the volume on which the output master file will reside.
UNITWRK=3330	The unit type of the volume on which temporary work files will reside.
VOLOUT=DNPL33	Volume serial number of the output master file residence volume.
VOLWRK=DNPL33	Volume serial number of the volume used by temporary work files in procedures FAMKWIC, FAMSOGAL and FAMSORT.

NOTES:-

1) if a parameter contains other than alphanumeric characters then it must be enclosed in quotes e.g. FILEIN='JOES.MASTER.FILE'.

2) DNPL33 is a scratch pack and as such is available for files which are only to be used for the duration of a job. However if the master file already resides on a disk volume on which there is space to place temporary files then this volume should be specified by means of the VOLWRK parameter.

3) it is possible to create a job which uses more than one of the procedures.

A) FAMCOUNT

This procedure executes the FAMULUS COUNT program which is used to

produce statistics on the vocabulary of the database. Parameters are:-

```
FILEIN='&&TEMP'
DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

B) FAMEDIT

This procedure executes the FAMULUS EDIT program which is used to update the database by adding, deleting and editing records. Parameters are:-

```
FILEIN='&&TEMP'
DISPIN=OLD
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
VOLOUT=DNPL33
SPACOUT='(CYL,(1,1))'
```

C) FAMGALLY

This procedure executes the FAMULUS GALLEY program which is used to obtain prints from the database. Parameters are:-

```
FILEIN='&&TEMP'
DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

D) FAMINDX

This procedure executes the FAMULUS INDEX program which is used to produce indexes to the contents of the database. Parameters are:-

```
FILEIN='&&TEMP'
DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
```

```
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

E) FAMKEY

This procedure executes the FAMULUS KEY program which is used to generate keywords from field(s) of the database. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

F) FAMKWIC

This procedure executes the FAMULUS KWIC program which is used to produce Keyword In Context Indexes from the database. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
UNITWRK=3330
 VOLWRK=DNPL33
SPACWRK='(CYL,(5,5))'
```

G) FAMMERGE

This procedure executes the FAMULUS MERGE program which is used to merge two existing master files to give one database. Parameters are:-

```
 FILEIN='&&TEMP'
FILEIN2='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
```



```
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

H) FAMMULT

This procedure executes the FAMULUS MULTI program which is used to create a new master file with one entry for each element of a field. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

I) FAMOSSY

This procedure executes the FAMULUS OSSIFY program which is used to output data in the format used by the EDIT program. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,CATLG)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

J) FAMSOGAL

This procedure executes the FAMULUS SORT and GALLEY programs which are together used to produce sorted print from the database. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
UNITWRK=3330
 VOLWRK=DNPL33
SPACWRK='(CYL,(5,5))'
```

K) FAMSORT

This procedure executes the FAMULUS SORT program which is used to produce a new master file sorted to a new order. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
UNITWRK=3330
 VOLWRK=DNPL33
SPACWRK='(CYL,(5,5))'
```

L) FAMSRCH

This procedure executes the FAMULUS SEARCH program which is used to search the database for records meeting conditions. Parameters are:-

```
 FILEIN='&&TEMP'
 DISPIN=SHR
FILEOUT='&&TEMP'
DISPOUT='(NEW,PASS)'
UNITOUT=3330
 VOLOUT=DNPL33
SPACOUT='(CYL,(5,5))'
```

5. PROGRAM CONTROL CARDS

The operation of FAMULUS programs is controlled by program control cards. They are used for various purposes, for example, to inform FAMULUS of the names of files or fields, to select portions of a file for processing, to request particular printed output formats, and many others. Some control cards are specific to certain programs, but many are used by more than one program. An effort has been made to generalize the control cards as far as possible, to achieve greater flexibility and more powerful facilities, without, it is hoped, changing the philosophy of the original system too much. A chart of the control cards available to each program is given in Appendix B.

All control cards contain a statement, which is one of a set of upper

case keywords recognized by FAMULUS, enclosed in slashes and punched in Column 1 onwards. Some control cards consist of nothing more than this, for example,

```
/ORIGINAL/
/WRITE TAPE/
/PUNCH/
/PRINT BY FIELDS/
/PRINT BY SUBJECTS/
```

The information conveyed by these cards is essentially of a yes/no nature. The majority of control cards, however, need to provide particulars of the action required. They contain a text which follows the second slash of the statement, though not necessarily immediately. If the whole text does not fit on one card it can be continued on the next anywhere from Column 1 onwards. The text of most statements is enclosed in parentheses, for example,

```
/FIELDS/(AUTH,TITL,PUB,DATE)
/DELETE/(3,15-19,46-48,101)
/SELECT/(1-500,575-690)
```

There are no enclosing parentheses, however, on the following:

```
/ID/PARTS INVENTORY
/NEW ID/CLASSIFIED INDEX
/SEARCH/SOIL STABILITY & (EROSION ↑ SLIDES)
/VOCABULARY/A*THE*IN*SOIL,STABILITY,EROSION
/SYNONYMS/EROSION=SLIDES,DUST
```

There is, in general, no fixed order in which control cards must appear, and the position of most control cards is not significant. A few, however, are context-sensitive, as follows:

/ID/	must be the first control card,
/CITATIONS/	must be the last control card and immediately precede the input data cards, if any,
/SEARCH/	must come after all other control cards.

Note that a program which uses the /CITATIONS/ card will not use a /SEARCH/ card and vice versa.

The use of the control cards is described further under each separate program, but it is appropriate at this point to introduce a few statements

which are common to most of the FAMULUS programs.

/ID/

Internal files are always labelled with an identifier. When an old master file is used as input to a program, FAMULUS checks the identifier written on the tape with the text of the /ID/ statement. If they fail to match, an error message is produced and the program terminates.

The identifier may be up to 100 characters in length. Blanks before and after the old master file name are ignored. Punch up to Column 80 and if required, continue on a second card, starting in Column 1.

Every program needs an /ID/ card, and it must be the first control card of all.

/NEW ID/

Whenever a new master file is created the identifier with which it is labelled is taken from a /NEW ID/ control card. The new identifier is punched after /NEW ID/ in the same way as on the /ID/ card. If this control card is not supplied, the /ID/ card is used instead, which means that the output file is labelled the same as the input file by default.

In most programs, the text from the /NEW ID/ card is printed as a title on the first line of every page before the page number. Even when no output file is being written this control card is still useful for entitling the printed output. The default title is again taken from the /ID/ card.

/FIELDS/

Field labels may be up to four characters in length, must begin with an alphabetic character, and the remaining characters must be either alphabetic or numeric. Lower case characters and special characters (such as punctuation) are not allowed. For example,

PUBL (legal field label)

PUB. (illegal field label)

The text of this statement consists of a list of field labels separated by commas, the whole enclosed in parentheses. The list supplies three

kinds of information: the number of fields, the order in which they occur in the records, and their names. This information is used in different ways according to the requirements of each program. For instance, in EDIT, the /FIELDS/ card is used to define the full record structure for the new master file; in GALLEY, it is used to define the number and order of the fields to be printed; in SORT, it defines the fields on which the records are to be re-sorted; in SEARCH and COUNT, it indicates the field or fields to be searched or counted; and in MULTI, only one field label is permitted on the card, that of the field on which the file is to be multiplied.

Note that if many fields are used, this 'card' may need several physical cards. These lines must not exceed 80 characters and may be conveniently split after any comma. Extra spaces may occur anywhere except inside field names.

/DESCRIPTOR FIELD/

The INDEX program is designed to work only on the descriptor field of a file. Descriptors are subject headings or index terms which often consist of more than one word. Within the descriptor field, therefore, descriptors are separated by a delimiter or break character, and may consist of any character string (including blanks and punctuation) apart from the break character. Descriptors may be any length, but will be truncated to 40 characters in the output from some programs. Also, descriptors which only differ from each other after more than 40 characters will be considered as identical.

Other programs, such as SEARCH, COUNT, MULTI, and KEY, also recognize a descriptor field and regard its contents differently from other fields. When working on the descriptor field these programs (like INDEX) identify items by means of the break character, and words, delimited by blanks or punctuation, are the basic items in other fields.

Only one field may be a descriptor field at any one time. There is no obligation to designate a descriptor field, but once defined, the identity of the descriptor field becomes part of the information carried on the file like the identifier or fields information, and there is no need to include the /DESCRIPTOR FIELD/ card again. This information cannot be altered on an old master file, but a /DESCRIPTOR FIELD/ statement will temporarily re-

define the descriptor field during the execution of many of the FAMULUS programs, reverting afterwards to the field defined on the file. If a new master file is produced, the descriptor field may be permanently re-defined on the output file by the /DESCRIPTOR FIELD/ statement, otherwise the old file descriptor field is assumed to hold good by default.

The delimiter or break character is often taken as comma by default, and this information is not carried on the file. If the break character is not a comma, therefore, the /DESCRIPTOR FIELD/ card will be necessary every time.

The text after the /DESCRIPTOR FIELD/ statement consists of a field name in brackets followed by a break character, also in brackets. The break character is not optional, even if it is a comma; its omission causes some programs to malfunction. The field name must be one of those defined on the old master file, or on the /FIELDS/ card if, as may be in the case of EDIT, there is no old master file. Any character, except colon (:), may serve as the break character.

/SELECT/

This control statement is used to select records out of the input file for processing. The text consists of a list of record numbers (not necessarily in ascending order) separated by commas, the whole list being enclosed in parentheses. Instead of a number, an item in the list may consist of a sequence of consecutive numbers, indicated by separating the first and the last in the sequence by a dash. The list is limited in size to 100 numbers, but a sequence only counts as two numbers, irrespective of its length. For example,

/SELECT/(1-500,575-690,526)

is valid and counts as 5 numbers.

This statement is always optional. If it is omitted, the whole file will be processed unless it exceeds 100,000 records. For processing to continue beyond this limit an appropriate /SELECT/ card is required.

6. DATA MANAGEMENT PROGRAMS

There are four data management programs available: EDIT, OSSIFY, MULTI and KEY.

EDIT is used to insert, delete, or alter citations in the data base and is also used for the original creation of the data base.

OSSIFY will punch out a deck of cards from any FAMULUS-produced file. The cards will be in the standard FAMULUS input format, with the field labels established by the most recent EDIT run. The principal uses are to produce back-up decks in case the file is completely lost and to produce card decks or magnetic tapes of the data base to send to other machines.

MULTI is designed to handle problems arising from multiple entries in a field - multiple authorship is a typical example. When two or more authors' names appear in the author field, a listing of the file (sorted by author) will not be a complete author catalogue, unless the file is first multiplied to produce a complete new record for each entry in the author field. This is the function of MULTI. (Some delimiter will have to have been used between the authors' names in the author field.)

The KEY program provides an automatic keywording facility for FAMULUS files, using existing fields as the source for key terms or descriptors. The program will scan one or more fields in a record and generate entries in a specified key field. Descriptors will be truncated to 40 characters.

6.1 EDIT

6.1.1. Creating an original master file

The program creates an internal FAMULUS file from card input and writes the file to the file indicated by the FILEOUT parameter in the FAMEDIT procedure. In this case, the /ID/, /FIELDS/, and /DESCRIPTOR FIELD/ control statements provide an identifier for the master file, the list of field labels, and the name of the descriptor field, if any. An /ORIGINAL/ control statement must be present to indicate that an original file is being created and that there is no old master file. The last control card must be /CITATIONS/, followed by the data cards for EDIT. Note that when an

original file is being created a /FIELDS/ card is necessary.

The usual control cards for an ORIGINAL run of EDIT are:

```
/ID/ data base name
/FIELDS/(....,....,....,....)
/DESCRIPTOR FIELD/(descriptor field name)
/ORIGINAL/
/CITATIONS/
        Citations (maximum 4000 characters each)
```

6.1.2. Updates

Once the file has been created fresh data may be added by omitting the /ORIGINAL/ control statement and supplying the existing master on the file indicated by the FILEIN parameter of the FAMEDIT procedure. The file is copied to the file indicated by the FILEOUT parameter of the FAMEDIT procedure and the new data on cards is added to the end of the original data base in the internal format. Corrections to the input file may be made via /REPLACE/ and /DELETE/ control cards as it is being copied. Note that no /FIELDS/ card is required if an old master file is used, unless the fields are to be renamed.

Adding new data to a large sorted operational file may call for a more complicated procedure. The new data can be written onto a temporary file and then sorted into the same order before being merged with the master file. This is more efficient than adding the new records to the master and sorting the whole file after every update.

6.1.3. Changing the file identifier and field labels

If no additions are to be made, the /CITATIONS/ control card is omitted and operations are limited to changes to the input file. /NEW ID/ renames the new master file. A /FIELDS/ card may be used to change the field names in this way; all the other programs require the labels on the /FIELDS/ card to match the old master file. Note that EDIT cannot be used like SORT to change the order of the fields in the record, nor can the number of fields be changed. It is wise, therefore, when creating the file, to declare more fields than are immediately required using dummy names which can be changed later when the need for another field arises. For example, a spare field may be needed as a key field for the KEY program. Dummy fields

incur virtually no time or space penalty and can be ignored during data preparations.

6.1.4. Deleting records

Complete records are deleted from the file by the /DELETE/ control card, similar in syntax but opposite in semantics to /SELECT/. A list of up to 500 record numbers is acceptable to /DELETE/. If two numbers in the list are separated by a hyphen instead of a comma they represent an inclusive sequence of records beginning at the first and ending at the second.

6.1.5. Corrections

Corrections are made by means of the /REPLACE/ statement. Up to 100 corrections are permitted in each run. The /REPLACE/ only occurs once, and is followed by details of each replacement in free format. Every change requires four pieces of information: the record number or range of numbers, in brackets, the name of the field which is to be changed, also in brackets, the text which is to be replaced, and the new material which is to take its place. Three asterisks are used to delimit the beginning of the first text, the end of the first and beginning of the second text, and the end of the second, respectively.

```
/REPLACE/(1)(AUTH)*SMITHH*SMITH*
(2)(TITL)*MATHIMATICAL*MATHEMATICAL*
(5)(ABST)**THIS PAPER INCLUDES A DISCUSSION OF
CURRENTLY IMPLEMENTED LINEAR PROGRAMMING TECHNIQUES
AVAILABLE FOR THE CDC 7600.*
(120-127)(AUTH)*MACPHEE,*MAC PHEE,*
```

These cards change SMITHH to SMITH in the author field of record 1. MATHIMATICAL to MATHEMATICAL in the title field of record 2, add an abstract to record 5, and change author's name from MACPHEE to MAC PHEE in eight consecutive records.

Regular changes for example, an alteration to a descriptor, are easily made by specifying a range of records. If the whole file is to be systematically changed, the upper limit of the range need not be specified exactly; any number greater than the number of records in the file will suffice. Note that it is permissible to specify a range, as in the last example, but a list of record numbers is not permitted. The corrections do not have to

be relevant to every member of the sequence of numbers.

When numerous corrections are to be made the 'one-card-per-correction' approach is most convenient. If the text extends beyond Column 80 it is continued in Column 1 of another card. The corrections can also be punched as a string, as follows:

```
/REPLACE/(1)(AUTH)*INCORRECT INFORMATION*CORRECT
INFORMATION*(2)(TITL)**INFORMATION TO BE ADDED*(5)
(AUTH)*MATERIAL TO BE DELETED**(9)(ABS)
**NEW MATERIAL TO BE ADDED*(25)(AUTH)*INCORRECT*
CORRECT*(25)(TITL)*INCORRECT*CORRECT*
```

Corrections should always be made to the latest version of a file, and the corresponding most up-to-date listing should be used to look up the numbers of the records to be changed. Corrections need not be input in ascending order of record numbers. The program makes one pass through the input file, taking each record in turn and scanning all corrections to find any that apply to it. The cited field is then scanned for the text between the first and second asterisk, and this is replaced by the text between the second and third asterisk. The length of the two texts need not be the same.

If the text to be replaced occurs more than once in the field, the replacement is performed each time. This feature is a convenience when multiple replacements are desired, but it can lead to accidental alterations unless care is taken to guard against ambiguity. This can usually be done by including sufficient context.

If there are two separate changes to be made within the same field, with considerable information separating them, make two replacements quoting the record number and field label each time.

6.1.6. Deletions and additions to a field

Deletions can be made by putting a string to be deleted between asterisks and punching the third asterisk immediately after the second, which means that the replacement text is a null or empty string. Material can be added to the end of a field by punching two asterisks before and the third after it, which means that a null string is deleted.

The use of * here suggests that they should be avoided as text characters.

6.1.7. Printed output

The /PRINT/ statement is used to control the printing of the file. The text on this card is similar to /SELECT/; a list of up to 100 numbers identifying records or sequences of records to be printed. The record numbers correspond to the output file. If no /PRINT/ card is supplied the default actions are as follows: records from an input file are not printed unless changes are made in them, and on data input, with a /CITATIONS/ card, the first 1000 records only are printed automatically. Printing of records can be suppressed altogether by /PRINT/(0).

Records are printed with explicit field labels as in GALLEY when the /PRINT BY FIELDS/ statement is used. This is the most convenient format for making corrections, and no other format is permitted with EDIT.

The maximum number of characters printed on a line is 128, which is also the default. The /WIDTH/ card reduces the line width to a number specified in brackets after the /WIDTH/, for example,

```
/WIDTH/(70)
```

This card reduces the line width to 70 characters. The minimum width is 20 characters. FAMULUS never ends a line in the middle of a word, unless the word is too long to fit on a line. Lines are left with spaces at the right-hand side if not completely filled.

6.1.8. Sample control cards for EDIT

```
/ID/DRACUNCULUS BIBLIOGRAPHY
/FIELDS/(AUTH,TITL,JRNL,PAGN,YEAR,LANG,SRCE,KEYW,BLNK,ABST)
/DESCRIPTOR FIELD/(KEYW)(,)
/ORIGINAL/
/CITATIONS/
```

6.2 OSSIFY

6.2.1. Description

Besides punching out a deck of cards, under certain conditions, it may be less expensive to use OSSIFY for the correction of a badly-organized

file, particularly if the file has long fields, such as abstracts or text-storage which must be deleted or changed extensively. For instance, if for some reason many citations had incorrect abstracts or if the abstracts were punched with the wrong citation, or if several fields in each citation had errors, it would be far more economical to have the faulty records punched out. Then, by making the corrections on these cards and deleting the original citations from the file an additions run could be made re-adding the corrected citations to the file.

OSSIFY recognizes only the /ID/ statement and the /SELECT/ statement which indicates the records to be punched. The record numbers should directly follow the /SELECT/, enclosed in parentheses. If this statement is not present, the entire file will be punched.

6.2.2. Sample control cards for OSSIFY

/ID/INPUT FILE IDENTIFIER

6.3 MULTI

The identifier of the old master file Specified by the FILEIN parameter of the FAMMULTI procedure must agree with the name on the obligatory /ID/ control card. The new master file identifier will be the same unless specified otherwise on a /NEW ID/ control card.

6.3.1. The multiplication field

The input file may be multiplied on one field only, but any field may be used. A record is generated in the output file for each entry in the multiplication field, which is specified either by a /FIELDS/ card containing only one field label or by a /DESCRIPTOR FIELD/ card, depending upon whether entries are to consist of separate words or of character strings bounded by a delimiting character. Any character may be defined as the delimiter, the default being a comma. A /DESCRIPTOR FIELD/ card will override any previously-defined descriptor field. If neither a /FIELDS/ nor a /DESCRIPTOR FIELD/ card is supplied the previously-defined descriptor field becomes the multiplication field, and if none exists the program terminates with an error message. All the fields in the output record remain the same as in the input record except the multiplication field, which contains only one of the words or phrases in the original. If the multiplication field in any record is empty no output record is created; this is equivalent to

multiplication by zero.

6.3.2. Printed output

Apart from the usual control card listing, etc., the printed output from MULTI is limited to the first ten records of the output file. This should normally be sufficient to check the operation of the program. To obtain more (or less) output the required number of records should be specified by means of a /PRINT/ control card.

Records are printed in the fields format, with the record number on the left and on the right the number of the input record from which it was derived. The number of characters printed per line can be reduced from the normal 128 by means of the /WIDTH/ card. The minimum is 20.

6.3.3. Sample control cards for MULTI

```
/ID/DRACUNCULUS BIBLIOGRAPHY
/DESCRIPTOR FIELD/(AUTH)(,)
/PRINT/(75)
/WIDTH/(80)
/SELECT/(1-2000)
```

6.4 KEY

6.4.1. The key field

The field which is to receive the key terms is specified on the /KEY FIELD/ card by its label enclosed in parentheses. Only one field label is permitted, and it must be one of the field labels already defined by EDIT when the old master file was created, though the field need not necessarily contain any information yet. When creating a new file it is wise to predefine more fields than are immediately required, just in case an unforeseen need like this arises.

If the key field of any record is not empty, entries will overwrite the material already there unless a key field is specified as one of the source fields (see below). No entry is duplicated in the key field, even if it occurs more than once in the source fields.

Terms are separated in the key field by a single blank unless a separator is given on the /KEY FIELD/ card. Any single character may be speci-

fied, enclosed by parentheses, following the field label. When a separator is supplied it is inserted followed by an extra blank between terms in the key field. Key terms will be truncated to 40 characters if they are longer. It is often possible to 'cheat' by including extra delimiters and having the key field separator as a blank.

The /KEY FIELD/ card is obligatory.

6.4.2. The source fields

Terms are identified as in MULTI, that is, strings bounded by a delimiter when scanning the descriptor field, and single words when scanning the other fields. Unlike MULTI, however, KEY can operate on more than one field.

The fields to be used as the source for single word terms are specified on a /FIELDS/ card, the descriptor field serving as the default. The /DESCRIPTOR FIELD/ card is also available to define a source field and delimiter for terms consisting of bounded strings. This field becomes the implicit descriptor field of the new master file, irrespective of the old master file descriptor field.

6.4.3. Stop and go lists

KEY has an internal dictionary to store terms which are input via control card statements. During processing of each record every term in the source fields is checked to find out whether it is in the dictionary or not. It is then only transferred to the key field depending upon whether the dictionary is considered to be a stop list of unwanted terms or a go list which excludes all other terms.

For example, the dictionary may represent a controlled thesaurus, that is, a go list of terms which are required. In this case, a /GO LIST/ control statement is used to introduce the thesaurus, which is punched on cards in exactly the same form as for the /VOCABULARY/ statement of COUNT, with terms separated by commas. When operating with a go list the program rejects any words it finds which are not in the list.

Alternatively, KEY may be made to work with a stop list by means of a /STOP LIST/ control card containing unwanted terms each followed by an asterisk. When the program operates in this mode these terms are rejected, but all terms not found in the dictionary are placed in the key field.

/GO LIST/ and /STOP LIST/ statements are mutually exclusive. If neither statement is used the program operates as if it were in stop list mode with an empty dictionary, with the result that all terms are transferred.

The syntax of stop and go lists has deliberately been made compatible with the punched card vocabulary provided by COUNT, so that these cards can be used for KEY with minimal changes to commas or asterisks. Starred words may be left in a go list, and they are ignored in order not to waste space in the dictionary. Similarly, words followed by a comma in a stop list are not added to the dictionary.

6.4.4. Synonyms

There is also a feature which limits the number of terms required on the thesaurus for the file. It is possible to define sets of two or more terms to be synonyms, one of which is considered the preferred synonym. When one synonym is found the preferred synonym of the set is entered in the key field.

The /SYNONYMS/ statement is used to specify synonyms. Each preferred term is followed by an equals sign and a list of synonyms separated by commas. There is a comma before the next preferred term.

6.4.5. Printed output

Apart from a listing of control cards and file identification information the printed output from KEY is limited to the first ten records, unless a /PRINT/ control card is used to vary it.

6.4.6 Sample control cards for KEY

```
/ID/DRACUNCULUS BIBLIOGRAPHY
/NEW ID/KEYED DRACUNCULUS BIBLIOGRAPHY
/FIELDS/(TITL,ABST)
/KEY FIELD/(KEY)(,)
/STOP LIST/A*AN*THE*OF*FOR*OR*IN*ON*TO*BY*FROM*
INTO*SINCE*THEN*WORM,DISEASE,PARASITE*THEREFORE*
/SYNONYMS/HELMINTH=HELMINTHIC,HELMINTHOLOGICAL,
HELMINTHOLOGIST,HELMINTHOLOGY,
INFECTION=INFECTED,INFECTIONS
/PRINT/(50)
/SELECT/(1-30,34,36,47-91,94-100)
```

7. FILE MANIPULATION PROGRAMS

There are two file manipulation programs: SORT and MERGE.

SORT re-arranges the records in a file into alphabetical order, using any field or combination of fields as the sort key. The sort key is the portion of the record which is used for alphabetical comparisons.

It is not possible to define a part of a field as the sort key. This limitation should be considered when designing the structure of the record. For example, in the case of a bibliographic file, if both author and title catalogues are to be produced by means of SORT/GALLEY jobs author and title information should not be incorporated in a single field. Conversely, if the date will never be required as a sort key it could be included, if desired, in one of the other fields.

MERGE will only accept input files having the same number of fields; however, the field names do not have to be the same. The master file and the additions file will be merged and written onto the new master file. The field labels on the new master file will be the same as those occurring on the old master file. The /ID/ will also be taken from the old master file, unless a new /NEW ID/ statement is included.

Any record which is not in correct alphabetical order will cause the MERGE program to halt processing, reject the remainder of the job, and print an error message. Therefore, only tapes which have been ordered in the same way by the SORT program should be merged. Unless specifically requested, MERGE does not print out the file. It prints the /ID/ of the old master file, the /ID/of the additions file, and the /ID/ of the new master file, along with the field labels for each. This will be followed by the number of records on the new master file and the familiar three large OK's.

The new master file can be printed using GALLEY.

7.1 SORT

7.1.1. Sort key comparisons

The order of two records is decided by comparing their sort keys character by character from left to right. As soon as a mismatch between a pair of characters is found the records are ordered on the basis of an internal collating sequence.

7.1.2. The collating sequence

It should be explained that characters are represented within computers by numeric codes which differ from machine to machine. FAMULUS, however, converts to its own internal collating sequence so that the same sort order can be obtained irrespective of the machine on which the program is run. A Table giving the FAMULUS character set may be found in Appendix A.

Blank precedes other characters. Care must be taken in certain cases to pad the data with blanks or noughts to ensure correct alignment of the keys. For instance, 15 will come before 5, but after 05. FAMULUS removes leading blanks, so another character must be used for padding at the beginning of a field.

SORT requires a master file in FAMULUS internal form, i.e. as produced by, for example, the EDIT program. It creates a new sorted file and the old master file remains unchanged. The sorted file is not printed. Only a listing of the program control cards and technical information such as the number of records processed is given, plus an indication that processing was successfully completed. GALLEY may be used to print the new file. During sorting, the words A, AN, or THE at the beginning of phrases are

disregarded in deciding alphabetical order, but they are printed.

7.1.3. Program control cards

The /ID/ control card contains an identifier which must match that of the old master file. This control card is obligatory, and must precede the rest, which are optional.

The /FIELDS/ card defines the sort key. Any or all of the field labels in the old master file may be given, in any order. When records are compared, if the first field is identical the second or subsequent fields are used to determine precedence.

On the new master file of sorted records the fields are ordered as on the /FIELDS/ card, and any fields not specified remain in the order already established in the old master file. If this card is absent, the implicit order of fields in the old master file is assumed. If GALLEY is used to print the sorted file, the fields will be printed in the new order established by SORT, unless another /FIELDS/ card is used for GALLEY.

The /SELECT/ card may be used to break large files into sections for more efficient sorting. It is recommended to sort up to 3000 records at a time and then merge the sorted sections. If this card is absent the whole file is sorted.

A /NEW ID/ card may optionally be used to specify a new identifier for the new master file.

7.1.4. Sample control cards for SORT

```
/ID/DRACUNCULUS BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR)
/SELECT/(1-2500)
/NEW ID/FIRST HALF
```

7.2 MERGE

MERGE provides the updating facility for FAMULUS-controlled files. It merges any 'master file' with an 'additions file'. MERGE can be used for two purposes - to update files by merging information from two files onto one new master file, or to join files belonging to two or more persons.

For instance, the various members of a working group may maintain separate literature files, but on occasion they may wish to make a master listing of all these files.

7.2.1. Program control cards

An /ID/ card for the master file (FILEIN parameter) and an /ID/ card for the additions file (FILEIN2 parameter), in that order, are mandatory.

A /NEW ID/ should be included if a different /ID/ from the old master file is desired for the new master file.

The new master file is not normally printed in full. If no /PRINT/ card is included ten records will be printed. Otherwise the records specified will be printed. A /PRINT/(0) card will suppress printing of records altogether.

A line width of 128 characters is normally used, but this may be altered to any value in the range 20 to 128 by the use of a /WIDTH/ card.

7.2.2. Sample control cards for MERGE

```
/ID/FIRST HALF
/ID/SECOND HALF
/NEW ID/DRACUNCULUS BIBLIOGRAPHY - AUTHOR CATALOGUE
/PRINT/(11-21)
/WIDTH/(65)
```

8. INFORMATION RETRIEVAL PROGRAMS

There are four information retrieval programs: GALLEY, INDEX, SEARCH, and KWIC.

GALLEY will print a FAMULUS file or selected parts of a file with a choice of several formats.

The index which the program INDEX provides is an alphabetical list of terms in the descriptor field of a file. Each term is accompanied by a

list of references to the records where it occurs, separated by commas. A record is referenced by its position in the master file, which is printed as an integer to the left of the record in the GALLEY listing of the file. The index is normally used, therefore, in conjunction with a complete listing for indirect access to records via an attribute other than the one which the file is ordered, such as, for example, subject categories.

SEARCH selects records from a FAMULUS file which satisfy a detailed request for information given on a /SEARCH/ card.

The KWIC program (Key Word In Context) generates a concordance for a FAMULUS file. An alphabetical list of words is produced, with each word centred in a line containing as much as possible of the context of its immediate left and right. Each line also contains a number referencing the record from which it comes.

8.1 GALLEY

8.1.1. File identification

The master file to be printed must be identified by an /ID/ control card with the correct file identifier. This is the only obligatory control card, and it must precede any others.

8.1.2. Printing portions of a file

Selected portions of a file may be printed by specifying the required records on a /SELECT/ card. This contains a list of record numbers separated by commas. Sequences of records are denoted by a dash instead of a comma between two numbers. The whole list is enclosed in parentheses.

Note that the record numbers need not be in order, but the records will be printed in file order, not the order on the /SELECT/ card.

8.1.3. Printed output formats

The standard default format prints the fields in sequence without starting a new line for each field, and also drops the field labels, simply leaving three spaces between fields. The records are numbered. This is a relatively concise form, but it can be confusing if records vary considerably in the fields which they possess.

Alternative formats are obtained by the /PRINT BY FIELDS/ or /PRINT BY SUBJECTS/ control cards.

The /PRINT BY FIELDS/ card produces an output with field labels, each field beginning on a new line, like the output from EDIT. The records are numbered.

The /PRINT BY SUBJECTS/ card will print a file without field labels but with subject headings. The first field named in the /FIELDS/ card if printed as a heading above the remainder of the record. The first field of subsequent records is only printed if it differs from the previous record, otherwise it will be truncated.

/PRINT BY FIELDS/ and /PRINT BY SUBJECTS/ are mutually exclusive options. If none of these cards is present the standard print format is produced by default.

8.1.4. Omitting fields

The /FIELDS/ card in GALLEY is used to define which fields of the record are to be printed, giving the user the option of omitting fields. The fields will be printed in the order in which they appear on the /FIELDS/ card, regardless of their order in the master file. If no /FIELDS/ card is supplied all the fields of the record are printed in the order implicit in the file.

8.1.5. Page heading

The /NEW ID/ card is available in GALLEY, but since no new master file is produced by this program it is only used to supply a heading which will be printed at the top of each page of output. If this card is omitted the master file identifier is printed as a page heading.

8.1.6. Page width

The required page width may be obtained by means of a /WIDTH/ card. It may range from 20 characters up to the normal 128. As a guide, ten characters occupy an inch on many line printers.

8.1.7. Sample control cards for GALLEY

```
/ID/FIELD REPORTS
/FIELDS/(NAME,NUMB,OCC,AD,DATA)
/SELECT/(1-5,15-20,25,30,11,8,23,21)
/WIDTH/(90)
/PRINT BY FIELDS/
/NEW ID/ SELECTED REPORTS.
```

8.2 INDEX

8.2.1. The descriptor field

The INDEX program is only designed to operate on the descriptor field of a file to produce a thesaurus of descriptor terms. The descriptor field may have been previously defined, most likely by EDIT when the file was created, but if not a /DESCRIPTOR FIELD/ control card is required. This card is also necessary if the default delimiter, comma, is to be overridden by another character.

An index of any other field can be produced by using a /DESCRIPTOR FIELD/ card to define it temporarily as the descriptor field with an appropriate delimiter. This overrides the previously nominated descriptor field, if any, as far as the present run of the INDEX program is concerned.

8.2.2. Printed index

The /NEW ID/ control card is available to specify a heading for the printed output. Descriptor terms longer than 40 characters are truncated during printing of the index. Terms are printed in alphabetical order, each with its list of references on the following line or lines.

8.2.3. Punched card index

An optional /PUNCH/ control card causes the index to be punched out on cards to produce a manual card index. This facility is particularly useful where a complete index of the whole file cannot be made because the file contains too many index terms.

In this case the file can be indexed in sections by including /SELECT/ control cards in separate runs to select successive sets of records. The punched card output from the different runs can then be merged, discarding duplicate descriptor headings, to produce a master index to the whole file.

INDEX can handle up to 2000 different descriptor terms each referenced up to 2500 times.

8.2.4. Sample control cards for INDEX

```
/ID/MASTER FILE
/NEW ID/ INDEX OF RECORDS 901 - 2000
/SELECT/(901-2000)
/DESCRIPTOR FIELD/(KEY)(;)
/PUNCH/
```

8.3 SEARCH

8.3.1. Searching subsections

The /SELECT/ card may be used to indicate the record numbers which are to be searched. It is sometimes possible to use this facility to narrow the area of search to one or more subsections of the complete file. For instance, if a file is in chronological order the record numbers delimiting any particular period of interest will be known from the GALLEY listing. Selecting a subsection of a file in this way obviously cuts down processing time but it has greater potential in conjunction with the /SEARCH/ formula to help define the set of records to be retrieved. In many cases, however, the attribute by which the file is ordered is not among the retrieval criteria, and there is then no alternative but to search the whole file. In this case no /SELECT/ card is required.

8.3.2. Defining the fields to be scanned

One or more fields of each record are scanned to determine whether it meets the criteria for retrieval, and a /FIELDS/ control card is used to list the relevant field labels. The fewer the fields that have to be scanned the more efficient the search will be. If this card is absent the descriptor field is used, and if no descriptor field has been defined the program terminates with an error message. If no descriptor field was defined when the file was first created a /DESCRIPTOR FIELD/ control card may be used to define one for the purpose of the SEARCH program, or if the descriptor field already exists, to change it. Remember that the method of identifying terms depends upon whether the field is a descriptor field or not.

8.3.3. Printed output

The printed output produced by SEARCH is normally a listing in full of all the records that satisfy the /SEARCH/ requests. The amount of printing can be reduced if desired by a /NUMBERS/ control card which results in record numbers only being listed. The /WIDTH/ card is available for changing the line width of the output. The /NEW ID/ card enables the page heading to be specified if it is to be different from the old master file /ID/.

The listing is printed in the GALLEY default format, by default, but can be obtained in fields by including a /PRINT BY FIELDS/ control card.

8.3.4. Creating a new file

If required, a new master file containing just the retrieved records can be created by using a /WRITE TAPE/ control card, and its /ID/ will be the same as the page heading of the printed output and is sent to the file indicated by the FILEOUT parameter of the FAMSRCH procedure.

Note that this is a new file in FAMULUS internal format, which is useable with other FAMULUS programs; it is different from that which is sent to the print file (which contains a readable listing of the retrieved records, page headings, number of records searched, etc.).

8.3.5 The search formula

Retrieval of records from the input file is achieved by means of a formula on the /SEARCH/ control card. The search formula is composed of terms combined with Boolean operators, punched in free format.

Terms are defined according to the usual convention, broadly speaking, as phrases bounded by a delimiter in a descriptor field and words elsewhere. In this program, however, there are also two additional features, truncation and qualifacation, which are described later.

The three logical operators permitted in a search formula are and, or, and not, represented by &, ↑, and # respectively.

Use of the and operator implies that the user only wants to retrieve records in which all the terms joined by and are present. If a single one of the specified terms is missing, the record will not be retrieved. The

or operator will retrieve records in which any of the specified terms are present. The not operator indicates that records containing the term which follows it will not be retrieved.

Brackets should be used to avoid ambiguity as to the order in which the operations should be performed. There is no limit to the depth to which brackets may be nested. Limits on the size of the search formula require that not more than fifty different terms (or sixty, including duplicates) for a total of 150 terms and operators be used in one formula, there is a limit, but it is large!

8.3.6. Vocabulary lists

The descriptor field usually contains a controlled vocabulary and there is then no difficulty in choosing the correct terms, but for searching any field with an uncontrolled vocabulary, it is useful to refer to a list of the vocabulary in the field produced by COUNT or VOCAB.

8.3.7. Truncation of terms

A title field, for instance, will contain an uncontrolled vocabulary, because you had no say in the choice of words used by the original authors in their titles. Several forms of the same word may occur, for example, REGENERATE, REGENERATING, and REGENERATION. In order to retrieve records containing any of these forms it is permissible to truncate on the right and to specify a common substring, for example, REGENERAT or even REGEN, though by using too short a form you run the risk of retrieving irrelevent words such as REGENT. Truncation on the left is not permissible.

8.3.8. Qualification of terms

The terms are normally searched for in the fields specified by the /FIELDS/ card. Sometimes there is a need to include a term from another field not already being searched, (for instance, an author or date). This is done simply by following the term with the field label enclosed in parentheses, for example,

SMITH(AUTH) or 1948(DATE)

8.3.9. Note

At least one /SEARCH/ card is obligatory. Not only is it the means by which criteria for record retrieval are expressed, it is also the signal

for processing of the master file to begin and must follow all the other control cards for SEARCH.

8.3.10. Multiple searches

Multiple searches of the file may be carried out by supplying more than one /SEARCH/ card. The input file is re-scanned for each search, and while there is no limit to the number of searches, the cost is obviously proportional.

8.3.11. Sample control cards for SEARCH

```
/ID/MASTER FILE
/FIELDS/(TITL,KEY)
/SEARCH/SOIL STABILITY & (EROSION ! SLIDES) & 196(DATE)
/SEARCH/(PANCHROMATIC ! INFRARED ! THERMAL ! RADAR)
& AERIAL & 70MM
/SEARCH/VOLCAN & EURO & @ (EXTINCT ! INACTIVE)
```

8.4 KWIC

8.4.1. Specifying fields

As in the case of the KEY program, any field or combination of fields may be specified by means of a /FIELDS/ card, and if no /FIELDS/ card is supplied the implicit descriptor field serves as the default. A /DESCRIPTOR FIELD/ card defines a field containing multi-word terms, and also the delimiting character which separates them.

8.4.2. Stop and go lists

A /STOP LIST/ or a /GO LIST/ card may be used to restrict the number of terms to be concorded. Since a line of output is generated for every term token found in the specified fields of the file the amount of printed output may be very large unless the frequently occurring terms are stopped.

Alternatively, if only some known terms are of interest they can be put on a /GO LIST/ statement which will exclude all others.

All terms to be stopped must be terminated with *, regardless of what list they are in.

8.4.3. Sample control cards for KWIC

```
/ID/UMRCC PROGRAM CATALOGUE.
/NEW ID/KWIC CONCORDANCE OF PROGRAM DESCRIPTION FIELD.
/FIELDS/(DESC)
/GO LIST/SYSTEM,MODEL,PROCESS,CONTROL
```

9. THESAURUS CONSTRUCTION PROGRAMS

There are two thesaurus construction programs: COUNT and VOCAB, however only the COUNT program has been implemented at Daresbury. The COUNT program lists the vocabulary in specified fields of a FAMULUS file in alphabetical order with a count of the number of times each item occurs.

The VOCAB program performs the same functions except that word frequencies are not counted.

Though requiring less space than COUNT, VOCAB is relatively undeveloped, and parts of the following discussion do not apply to it, notably the /STOP LEVEL/ feature. A punched card vocabulary list is produced automatically, and the /PUNCH/ control card is not available. The /NEW ID/ and /DESCRIPTOR FIELD/ control cards are also unavailable. It is assumed that COUNT will normally be preferred.

9.1 COUNT and VOCAB

9.1.1. Uses of vocabulary lists

Vocabulary lists have a number of uses, for example, for data validation, search formula construction, thesaurus building, and simple research on the information contained in the field.

9.1.2. Error detection

In the early stages of data base construction, a vocabulary list will throw errors, such as misspellings, into prominence, so providing a checklist of errors for use in data correction. Proof-reading is not eliminated, because the errors still have to be located, but they can at least be traced to one of the fields, which narrows the area of search. INDEX and KWIC can also be useful for error detection; they also give record numbers directly.

9.1.3. Breakdown of vocabulary by fields

The fields whose vocabulary is required are specified on a /FIELDS/ control card, the descriptor field being the default. A /DESCRIPTOR FIELD/ card will override any existing descriptor field. This feature allows vocabulary items consisting of phrases separated by any specified delimiter.

Vocabulary lists are often required to help construct search formulae, and then it is useful to know the fields in which particular items occur. A breakdown of the vocabulary by fields is not provided, however, and the only way to achieve this result is to run the program repeatedly with a /FIELDS/ card specifying a different field each time. Otherwise if more than one field is specified, a combined list is produced which will not show which field a word came from.

9.1.4. Stop-list compilation

Vocabulary lists are also useful for constructing a controlled thesaurus of index terms for use in the descriptor field. If the /PUNCH/ control card is supplied, a punched card deck of the vocabulary is automatically produced, consisting of a list of words in alphabetical order, separated by commas. These 'cards' may be modified and used in a later run following a /VOCABULARY/ statement in order to specify a stop list of trivial words. Trivial and non-trivial vocabularies are then printed in separate lists. The modification to the card deck simply involves replacing the comma following every trivial word with an asterisk. This deck can also be used as a STOP or GO list by the KEY program.

There is also provision for automatically transferring words from the vocabulary to the trivial list if they occur too frequently, for example,

/STOP LEVEL/(25)

This card will put any word occurring 25 times or more into the trivial word list in the printed output, and will mark it as trivial by an asterisk in the punched card output, if any.

A /VOCABULARY/ card should contain at least one trivial word which actually exists in the master file. (This is necessary as there is an error in the current program.)

9.1.5. Vocabulary lists

If any words are marked as trivial a list of trivial words in alphabetical order is printed first. The non-trivial vocabulary list is printed next, followed by the complete dictionary in order of word frequency with trivial words distinguished by a following asterisk. The word lists are printed in three columns across the width of the page. Each word, or phrase in the case of descriptor field vocabulary list, is followed by its actual and percentage frequency of occurrence. If the actual frequency exceeds four decimal digits, asterisks are printed in lieu.

Any item which is too long to be accommodated in the column is truncated to 33 characters. This is not likely to occur except when counting a descriptor field, because the longest word in the Oxford English Dictionary consists of 28 letters only. The VOCAB program provides no word frequency count and therefore allows up to 40 characters before truncating. The extra seven characters may sometimes be just enough to resolve ambiguity which is the main occasion for using VOCAB in preference to COUNT.

9.1.6. Statistics of word types

Dictionary statistics are provided separately for the words in the stop list and for the remainder of the vocabulary. Both sets of statistics consist of a table of word-length distributions followed by the number of entries. Their average and maximum lengths, and details of the dictionary size. The word-length distribution table comprises four rows and ten columns. Each value in the table represents the number of dictionary entries having a certain number of characters. The first row gives the values for terms having one to ten characters; the second, for 10 to 20 characters; the third, for 21 to 30 characters; and the fourth, for 31 to 40 characters.

9.1.7. Statistics of word tokens

The distinction between types and tokens is fundamental. Word types hardly require explanation; they are simply the different words found in the master file and entered into the program's internal dictionary. When different instances of a single word type are encountered in the file they constitute separate word tokens.

Statistics of word tokens are provided, as for word types, both for trivial and for vocabulary words. In each case the distribution of word

frequencies is tabulated under six headings across the page as follows:

R	The rank of N, increasing by one on successive lines.
N	The number of tokens of any one word type, in decreasing order.
F(N)	The number of word types having N tokens. This tends to increase as N decreases.
SIGMA(F(N))	The cumulative sum of word types having N or more tokens.
N*F(N)	The number of word tokens of rank R.
SIGMA(N*F(N))	The cumulative sum of word tokens of the SIGMA(F(N)) most frequent word types up to an including rank R.

9.1.8. Dictionary capacity

The dictionary limits on vocabulary size are either 4500 entries or 45000 characters. If either limit is exceeded the program will not proceed with the reading of the master file, but will output the results obtained up to that point.

This situation can easily arise with large files, especially if several fields are being scanned in a single job. This is another reason for scanning one field per run, though even this may not solve the problem for very large files. In such a case the vocabulary of the unprocessed portion of the file may be obtained by a subsequent run using the /SELECT/ card. The starting point for the second run is obtained form the figure for the last citation inspected in the output from the first run.

9.1.9. The last citation inspected

Following the file identification information on the printed output the number of the last citation which was inspected is given. This will be the last record in the file, or the last record specified on a /SELECT/ card, for the last record read before the capacity of the program was exceeded.

9.1.10. Sample control cards for COUNT

```
/ID/MASTER
/NEW ID/ABSTRACT VOCABULARY LIST
/FIELDS/(ABST)
/PUNCH/
/VOCABULARY/A*THE*AN*CRISIS,OF*IN*
/STOP LEVEL/(20)
```

APPENDIX A

FAMULUS Character Set

Precedence	Characters	Precedence	Characters
0	blank	32	F
1	.	33	G
2	,	34	H
3	(	35	I
4	)	36	J
5	+	37	K
6	-	38	L
7	*	39	M
8	/	40	N
9	$	41	O
10	£	42	P
11	=	43	Q
12	<	44	R
13	>	45	S
14	"	46	T
15	:	47	U
16	'	48	V
17	]	49	W
18	↑	50	X
19	&	51	Y
20	←	52	Z
21	!	53	0
22	;	54	1
23	%	55	2
24	?	56	3
25	#	57	4
26	@	58	5
27	[	59	6
28	B	60	7
29	C	61	8
30	D	62	9
31	E	99	all others

Note on the Character Set

The characters are listed in the FAMULUS internal collating sequence, that is, in the order of precedence used by SORT. Characters which are not defined in this list may be used (if they are available), and they come equal last in the order of precedence.

APPENDIX B

Table of Program Control statements

	EDIT	OSSIFY	SORT	MERGE	MULTI	GALLEY
ID	*	*	*	**	*	*
FIELDS	*		*		O	O
SELECT		O	O		O	O
NEW ID	O		O	O	O	O
PRINT	O			O	O	O
CITATIONS	O					
DELETE	O					
REPLACE	O					
ORIGINAL	O					
SEARCH						
VOCABULARY						
DESCRIPTOR FIELD	O				O	
WIDTH	O			O		O
PUNCH						
NUMBERS						
PRINT BY FIELDS						O
STOP LEVEL						
SYNONYMS						
KEY FIELD						
PRINT BY SUBJECTS						O
STOP LIST						
GO LIST						
WRITE TAPE						

* Obligatory O Optional

Table of Program Control Statements (Continued)

	INDEX	SEARCH	VOCAB	COUNT	KEY	KWIC
ID	*	*	*	*	*	*
FIELDS		O	O	O	O	O
SELECT	O	O	O	O	O	O
NEW ID	O	O		O	O	O
PRINT					O	
CITATIONS						
DELETE						
REPLACE						
ORIGINAL						
SEARCH		*				
VOCABULARY			*	*		O
DESCRIPTOR FIELD	O	O		O	O	O
WIDTH		O				O
PUNCH	O			O		
NUMBERS		O				
PRINT BY FIELDS		O				
STOP LEVEL				O		
SYNONYMS					O	
KEY FIELD					*	
PRINT BY SUBJECTS						
STOP LIST					O	O
GO LIST					O	O
WRITE TAPE		O				

* Obligatory O Optional