OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION

18263

USER'S GUIDE
TO THE COOPERATIVE RESEARCH DATABASE
FOR THE SUPPORT OF JOINT RESEARCH
IN THE FIELD OF FINE CHEMICALS

rel. 1.0

Vienna, December 1989

TABLE OF CONTENTS

0. INTRODUCTION

This document is a manual dealing with the cooperative database for the support of joint research in the field of fine chemicals. Later on the database will be called PETRO.

The PETRO database is implemented for IBM PC XT/AT and compatible microcomputers. The software is based on the Micro CDS/ISIS, ver. 2.3, package [1] developed by UNESCO, 1989.

Since Micro CDS ISIS is logically compatible with the mainframe CDS ISIS the PETRO databases can be easily transportted from microcomputers to mainframes, and vice versa, if needed.

Logically, the PETRO database can be considered as composed of three interrelated files concerned with

- enterprises
- products
- processes

Moreover, a file containing tress describing chemical processes also takes part in the PETRO database.

In the system a family of tools for special operations has been designed. It refers to:

- setting up relationships between data units representing enterprise, products and processes;
- visualization and manipulation of the structured data, in particular, trees representing processes.

This manual is composed of three basic parts. The first one gives general rules for entering data into worksheets. The second part presents the main functions available for a user of the databases, such as data entry and retrieval. The las part (Chapters 9 and 10) contains some recommendations for a database administrator and information on manipulations dealing with trees. Technical specifications of the equipment and software required to run the [illegible] are given in Appendix 1.

It is believed that the described software is self-explanatory and friendly enough to the user due to its menu-driven philosophy. This manual should be considered as a user reference rather than a detailed document for study, as much information is displayed on the screen. However, knowledge on the usage of the Micro CDS ISIS and its various features and functions is mandatory. Moreover, the

user of the package must be familiar with the chemical issues, in particular, as far as the chemical terminology is concerned. One has to realize that controlling and validation procedures implemented within the PETRO software are limited. therefore, the user is responsible for the quality of input and, as a result, the final output.

It has to be stressed that the manual refers to the PETRO software, rel. 1.0, which is not supposed to be exploited as a final product. This version was conceived as a basis for collecting remarks and comments on the database structure and further development of the software, as well as, to start inputing the data. Next versions are expected in the comming year. Further work is presented in Chapter 11.

This manual is also available on a diskette as the file PETRO.CHI and can be used under the CHI-WRITER text-processor.

1. GETTING STARTED

(a) Installing the PETRO software package

The PETRO software is distributed on 3 diskettes labelled PETRO01, PETRO02 and PETRO03.

The diskettes contain:

PETRO01 - the ISIS program and a batch file INSTALL.BAT
PETRO02 - the MICRO CDS ISIS files and programs
PETRO03 - the PETRO database files and INPETRO.BAT batch file

Installation of the PETRO software is done automatically by the INPETRO.BAT program. However, before invoking this program some preparatory steps must be performed. They are described below.

Step 1

You should be aware that certain screen formatting functions performed by the PETRO software will not be done properly if the ANSI.SYS driver is installed, although all other functions of the system will be executed correctly. Furthermore, you must make sure that your DOS system is set up to support the number of files required by the PETRO software. Both of these requirements are contained in the DOS file called CONFIG.SYS. Switch to the root directory and have this file displayed on the screen doing as follows

TYPE CD\ <CR> (*<CR> stands for pressing the ENTER key.*)
TYPE CONFIG.SYS <CR>

If this file does not exist you must create one containing the following lines

BUFFERS = 20
FILES = 20

Use EDLIN or another text editor to do this.

If the CONFIG.SYS file has been already created check whether the above records are included and that the number of FILES and BUFFERS is at least 20. If not, modify the file by adding the above lines or increase the values. Then check if the file contains the following command

DEVICE = ANSI.SYS

If it exists you should delete it for proper software functioning. However, if some other software package installed on your computer

requires the ANSI.SYS driver, you must keep a copy of the original CONFIG.SYS. before deleting the above command.

We suggest in this case that you keep two copies of CONFIG.SYS, e.g. one called CONFIG1.SYS and the other CONFIG2.SYS. Before using the PETRO software you check out which CONFIG.SYS you have. If it is not the correct one replace it.

Step 2

From the root directory, create a new directory called ISIS, in case it does not exist, then, switch to this. Type

```
CD\              <CR>
MD ISIS          <CR>
CD\ISIS          <CR>
```

Now, you are in the ISIS directory.

Step 3

From the ISIS directory create new sub-directories:

```
MD ISISMENU      <CR>
MD ISISMSG       <CR>
MD ISISWORK      <CR>
MD PETRO         <CR>
```

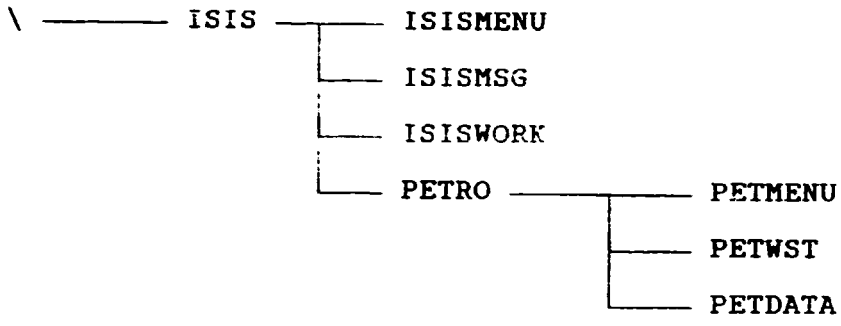
After that attach to the PETRO directory by typing:

```
CD PETRO         <CR>
```

and create again three directories:

```
MD PETMENU       <CR>
MD PETWST        <CR>
MD PETDATA       <CR>
```

As a result we receive the following directory tree:



Step 4

Copy INPETRO.BAT program to the hard disk. To do this, mount the PETRO03 diskette on drive A:, and type

```
CD\ISIS                 <CR>
COPY A:INPETRO.BAT      <CR>
```

Step 5

Run the I program by typing

```
INPETRO     <CR>
```

As this procedure is executed it will ask you to mount each diskette when required.

Step 6

You must now establish an access path to the subdirectory \ISIS. To do this you have to edit the file AUTOEXEC.BAT (or create one if it does not exist) in the root directory, and change or add the appropriate PATH command. Any familiar text editor, e.g. EDLIN, can be used for this purpose. If the PATH command does not exist you have to write it

```
PATH C:\ISIS
```

If such a command already exists you have to add a path only, e.g.

```
PATH C:\DOS;C:\GAMES;C:\ISIS
```

(b) Invoking the PETRO services

To invoke PETRO functions you should type

```
CD\ISIS          <CR>
ISIS             <CR>
```

You will be prompted to provide your identificator (password), which for the database administrator is MSTR and for a standard user USRS.

From this moment the program is driven by menus and worksheets which were designed on the basis of the Micro CDS/ISIS package.

Note that operations performed within the ISIS.EXE program may lead to temporary inconsistencies between the database files. Therefore, the system asks you, when leaving the ISIS.EXE program, whether you want to update the inverted file. The recommended answer is "Y".

Example

```
Database name: PETRO
       0 Records created
       3 Records updated
       0 Records deleted
Do you want to update the inverted file now (Y/N)?  Y
```

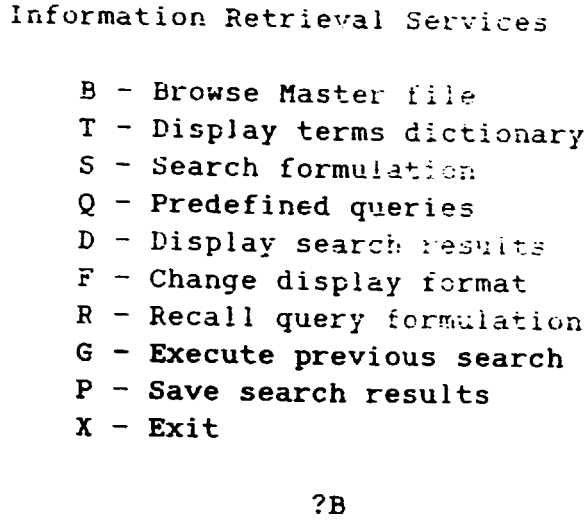


Fig.1. The menu "Information Retrieval Services" - EXGEN

2. BROWSING THE DATABASE

To enable one to have a look at the PETRO database the function "Browse Master file" is provided. In order to use this function you should

- invoke the ISISRET service (Information Retrieval Services) from the main ISIS menu (EXISI) by typing
  S

  The menu EXGEN - "Information Retrieval Service" will be displayed (see Fig. 1) as a result;

- Select appropriate display format. There are 3 formats available in PETRO, corresponding to various types of information, viz.

  * format for enterprises (ENTRP),
  * format for products (PROD),
  * format processes (PROC),

  To select a format you have to choose the option F from the menu EXGEN. As a result you will see the menu shown in Fig. 2. The arrow indicates the format to be used. To select the desired format use down and/or up arrows and press <CR>. To leave the menu press X.

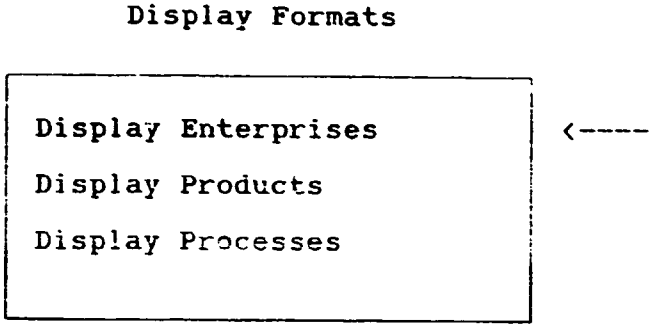


Fig. 2. Display formats available in PETRO

- choose the B option ("Browse Master file") by typing

  B

- after that, you will be prompted to provide the starting MFN (document number) for browsing. At the bottom of the screen you will find the following message

  Starting MFN for browse?

  Type an appropriate number as a sequence of 1-5 digits. Press <CR>

Then the first document chosen for browsing is to be displayed on the screen. To continue browsing press <CR>. To stop the browsing strike the "ESC" key and you will return to the menu "Information Retrieval Services".

3. RETRIEVAL FROM THE DATABASE

In order to perform retrieval you have to use the ISISRET service (see Chapter 2). You can invoke this service from the main ISIS menu (EXISI) by typing

S

The menu EXGEN - "Information Retrieval Service" will be displayed (see Fig. 1) as a result.

You have two ways to formulate and submit your query for processing. The first one is based on the built-in MICRO CDS ISIS search language and available through the option S on the menu EXGEN. The second option contains four predefined query skeletons which have to be provided with actual query key-words by the database user. This option can be employed by selecting the O option from the EXGEN menu.

Option S

After choosing the S option (Search formulation) you can type your query and strike <CR>.

For syntactic rules dealing with the query formulation see the description of the CDS ISIS search language given in detail in the manual [1]. Some information in this respect is provided in Appendix 3.

As a result of searching in your database the system will present you a query report which, for the query FRANCE * CONCRETE + USA, could look as follows

```
Set        1: FRANCE * CONCRETE + USA
P=         2  FRANCE
P=         2  CONCRETE
T=         1 -#2: FRANCE * CONCRETE
P=         3  USA
T=         3 -#3:  #2 + USA
T=         3 - #1: #3
```

This data means that there are 2 access points for the primitive terms FRANCE and CONCRETE, one access point for sub-expression FRANCE * CONCRETE (later referred to as a subexpression #2), 3 access points for the primitive term USA, 3 access points for the expression #2 + USA (later referred to as #3) and 3 access points for the whole query (later referred to as #1), which is identical with the expression #3. Press <CR> to return to the main menu.

If you wish to display the results of the search on the screen you should choose an appropriate display format (see Chapter 2) and next select the D option by typing

D

The hits are displayed on the screen in the same way as during browsing (see Chapter 2).

If you need to recall former query formulations, use the R option by pressing

R

and the system will display the following information: number assigned to each query, the number of hits and query formulations. The number assigned to query(ies) of interest is necessary when you are to pick up the G option (Execute Previous Search). The system prompts you to give this number by displaying

Set

The chosen formulation will be displayed and you can edit it (i.e. correct formal mistakes or make it more precise). If the formulation satisfied you, strike <CR> to execute it. You can again display the search results (using the D option).

Dictionary list **Data Base Name: XXXXX**

- BRUSHING	- CHEMICAL
+ BUILDING	- CHEMICAL ELEMENTS
* BULGARIA	- CHEOPS GROUP
- CALIBRATION	- CIVIL
- CANADA	- COATING

Fig. 3. A part of the a database dictionary.
The query BUILDING * BULGARIA is defined

To facilitate the query formulation the option T (Display terms dictionary) is provided. It allows to display the full dictionary of search terms, in lexicographical order, related to the database in question. You can choose the starting point of the dictionary browsing: (i) if you strike <CR> the dictionary will be presented starting from the very brining; (ii) if you enter any sequence of letters, followed by <CR> then the relevant part of the dictionary will appear. The cursor position is controlled by the arrow ←, →, ↑, ↓ keys; to proceed to the next page strike the PgDn key (note

that it is impossible to return to the previous page).

The term indicated by the cursor can be marked using either ^, + or * keys which correspond to the NOT, OR, AND boolean operators, respectively. In that way the query formulation can actually be facilitated (cf. Fig. 2). To terminate this phase of defining the query press X. You will then be brought to the same place as after choosing the option S in the main menu. Now, you have your query displayed explicitly and you can modify it using the editor available.

Option Q

After choosing the Q option (Predefined queries) you will see the box containing 4 types of queries. The box is presented in Fig. 4. Select a relevant query by means of the moving arrow and strike <CR>. You will see a new screen where your query is displayed again. Now you have to enter a key-word and press <CR>. Note that the $ signe for query truncation is not allowed. Also, you cannot use boolean expressions as queries.

Having processed the query you will see the query search report. To continue press <CR>. In order to display the hits select an appropriate display format (see Chapter 2) and next choose the option D from the menu EXGEN.

It has to be noted that a query processor employs slightly different display formats than those which are available by means of the option F on the menu EXGEN. They are as follows:

query 1 uses format ENTRP0
query 2 - not implemented
query 3 - not implemented
query 4 uses format ENTRP5

NOTE! The processing of the second and the third query from the box has not been implemented yet.

In order to prepare the results of search for printing select the P option from the menu and then you will be prompted to give the save file name. You should answer by typing

PETRO <CR>

The results of retrieval are to be stored in this save file and may be printed out using the ISISPRT service (see Chapter 7) and exported by means of the ISISXCH service (see Chapter 9, p.(e)).

QUERIES

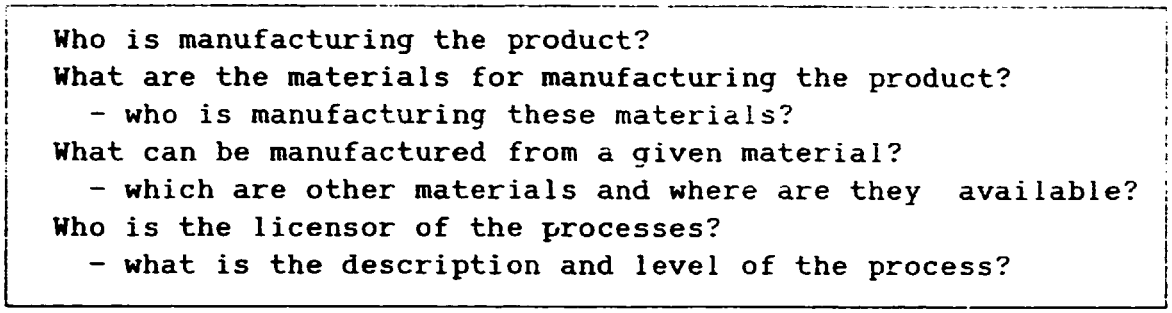


Fig. 4. Predefined queries

Data Entry Services

L - Change dialogue language
W - Select another worksheet
N - Create new record
E - Edit record (or range)
R - Edit last search results
P - Recall last record modified
D - Define default values
C - Clear default values
X - End Data Entry
?

Fig. 5. The "Data Entry Service" menu

4. DATA ENTRY

General rules concerning data entry into the PETRO worksheets are given in this chapter. Some of them are briefly recalled on the worksheet level after pressing the key 'F1' for help.

To start the process of data entry for PETRO you should choose the option E from the main ISIS menu (EXISI) what brings you to the the menu "Data Entry Service", menu EXE1, which is displayed in Fig. 5.

Three data input worksheets are given to the disposal of PETRO users. It is up to the user to pick up one of them for data inputting. The worksheets, however, have been designed to cover documents occuring in the PETRO database. The correspondence between the type of the record (document) and the name of the worksheet is given below:

record type	name of worksheet
no record	PETRO
enterprise	ENTRP
product	PROD
process	PROC

The option W on the EXE1 menu allows you to specify the name of a worksheet to be used for data input. After pressing W you will see the menu which is shown in Fig. 6. Use down and/or up arrows to pick up an appropriate worksheet and press <CR>. To leave the menu strike X, as usually. The PETRO worksheet is provided when entering the database for the first time without prior selection of worksheet and does not support any document type, it is dummy.

To create a new record press N. Then you will see the first page of the data input worksheet. You should fill in subsequent fields following the format of the worksheet. Whenever in doubt how to fill out a field strike the F1 key, before filling, for help. If you want to skip over a field press <CR>.

DATA ENTRY WORKSHEETS AVAILABLE

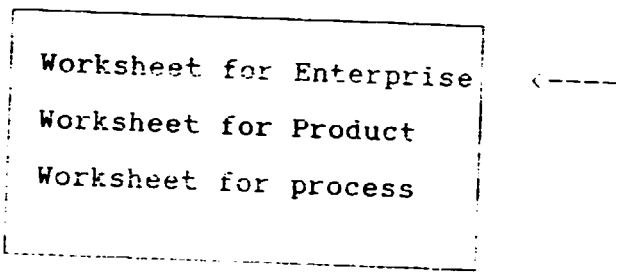


Fig. 6. PETRO worksheets menu

When filling out a page is completed, in the lower part of the screen you will see the following text

- Next page B - Previous page M - Modify
R - Screen restore X - Exit D - Delete C - Cancel
R - End revise

To proceed to the next page press <CR>. If you have made some mistakes you can correct them with the M option.

Eventually, you will reach the last page of the worksheet (there is a note "END"). Now press X to finish the creation of your record.

5. UPDATE

When updating existing records you should take into account the rules given for data inputting (Chapter 4).

To update the existing records choose the option E from the menu EXGEN - "Data Entry Services" (cf. Fig. 3). You will be then prompted to give the number or a range of numbers of documents to be modified. In the latter case the numbers have to be separated by a space. In both cases, next, press <CR>. After that you will see the first page of the document. You can either proceed to the next page by striking <CR> or start updating by pressing M and then correcting the fields.

The cursor blinks at the beginning of the first field of the page. To leave this field unchanged press <CR>. To correct the field simply enter an appropriate text. If you find that your corrections were not adequate press C which cancels all modifications made in the document till pressing C. The original values of the fields in the document will be restored. Again, you are prompted to give the number or a range of numbers of documents to be modified. It is possible to reiterate the process of updating by striking M again.

As for the data entry (Chapter 4) the help is provided by pressing the F1 key.

6. DELETE

To delete existing records choose the option E from the menu EXGEN - "Data Entry Services" (cf. Fig. 3) and answer to the prompts as for updating (Chapter 5). From the options which are displayed at the bottom of the screen pick up the D one.

If some of the records from the range do not exist you will see at the bottom of the screen the following message

Record deleted. Enter one of the following action codes:
R - Undelete N - Recreate <CR> - Leave deleted MFN =

Press ENTER to confirm deletion.

Sorting and printing

L - Change dialogue language
S - User print worksheet
P - System print worksheet
X - Exit

?

Fig. 7. The "Print and Sort" menu

7. PRINTING AND SORTING

Before printing you should check that your printer is properly connected, has sufficient quantity of paper loaded, is in "On-line" status and its system name is LPT1. The latter can be tested and, if needed, corrected from the DOS level [2].

In order to use the print and sort functions you should activate the ISISPRT service from the main CDS ISIS menu, viz EXISI, by pressing

P

The menu EXPRT - "Sorting and printing" will appear (cf. Fig. 7). Proceed to printing by choosing the option S (User print worksheet). The system will display the menu shown in Fig. 8(a) and you should decide whether you want to print hits or the whole database. Use up and/or down arrow and strike <CR>to select the desired option. To leave the menu press X.

After that you will see the next menu (Fig.8b or 8c or 8d) containing the typess of indexes to be printed. Again, you have to select, according to your needs, one of indexes available in the same way as for the previous menu.

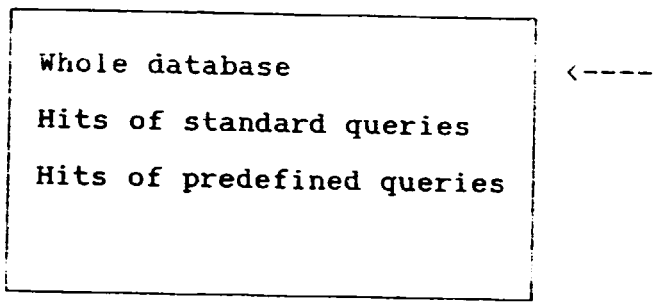
Finally, you will see a pre-defined (already filled out) worksheet. Skip over all the fields by pressing the key PgDn or <F> until you see a message made up of four options at the bottom of the screen. Press X for printing.Some system messages are to be displayed on the screen and the printing starts provided that your printer is in "On-line" status and its system name is LPT1.

NOTE! The indexes for printing hits of predefined queries for the second and the third type are not implemented.

(a)

PETRO Indexes - Printing

Do you want to generate indexes from hits or whole database?



X - Exit

Select your index using down and/or up arrows

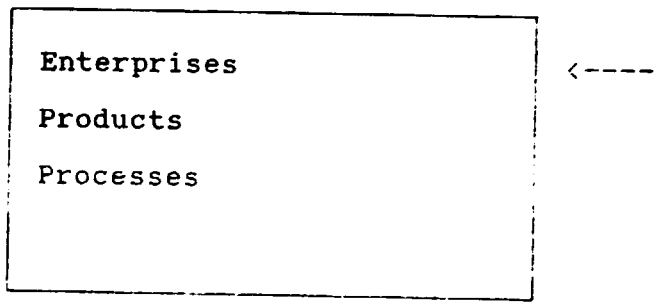
Confirm you choice by pressing ENTER

Strike X to exit

(b)

PETRO Indexes - Printing

WHOLE DATABASE



X - Exit

Select your index using down and/or up arrows

Confirm you choice by pressing ENTER

Strike X to exit

(c)

PETRO Indexes - Printing

HITS OF STANDARD QUESTIONS

Enterprises <----
Products
Processes

X - Exit

Select your index using down and/or up arrows

Confirm you choice by pressing ENTER

Strike X to exit

(d)

PETRO Indexes - Printing

HITS OF PREDEFINED QUESTIONS

Manufacturer(s) of a given product <---
Materials for manufacturing the product
- and manufacturer(s) of these materials?
Products manufactured from a given material
- other materials needed for the product and the manufacturers
Licensor(s) of a given process
- description and level of the process

X - Exit

Select your index using down and/or up arrows

Confirm you choice by pressing ENTER

Strike X to exit

Fig. 8. Menus for printing
(a)PETRO indexes menu
(b) PETRO indexes for whole database
(c) PETRO indexes for hits of standard queries
(d) PETRO indexes for hits of predefined queries

8. INVERTED FILE GENERATION

For retrieval and in order to maintain the database consistency it is necessary to have the inverted file updated after each modification of the database contents. Basically, there are two ways of updating the inverted file. First, you can update it when leaving the ISIS.EXE program after modifying the database. Second, you can invoke the ISISINV service (Inverted file services) from the main menu EXISI by typing

I

You will see the menu EXG1 - "Inverted File Services" which is also depicted in Fig. 9.

To update the inverted file you have to choose the option U. You are prompted to give the database name. After some system messages you will see the text

Inverted file update completed

Press <CR>.

The option F is used for full inverted file generation. For a database which contain a substantial amount of documents the process of the inverted file generation may be a lengthy one.

The option P is very useful for a database administrator as it allows to obtain the actual list of indexing terms occurring in the database documents. We will discuss this more exhaustively in Chapter 9.

Inverted File Services

L - Change dialog language
F - Full inverted file generation
U - Update inverted file
B - Inverted file backup
G - Create unsorted link file
S - Sort link file
[illegible] - Load inverted file
D - Dump inverted file
P - Print search term dictionary
E - Exit

- ?

Fig. 9. The "Inverted File Services" menu

9. DATABASE MAINTENANCE

This chapter is dedicated to a database administrator, who is responsible for the system reliability and maintenance. The structure of the PETRO database, viz. its FDT and FST tables are given in Appendix 2. Also, the names of print worksheets, as well as, printing and display formats (*.PFT) are presented in this Appendix.

(a) The PETRO software protection

The PETRO software is distributed on 3 diskettes. The first two diskettes contain the CDS MICRO ISIS package as it is distributed by UNESCO (except for the INSTALL.BAT file) whereas the third one contains system files specific for PETRO. Note that no data (records) pertaining to the PETRO databases itself are stored on these diskettes. As the first job recommended to be done after receiving theses diskettes is to prepare the verified copies of them. To do so check out whether they are protected against writing and if not, protect them. Then insert the diskette labelled PETRO01 into the drive A and insert an unprotected diskette into the drive B. Type the DOS command

DISKCOPY A: B: <CR>

When finished answer N to the DOS question and type

DISKCOMP A: B: <CR>

If the comparison is OK protect and label the diskette from the drive B. Otherwise, repeat the process of copying. Repeat the process for the diskettes PETRO02 and PETRO03.

(b) Database building

There are several ways of database building. The first one has been described in Chapters 4-6. The second one is concerned with the import of data. Basically there are two possible cases

(i) The database is empty (i.e. only the diskettes PETRO1, PETRO02 and PETRO02 were copied to your computer), then you should invoke the ISISXCH - "Master file services" from the main ISIS menu (EXIST by typing

M

You will see the menu EXCH - "Master file services" which is presented in Fig. 10.

To load the external database (in ISO 2709 format) type I. As a result you will see the input form which is depicted in Fig. 11.

```
                    Master file services

                  L - Change dialogue language
                  B - Master file backup
                  R - Master file restore
                  C - Reorganize Master file
                  I - Import external file
                  E - Export CDS/ISIS file
                  X - Exit
                          - ?
```

Fig. 10. The "Master file services" menu

```
              Data interchange services - Import parameters

Data base name                  PETRO____

                          ISO File parameters

Input ISO file name   MST.ISO__          field separator          #
                                         record separator         #

                          Input options

First MFN to be assigned        1_____     Input tag containing MFN _
Load/Merge/Update   L

Reformatting parameter file     _______    Gizmo conversion file    ____
```

Fig. 11. The input form for data import

To skip over a field and leave it unchanged strike <CR>. After reaching the bottom of the screen press X. The prompt: "Backup drive and/or directory?" will appear. The name of the drive which contains the diskette with the data has to be given followed by pressing <CR>.

Example

```
Backup drive and/or directory? A:
Mount diskette number 1 on drive a:  X - no more diskettes, <CR>
to continue after mounting new diskette                    <CR>
Master file exists and will be cleared, OK (Y/N) Y

MFN 1 added
MFN 2 added
```

```
MFN 3 added
MFN 4 added
MFN 5 added
MFN 6 added
MFN 7 added
MFN 8 added

Mount diskette number 2 on drive a: X - no more diskettes,  <CR>
to continue after mounting new diskette      X
Next MFN to be assigned: 9
```

Note that the file stored on the diskette(s) mounted on the drive must be MST.ISO. To return to the menu strike <CR>.

(ii) Some records have been already stored in a database. Then you should read the MFN number displayed at the bottom of the menu "Master file services". After that your steps are the same as described in (i), except for that you put

- M into the field "Load/Merge/Update";
- MFN+1 into the field "First MFN to be assigned", e.g. if MFN is 7, you write 8.

The third possibility of the database building is concerned with the "Master file restore" option (R) on the menu "Master file utility services". To perform it you must have diskette(s) prepared under the option B (master file backup) available on the same menu (see below).

To proceed, strike the key R. Comply with the system prompts concerning diskettes. Beware! the order of diskettes is critical, i.e. they must be mounted in the same order they were produced.

(c) Backup

To avoid the consequences of an accidental database destruction it is recommended to backup its contents from time to time. For this purpose you can use the option B from the EXCH - "Master file services" menu. After typing B you will be prompted to give the database name. The next prompt is "Backup drive and/or directory?". It is recommended to type

```
A:  <CR>
```

which means that a backup file is to be stored on the diskette mounted in the drive A. After usual prompts concerning diskettes and system messages press <CR> (after an arrow which is displayed on the screen). As a result of backup function you will have your backup file PETRO.BKP on the diskette(s).

If there are more diskettes than one needed for backup they should be cautiously labelled and numbered to ensure correct restore (see (b)).

(d) Transfer of database to other centers

To prepare the diskette(s) containing the databases (or their increments) you should invoke the ISISXCH service from the main ISIS menu and, then, choose the option E from the menu EXCH - "Master file services". As a result you will see an input form presented in Fig. 12. As usually, you can skip over the fields by striking <CR>.

```
                Data interchange - Export option parameters

Data base name                  PETRO

                              ISO File parameters

Input ISO file name   MST.ISO__          field separator          #
                                         record separator         #

                                 Input options

MFN limits  1/32000____   Save file name ___   Hit file  (Y/N)? N

Reformatting FST _________          Gizmo conversion file _______
Renumber records from ______
```

Fig. 11. The input form for data export

The range of records to be exported has to be put into the field "MFN limits", e.g. if you want to export records having MFN numbers 123, 124, 125, 125, 127 write 123/127 in this field. If a file containing results of searching is to be exported put into the field "Save file name" the name of your save file which should be PETRO (cf. Chapter 3, usage of the P option - "Save search results" on the menu "Information Retrieval Services" and overwrite Y in the field "Hit file(Y/N)?". If the records from the master file are supposed to be exported leave the fields "Save file name" and "Hit file(Y/N)?" unchanged.

After reaching the message area at the bottom of the screen, press X, and you will see the prompt: "Backup drive and/or directory?".

It is recommended to type

A: <CR>

which means that the database will be unloaded to the diskette that is mounted on the A drive. Comply with the system prompts concerning diskettes. After that the system will communicate which records are unloaded.

Example

```
MFN 1 unloaded
MFN 2 unloaded
MFN 3 unloaded
MFN 4 unloaded
MFN 5 unloaded
MFN 6 unloaded
MFN 7 unloaded
MFN 8 unloaded
MFN 9 unloaded
Total output records 9
```

To continue press <CR>. Now the exported data are stored in the MST.ISO file on the diskette.

Beware! Each file created by this option on diskette(s) is named MST.ISO. So, to avoid confusion it is strongly recommended to unload every database on a separate set of diskette(s).

(e) Printing search term dictionary

To perform this function choose the option P from the menu EXG1 - "Inverted File Services" (see Chapter 8). After that the text:

"Dictionary list completed "

is displayed. To continue press <CR>. The dictionary is stored in the form suitable for printing in the file IFLIST.LST. You can print this file using the DOS command

PRINT IFLIST.LST

Example

```
C:\PETRO\WORK> PRINT IFLIST.LST
Name of list device [PRN]: LPT1
Resident part of PRINT installed
C:\PETRO\WORK> IFLIST.LST is currently being printed
C:\PETRO\WORK>
```

The dictionary may be distributed to the users who have no direct access to the computer, in order to facilitate them formulating the queries and get the overall knowledge about the database contents.

10 . PROCESSING TREES

First attempt was made to provide a facility to support chemical trees. Two programs coded in MICRO CDS ISIS have been developed, viz.:

CHEM - the preliminary version of the program for visualisation of process trees, and all operations on CHEM database;
CHEM1 - temporary program for creating CHEM database.

In the final version of PETRO software only one program supporting trees will be used.

Below are some remarks dealing with the creation of trees (links) by means of the CHEM program.

The two fields may be used to add a link to the node displayed in the upper corner (source node). The first field (3 characters) indicates type of link, the second one (30 characters) - the target node.

Notes: (1) Only direct links have to be specified!
(2) The target node has to be defined before any links are specified!

Types of links that you can add to the given node are as follows:

USE - this link is obligatory if the source node specifies an ascriptor. In this case in the target node the proper process name or the proper material has to be referenced; No other links types can be specified for ascriptor source node. When used, it automatically creates the opposite link of the type UF. An example of this type of link is:

ETHYL ALCOHOL USE ETHANOL

UF - this link indicates ascriptors of a given source node. It is opposite to USE. When used, it automatically creates the link of the type USE in the target node.

POF - (Part OF)-type link is used to indicate all the chemicals where the source node is a direct compound; the link cannot be used for ascrip- tors, i.e. neither source node nor target node can be ascriptor.

Example:

METHANOL POF METHYLAMINE

Note: The system automatically settles the opposite link (COF) between the target and source nodes.

COF - (Consist OF)-type link is used to indicate all the component chemicals for the source node chemical. As in the case above the link cannot be used for ascriptors.
Example:

METHYLAMINE COF AMMONIA

Note: The system automatically settles the opposite link (COF) between the target and source nodes.

EQU - (EQUal). This type of link is intended to be used only for the processes (roots in the trees). For the source node they indicate another proce- sses which are equivalent to the given process (from the point of view of final products).

PRT - (PRoducT). The link indicates final product for the process specified by the source node;

Notes: (1) The source node must be the process name (tree root);
(2) The system automatically settles the opposite link (PRS) between product and process nodes.

EXAMPLE

CARBOFURAN SYNTHESIS PRT CARBOFURAN

PRS - (PRocesS). The link indicates the process which gives rise to the specified product;

Notes: (1) The source node must be the product name (inner node);
(2) The system automatically settles the opposite link (PRT) between product and process nodes.

11. FURTHER WORK

In order to complete the PETRO software development the following tasks have to be undertaken.

1. Development of programs for sophisticated query formulation and retrieval taking into account relations occuring between various types of records stored in the database.

2. Elaboration and implementation of graphical representation of trees and on-screen manipulations, in particular:
   - completion of the program CHEM for visualization and manipulations of trees;
   - establishing extensive HELP procedures for the CHEM program.

3. Elaborating procedures for automatic loading the data from trees to PROD database records.

4. Design and implement tables and/or rules for enterprise codes by UNIDO.

5. Implement tables for Country codes, according to ISO.

6. Development of sorting and printing formats and programs.

7. Establishing data protection features. Distinction between a casual user (only retrieval and printing allowed) and a database administrator has to be made.

8. Development of mailing system on the basis of "ENTRP" records stored in the database.

9. Development of user-friendly interface to manipulate the data.

10. Setting up a testing database and performing extensive testing.

11. Preparing a user's guide (manual)

12. Providing a training to a UNIDO staffer on the operation of the database.

The estimated manpower for accomplishing the above work is:

1.5 mm, including 1 week for training

It is also recommended to establish a thesaurus and an appropriate program to support data entering and retrieval.

12. REFERENCES

1. Mini-micro CDS/ISIS, Reference manual (Version 2.3), UNESCO, Paris, 1989.
2. PC's Handbook for 16 Bit PC User's, Microway Enterprises Co. Ltd.

Appendix 1

Technical Specification

The minimum hardware requirements for running MICRO CDS/ISIS are the following.

IBM PC XT/AT or compatible microcomputer equipped with:

- 640K RAM memory
- 1 floppy disk drive
- 1 hard disk
- 1 monitor (monochrome or color)
- 1 printer

The MS-DOS operating system, ver. 2.0 (or higher) has to be installed on a microcomputer.

The following MICRO CDS/ISIS restrictions are, *inter alia*, currently in effect:

max. number of records	16 millions
max. size of record	8000 characters
maximum number of stopwords	799

PETRO Structure

Below is given the FDT table.

```
W:PETRO ENTRP PROD  PROC
F:PETRO ENTRP PROD  PROC  ENTRPOENTRP5
S:PETRO
***
Filled out by                                  1 25 0 0
Date of entering data          99-99-99        2 8 3 0
Subject information received                   3 1 1 0
Date of last request sent      ra              4 20 0 1
Status of Institution                          5 1 1 0
Record type                                    6 15 0 0
Country code                                   10 4 2 0
Company name                                   20 80 0 0
Acronyme                                       21 25 0 0
Enterprise code  (by UNIDO)                    22 25 0 0
Activity/Manufacturing                         25 1 1 0
Activity/R&D                                   26 1 1 0
Activity/Engineering                           27 1 1 0
Activity/Contracting                           28 1 1 0
Activity/Training                              29 1 1 0
Mailing Address                                30 240 0 0
Institution Address                            35 240 0 0
Tlx                                            36 30 0 0
Phone                                          37 40 0 0
Fax                                            38 40 0 0
E-mail                                         39 100 0 1
Contact person                                 41 60 0 1
His/her telephone                              42 40 0 1
Product                        npemlx          100 300 0 1
Processes for licence          stl             120 320 0 1
Processes in use               stl             130 320 0 1
Product name                                   200 80 0 0
Product code ISIC                              210 20 0 0
Product code SITC                              213 20 0 0
S-list of trees                                220 80 0 1
I-list of trees                                230 80 0 1
E-list of trees                                240 80 0 1
Manufacturer Id = Enterp. code                 250 40 0 1
Process name                                   300 160 0 0
Process description                            303 400 0 0
Licensor ID = Enterprise code                  310 40 0 1
Main product                                   320 80 0 0
```

```
Capacity range from          vm                  330 20 0 0
Capacity range to            vm                  333 20 0 0
Standard capacity            vm                  335 20 0 0
Unit production cost                             337 20 0 0
Investment cost for s-capacity                   339 20 0 0
Main equipment               nqs                 340 320 0 1
Row material(s)              nq                  360 320 0 1
Utilities/Electrical energy                      370 10 0 0
Utilities/Steam                                  373 10 0 0
Utilities/Water                                  375 10 0 0
Utilities/Other energy                           379 10 0 0
Manpower/Manager(s)                              390 10 0 0
Manpower/Engineer(s)                             393 10 0 0
Manpower/Operator(s)                             395 10 0 0
Manpower/Others                                  397 10 0 0
Required space/Site area                         400 10 0 0
Required space/Building area                     403 10 0 0
Dummy00                                          997 200 0 1
Dummy0                                           998 160 0 0
Dummy                                            999 200 0 1
```

The PETRO.FST table is as follows

```
6 0 mhu,"TYPE="v6
10 0 "CC="V10
22 0 "EC="v22
25 0 mhu,"MANUF="v25
26 0 mhu,"R&D="v26
27 0 mhu,"Eng="v27
28 0 mhu,"CONTR="v28
29 0 mhu,"TR="v29
100 0 mhu,(!PROD=!v100^n/)
120 0 mhu,(!PSL=!v120^s/)
120 0 mhu,(!PTL=!v120^t/)
130 0 mhu,(!USL=!v130^s/)
130 0 mhu,(!UTL=!v130^t/)
200 0 mhu,"PNAME="v200
210 0 mhu,"PCI="v210
213 0 mhu,"PCS="v213
220 0 mhu,(!S=!v220/)
230 0 mhu,(!I=!v230/)
240 0 mhu,(!E=!v240/)
250 0 mhu,"MID="v250
300 0 mhu,"PROC="V300
310 0 mhu,"MID=",V310
320 0 mhu,"MP=",V320
```

```
360 0 (!RAW=!,v360^n/)
```

Names of worksheets for printing (whole database and hits) as well as names of display/printing formats (*.PFT) are presented in the following table.

	Enterprises	Products	Processes
All	EYD0	EYD1	EYD2
Hits standrd.	EYH0	EYH1	EYH2
Format	ENTRP	PROD	PROC

For Hits of predefined queries the worksheets and display formats are the following

	Query 1	Query 2	Query 3	Query 4
Hits predef.	EYQ0	EYQ1	EYQ3	EYQ5
Format	ENTRP0	not implemented		ENTRP5

The display formats are as follows:

ENTRP.PFT

```
IF V6='ENTERPRISE' THEN MFN(4),C50,'Enterprise code: 'v22/,V20,"
("v21")"/,"MAILING ADDRESS: "v30(18,18),/"ENTERPRISE ADDRESS:
"v35(21,21)/"Tlx: "v36,"   Tel.: "v37/,"Fax: "v38,"   E-mail:
"v39/,"CONTACT PERSON(s): "d41,(v41(19,19),!;  Tel.: !v42/),#
 ' --- PRODUCT(s)
----------------------------------------------------------'/'
Quantity        Price (in "V2")'
Produced  Exported   Local   Export'/,      IF A(V100^n) THEN 'None
products are manufactured'/ FI,                          (V100^n,! (in
!V100^m!)!,C37,V100^p,C47,V100^e,c58,v100^l,C66,V100^x/)#%,       ' ---
PROCESS(es)----------------------END PRODUCT------------------ LEVEL
--'/ IF A(V120^s) THEN 'None process is offered for licence'/ FI,"
(for license)"D120/(V120^s,C40,V120^t,C70,V120^l/),"   (in
use)"D130/(V130^s,C40,V130^t,C70,V130^l/)#%
```

```
' ---- ACTIVITIES
-----------------------------------------------------------'/
,C5,"Manufacturing: "V25,C30,"R & D:"D26,C43,V26#%C5,"Engineering:
"V27,C30,"Contracting:"D28,c43,V28#%C5,"Training:      "V28,## FI
```

ENTRP0.PFT

```
IF V6='ENTERPRISE' THEN MFN(4),C50,'Enterprise code: 'v22/
"PRODUCT: "V998," is manufactured by:"D998/%,
V20," ("v21")"/,"MAILING ADDRESS: "v30(18,18)/,
"ENTERPRISE ADDRESS: "v35(21,21)/,
"Tlx: "v36,"   Tel.: "v37/,"Fax: "v38,"   E-mail:  v39/,
"CONTACT PERSON(s): "d41,(v41(19,19),!;  Tel.: !v42/) ## FI
```

ENTRP5.PFT

```
IF V6='ENTERPRISE' THEN MFN(4),C50,'Enterprise code: 'v22/,V20,"
("v21")"/,"MAILING ADDRESS: "v30(18,18),/"ENTERPRISE ADDRESS:
"v35(21,21)/"Tlx: "v36,"   Tel.:"v37/,"Fax: "v38,"   E-mail:
"v39/,"CONTACT PERSON(s): "d41,(v41(19,19),!;  Tel.: !v42/),
 'PROCESS: 'V998,(IF V997^S=V999 THEN !      Level: !V997^l FI)/,
 ref(L(!PROC=!V998),!DESCRIPTION: !V303) ## FI
```

PROD.PFT

```
IF V6='PRODUCT' THEN
MFN(4),/'PRODUCT: ',V200/,"ISIC code: "V210,"    STIC code: "V213/,
"End product in the following process(es):"D240/,(V240(5,5)+!; !)/
"Used as a row material in the following
process(es):"D220/,(V220(5,5)+!; !)/   "Used as an intermediate in the
following process(es):"D230/,(V230(5,5)+!; !)/  "The product is
manufactured by:",D250/(ref(L(!EC=!V250),"   "V20," ("v21")"/,  "
MAILING ADDRESS: "v30(18,18),/%"   ENTERPRISE ADDRESS: "v35(21,21)/%
Tlx: "v36,"   Tel.: "v37/,"   Fax: "v38,"   E-mail: "v39/,"   CONTACT
PERSON(s): "d41,(v41(22,22),!;  Tel.: !v42////)##% FI
```

PROD1.PFT

```
IF V6='PROCESS' THEN MFN(4),/"PROCESS:
",V300/,"DESCRIPTION:"D303/,V303/,"MAIN PRODUCT: ",V320/,"ISIC code:
"V210,"    STIC code: "V213/%,"Licensor of the product: ",D310/
 (ref(L(!EC=!V310),"   "V20," ("v21")" /,"   MAILING ADDRESS:
"v30(18,18),/"   ENTERPRISE ADDRESS: "v35(21,21)/"   Tlx: "v36,"
Tel.: "v37/,"   Fax: "v38,"   E-mail: "v39/,"   CONTACT PERSON(s): "d4
```

```
,(v41(22,22),';  Tel.:';v42//))))/        "Capacity range FROM:
"V330^v,X2,V330^m,"/year "D330,"TO:  "V333^v,X2,V333^m,"/y
ear"D333/,"Standard capacity: "V335^v,X2,V335^m,"/year"D335/,
"Investment cost for standard capacity: "D339,V339, " USD"D339/,
"Unit production cost: "V337, "USD per kg"D337/,
"------ MAIN EQUIPMENT --------------- QUANTITY----- SPECIFICATION
-----------" D340/,(V340^n,C42,V340^q,C51,V340^s(51,50)/),
 "------ RAW MATERIALS ---------------- QUANTITY ----- per kg of
product"D360/,  (V360^n,C40,V360^q/),
 '------ UTILITIES:'/,
 "Electrical energy: "V370,"  kwh/kg"D370/,
 "Steam:             "V373,"  kg/kg"D373/,
 "Water:             "V375,"  m3/kg"D375/,
 "Other energy:      "V379,"  Mjoule/kg"D379/,
 '----- MANPOWER (number of persons):'/,
 "Managers:  "V390,X10,"Engineers: "V393/,
 "Operators: "V395,X10,"Others:    "V397/,
 '----- REQUIRED SPACE (in m2):'/,
 "Site area: "V400,X10,"Building area: "V403 ##% FI
```

THE CDS/ISIS SEARCH LANGUAGE

1. Introduction

The retrieval language of CDS/ISIS is based on boolean algebra, which provides a convenient way of expressing logical operations between classes. Each descriptor or access point used to describe the subject contents or other characteristics of a record, in fact, can be viewed as representing the class of all those records associated with that access point. Thus by expressing logical operations between access points you can define precisely the class of records to be retrieved in response to your needs.

In the following paragraphs, the examples are based on an actual data base for the sake of clarity. They are not meant to show how to formulate specific queries for any data base, but simply to illustrate the capabilities of the system.

It is important, therefore, that you become familiar with the access points available for the data base you are querying and with the rules followed to index the records in that data base.

2. Types of operands

In formulating the boolean expression you may use three types of class identifiers: precise access points, right-truncated access points, and ANY terms.

2.1 Precise access points

A precise access point is any searchable element defined for a given data base, such as subject descriptors, key words, key phrases, words in titles, author names, etc. You must be familiar with the access points available for each data base at your installation.

When you use a precise access point you must specify it in exactly the same way it is known to the system. Even minor variations in spelling will cause the system to reject it. You should normally have a list of access points at hand when you formulate your query.

Thus, for example, if the term known to the system is COLOR (American spelling), you may not use the English spelling COLOUR as this will be rejected.

An important point to be remembered is that if an access point contains parentheses or any one of the logical operators (*, +, (&), (F), * . *, * $ *, -) or begins with a number sign ($) you must enclose it in double quotation marks (") in order to avoid a possible ambiguity. Thus if your access point reads:

GERMANY (FEDERAL REPUBLIC)

you must enter it as follows:

"GERMANY (FEDERAL REPUBLIC)"

or else the system will issue a syntax error message.

2.2 Right-truncated access points

Instead of specifying a precise access point, you may just give a root. This technique, referred to as root searching or right truncation, allows you to search on leading sequences of characters. The system will automatically perform a logical OR operation between all access points having the specified root.

Right-truncation is indicated by placing a dollar sign ($) immediately after the last root character. Assume, for example, that your list of access points contains the following set of terms:

FILE ORGANIZATION
FILM
FILM INDUSTRY
FILM LIBRARIES
FILM-MAKER
FILM-MAKING
FILM-MAKING TRAINING
FILMSTRIP
FILTRATION

Then the specification:

FILM$ is equivalent to:

(FILM + FILM INDUSTRY + FILM LIBRARIES + FILM-MAKER
+ FILM-MAKING + FILM-MAKING TRAINING + FILMSTRIP)

FILM-$ is equivalent to:

(FILM-MAKER + FILM-MAKING + FILM-MAKING TRAINING)

As for precise access points, if your root contains parentheses, any one of the logical operators (*, +, (G), (F), " . ", " $ ", ¬), or begins with a number sign (#) you must enclose it in double quotation marks (").

For example: "GERMANY ($"

Because of the PROXIMITY operator " $ ", if a right truncation is to be made on a descriptor followed by a blank, you must enclose it in double quotation marks (").

Continuing the 'FILM' example from above "FILM $" is equivalent to:

(FILM + FILM INDUSTRY + FILM LIBRARIES)

2.3 ANY Terms

An ANY term is a collective term standing for a pre-defined set of access points. Whenever you use an ANY term in your search formulation, the system will automatically OR together all access points of the cluster associated with that ANY term.

An ANY term consists of the word 'ANY' followed by a unique identifier, usually mnemonic, assigned to the associated cluster. For example, the term ANY BENELUX COUNTRY could be used to retreive records indexed with the individual name of any country in the Benelux group (BELGIUM, NETHERLANDS, LUXEMBOURG).

Before an ANY term can be used in a search formulation, its meaning must be defined by specifying the associated cluster. Note that not all data bases need implement this facility. Therefore before attempting to use an ANY term you must ensure that the facility is available for the data base you are querying and, if so, that the ANY term you intend to use is actually defined.

2.4 The "dot" feature

This facility permits the assignment of a higher level of importance to one or more terms in the query formulation, than to others. The facility may only be used when querying data bases indexed according to the ISIS indexing technique, where the index terms are entered in such a way that the most important terms appear first and terms of secondary importance appear last (the two groups being seperated by an appropriate seperator) or seperate tags are used for the terms of different importance.

By using the dot feature you may ask the system to retrieve only those records in which the specified terms appear amongst the set of most important terms.

To request this facility you simply enter a dot (hence its name) immediately before the term. For example:

.EDUCATION OF WOMEN * ITALY

In this case, the only records that will be retrieved are those records in which the term EDUCATION OF WOMEN appears amongst the important terms. The effect of the dot feature is to narrow the scope of the search and increase the relevance of the retrieved records.

When using this feature you should be aware of the following:

(a) when the operand is enclosed in double quotes the dot must immediately precede the opening quotes. For example:

.*GERMANY (FEDERAL REPUBLIC)*

(b) when the operand is a cluster (right-truncated access point or ANY term). the system will apply the dot to all members of the cluster.

Thus for example:

.ANY BENELUX COUNTRY

is equivalent to:

.BELGIUM + .NETHERLANDS + .LUXEMBOURG

3. Boolean operations

3.1 Logical "OR" (inclusive)

The logical OR is the class union operator. The result of a logical OR between two classes is the class obtained by merging the two classes, retaining common elements, if any, only once.

Thus if A and B are two descriptors, representing the two classes of documents indexed with descriptors A and B respectively, the logical OR between these two classes is the class of documents indexed with descriptor A or descriptor B or both.

The logical OR, therefore, is used to broaden the scope of the search and will, in general, increase the number of hits. The symbol used to indicate the logical OR operation is the plus sign (+).

Thus, for example, to retrieve documents about the Benelux countries one could use the logical OR operator as follows:

BELGIUM + NETHERLANDS + LUXEMBOURG

Note that the order in which the three countries are given is irrelevant.

3.2 Logical "AND"

The logical AND is the class intersection operator. The result of a logical AND between two classes is the class containing only those elements which are common to both classes.

Thus, if A and B are two descriptors, representing the two classes of documents indexed with descriptors A and B respectively, the logical AND between these two classes is the class of documents indexed simultaneously with both descriptor A and descriptor B.

The logical AND, therefore, is used to narrow the scope of the search by

requiring the co-occurrence of descriptors and will, in general, decrease the number of hits. The symbol used to indicate the logical AND operation is the asterisk (*).

Thus, for example, to retrieve documents generally about on-line information retrieval systems one might use the logical AND operator as follows:

ON-LINE SYSTEMS * INFORMATION RETRIEVAL

Note that the result of a logical AND between two classes may be an "empty" class, i.e. a class containing no elements. In the example above, the "empty" class would be obtained if no document existed in the file to which both the descriptor ON-LINE SYSTEMS and the descriptor INFORMATION RETRIEVAL were assigned. Also note that the order in which the descriptors are given is irrelevant.

3.3 Field level and proximity search operators

These operators are more restrictive types of the logical AND operator and are particularly useful for natural language searching.

The field level and proximity search operators are:

(G) same repeatable field (any occurrence), for example:

A (G) B will retrieve all records with both A and B in any field or in the same group of occurrences of a repeatable field;
A (G) B/(20) will retrieve all records with A and B in field 20;

(F) same field or occurrence, for example:

A (F) B will retrieve all records with both A and B in the same occurrence of any field;
A (F) B/(20) will retrieve all records with A and B in the same occurrence of field 20;

. proximity: not more than 'n' words apart, where 'n' is the number of periods plus one. For example:

A . B adjacent
A . . B at most one word between A and B
A . . . B at most two words between A and B
etc.

$ proximity: exactly 'n' words apart, where 'n' is the number of dollars plus one. For example:

A $ B adjacent
A $ $ B exactly one word between A and B
A $ $ $ B exactly two words between A and B

Note that the operators . and $ must be preceded and followed by at least one

space.

3.4 Logical "NOT"

The logical NOT is the class exclusion operator. The result of a logical NOT between two classes is the class containing all the elements of the first class which do not also belong to the second class.

Thus, if A and B are two descriptors, representing the two classes of documents indexed with descriptors A and B respectively, the logical NOT between A and B is the class of documents indexed with descriptor A but not simultaneously with descriptor B. The symbol used to indicate the logical NOT operation is the not sign (¬).

The logical NOT should be used with extreme caution, since it may easily result in the inadvertent loss of relevant material.

For example, a search request might concern documents about disadvantaged groups but excluding references to disadvantaged children. One might be tempted to formulate the query as follows:

DISADVANTAGED GROUP ¬ DISADVANTAGED CHILDREN

It is quite possible, however, for a document on the subject to contain a section on disadvantaged children. This document would then be missed if one formulated the query as above. Note further, that unlike the logical OR and the logical AND, the logical NOT operation does not give the same results if the two operands are inverted.

Thus, A ¬ B is not the same as B ¬ A (except in the particular case in which A and B represent the same class).

4. Boolean expressions

4.1 Syntax

You may form complex boolean expressions by combining the use of the boolean operators (+, *, (C), (P), " . ", " $ " and ¬).

As in normal algebra, you may use parentheses to alter the order of evaluation. In the evaluation of expressions the priority of the operators is as follows:

(highest) " $ " and " . "
(P)
(C)
" and ¬
(lowest) *

Thus, for example, to evaluate the following expression:

A + B • C

the system will first evaluate B • C and then evaluate the logical OR between A and (B • C).

Whereas to evaluate the following:

(A + B) • C

it will first evaluate A + B and then the logical AND between (A + B) and C. You may "nest" parentheses, if required, as in the following example:

((A + B) • C + (D + E) + F) ¬ G

In forming boolean expressions you must observe certain simple syntactical rules:

(a) no two logical operators (•, +, ¬, (G), (F), " . ", " $ ") may be adjacent to each other, except for repeats of " . " and " $ " (which may not, however, be mixed together);

(b) the parentheses used must be balanced, i.e. the number of open parentheses must be equal to the number of closed ones and each open parenthesis must have a matching closed one;

4.2 Operand qualifiers

You may use qualifiers to specify the field or group of fields in which you want one or more descriptors to appear.

This is particularly useful for data bases which contain the same data in a number of different fields.

The qualifier has the following general formats:

descriptor/(t1,t2,t3,...)

where 't1', 't2', 't3', ... is the set of fields from which you want the descriptor to originate.

Operand qualification can be used in conjunction with Boolean operators to restrict the search to the specified field(s) and can also be applied to a right-truncated access point or an ANY term.

For example, consider a bibliographic data base where descriptors are generated, on a word by word basis, from all the fields present and without prefixes.

By simply ANDing it is possible to retrieve any records that contains the

descriptors INTERNATIONAL, EXPERT and MEETING.

INTERNATIONAL • EXPERT • MEETING

However, the number of records that satisfied this query would be very large, as each word could have appeared in any of a number of fields eg. title, corporate body, subject, etc.

By using the (F) field level operator

INTERNATIONAL (F) EXPERT (F) MEETING

the number of records retrieved would be reduced, but it is still not certain as to which tag the records retrieved contain the information, eg. subject meeting, meeting - main entry or even the title.

However, by adding the qualifier "/(62)":

INTERNATIONAL (F) EXPERT (F) MEETING /(62)

only those records that have all the descriptors in the same occurrence of field 62 will be retrieved.

When the operand is a right-truncated access point or an ANY term, the system will apply the qualifier to all members of the corresponding set.

Thus for example:

ANY BENELUX COUNTRY/(64)

is equivalent to:

BELGIUM/(64) + NETHERLANDS/(64) + LUXEMBOURG/(64)

5. Search formulation

The basic building block of a search formulation is a boolean expression as defined above, which we shall call a "search expression". Each time you select option '8' in menu XREM, you are in fact creating a new search expression.

In response to a search expression entry the system will:

(a) assign a unique number to the search expression you have just entered (provided it contains no syntactical errors).

(b) display at the terminal the number of postings for each term in the expression, for each sub-expression and for the whole of the expression.

Thus, for example, if you submitted the search expression (ITALY + FRANCE) • ART, the system reply would be as follows (note that the line numbers on the left do not appear on the screen; they have been included for easy reference to each line in the explanation that follows).

```
1     Set 1: (ITALY + FRANCE) • ART
2     P=   488   ITALY
3     P=  1865   FRANCE
4     T=  2192 - $2: ITALY + FRANCE
5     P=    84   ART
6     T=     8 - $4: ART • $2
7     T=     8 - $1: $4
```

Line 1: Contains the search expression number assigned by the system to the search expression (1 in this case).

Lines 2-3 Contain the number of postings (P=nnnnnn) for each term used in the innermost sub-expression.

Line 4: Contains the number of records (T=2192), the sub-expres- sion number ($2:) and the matching sub-expression (ITALY + FRANCE). In this example there are 2192 records in the data base containing the term ITALY or the term FRANCE, or both.

Line 5: Contains the number of postings (P=nnnnnn) for each term used in the next level of the expression, in this case the outer-most level.

Line 6: Contains the number of records (T=8), the sub-expression number ($4:) and the matching two sub-expressions ie. ART + $2 (from line 4).

Line 7: Contains the number of records (T=8), the query expression number ($1:) and the sub-expression number ($4) from line 6.

If the search expression contains an ANY term, the postings of all access points assigned to the ANY term as well as the aggregate posting value for the ANY term will be displayed.

Similarly, if the search expression contains a right-truncated access point, all the postings of the individual access points as well as the aggregate posting value for the truncated term will be displayed.

If you use a term which is not a valid access point, the system will set its posting value to zero and flag the term with the message:

```
** NOT FOUND **
```

As was mentioned above, the system assigns a unique number to each search expression. It also saves the records matching each search expression. In later search expressions you may refer to previously entered ones by simply using the search expression number assigned by the system preceded by a number sign (#).

This facility allows you to build the final query step by step. Breaking the query into elements also allows you not only to check at each step the number of records retrieved, but also to display at the terminal the records themselves (see option 'D' in xXGEN menu), so that you may verify at any time the logical validity of the search formulation in terms of the relevance of the records retrieved.

Suppose, for example, that after displaying the records retrieved by the query "ART • (ITALY + FRANCE)" you realized that some irrelevant records were retrieved because they dealt with the conservation of monuments. You could then modify the formulation by introducing the following search expression:

#1 ~ CONSERVATION OF MONUMENTS

A reference to a previously defined search expression is called a "backward reference". X DFONT='7';

Note that it is possible to apply either the dot feature or tag qualification, or both, to a backward reference.

For example,

.#1/(64)

would restrict the search to those primary terms as listed in query element 1 and originating from tag 64.

There are no set rules for formulating a query. You will develop your own habits as you gain experience in using the system. In general, an inexperienced person should not try to use complex query elements immediately, but rather start with simple ones, which are then combined to produce the final formulation. By referring again to the example above, the same search could have been formulated in steps, as follows:

search expression #1:	ART
search expression #2:	ITALY + FRANCE
search expression #3:	CONSERVATION OF MONUMENTS
search expression #4:	#1 • #2 ~ #3

```
rc:=find(t);
findterm:=rc;
if rc=0 then
   if nxtpost<0
      then findterm:=1
      else begin
           mfn:=posting('MFN');
           rc:=record(mfn);
           findterm:=rc;
           if rc=0 then
              begin
              display(1,1);
              action:=decide(0);
              end;
           end;
end;


Function FLDUC(k: real): string;

{-------------------------------------------------------------}
{ Returns k-th field of record converted to upper case        }
{-------------------------------------------------------------}

var f: string;
begin
f:=field(k); uc(f);
flduc:=f;
end;


Function CHKREL(t: string): real;

{-------------------------------------------------------------}
{ Check if a relation already exists                          }
{-------------------------------------------------------------}

var i,n: real;
begin
n:=nfields; i:=1;
while (i<=n) and (flduc(i)<>t) do i:=i+1;
if i>n then chkrel:=0
       else chkrel:=i;
end;


Procedure UPDINVF;
```

```
{-------------------------------------------------------------}
{ Update inverted file (screen is clear because FST is displayed)}
{-------------------------------------------------------------}

begin
cleardata;
updif;
end;


Procedure CREATERM;

{-------------------------------------------------------------}
{ Create new thesaurus term                                   }
{-------------------------------------------------------------}

var tuc: string;
    rc,np: real;
begin
term:=''; clearmsg;
displt;
clearmsg; write('Enter new term');
rc:=edit(term,30,2,2,30,1,' ');
if term<>'' then
   begin
   tuc:=term; uc(tuc); rc:=find(tuc); np:=-1;
   if rc=0 then np:=nxtpost;
   if (rc=0) and (np>0)
      then errmsg('Term already exists')
      else begin
           mfn:=newrec;
           rc:=fldadd(tag[1],1,term);
           update; updinvf;
           writeln('allready updated');  {test}

           action:='S';
           end;
   end
   else action:='T';
end;


Procedure ADDREL;

{-------------------------------------------------------------}
{ Add new relation to a term                                  }
{-------------------------------------------------------------}
```

```
var r,rt,rtu: string;
    rc,i,rtag: real;

Function ADDIT: real;
var tt,ir: string;
    n,k: real;
    relmfn: real;

Procedure RELADD;
var rc: real;
begin
n:=nocc(rtag); k:=1;
while (k<=n) and (flduc(fieldn(rtag,k))<rtu) do k:=k+1;
rc:=fldadd(rtag,k+1,rt); update;
end;

begin
if (find(rtu)<>0) and (substr(r,1,3)<>'TXT')
   then begin
        addit:=1;
        errmsg('Related term does not exist');
        end
   else
if (chkrel(rtu)<>0) and (substr(r,1,3)<>'TXT')
   then begin
        addit:=1;
        errmsg('Relation already exists');
        end
   else
begin
rtag:=tag[(rtag-1)/3+1];
reladd;
ir:=substr(invrel,(rtag-1)*3+1,3);
if ir<>'   ' then
   begin
   k:=nxtpost; relmfn:=posting('MFN');
   rtag:=tag[(position(rel,ir,1)-1)/3+1];
   rt:=field(fieldn(tag[1],1)); rtu:=rt; uc(rtu);
   k:=record(relmfn);
   reladd;
   end;
k:=record(mfn);
addit:=0;
end;
end;

begin
box(18,10,3,5,1); box(18,14,3,52,1);
```

```
cursor(19,1); write('Relation');
r:=''; rt:='';
repeat
clearbox(19,15,1,50,1);
clearmsg: write('Enter link code: ');
for i:=2 to maxt do write(substr(rel,(i-1)*3+1,3),'  ');
cursor(23,1);
writeln('For HELP enter H in the relation field and press ENTER');
REPEAT
  clearbox(19,11,1,3,1);
  r:='';
  rc:=edit(r,3,19,11,3,1,' '); uc(r);
  IF r='H' then help('REL');
UNTIL r <> 'H';
rtag:=position(rel,r,1);
if rtag=0 then write(chr(7));
until (r='') or (rtag>0);
repeat
i:=0;
if rtag>0 then
   begin
   clearmsg;
   rc:=edit(rt,30,19,16,30,1,' '); rtu:=rt; uc(rtu);
   if rtu<>'' then i:=addit;
   end;
until i=0;
action:='S';
end;


Procedure DELREL;

{-------------------------------------------------------------}
{ Delete a relation                                           }
{-------------------------------------------------------------}

var rtag,rc,k,relmfn: real;
    rt,rtu,ir: string;
begin
rtag:=fieldn(dt[cl],doc[cl]);
rt:=field(rtag); rtu:=rt; uc(rtu);
rc:=fldde1(rtag);
update;
ir:=substr(invrel,(dt[cl]-1)*3+1,3);
if ir<>'   ' then
   begin
   rc:=find(rtu);
   if rc=0 then
```

```
      begin
      k:=nxtpost;
      if k>=0 then
         begin
         relmfn:=posting('MFN');
         rtag:=tag[(position(rel,ir,1)-1)/3+1];
         rt:=field(fieldn(tag[1],1));
         rtu:=rt; uc(rtu);
         rc:=record(relmfn);
         if rc=0 then
            begin
            k:=chkrel(rtu);
            if k>0 then
               begin
               rc:=flddel(k);
               update;
               end;
            end;
         end;
      end;
   end;
k:=record(mfn);
action:='S';
end;


Procedure DELTRM;

{-----------------------------------------------------------------}
{ Delete a thesaurus term                                         }
{-----------------------------------------------------------------}

begin
if nfields>1
   then begin
        errmsg('Cannot delete term with relations. Delete all
relations first.')
        action:='S';
        end
   else begin
        rc:=flddel(1);
        update; updinvf;
        action:='T';
        end;
end;


Procedure SHOWDICT;
```

```
{-------------------------------------------------------------}
{ List dictionary                                             }
{-------------------------------------------------------------}

var i,ii,k,sc: real;
    tp: array[1..16] of real;
    ts: array[1..16] of real;
    pg,ft: string;

begin
ft:=term;
repeat
pg:=''; i:=1; sc:=find(ft);
repeat
tp[i]:=size(pg)+1; ts[i]:=size(ft);
pg:=pg+ft;
ft:=nxtterm; i:=i+1;
until (i=17) or (ft='');
i:=i-1;
for k:=1 to i do
    begin cursor(k+4,5); writeln('  ',substr(pg,tp[k],ts[k])); end;
k:=1;
repeat
ii:=k;
chattr(1,k+4,5,30); term:=substr(pg,tp[k],ts[k]);
sc:=kbdkey(action); uc(action);
if action=chr(13) then k:=k+1 else
if action='B' then if k>1 then k:=k-1;
chattr(0,ii+4,5,30);
until (position('CPSTX', action,1)>0) or (k>i);
page(1);
until (position('CSTX',action,1)>0) or (term='');
end;



{------------------------- Body of program CHEM ---------------}

begin

maxt:=9;                              { Number of defined relations }
rel:=   '      USEUF POFCOFEQUPRTPRS';       { Name of relations }
invrel:='      UF USECOFPOFEQUPRSPRT';{ Name of inverse relation }
fullname:='Use:       Acronims:  Part of:
  Consists ofEquiv. to: End productPro
for i:=1 to maxt do tag[i]:=;;              { Tag of relation }
```

```
maxl:=15; q:='';
lbname:=dbn; { save currently selected data base }
if dbname<>'CHEM' then open('CHEM');
clear;
box(11,22,3,22,3);
cursor(12,24);
writeln('Trees of chemicals');
cursor(24,1);
writeln('Press any key');
action:=inkey;
if maxmfn=1 then action:='C' else action:='T';

repeat

case action of

'T': { Term selection }

     begin
     clearmsg;
     write('Select chemical name');
     term:=''; displt;
     cursor(2,2); readln(term);
     if term='' then action:='X' else
     if (substr(term,size(term),1)='$') or (findterm(term)<>0)
        then action:='L';
     end;

'L': { List of thesaurus terms }

     begin
     uc(term);
     rc:=find(term);
     page(1);
     clearmsg;
writeln('Y [Next]        B[previous]       P[age]        S[elect]');
     write  ('C[reate node]    T[erm select]     X[exit]');
     savescr(1);
     showdict;
     if term='' then action:='L';
     end;

'S': { Display term relations }

     begin
     rc:=findterm(term);
              WRITELN('NOW HERE');
                                   WRITELN('rc=',rc);}
```

```
     if rc<>0 then action:='L';
                    {      WRITELN('action='.action);}
     end;

'A': { Add a relation }

     addrel;

'C': { Create a new term }

     createrm;

'D': { Delete a term or a relation }

     if cl=1 then deltrm else delrel;

'Q': { Select term for searching }

     begin
     s:=field(fieldn(tag[dt[cl]],doc[cl]));
     if size(s)+size(q)+3>255
        then begin
             write('');
             action:='?';
             end
        else begin
             if q<>'' then q:=q+' + ';
             q:=q+s;
             action:=decide(cl+1);
             end;
     end;

'?': { Display current query }

     begin
     savescr(2);
     box(16,8,6,66,2); clearbox(17,9,4,64,1);
     cursor(17,9); lq:=size(q);
if lq=0 then write('No chemicals currently selected for querying')
 else
        begin
        k:=1; k1:=17;
        repeat
        if lq>64 then i:=64 else i:=lq;
        writeln(substr(q,k,i));
        k:=k+i; lq:=lq-i;
        k1:=k1+1; cursor(k1,9);
        until lq=0;
```

```
        end;
      clearmsg; write('Press any key to continue');
      s:=inkey;
      page(2);
      action:=decide(cl+1);
      end;

end;
until action='X';

if dbname<>'CHEM' then
   begin
   open(dbname);
   if size(q)>0 then
      begin
      clear;
      clearmsg; write('Edit search expresssion or press Enter');
      rc:=edit(q,254,2,1,254,0,' ');
      if size(q)>0 then rc:=search(q);
      end;
   end;
option :=' ';  {rhr}
end.
```