TABLE OF CONTENTS

LIST OF FIGURES

DATA SEARCHING AND DATAFILE STRUCTURE

USER INTERFACE SCREENS

PROGRAM STRUCTURE

Introduction

One of the most frustrating aspects of dealing with Indoor Air Pollution is the difficulty in acquiring information linking pollutants and specific source materials. Although there has been considerable research done in this area, the field of indoor air pollution is so new that it is often difficult to find relevant scientific articles dealing with this subject. Even with access to large university research libraries and on-line computer bibliographic databases it is very difficult to find articles dealing with indoor air pollution in general, and in particular those dealing with specific air pollutants and their sources.

Unfortunately, it is often the people who need these data most that have the least access to even these difficult to use resources. Architects, policy makers, builders, and Industrial Hygienists frequently have neither the hardware, time, expertise nor funds to do a comprehensive search for this type of data. On-line database access charges alone could run into the thousands of dollars before receiving the first list of applicable information. As a result, it is not expected that present on-line databases can offer adequate resources to these people in the area of indoor air pollution sources.

The recent proliferation of powerful desk-top personal computers and corresponding advances in software for these computers have made it possible to design and implement a stand-alone computerized database allowing quick access to this indoor air pollution data. Results

from scientific research concerning thousands of source-pollutant relationships can now be made available in a form that allows not only for the presentation of the bare-bones pollutant-emission data, but also for a comprehensive summary of research methods, test conditions, analysis methods, and bibliographic references. With the equipment that is available to almost any office today, a thorough and comprehensive database of indoor air pollution source evaluations can be put in the hands of anyone who would find it useful.

In a recent publication J. Crum of the US EPA at Research Triangle Park, NC introduced the concept of a computer program to accomplish these goals and an actual program called Indoor Air Source Emissions (IASE) database.[1] This program was written in *dBase III*[R] and was intended for use with IBM[R] and compatible personal computers. Although demonstrating an approach to finding and utilizing the broadly based literature on indoor air emission sources via personal computer, the program had serious flaws. It was felt that much could be done to increase the speed and versatility of such an indoor air pollution database system. This paper reports on progress toward these goals at UNC.

In order for any database of indoor air pollution source data to be most useful, it was decided that it should meet several goals:

1) It should be useful to users with a wide range of interests, including researchers, architects, policy makers, manufacturers, builders, designers, and modelers.
2) It should be as widely accessible as possible. This has been a major consideration in determining the best

combination of relational database management system and computer hardware to accomplish this goal.

3) It must be able to satisfy a broad range of needs. For example, some users may be interested only in bibliographic information, whereas others may wish to use the retrieved data to analyze specific indoor air pollution problems.

4) The system must be "user friendly"; *i.e.*, uncomplicated and undaunting to operate. This allows the data to be accessible to persons whose training is in areas other than in the use of computers.

5) Since it is anticipated that the amount of emission rate data will grow rapidly in the near future, the database must be easy to update and modify. Easy data entry also helps to minimize entry errors, another important consideration.

With these design goals in mind, a database containing source specific data from research articles has been compiled, along with controlling computer programs which will allow for quick and easy access to this indoor air pollution data. This database system has been titled dMAPS, the *database of Micro-environmental Air Pollution Sources*. The remainder of this paper describes dMAPS' function, design, and methods.

The need for an Indoor Air Pollution Database

In recent years the recognition that non-occupational indoor environments may contain air pollutants in concentrations which may be harmful to human health has led to increased interest in the recognition and identification of the sources of these pollutants. The trend toward home and office designs which employ energy conservation techniques such as tight construction, weatherization, and reduced ventilation has led to an increased opportunity for accumulation of air pollutants produced within these structures.

Identification and measurement of low level pollutants in non-industrial indoor air can be extremely difficult. These pollutants are not associated with specific manufacturing processes as they are in the industrial environment, but are instead produced by common materials within the environment itself. Since they are usually present in concentrations far lower than the recognized acceptable levels for occupational exposures they are not readily detected using traditional Industrial Hygiene sampling methods. Since extremely sensitive and expensive collection and analytical techniques must be employed to determine the concentration of a typical indoor air pollutant, it is very important to identify the most probable pollutants present before sampling is begun. The 'shotgun' approach to sampling and analysis is not appropriate because of the large number of compounds present in most indoor air, and the difficulty in determining which of these common low-level pollutants may be the cause of a problem.

The Industrial Hygienist traditionally uses the Recognition-Evaluation-Control strategy in determining the proper sampling strategy in an industrial environment. In an orthodox industrial environment the Industrial Hygienist typically evaluates manufacturing processes, recognizes the pollutants which may be emitted in these processes, and then recommends control strategies. Indoor air pollution complaints in a non-manufacturing occupational environment produce a different set of problems which the Industrial Hygienist often finds difficult to approach. The Recognition-Evaluation-Control method is often short-circuited in the recognition stage since there are no manufacturing processes from which obvious pollutants may be identified. It may be for this reason that Industrial Hygienists often find it difficult to determine a logical starting point to deal with indoor air pollution problems.

Because most indoor air pollutants are generated by materials both within and comprising the building's structure, one approach to evaluating indoor air pollution levels is the treatment of physical materials in the indoor environment just as if they were small manufacturing processes. Treating them as such re-establishes the recognition, evaluation, and control procedures familiar to Industrial Hygienists. Thinking of the various materials in the environment as pollution producing processes requires only that each material be associated with the pollutants that it may produce. Once back on familiar ground, the Industrial Hygienist should find the job

of formulating an indoor air pollution sampling strategy more manageable.

The dMAPS system has been designed to offer a starting point for the evaluation of indoor air pollution by identifying pollutants which have been shown to be associated with specific source materials. Data from published scientific articles dealing with definite pollutant/material relationships have been included in the dMAPS database which at this time includes almost 2,000 of these relationships.

As a further assistance in formulating a sampling strategy, the Industrial Hygienist may make use of dMAPS' features for display of information such as test method, analytic method, emission and environmental data for any pollutant/material relationship within the database. Table 1 details the data types in the dMAPS database that can be displayed.

TABLE 1

Bibliographic Information	Pollutant Information
Author(s)	Pollutant name
Journal	Emission measurement
Title	Comments
Publication Date	User Defined Field #1
Journal Reference	User Defined Field #2
Abstract	Units
Test Conditions	**Sample Description**
Pre-Conditioning	Sample Type
Test Duration	Sample Rate
Test Description	Sample Duration
Operating Parameters	Sampling Method
Temperature	Sample Size
Humidity	Analytical Method
	Other Conditions
Source Description	
Category	
Group	
Material	

Table 1. Listing of the fields in which data is stored in the dMAPS database. Any of this information is available to the user for any pollutant-source relationship.

The system also contains abstracts of all articles entered into the database, and displays full bibliographic references which are always associated with any presented data. Data is never displayed without the bibliographic information needed to reference the originating publication.

It is important to remember that dMAPS is not a replacement for review of the original articles. Although dMAPS has been made as complete as possible, space limitations prevent inclusion of the articles original text in the database. Comment fields have been included so that the most obvious characterizations of the article may be included, but these cannot replace the information and interpretation of the original author. It is strongly recommended

that the original articles be reviewed before final decisions are made on indoor air pollution control strategies, or before data from dHAPS is included in original research.

dMAPS Structure

Data in the dMAPS programs are contained in five datafiles. Each of these datafiles holds data concerning a particular type of data. For example, the SOURCE.DBF datafile holds data concerning source materials which were evaluated for emissions, while the BIBLIO.DBF datafile holds data concerning the bibliographic information about the publication from which the data have been collected. The information in these five datafiles is related to the others by key fields.

Each datafile contains units called records. In the Biblio datafile each of these records contain several types of information (Authors, Journal, Article Title, and Publication date) about one particular publication. If information has been entered for forty different articles, there will be forty records, each having information about a different article. There is an additional field in each record, unique for each record and normally invisible to the user, which contains a unique identification number for each article.

Information about sources of indoor air pollutants are held in the Source datafile, just as bibliographic information is held in the Biblio datafile. However, each record in the Source datafile also contains a small piece of the Biblio datafile: the ID number of the article in which its data was found. Therefore, any published article listed in the Biblio datafile which deals with several source materials will be linked to only those records in the Source datafile

which have its ID number. Conversely, due to their common ID numbers, any specific record in the Source datafile can immediately be associated with its parent record and all the parent records' data in the Biblio datafile.

This type of structure means that for every record in the Source datafile, and therefore every piece of data in the Source datafile, it is possible to immediately have access to all of the information about its bibliographic origin held in the Biblio datafile. It is this type of ID number linking which associates every piece of information in all five datafiles with the information about its publication article.

Since the dMAPS database contains five datafiles, the relationship between them becomes more complex than the example above. Each of these datafiles is related only to the datafile above and below it. The Source datafile for example, is related to the Biblio datafile as described above, but is also related to the Cond datafile in exactly the same way that the Biblio datafile is related to the Source datafile. In other words, each record in any datafile is related to one and only one parent record in the datafile above it. In turn, that parent record is related to only one grandparent record in the datafile above it. This chaining continues until the topmost datafile Biblio is reached, which contains the most general information associated with any piece of data. This chaining allows every individual record in any of the datafiles to have constant access to all of the more general information concerning it which has

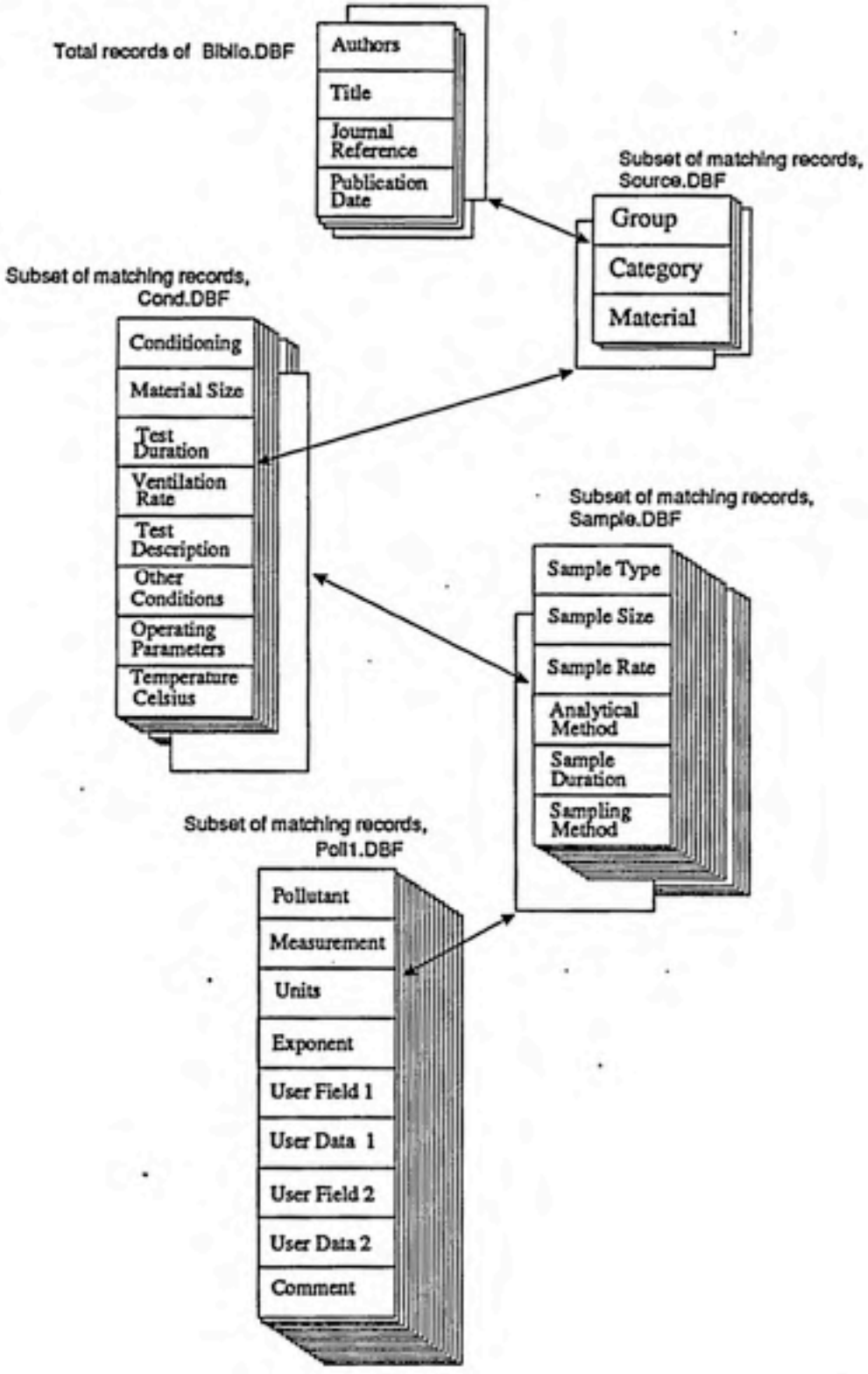


Figure 1: Representation of the relationship between databases in dMAPS. Each specific item (record) in each database is associated to related records in the database immediatley below it. This structure allows information to be related in data searches from general to specific; or once specific items are selected by the user, upward, from specific items to their general characteristics.

been entered into the other dMAPS datafiles. Figure 1 is a graphical representation of the logical linking of the five datafiles, while Figure 2 displays the same relationship in a different manner. Figure 2 also lists the fields in the datafiles which hold the ID numbers, and indicates how they relate the datafiles to those above and below.

Even though it would be more direct and would result in simpler datafile structures and searching programs to put all the information in one big datafile where all information from one bibliographic article is constantly associated, it would take up far too much storage space to replicate all the bibliographic information for each of the different conditions or pollutants covered in all the articles. It is therefore advantageous to relate the datafiles to each other via common fields so that the information common to many pieces of data need be contained in only a few records, not several thousand.

There are five datafiles in the dMAPS program, each related to the datafile immediately above it in the datafile hierarchy in the same way that the Source datafile is related to the Biblio datafile. This relationship is used in two ways in the dMAPS program. First, this structure allows the dMAPS program to access any piece of data in any of the five datafiles dealing with any specific published article; and secondly, as any specific piece of data in any of the datafiles is accessed, the relationships may be traced upward through all the datafiles, relating all associated data and ending in the Biblio

datafile with the associated data concerning the originating publication.

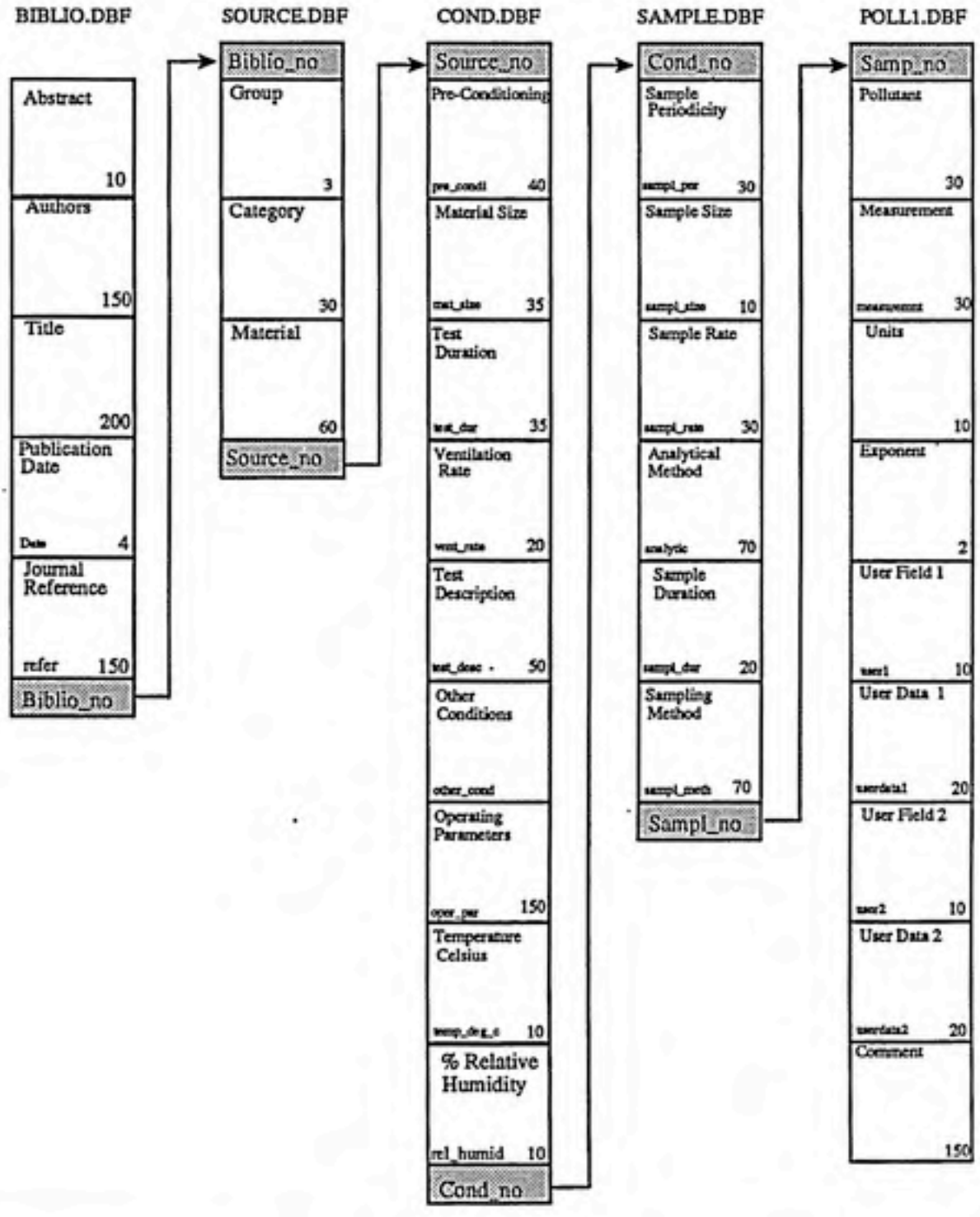


Figure 2: Contents of the five dMAPS databases and their linking fields. Each shaded field at the bottom of each database column is unique in that database, but is not unique in the following database. For example, the Cond_no field in COND.DBF will have only one record with a value of 4, while the Cond_no field in the SAMPLE.DBF may contain many records with the value 4. Small entries in boxes are actual field names used by dMAPS, numerals are the number of characters allowed in that field.

The following are descriptions of the data fields contained in the dMAPS program databases. The dMAPS program contains five databases, BIBLIO, SOURCE, COND, SAMPLE, AND POLL1. The numbers enclosed in parentheses after every field name indicate the number of spaces allocated in the database to that field. It is this data which is related by key fields in the manner previously described. Field names which are prefaced by an asterisks are fields which may be used for searches of the data.

BIBLIO DATABASE

This database deals with the information concerning the published article in which the data was found. All data in the other databases ares associated with a parent article contained in this database.

Abstract (10)

This is a special field in dBASE III called a memo field. Employed in every BIBLIO record, it allows large descriptive narratives to be associated with each individual record with out taking up large amounts of storage space. In this case the field contains the abstract of the published article associated with the record being viewed. The abstract is physically located in the file BIBLIO.DBT, but is linked to the database via this field.

**Authors (150)*

This field holds the names of the authors of the article. Listed in the order in which they appeared on the original

article heading, all authors are listed except when the length would exceed the 150 character maximum length.

Title (200)

The original title of the article, book, or publication as presented in the original publication.

**Date (4)*

The year the article was originally published.

Refer (150)

This field holds information concerning the reference publication from which the article was gathered. As well as the publication, other reference information is included in this field such as reference publication volume number and the pages the article occupies in the publication.

The previous four fields, when appended together by the dMAPS program, form a classical bibliographic reference: Authors, Title, Journal Reference, and Publication Date.

Biblio_no (4)

The number linking this database with the source database, and indirectly with all four other databases. Each bibliography record in the Biblio database receives a unique biblio_no which associates all data throughout the database with its original journal article. See Figure 1 for the ranking of the databases.

SOURCE DATABASE

This database has information concerning the indoor air pollution source classes.

**Group (15)*

This is the broadest division for indoor air pollution sources and generally includes such classifications as Water, Combustion, Consumer Products, and Structural Materials.

**Category (30)*

This classification is somewhat more specific than Group, but still is related to broad types of indoor air pollution sources. Types of entries in this area are Textiles, Insulating Materials, Plastics, and Applied Coatings.

**Material (60)*

This field is the classification that deals with the most specific sources of indoor air pollutants, and includes specific source materials such as Plywood, Paint, Carpet, and Cigarettes.

Source_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, COND. See Figure 1 for the ranking of the databases.

Biblio_no (4)

A field containing the bibliography number from which the record is associated. Many records in this database may share the same biblio_no.

COND DATABASE

Information concerning the experimental conditions are contained in this database. Either material condition or environmental condition factors effecting emissions from are in this database.

Pre_Condi (40)

Any **pre-conditioning** of the source material before the beginning of the test is entered here.

Test_dur (35)

The **test duration** is entered here. This is not the same as the sampl_dur field in the sample.dbf, which is the duration of the sampling activity, but is the duration of the entire test under any particular condition.

Test_desc (50)

A short verbal **test description**, including any information which may be useful to the dMAPS user.

Oper_par (37)

Any **operating parameters**, environmental or otherwise, which may have had an effect on the outcome of results of the test.

Temp_Deg_C (30)

The **temperature in degrees Celsius** under which the test was conducted.

Rel_humid (10)

The percent **relative humidity** under which the test was performed.

Vent_rate (20)

The **ventilation rate** of the material during the time the measurements were being taken, or during the time of the entire experiment, whichever is more appropriate.

Mat_size (35)

Material size, the surface area or any other measurement identifying the amount of material evaluated.

Other_Cond (150)

Any **other operating conditions** which may be of use in evaluating the data. Any special conditions of the test, special environmental conditions, or comments about the test are included in this field.

Cond_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, SAMPLE. See Figure 1 for the ranking of the databases.

Biblio_no (4)

A field containing the bibliography number from the previous ranking database with which the record is associated. Many records in this database may share the same biblio_no.

SAMPLE DATABASE FIELDS

This database holds information concerning sample collection procedures and analytic methods.

Sampl_Per (30)

This is the **sample periodicity**: Continuous, Periodic, Integrated, etc.

Sampl_Rate (30)

This field lists the **sampling rate** at which the pollutant was measured, such as liters/minute.

Sampl_Dur (20)

The **sample duration**, the length of time a sample was taken for evaluation of the emissions from a particular material/condition combination.

Sampl_Meth (70)

This is the method used to collect or measure the pollutant emitted by any particular source material. The **sample method** includes entries such as charcoal tube, direct reading colorimetric tubes, or impinger collection into distilled water.

Sampl_Size (10)

Sample size, the amount of sample taken for analytical measurement. This is not the statistical sample size, nor is it the size of the material from which pollutants were measured.

Analytic (70)

The **analytic method** used to determine the concentration of pollutant collected in the sample device. When direct reading instruments are used to determine the pollutant concentration, this field can be the same as the sample method or the method that the direct reading instrument uses to determine concentration.

Samp_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, POLL1.

Cond_no (4)

A field containing the condition number in the previous ranking database with which this record is associated. Many records in this database may share the same cond_no.

POLL1 DATABASE

This database contains information concerning the final test results, including measured emissions for every pollutant in the database. Also included here are the user defined labels and data fields, and

specific comments concerning any individual pollutant and its test condition.

**Pollutant (30)*

This is the chemical name of the pollutant that was evaluated under the conditions of the experiment.

Measuremnt (30)

The air concentration of the pollutant emitted by the source material being tested.

Units (10)

The units of concentration of the pollutant measured as reported by the original article. This may be in PPM, Mg/m3, or any other appropriate units. If both CGS and English units are reported, CGS units are preferred.

Exponent (3)

When measured concentration is extremely large or extremely small, there may not be enough room in the measuremnt field to accommodate these numbers. If this is the case the exponent field may be employed to allow meaningful entries in the measuremnt field.

Comment (150)

This comment field is associated with each pollutant/measurement value and is available for detailed description of how this particular test or condition is unique.

Userdata1 (20)

This is one of two user definable fields, and is called the User Defined Label in the data entry screen. The user may enter any label into this field, and associate any data with it in the User1 field, which is called Data in the entry screen. Any type of information or comment concerning this pollutant and this condition may be entered in these fields.

Userdata2 (20)

Second of two user definable fields.

User1 (10)

The data field associated with the Userdata1 field which contains its label.

User2 (10)

The data field for the Userdata2 label.

Samp_no (4)

A field containing the sample number which associates records in this database with a particular samp_no from the previous ranking database, SAMPLE. Many records in this database may share the same samp_no.

DESCRIPTION OF PROGRAM SEARCH LOGIC

The dMAPS program allows searches of the database on seven key fields. These fields are *Group, Category, Material, Pollutant, Authors, Date of Publication, and Journal title.* The *Group, Category, and Material* fields allow searches for data associated with indoor air pollution sources, while the *Pollutant* field allows searches for specific pollutants. The *Authors, Date, and Journal* fields allow searches for data associated with specific published articles.

The program also allows any combination of these fields to be searched simultaneously. For example, the user may query the program to find all information concerning the emission of the pollutants *formaldehyde* and *hexane* from the source material *carpet*.

The five separate databases in the dMAPS data section are related to the others through identifying fields (Figure 2). Although this type of structure requires a complex data relation formula, it is necessary to use this technique to save disk storage space. A more detailed description of database relationships in dMAPS is given in the dMAPS Structure section.

Although there are five databases in dMAPS, searchable fields are found in only three of them. The *Author, Date, and Journal* fields are located in the BIBLIO database, *Pollutant* is located in the POLL1 database, and *Group, Category, and Material* are in the SOURCE database. Which databases are searched depends upon the user's search criteria. If the user picks only a pollutant, dMAPS will search only the POLL1 database, while if the user picks a date and a pollutant, the program will search the POLL1 database and then the BIBLIO database finding information meeting the user's search criteria in each. As matching information is found in each database

the program assembles a character string identifying which published article is associated with the match. After the searches are complete these strings are then compared according to the logic operators the user has specified.

Single Database Searches

The internal programming algorithms used to search for user queries has been written to provide the speediest search possible. In order to provide this fast search, the program begins searching every record sequentially until a match is found, and then "remembers" the article from which the data came. The program then resumes the search. To speed the search however, the program now ignores all records associated with that article. This is useful since once an article is identified as containing information of interest to the user it is redundant and time consuming to repetitively re-identify it.. As other matches are found at other locations in the database, these new article ID numbers are also remembered and subsequently ignored.

At the end of the search, these "remembered" article numbers are used to allow the data display portion of the dMAPS program to quickly jump between sections of the database which contain information concerning the user's needs. Areas which are not associated with the user's search criteria are quickly skipped. For example, if article number ten was the only article which dealt with the user's query, then only the data derived from that article will be accessed.

Multiple Database Searches

If a user specifies two search fields that are in separate databases, the above procedure is carried out with each database. The result is string A containing the ID numbers of the articles matching the first

criterion, and String B containing the ID numbers of the second criterion. The program is then able to evaluate the interaction of these two strings via any logical operator the user has specified. See Figure 3.

If the logical operator is "OR", as is the case when the user requests information concerning CARPET or FORMALDEHYDE (A union B) String A is compared to String B and duplicate ID numbers are deleted. A third string, String C, is then created containing all of the ID numbers from both strings. This operation, A union B, is performed in the program COMPOR.

If the logical operator is "AND" (A intersect B) the program determines which article ID numbers are common to both strings. This results in a String C which contains ID numbers for articles which have information concerning search criteria A and search criteria B. Comparison for A intersect B is performed in the COMPAND program.

When three different databases are searched, the process is nearly identical. In this case however, when string A is compared to string B the result is stored to a new string A. The string from the third database is compiled, stored to a new string B and then compared to A. When the program has searched through all the appropriate databases, the final stored string contains a list of all articles which contain information about the user's search criteria. Although this sounds complicated, it practice it is quite straight-forward as can be seen in Figure 3.

The above searching method speeds operation of the search in two ways. First, skipping records is much faster than evaluating them to determine if they match a search criteria, and secondly, this method allows several databases to be searched and the logical relationships between the databases found without linking them on common key

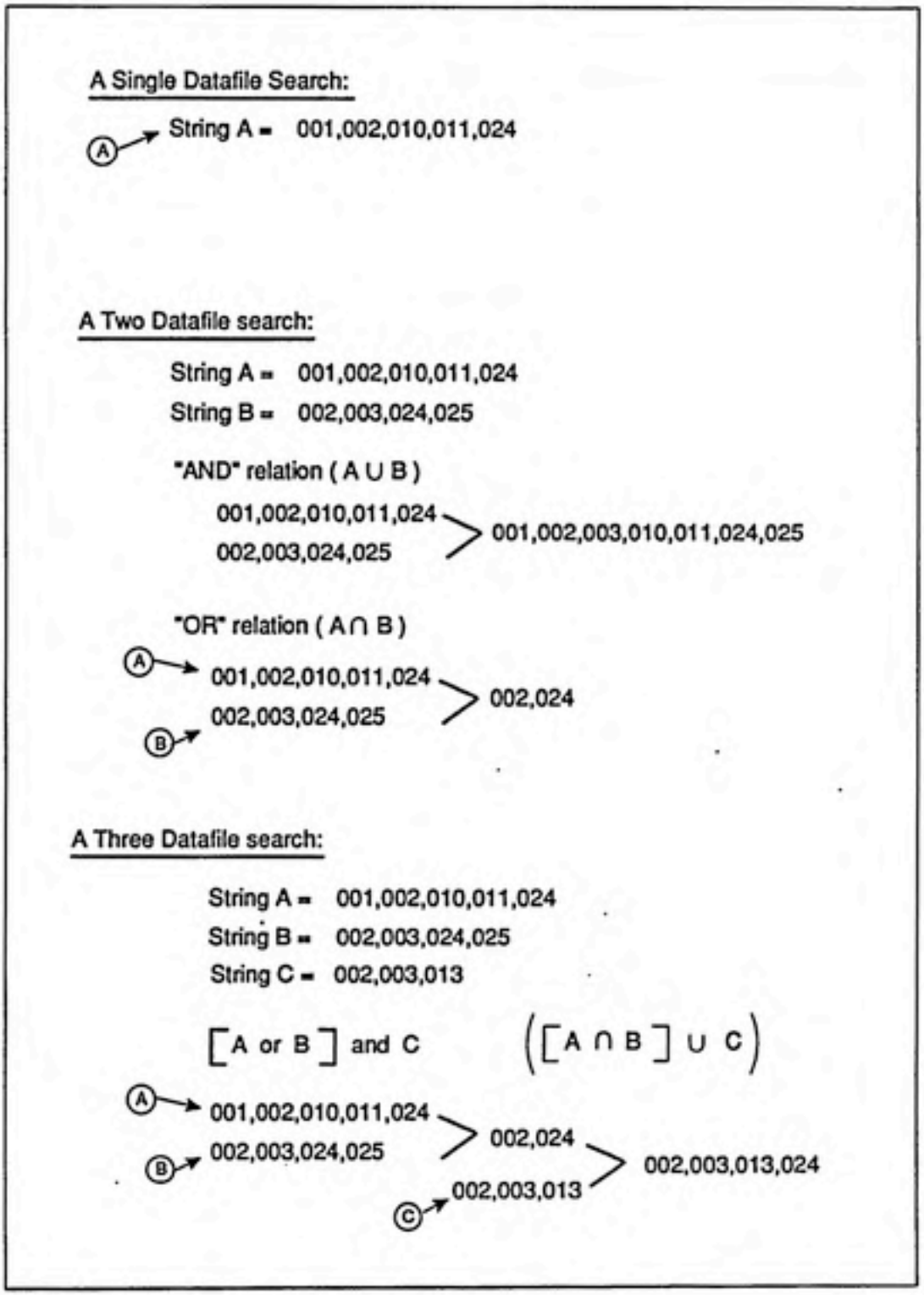


Figure 3: The method by which the dMAPS program determines which datafile records meet the user's criteria. The three digit numbers represent the bibliography numbers which have information matching the individual search criteria. The matching articles are listed as strings A, B, and C which are compared to each other according to the logical relationships assigned by the program or the user.

fields. Normally, searches conducted on databases constantly linked by common key fields are the easiest way to search multiple databases in dBASE III Plus, but the method is very slow since the pointers in each linked database must be updated every time the pointer is moved in any of the databases. In dMAPS this would requires opening and closing of each of the five databases in turn to move the pointer. Even though it is done automatically, the opening and closing of these databases takes considerable time in large databases. The method used in dMAPS uses key fields between databases, but does not link the databases together. Comparisons are performed after each database has been searched.

In summary, in order to make searching of the dMAPS databases as fast as possible a dedicated searching program was developed to search each database separately, compile lists of matching records in each database, and then to logically compare these lists against each other. The result is a list of articles which contain information related in the manner the user specified in his or her search criteria. Although requiring several program modules in the dMAPS program, this method of searching provides an average searching speed sixteen times faster than the standard dBASE III Plus searching method. This considerable improvement is speed means that a search taking 37 seconds using the developed search procedure would take about 10 minutes using the standard dBASE III Plus method.

USER INTERFACE SCREENS

The following pages contain representations of the dMAPS user interface screens. If the user is interested only in using dMAPS as a tool, and is not interested in the means by which the dMAPS program accomplishes its goals, then these screens are the only parts of dMAPS that the user sees. To this type user the following screens are essentially dMAPS. All database relations, program relations, search algorithms and data manipulations are transparent and need not be considered by this type of user.

These screens have been designed to lead the user through the steps to extract meaningful information from the database. Menus are provided rather than asking the user to know or learn dBASE-like commands. Additionally, the user is not required to be particularly aware of the type of information that is in the database. By selecting broad or narrow search parameters from the menus, the user is presented with lists of available options or choices. It is not even necessary that the user be able to spell formaldehyde to find articles dealing with it.

The following screens are presented generally in the order in which they would appear to a user conducting a search for a pollutant. To aid in clarity of this written presentation not all screens have been shown, and some instructions which are normally included in the screen versions have been omitted.

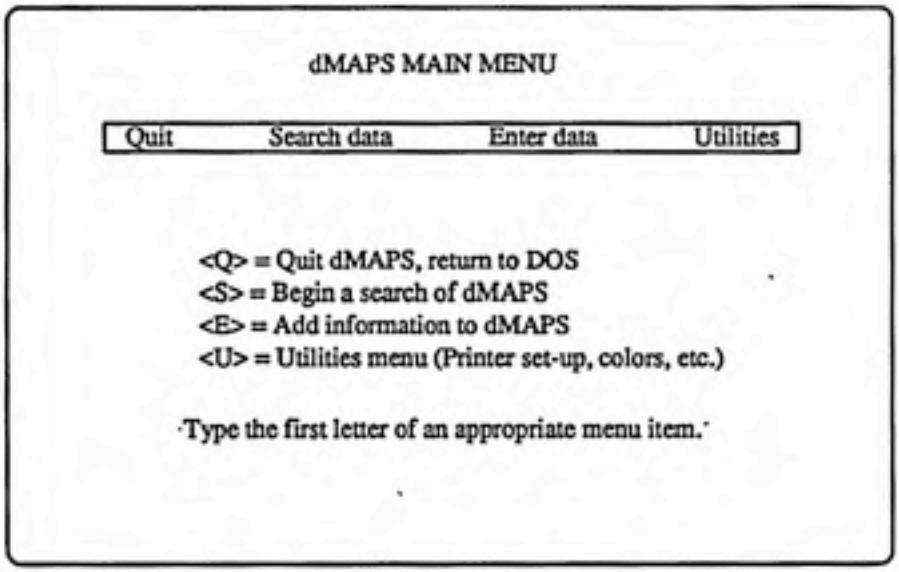


FIGURE 4: MAINMENU SCREEN

In this screen, the first after the title screen, the user is offered four options:

1. Quit
2. Enter
3. Search
4. Utility

1.)This screen is the only one from which it is possible to quit the program. The Quit option allows the user to exit the dMAPS system to DOS.

2.)The *Enter* option causes program execution to branch to the data entry portion of the program where the user may enter new data to the database, or alter data already entered.

3.)If the user selects *Search* the program branches to the data searching modules of dMAPS where the user specifies the searching parameters for data retrieval.

4.)The *Utility* choice allows for the selection of monochrome or color displays, selection of screen colors and attributes, and for other program related functions. Any features specified here are stored so that they are called automatically whenever dMAPS is subsequently re-started.

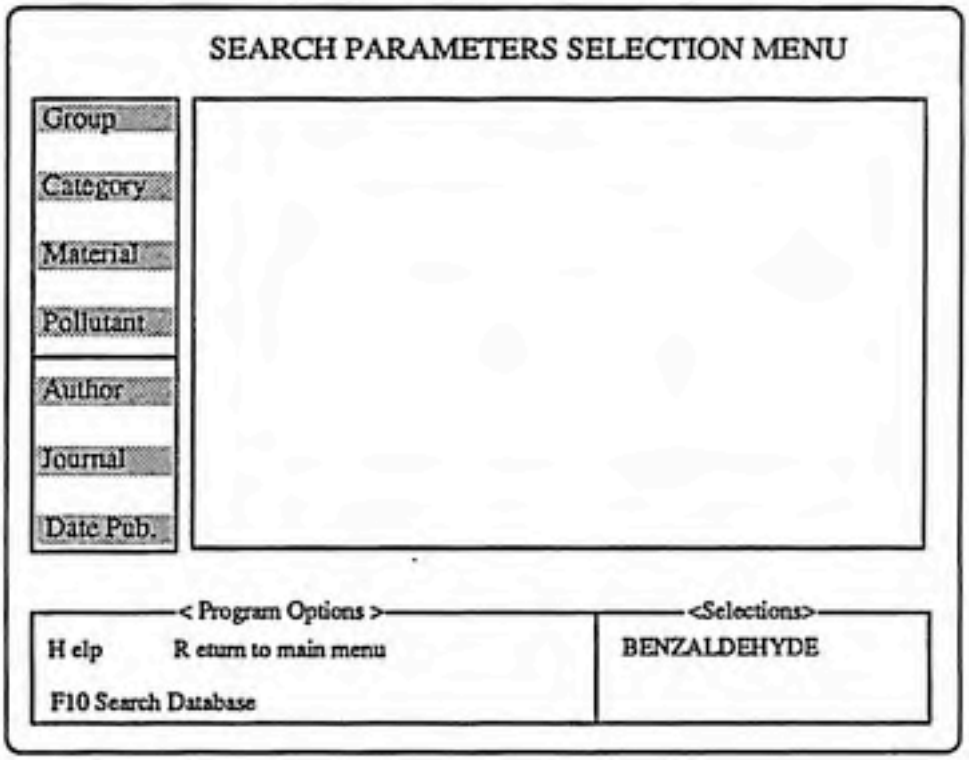


FIGURE 5: SEARCH PARAMETER SELECTION SCREEN:

This screen is the main screen of the search parameter selection module. From this screen the user picks the areas of the database to search by typing the first letter of one of the items listed at screen left. Depending on which selection the user makes, the program routes the program to various secondary selection screens from which more detailed search information is chosen by the user.

Other options in this screen are displayed in the Program Options box at the lower left. By typing the appropriate letter the user may get immediate help, return to the previous menu (the main menu), or search the database if a search parameter has been chosen.

The Selections box at the lower right displays search parameter selections as the user makes them. In this screen the user has chosen BENZALDEHYDE as a search parameter and now has the option of searching the database by pressing F10, or of specifying more selection parameters by pressing the first letter of one of the search areas. Up to three search areas may be selected simultaneously.

The user Program Options to the left of the selection menu is present on all screens where the user has to make a choice on how to proceed, and displays instructions on what options are available

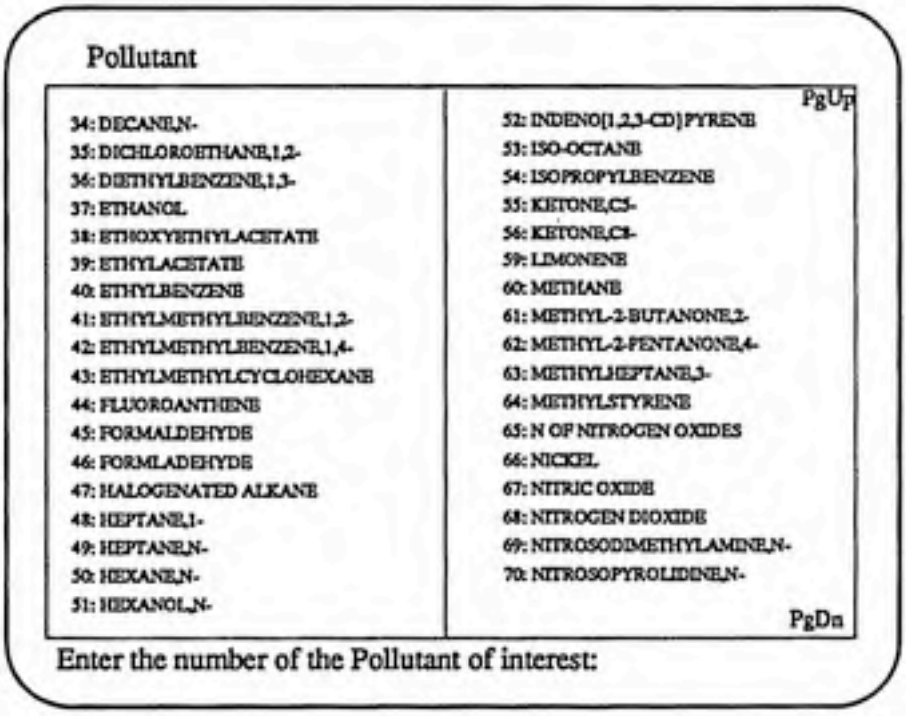


FIGURE 6: MENU SELECTION SCREEN:

This is an example of a menu type secondary selection screen. This type of menu appears whenever the user has chosen Material Group, Material Category, Specific Material, or Pollutant from the main selection menu. The items in each of these menus accurately display all of the available selections in that area. For example, the screen shown above appears when Pollutant has been selected in the main selection menu.

Even though each pollutant is only listed once, it may be associated with many articles or sources in the actual database. An intermediate screen allows the user to pick a single letter, with only pollutants beginning with the chosen letter being displayed in the above screen. This greatly increases display speed..

Typing the number associated with any pollutant will cause the program to return to the main selection menu with that pollutant appearing in the Selection box, at which point. At this point the user may search for the selection by pressing F10.

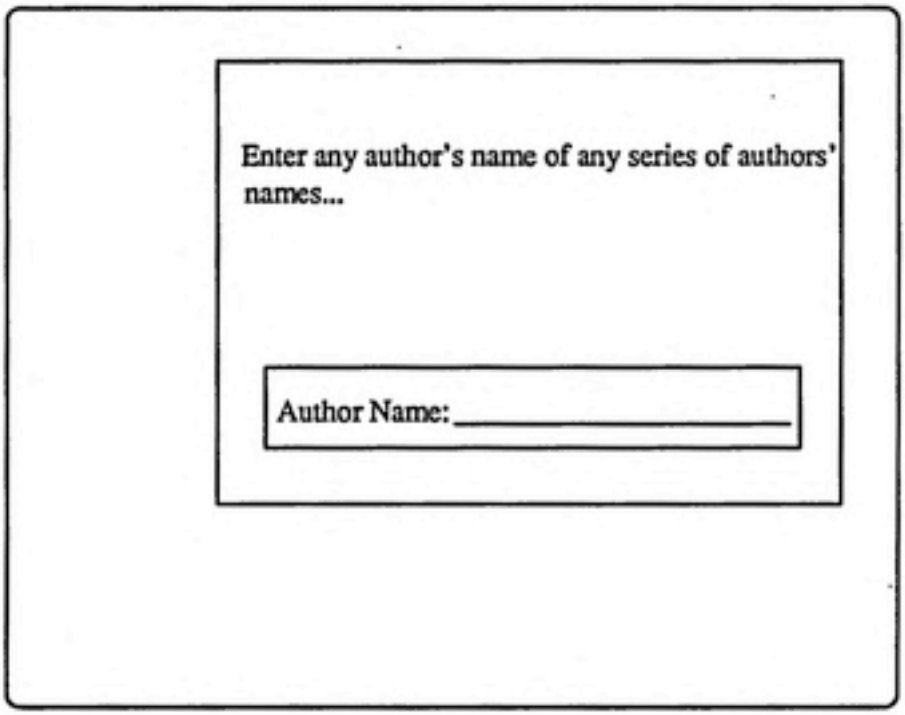


FIGURE 7: FREE-ENTRY SCREEN

An example of a free entry screen is shown in the figure above. The user may enter any author's name at this point and the program will determine how many articles by that author are present. The user may then incorporate this name into the program as a search parameter or may enter another name or series of names. This type of menu is also used in seléction of publication dates and journal titles.

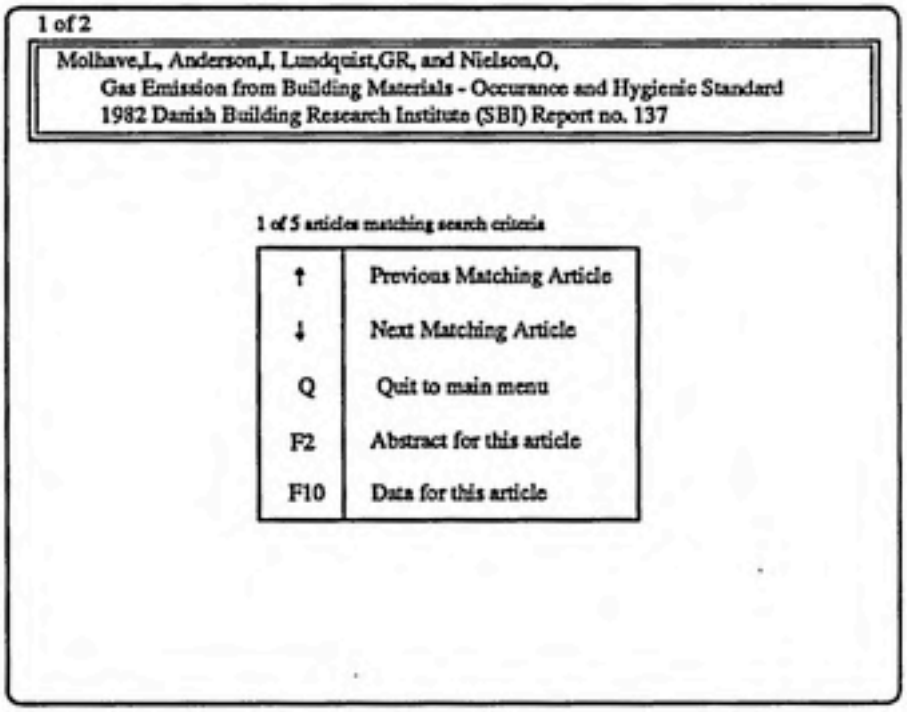


FIGURE 8: BIBLIOGRAPHY DISPLAY SCREEN

This screen appears after a data search has been performed by the program, and is the first point at which the user begins to see the contents of the database which deal with his or her search criteria. The first article that contains data concerning the search criteria is displayed at the top of the screen. The location in the list of articles is displayed slightly above the menu box. In this case the location is at the first of five articles which contain data meeting the search criteria.

Options at this point are to page through the other four matching articles with the arrow keys, look at the data for this article, view an abstract for this article, or return to the main menu.

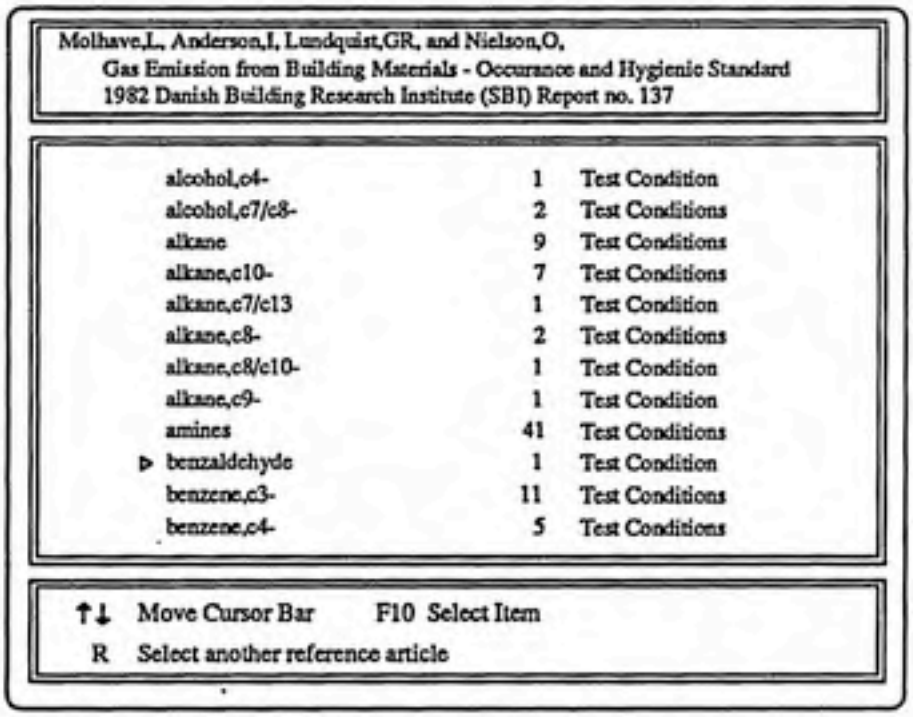


FIGURE 9: DATA SUMMARY SCREEN ONE

This screen begins the display of data from any article that the user has chosen from the previous screen. Each pollutant in the article is listed along with the number of different test conditions for each pollutant. As can be seen in the above screen, the amount of data in a single article can be very large. This article lists over 70 pollutant test conditions on this screen alone, with more screens available (PgDn).

To see data concerning a specific pollutant, the user moves a highlight bar through the available choices. By pressing F10 the data for only the highlighted pollutant is displayed.

If any pollutant/condition group contains a user specified search item, for example a specific source chosen by the user, it is marked with a delta. Although all pollutant/conditions are displayed for each article, only those groups so marked contain items that meet the user's search criteria. In the above screen, benzaldehyde has been marked as meeting the user's search criteria.

<<<Pollutant = AMINES>>>

Material	Measured Emission
vinyl coated wallpaper	not detected
floor varnish	not detected
plywood	not detected
chipboard	3 ppm
caulking compounds	not detected
needle felt	not detected
insulation sheets	not detected
floor/wall covering	not detected
calcium silicate sheet	not detected
wall covering	not detected
insulation foam	not detected
jointing compound	not detected
needle felt	not detected
linoleum floor covering	not detected

FIGURE 10: DATA SUMMARY SCREEN TWO

This type of screen can be presented for every pollutant/condition entry in data screen one, the previous screen. The above screen shows the results as if the user had chosen AMINES in the previous screen. All source materials and measured emissions are displayed for each of the 41 conditions listed on the previous screen.

If any of these materials are of interest to the user he may move the highlight to that material and press F10 to view data concerning the details of the test.

By positioning the highlight on a material of interest with the cursor keys and pressing F9, the user can restrict the presentation of data in the next screen to only those that specifically deal with the material that was highlighted. All other materials are ignored. An F10 will present data concerning this pollutant for all of the displayed materials.

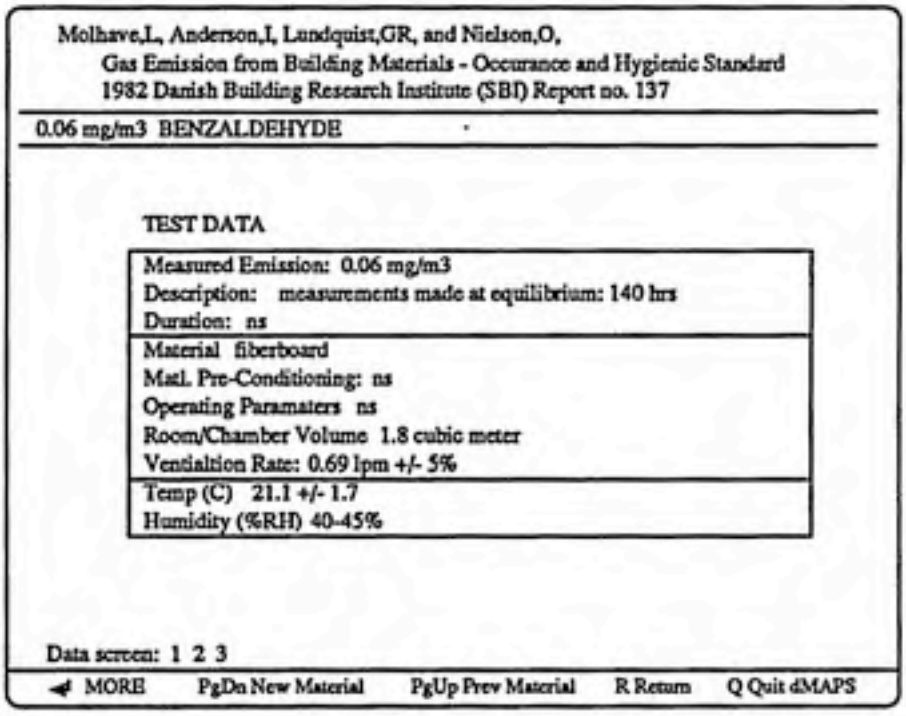


FIGURE 11: DATA OUTPUT SCREEN ONE:

This is the first of three data display screens which show the entire data set for this pollutant/material/condition relationship. This screen shows the test data and results given in the article listed at the top, the article which has been chosen previously by the user. By using the keyboard cursor keys the user can page through all three screens: this screen of test data, screen 2 which is sampling information, and screen 3 which includes comments and user defined fields.

When the user presses PgDn, data for the next set of conditions for this pollutant replaces the data in the display. If the user had accessed this screen by pressing F9, the display would shift to the next occurance of data concerning benzaldehyde from fiberboard. If the user had pressed F10 to access this screen, a PgDn would display the next data concerning the pollutnat benzaldehyde, without regard to what specific source material it may have been associated with.

Since much of this data is similar, any data that is different from the previous screen is highlighted.

DATA VIEWING

After the user has entered a correct search criteria and the dMAPS program has searched the database for data concerning those criteria, the amount of data that is available for presentation can still be very large. For this reason after the user has specified certain areas of interest a series of data output summary screens are presented. The need for these screens is obvious when it is realized that even one article may contain hundreds of separate entries. For example, if an article evaluates 10 source materials at two temperatures, and in each instance analyzes for 10 pollutants, the result would be 200 separate records associated with this one article. Clearly, to merely present the user with all the data at this point would make it very difficult to gain useful information from the database.

To allow the user to see data more specifically useful to his or her needs, the dMAPS program employs a series of data summary screens which allow the user to progressively filter the information to more manageable amounts. The procedure for data display is as follows:

1. Select a specific article which has been found,
2. Select a specific pollutant in that article,
3. Select source material(s) for that pollutant,
4. View data.

Figure 8 is a representation of the screen to select individual articles. By using the arrow keys the user can page up or down to the articles that dMAPS has found that match the user's search criteria. The user can then view the data in the selected article. At the present time there is no way to cross the article boundaries and view data that are in articles other than the selected article. To view information held in other articles, the user returns to this screen and selects another article.

The next screen, Figure 9,. is the first data summary screen and presents the first level of summary data for the selected article. As the user continues through this and the following summary screens the data displayed becomes more complete. This screen presents a list of all the pollutants dealt with in the article and the number of test conditions under which emissions of this pollutant were measured. The user selects a pollutant of interest by highlighting a pollutant by pressing arrow keys and then pressing F10.

Note that some of the pollutants shown are marked with a delta. If the user had originally specified a material in the selection section, a delta appears when that criteria is met in one of the test conditions for that pollutant. For example, if the user had specified "carpet" as a source material, the dMAPS program marks every pollutant in this article that has had its carpeting emission levels evaluated. The user may then choose only these tests to view in detail.

The next screen, Figure 10, is reached when the user selects a pollutant from the previous screen. Listed here are the source materials and the measured emissions from that material. If the user wishes to proceed, pressing F9 will present detailed information concerning all of the listed source materials, or F10 will allow the listing of only those materials similar to the highlighted material. The user may move the highlight to any specific material with the arrow keys before pressing F10.

If the user choses to continue, the next series of screens presents the detailed data available from that article/pollutant/source material combination. The format of these screens are shown in Figure 11. Three of these screens are available and present all the data in the datafiles for this particular combination. A more detailed description of these screens can be found under each Figure in the User Interface Screens section.

It is hoped that future versions of dMAPS will allow for convenient printing of the data found in the search procedures. Presently the only printed output is the Print Screen function of the computer. Printing of data will allow larger amounts of data to be output at one time than can readily be done from the video display, and will greatly enhance the ease of access to data.

Data Entry

Data entry for dMAPS is achieved by entering data and bibliographic information from articles in a special module of the program. The information entered in this area is stored in a file that is not incorporated into the full dMAPS datafiles until the user requests it. This method speeds up data entry and makes the data more easily appended or changed.

The data entry module is reached from the main menu screen, while the command to update the dMAPS datafiles with the newly entered information is in the Utilities module. These two actions are separated in order to discourage the frequent updating of the main datafiles for just a few newly entered records. Frequent updating is undesirable since it takes time to order and index the newly updated datafiles, and also because it becomes more difficult to correct entry errors once the new data has been incorporated into the main datafiles.

As can be seen in Figure 13, the data entry screens are divided into sections with each section corresponding to one of the dMAPS datafiles. Although the user must keep track of the structure of the data entered into the program, effort has been made to make this task as easy as possible. Repetitive typing has been minimized, and the input screens have been organized in such a way that user can most easily determine the datafile structure.

In order to do this the data entry screens have the entry blanks arranged in order of increasing specificity. The bibliographic

information comes first since this data will not change throughout the article being entered. The next section contains the entry blanks concerning the sources of pollutants covered in the article, while the next section contains information concerning the different conditions under which the sources were evaluated. The final two sections contain information about the sampling method for each of the source/condition relationships mentioned above and, finally, about the evaluated pollutant and associated test results.

This structure makes it relatively easy to enter large amounts of data for single articles. When the user first begins entry for an article all of the entry blanks are empty, and the user fills in all the appropriate blanks. The user then calls up another screen to fill in, but since much of the information will be identical the next screen appears with the new entry blanks filled with the old information from the previous screen. The user may then change only those items that require it. The first screen, completely filled in by the user, and every subsequent screen partially filled in by the user constitute one record each in the dMAPS datafiles.

This arrangement is such that if any item is changed only items in its immediate section or in the sections below it need be updated. For example, in Figure 13 it can be seen that if an author conducted research on two pollutants under identical conditions and used identical sampling methods, only fields in EXPERIMENTAL RESULTS and below are candidates for change, everything above would remain the same. On the other hand, if the user began entering data from an article concerning a different source material, all entry blanks

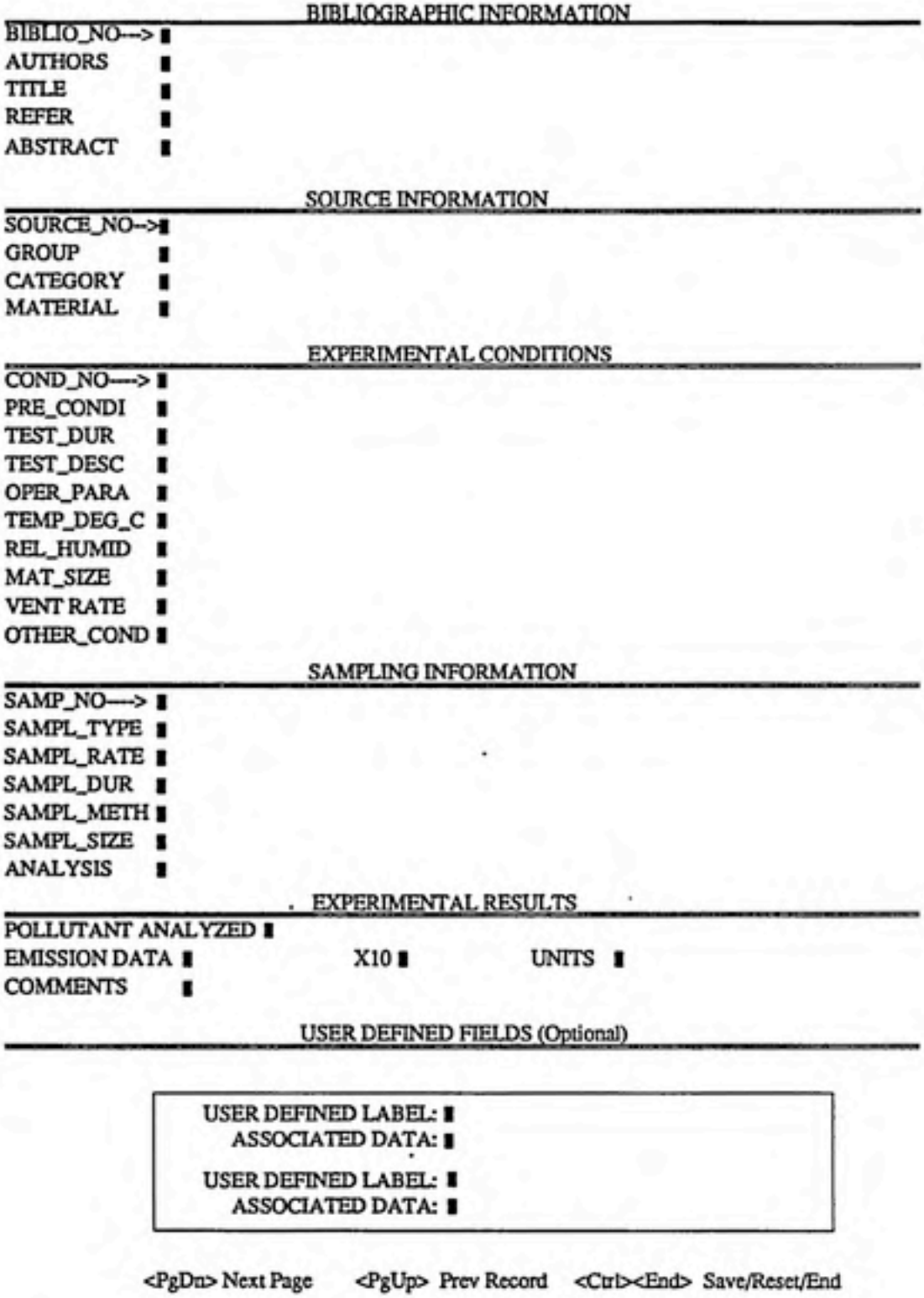


Figure 13: Representation of the data entry screen accessed from the Entry selection of the main menu. Data entered here is stored to a temporary datafile until the user appends it to the main datafiles.

within and beneath the section containing SOURCE INFORMATION would be candidates for change. Again, no areas above it would have to be altered.

The hierarchical format of the data entry screens make it almost as fast to enter data for an article dealing with 10 pollutants as for one dealing with only one pollutant. Although it may be possible on future versions of dMAPS, the program is not now able to automatically determine the relationships between data in an article. This task must be done by the user. Again, this is made relatively simple by the hierarchical structure of the data input screen. An easy to remember rule is that when any item is changed, the corresponding key number in that section and in all sections below it must be increased by one. This rule relieves the user of most of the task of keeping track of the relationships between article data.

As the user enters data from articles, the data is stored in a temporary datafile called UPDATE.DBF. This file is stored and appended to until the user determines that its information should be incorporated into the main dMAPS datafiles. This makes it easy to stop in the middle of entering data for an article and to resume later. The program either picks up where the user last stopped, or presents a new blank screen to begin another article, according to the user's response to a series of questions at the beginning of the data entry screens. The key for the Biblio_no field is automatically entered for the user according to the last entry.

For additional information concerning the best procedures for entry of new data and a tutorial lesson, consult the dMAPS Manual.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

This program has been structured so that it may serve as a starting point for further development. The program is in modular form, meaning that different functions such as data entry, data searching, and data display are in separately written sub-programs called Procedures within the dMAPS main program. This form makes it convenient to add other procedures to add other operations or features. Recommendations for additional procedures include:

1. A module to allow printing out the data via a printer in a form more convenient than the screen display format.

2. A module to transfer data from the dMAPS program, either found data or data *in toto*, to a Lotus 1-2-3 spreadsheet. This will serve two purposes. First, having the data in a spreadsheet format familiar to many computer users will allow easier analysis of the data, and secondly, it will allow the output of selected areas of the data into a structure which will allow easier viewing of grouped data. DBASE III Plus and Lotus 1-2-3 each have their own strengths in data manipulation, and a procedure to allow data to be transferred to a spreadsheet from dMAPS will greatly increase usefulness of the data contained in dMAPS.

3. It is presently necessary to view data grouped by published article. This is only one way that data output may be structured,

and other output formats may be more convenient for different users. Specifically, displaying data across bibliographic reference boundaries would be particularly useful, presenting data on one source material, pollutant, or other key field across the entire database.

4. Two versions of the program have been written, one for DBASE III Plus, and one for the compatible database program Foxbase Plus. The Foxbase Plus version is superior to the dBASE version, particularly in areas such as disk-space storage requirements and speed. Additionally, the dMAPS program written for Foxbase may be compiled to run without the presence of the entire Foxbase program, but with a Run-time version which may be distributed to users without charge. This greatly increases the usefulness and audience of the dMAPS program. I recommend that Foxbase Plus be used for all future developmental work.

5. Conversion programs are available to convert DBASE (and Foxbase) programs into the C programming language. This would be an extremely desirable task for three reasons. Since programs in C are able to run under many different operating systems, encoding dMAPS into C would greatly increase the available audience by allowing the program to run under non-DOS operating systems such as UNIX. Conversion to C would also mean that dMAPS would exist as executable files requiring no additional software. Finally, programs in C are extremely fast. Program operations now taking 17 seconds in DBASE and 8 seconds in the faster Foxbase programs would execute almost instantaneously in a C version.

6. The present version contains data from forty articles dealing with indoor air pollution. In order for the program to remain useful, data must continue to be entered as more articles are published and found in literature searches. As stated in the introduction, private individuals have neither the time, money, nor skills to do comprehensive literature searches for new data. Indeed, that lack of ability has been the main reason to develop dMAPS. It would be a serious mistake to now burden the users by requiring that they alone keep the dMAPS program datafiles up to date. This responsibility should fall on the group which has ultimate control of dMAPS development, be that the University of North Carolina or the US EPA.

APPENDIX A

dMAPS DATABASE STRUCTURE AND SELECTED RELATIONS

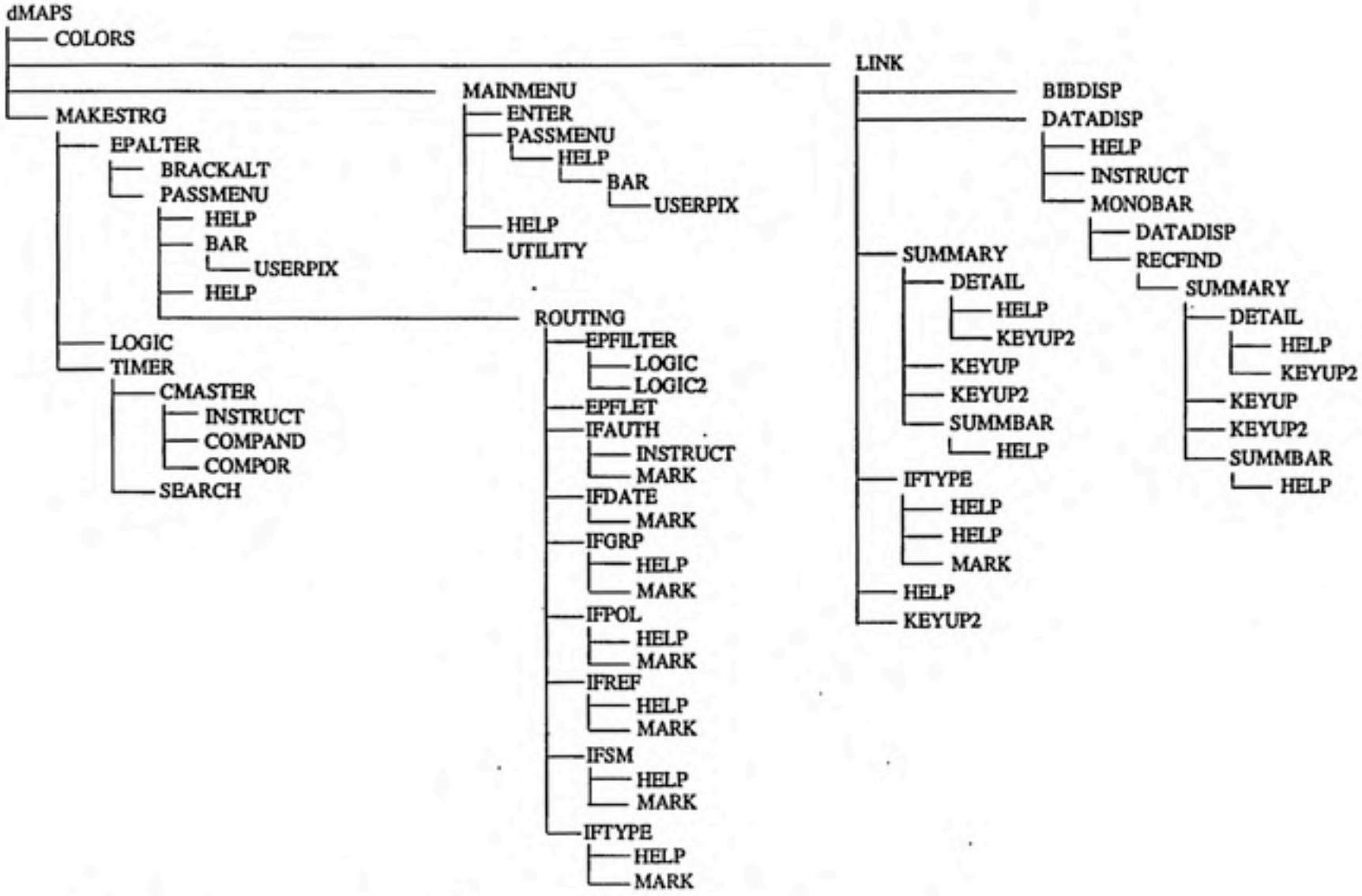


Figure 14: Representation of the relationships between all programs of the dMAPS database.

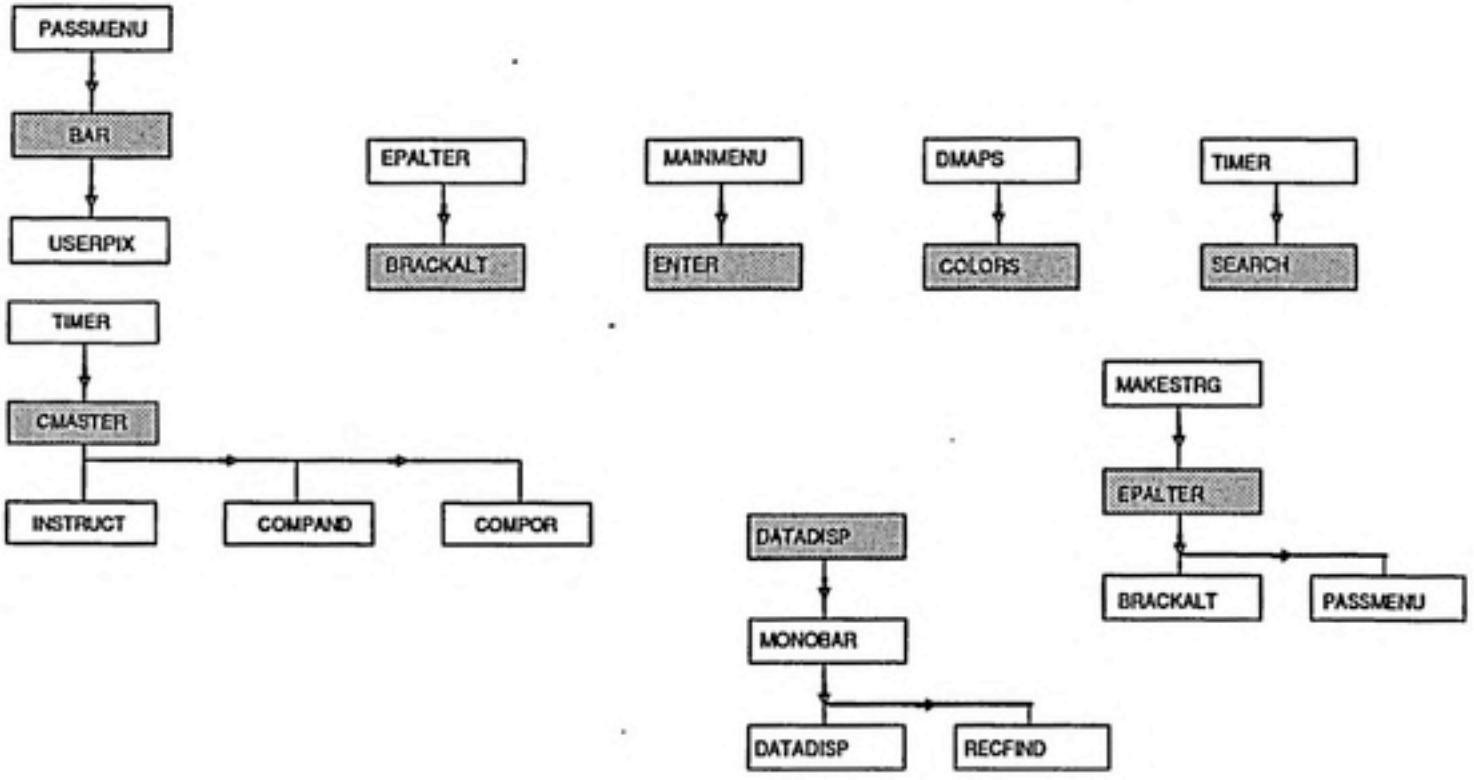


Figure 15: Relationships between selected calling and called programs.

BAR
SEARCH
CMASTER
COLORS
ENTER
TIMER
DATADISP
EPALTER

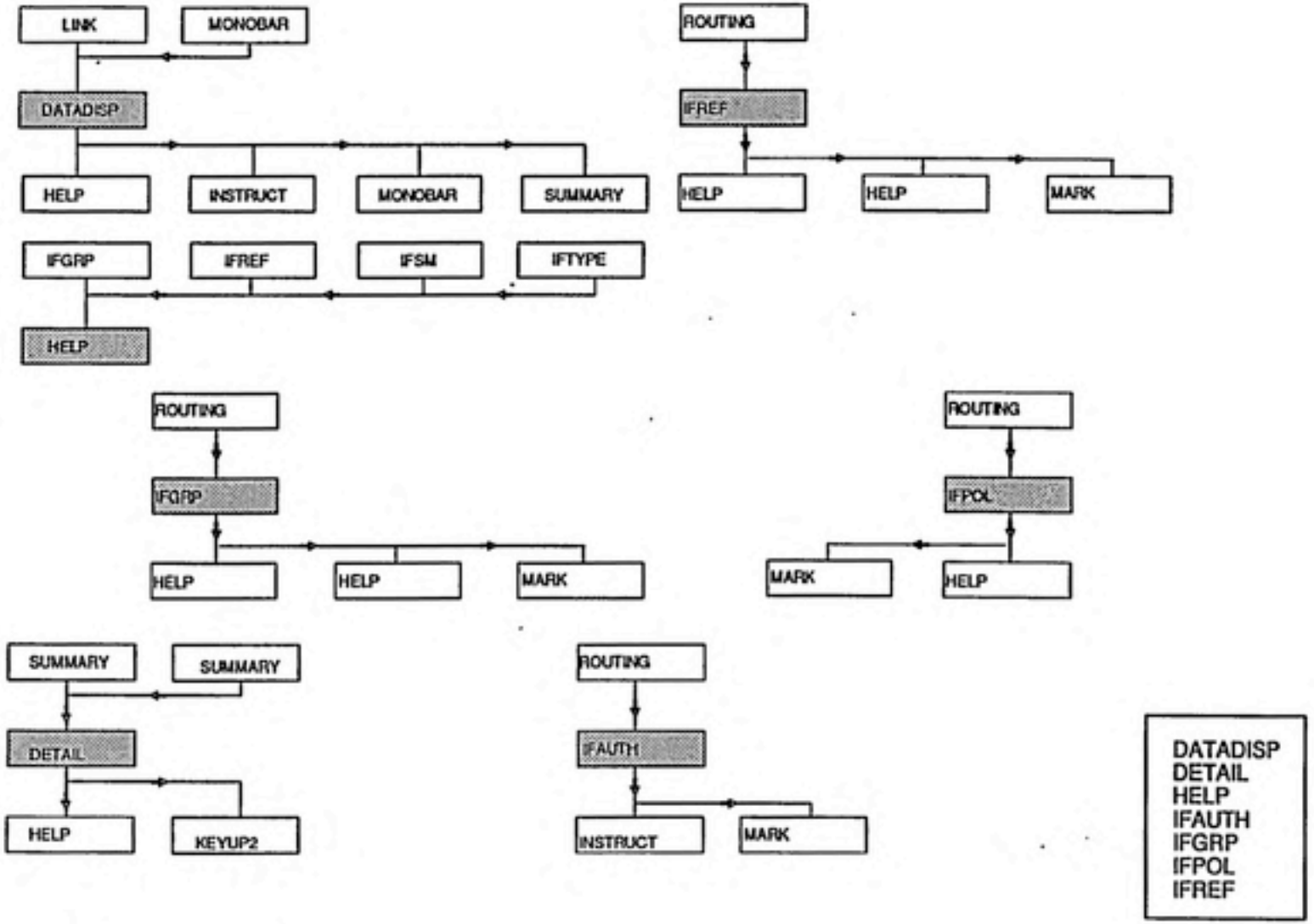
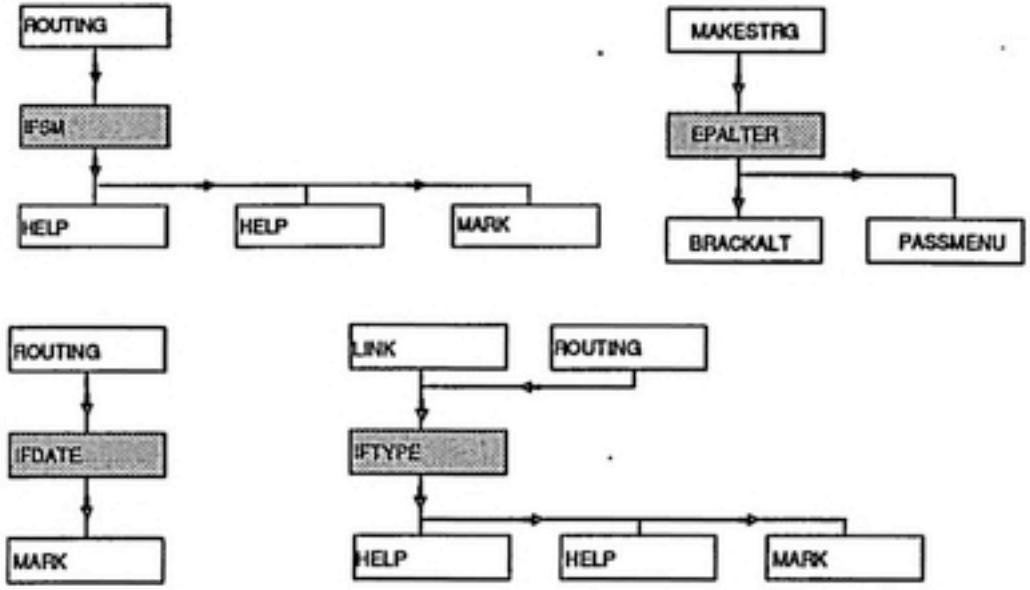


Figure 15: (Continued...)



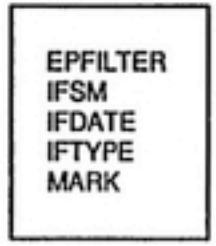


Figure 15: (Continued...)

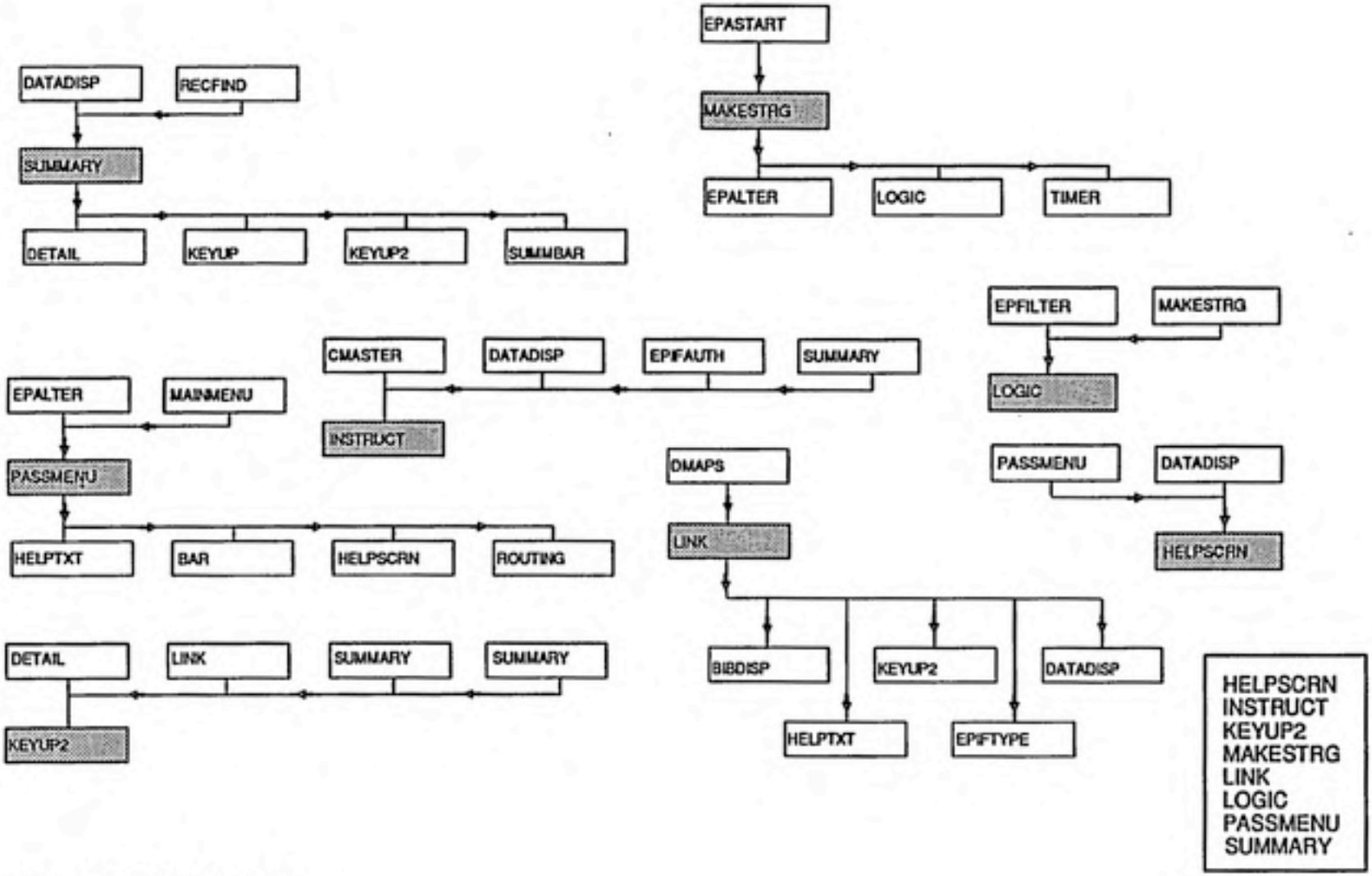


Figure 15: (Continued...)

APPENDIX B
dMAPS PROGRAM CODE

INDEX OF PROGRAMS LISTED IN APPENDIX B

```
clear
set scoreboard off
set procedure to dmaps2
PROCEDURE HELP
PARAMETERS SCREEN
set color to bg/n
@ 12,0 CLEAR to 24,79
@ 12,0 to 24,79 double
DO CASE
    CASE SCREEN=5.or.screen=11.or.screen=10.or.screen=7
        @ 13,1 SAY 'MENU SELECTION SCREENS:'
        @ 14,5 SAY 'The items displayed in screens of this type are contained in the database '
        @ 15,5 say 'in one or more published articles.  When you select a number corresponding'
        @ 16,5 say 'to an entry for which you wish to review the data, the dMAPS program saves'
        @ 17,5 say 'that entry and will later search through all articles for occurances.  You'
        @ 18,5 say 'may make up to three selections which may be from the same menu or from'
        @ 19,5 say 'different menus.'
        @ 22,5 say ''
        WAIT
    CASE SCREEN=2
        @ 10,1 SAY 'HELP SCREEN 2, CONCERNING BRACKET SELECTIONS'
        WAIT
    CASE SCREEN=3
        @ 13,1 SAY 'PRELIMINARY POLLUTANT SELECTION SCREEN:'
        @ 14,5 SAY 'In order to eliminate the need to display all pollutants contained in the '
        @ 15,5 say 'databases a method has been chosen to display only the pollutants that   '
        @ 16,5 say 'begin with any particular letter.                                        '
        @ 18,5 say 'To display all pollutants beginning with any letter, type that letter and '
        @ 19,5 say 'then < Enter>.  To display all pollutants, press <Enter> without making a '
        @ 20,5 say 'letter selection'
        WAIT
        @ 3,50 say ''
    CASE SCREEN=4
        @ 13,1 SAY 'AUTHOR SEARCH PROCEDURES:                                        '
        @ 14,5 SAY 'Searching for authors is done differently than the searches for pollutants'
        @ 15,5 say 'and materials.  Searching for authors is done from a free entry screen in '
        @ 16,5 say 'which you are free to enter any string of characters.            '
        @ 17,5 say 'Some author searching pointers:                                  '
        @ 18,5 say "The program searches the author's names for any string you input.  "
        @ 19,5 say 'Capitalization and punctuation must be exact. Therefore, typing man will '
        @ 20,5 say 'find Auman and Amana, typing Man will find Manson and Manteo, but not    '
        @ 21,5 say 'Auman or Amana.                                                  '
        @ 22,5 say 'All surnames are followed by commas, so to find Smith and not Smithers,  '
        @ 23,5 say 'enter the string Smith,.'
        set color to bg+/n
        @ 19,69 say 'man'
        @ 20,34 say 'Man'
        @ 23,22 say 'Smith,'
        set color to w*/n
        @ 20,48 say 'Man'
        @ 20,59 say 'Man'
        @ 20,12 say 'man'
        @ 20,21 say 'man'
        @ 23,69 say ''
        set color to bg/n
        WAIT
    CASE SCREEN=5
        @ 10,1 SAY 'HELP SCREEN 5, FOR POLLUTANT ENTRY HELP'
        WAIT
    CASE SCREEN=12
        @ 10,1 SAY 'HELP SCREEN 6, UNASSIGNED '
        WAIT
    CASE SCREEN=7
        @ 10,1 SAY 'HELP SCREEN 7, FOR IFGRP.DBF, CONVENTIONS FOR GROUP SEARCHES'
        WAIT
    CASE SCREEN=8
        @ 10,1 SAY 'HELP SCREEN 8, UNASSIGNED'
        WAIT
    CASE SCREEN=9
        @ 10,10 SAY 'HELP SCREEN 9, FOR REFERENCE JOURNAL SEARCH HELP'
        WAIT
    CASE SCREEN=10
      .@ 10,1 SAY 'HELP SCREEN 10 FOR SOURCE MATERIAL HELP'
        WAIT
    CASE SCREEN=11
        @ 10,1 SAY 'HELP SCREEN 11, FOR CATEGORY SEARCH HELP'
        WAIT
    CASE SCREEN=6
        @ 13,1 SAY 'SEARCH PARAMETERS SELECTION SCREEN'
        @ 14,5 SAY 'Selections are made here concerning which areas of the databases are to be'
        @ 15,5 say 'searched.  The menu at screen left contains all searchable areas.  After '
        @ 16,5 say 'one of these areas has been selected by typing its first letter, the dMAPS'
        @ 17,5 say 'program will prompt you for specific information on each.  You may make up'
        @ 18,5 say '3 simultaneous searches on any of the entries. As you make selections the '
        @ 19,5 say 'program displays them in the bottom right.                       '
        @ 20,5 say '                                                                 '
        @ 21,5 say 'Corrections to incorrectly entered search criteria may be made from the  '
        @ 22,5 say 'next menu screen, accessed by pressing "F10".                    '
        @ 23,5 say '                                                                 '
        WAIT
ENDCASE
RETURN
```

```
procedure readhelp
parameters helpscm,variable,x,y
**helpscrn=number of the helpscreen required
**variable=variable of read "get"
**x=row number of get
**y=column number of get
***help loop
store .f. to entered
*store row() to r
*store col() to c
set color to &rdwh
@ 24,0 say ' (Help available: <Ctrl-H>) '
set color to &blwh
DO WHILE .not. entered
    @ x,y get variable
    read
    IF READKEY()=0
        call savescr with 's1'
        DO HELP WITH helpscm
        call savescr with 'r1'
        LOOP
    Else
        entered=.t.

        return
    endif
enddo

***end help loop

*****************************************************************
*                BAR                *
*****************************************************************
PROCEDURE BAR
*// BAR:  displays the user selection bar.  No user inputs
*
public last
SET COLOR TO &blwh
CLEAR
SET COLOR TO &blwh
@ 2,17 clear to 2,78
@ 2,18 SAY 'SEARCH PARAMETERS SELECTION MENU'
if pass=1
    @ 3,16 to 17,78
    @ 3,1 to 17,15
    @ 3,7 SAY 'MENU'
    @ 11,1 say chr(195)
    @ 11,2 to 11,14
    @ 11,15 say chr(180)
    set color to &bwhbl
    @ 4,2 say '  Group     '
    @ 6,2 say '  Category  '
    @ 8,2 say '  Material  '
    @ 10,2 say '  Pollutant  '
    @ 12,2 say '  Author    '
    @ 14,2 say '  Journal   '
    @ 16,2 say '  Date pub.  '
    set color to &blwh
    @ 19,0 clear to 23,79
    @ 19,0 to 23,79 double
    @ 19,50 say chr(203)
    @ 20,50 to 22,50 double
    @ 23,50 say chr(202)
    @ 19,16 say '<              >'
    @ 19,18 say "Program Options"
    set color to &rdwh
    @ 20,10 say 'R'
    @ 20,2 say 'H'
    set color to &blwh
    @ 20,4 say 'elp'
    @ 20,12 say 'eturn to main menu'
    @ 21,2 say 'Help usually available by pressing <Ctrl-H>'
    set color to &bwhbl
    @ 4,4 say "G"
    @ 6,4 say "C"
    @ 8,4 say "M"
    @ 10,4 say "P"
    @ 12,4 say "A"
    @ 14,4 say "J"
    @ 16,4 say "D"
    set color to &blwh
    call savescr with "S5"
endif
call savescr with "R5"
    do USERPIX
call savescr with "S5"
if reentry
    set color to &blwh
endif
RETURN
*End of bar
```

```
*************************************************************
*                    BIBDISP                    *
*************************************************************
PROCEDURE BIBDISP
    set color to &blwh
if point<1
    close all
    return to master
endif
use biblio
goto point
set color to &blwh
store 73 to x
    store trim(biblio->title) to line
    @ 1,0 clear to 6,79
    @ 1,2 say substr(biblio->authors,1,70)
    STORE SUBSTR(TITLE,1,79) TO TITLEVAR
set color to &bkwh
    if len(line)<x
    @ 2,5 say line
    else
    do while substr(line,x,1) <> ' '.and.x>1
        x=x-1
    enddo
    @ 2,5 say substr(line,1,x)
    store substr(line,x+1) to line
    store 73 to x
    if len(line)<x
    @ 3,5 say line
    else
    do while substr(line,x,1) <> ' '.and.x>1
        x=x-1
    enddo
    @ 3,5 say substr(line,1,x)
    store substr(line,x+1) to line
    @ 4,5 say line
    endif
    endif
set color to &blwh
if col()<73
    @ row(),col()+1 say trim(biblio->date)
else
    @ row()+1,5 say trim(biblio->date)
endif
    store 77-col() to x
    store trim(biblio->refer) to line
    if len(line)<x
    if col()+1<5
    @ row(),4 say ''
    endif
    @ row(),col()+1 say line
    else
    do while substr(line,x,1) <> ' '.and. x>1
        x=x-1
    enddo
    if x>1
    @ row(),col()+1 say substr(line,1,x)
    store substr(line,x+1) to line
    endif
    store 73 to x
    if len(line)<x
    @ row()+1,5 say line
    else
    do while substr(line,x,1) <> ' '.and. x>1
        x=x-1
    enddo
    if x>1
    @ row()+1,5 say substr(line,1,x)
    store substr(line,x+1) to line
    endif
    @ row()+1,5 say line
    endif
    endif
@ row()+1,0 to row()+1,79 double
store row()+1 to last
call savescr with 's1'
set color to &blwh
call savescr with 's2'
return
*End ofBibdisp

*************************************************************
*                    CMASTER                    *
*************************************************************
PROCEDURE CMASTER
*//CMASTER
* COMPARES INDIVIDUAL BIBLIO STRINGS TO FORM A MASTER
* STRING USED BY DISPLAY PROGRAMS
* COMPARE NORMALLY (FROM RIGHT TO LEFT)
DO WHILE .T.
    DO CASE
** IF NO BRACKETS
```

```
    CASE BRACKET='N'.OR.BRACKET='R'.or.bracket=' '
    if timer2 <> ' '
*** IF TIMER2 BLANK
    else
        store timer1 to s1
        exit
    endif
** IF T2 <> ' '
    DO CASE

************************************************
** IF T2 <> ' '(.AND. B2 = AND)
        case BOOL2='AND'
*** IF T2 <> ' '(.AND B2 = AND .AND. T3 <> ' ')
            if timer3 <> ' '
                store timer3 to s1
                store timer2 to s2
                DO COMPAND
                TIMER3=' '
                BOOL2=' '
            endif
*** IF T2 <> ' '(.AND . B1 = AND)
            if BOOL1='AND'
                store timer1 to s2
                DO COMPAND
            ENDIF
*** IF T2 <> ' '(.AND. B1 = OR)
            if bool1='OR '
                store timer1 to s2
*           store timer2 to s1
                DO COMPor
            endif
** IF T2 <> ' '
                TIMER2=' '
                BOOL1=' '
        case bool2=' '
** IF B2= ' '
            store timer2 to s1
            store timer1 to s2
            DO CASE
** IF B2=' ' .AND. B1=AND
                case BOOL1='AND'
                    DO COMPand
** IF B2=' ' .AND. B1=OR
                case BOOL1='OR '
                    DO COMPor
** IF B2=' '.AND. B1=' '
                case BOOL1=' '
                    exit
            ENDCASE
    case bool2='OR '
** IF B2=OR .AND. T3 <> ' '
            if timer3 <> ' '
                store timer3 to s1
                store timer2 to s2
                DO COMPor
            ENDIF
** IF B2=OR .AND. T3=' '
            IF TIMER3=' '
                store timer2 to s1
            endif
                DO CASE
**IF B2=OR .AND. B1=AND
                    case bool1='AND'
                        store timer1 to s2
                        DO COMPand
** IF B2=OR .AND. B1=OR
                    case bool1='OR '
                        store timer1 to s2
                        DO COMPor
                ENDCASE
    ENDCASE
    * COMPARE LEFT TO RIGHT
    CASE BRACKET='L'
    DO CASE
        case BOOL1='AND'
            store timer2 to s1
            store timer1 to s2
            DO COMPand
            if timer3 <> ' '
                if BOOL2='AND'
                    store timer3 to s2
                    DO COMPand
                else
                    store timer3 to s2
                    DO COMPor
                endif
            endif
        case bool2=' '
            store timer2 to s1
            store timer1 to s2
            DO CASE
```

```
                case BOOL1='AND'
                    DO COMPand
                case BOOL1='OR '
                    DO COMPor
                case BOOL1=' '
                    exit
            ENDCASE
        case bool1='OR '
            store timer2 to s1
            store timer1 to s2
            DO COMPor
            if timer3 <> ' '
                DO CASE
                    case bool2='AND'
                        store timer3 to s2
                        DO COMPand
                    case bool2='OR '
                        store timer3 to s2
                        DO COMPor
                ENDCASE
            endif
    ENDCASE
ENDCASE
exit
enddo
store s1 to xstr
set color to &blwh
do instruct
store Ltrim(str(len(s1)/4)) to count
if count <> '1'
    if count='0'
    @ 20,1 say 'No articles have data matching search criteria.'
    else
    @ 20,1 say 'Selection criteria matched in '+count+' articles.'
*@ 20,1 SAY 'Data from '+count+' articles match your search criteria.'
    endif
else
    @ 20,1 say 'Selection criteria matched in one article.'
endif
@ 22,51 SAY SEC
@ ROW(),COL() SAY ' Seconds searching time.'
i=1
do while i<100
    i=i+1
enddo
    set color to &flashblwh
    @ 24,0 clear to 24,79
    @ 22,2 say chr(16)
    set color to &blwh
if val(count)=0
@ 22,3 say "Press any key to return to main menu..."
call cursoff
@ 23,79 say ''
i=0
do while i=0
    call bell with "2,2500"
    ct=1
    do while i=0.and.ct<10000
        i=inkey()
        ct=ct+1
    enddo
if i=35
    i=0
endif
enddo
else
endif
clear
call curson
if len(s1)<=1
    return to master
else
    return
endif
*End ofcmaster

*****************************************************************
*                    COLORS                    *
*****************************************************************
PROCEDURE COLORS
public flashblwh,ylwh,bwhrd,bwhbk,ylcy,bwhbl,;
ylbl,whbl,rdwh,bkwh,blwh,whwh,blank
store .t. to colormon
if colormon
*BLUE BACKGROUND
store 'b/w' to blwh
store 'n/w' to bkwh
store 'r/w' to rdwh
store 'w/b' to whbl
store 'gr+/b' to ylbl
store 'w+/b' to bwhbl
store 'gr+/bg' to ylcy
```

```
store 'w+/n' to bwhbk
store 'w+/r' to bwhrd
store 'gr+/w' to ylwh
store 'b*/w' to flashblwh
store 'w/w' to whwh
else
*BLACK/white
store 'w/n' to blwh
store 'w+/n' to bkwh
store 'n+*/w' to rdwh
store 'w/n' to whbl
store 'w+/n' to ylbl
store 'w+/n+' to bwhbl
store 'w+/n' to ylcy
store 'w+/n' to bwhbk
store 'w+/n' to bwhrd
store 'w+/n' to ylwh
store 'w+/n' to flashblwh
store 'w/w' to whwh
endif
*End ofcolors

*************************************************************
*                  COMPAND                  *
*************************************************************
PROCEDURE COMPAND
*// COMPAND
    ctr=1
    IF S1=TIMER1
    STORE '2ND' TO BOOL2
    ENDIF
    IF BOOL2='2ND'
    STORE 'FINISH' TO BOOL2
    ENDIF

* COMPARE STRING S1 TO STRING S2
*EXTRACT 4 CHARACTER RECORD NUMBERS FROM STRING OF FOUND
*BIBLIO #'S
    do while len(s1)>ctr
        store substr(s1,ctr,4) to sub1
*IF EXTRACTED RECORD # IS IN SECOND STRING ALSO
        if sub1 $ s2
*SKIP TO NEXT FOUR-CHARACTER RECORD NUMBER
        ctr=ctr+4
        loop
*delete from s1 if not also in s2
                else
            store stuff(s1,ctr,4,'') to s1
        endif
    enddo
RETURN
*End ofcompand

*************************************************************
*                  COMPOR                   *
*************************************************************
PROCEDURE COMPOR
*// COMPOR
CTR=1
DO WHILE LEN(S1)>CTR
    STORE SUBSTR(S1,CTR,4) TO SUB1
    IF SUB1 $ S2
        CTR=CTR+4
        LOOP
    ELSE
        STORE STUFF(S2,1,0,SUB1) TO S2
        CTR=CTR+4
    ENDIF
ENDDO
S1=S2
S2=''
*End ofcompor

*************************************************************
*                  DATADISP                 *
*************************************************************
PROCEDURE DATADISP
*//DATADISP -> DISPLAYS POLL/COND RELATIONS
* CALLED FROM LINK.PRG
set typeahead to 0
IF FIRSTIME
store .t. to firsttime
store recno() to bookmark
set index to POL-BIB
seek number
store recno() to start
set color to &blwh
@ 0,0 to last-2,79 double
@ last-1,0 clear to 24,79
@ last-1,0 to 18,79 double
store last to memlast
```

```
do instruct
* DISPLAY USER OPTIONS
set color to &rdwh
@ 20,1 say CHR(24)+' '+CHR(25)
@ 22,2 SAY 'R'
@ 20,23 say "F10"
set color to &blwh
@ 20,27 say "Select 'Cursored' Item"
@ 20,5 SAY 'Move Cursor bar'
@ 21,0 say chr(204)
@ 21,50 say chr(185)
@ 21,1 to 21,49 double
@ 22,5 say "Select another Reference for display"
@ 24,0
@ 24,0 say '"'+chr(16)+'" Indicates group contains undisplayed item that meets search criteria.'
ENDIF
STORE .F. TO FIRSTIME
last=memlast
store biblio_no to x
@ last,10 say ''
row1=last
page=1
pagect=0
goto start
store start to beginning,lastrec
do while .t.
store start to beginning
store recno()-pagect to start
if write
    do while biblio_no=x.and.(.not.eof())
    set color to &blwh
    @ row1,10 say pollutant
    store recno() to firstrec
    y=pollutant
    store 1 to sampCT
*make string to compare for hidden match
            store ltrim(str(source_no))-',' to snumber
*COUNT/DISPLAY CONDITIONS FOR THIS POLLUTANT
    DO WHILE POLLUTANT=Y
        if snumber$bigstr &&.or.snumber$bigstr2.or.snumber$bigstr3
            set color to &rdwh
            @ row(),8 say chr(16)
            set color to &blwh
        endif
        SKIP
        store ltrim(str(source_no))-',' to snumber
            @ row(),39 SAY SAMPCT
            SAMPCT=SAMPCT+1
        ENDDO
        pagect=pagect+(SAMPCT-1)
        if sampct=2
            @ row(),50 say 'Test Condition'
        else
            @ row(),50 say 'Test Conditions'
        endif
    set color to &blwh
    *page=# of display screen
    *pagect=# of records counted
    if row()=17
        page=page+1
         * store recno()-pagect to start
        set color to &ylwh
        @ 17,70 say chr(25)+' PgDn '+chr(25)
        SET COLOR TO &blwh
        store recno() to lastrec
    *store screen
        CALL SAVESCR WITH 'S2'
    *keyboard control of cursor and program handed to monobar
        do monobar with last,18,pagect,beginning,start
        last=memlast
        row1=last-1
        set color to &blwh
        @ last,1 clear to 17,78
        if more
            more=.f.
            return
        endif
    ENDIF
    if page>1
        set color to &ylwh
        @ last,70 say chr(24)+' PgUp '+chr(24)
        set color to &blwh
    else
        @ last,70 clear to last,78
    endif
    row1=row1+1
    call savescr with "s2"
    ENDDO
    else
    goto lastrec
    endif       &&if not write
    call savescr with "r2"
```

```
store recno() to lastrec
do monobar with last,18,pagect,beginning,start
last=memlast
    if more
        return  &&to link
    endif
DO SUMMARY
last=memlast
set typeahead to 20
select 5
set index to biblio_no
seek bookmark
counter=counter-1
set typeahead to 20
enddo
*End ofdatadisp

*************************************************************
*                  DETAIL                   *
*************************************************************
procedure detail
* "rec" ASSIGNED IN monobar AS FIRST MATCHING RECORD
goto POINTER
store source->material to materl
store poll1->pollutant to pollut
store biblio_no to go
WINDOW=2
SET STATUS OFF
store recno() to recd
* BEGIN DISPLAY OF Authors
set color to &blwh
store 73 to x
    store trim(biblio->title) to line
    @ 0,2 say trim(biblio->authors)
* MAKE VARIABLE FOR DATA DISPLAY IN SUMMARY.PRG
    STORE SUBSTR(biblio->TITLE,1,79) TO TITLEVAR
* FIRST LINE OF TITLE
    if len(line)<x
    @ 1,5 say line
    else
    DO WHILE substr(line,x,1) <> ' '.and.x>1
        x=x-1
    ENDDO
    @ 1,5 say substr(line,1,x)
    store substr(line,x+1) to line
* SECOND LINE OF TITLE
    store 71 to x
    if len(line)<x
    @ 2,5 say line
    else
    DO WHILE substr(line,x,1) <> ' '.and.x>1
        x=x-1
    ENDDO
    @ 2,5 say substr(line,1,x)
    store substr(line,x+1) to line
* THIRD LINE OF TITLE (LAST)
    @ 3,5 say line
    endif
    endif
set color to &blwh
* SAY DATE ON PRESENT LINE IF SPACE OR NEXT LINE IF NOT
if col()<72
    @ row(),col()+1 say trim(biblio->date)
else
    @ row()+1,5 say trim(biblio->date)
endif
* BEGIN DISPLAY OF REFERENCE INFO
    store 77-col() to x
    store trim(biblio->refer) to line
* FIRST LINE OF REFERENCE
    if len(line)<x
    if col()+1<5
    @ row(),4 say ''
    endif
    @ row(),col()+1 say line
    else
    DO WHILE substr(line,x,1) <> ' '.and. x>1
        x=x-1
    ENDDO
    if x>1
    @ row(),col()+1 say substr(line,1,x)
    store substr(line,x+1) to line
    endif
* SECOND LINE OF REFERENCE
    store 71 to x
    if len(line)<x
    @ row()+1,5 say line
    else
    DO WHILE substr(line,x,1) <> ' '.and. x>1
        x=x-1
    ENDDO
    if x>1
```

```
    @ row()+1,5 say substr(line,1,x)
    store substr(line,x+1) to line
    endif
* THIRD LINE Of REFERENCE (LAST)
    @ row()+1,5 say line
    endif
    endif
* DRAW DOUBLE LINE AT BOTTOM OF REFERENCE
@ row()+1,0 to row()+1,79 double
store row() to last
SET COLOR TO &rdwh
@ LAST,0 CLEAR TO LAST,79
store poll1->measuremnt to pt1
store ' '+poll1->units to pt2
@ last,1 say pt1
@ last,col() say pt2
@ last,col()+1 say trim(upper(poll1->pollutant))
if recd<>recno()
    @ row(),0 say 'WARNING: Program error.  Bibliography and data may not match!!'
    @ row()+1,0 say recno()
    @ row(),col()+1 say recd
endif
*background
set color to &blwh
@ last+1,0 to last+1,79 double
@ last+2,0 clear to 23,79 &&<$>
*<$>
    @ 23,0 say recno()
    @ 23,col()+1 say recd
    @ 23,col()+1 say 'pgm note: search for <$> to find '
store .t. to passone
m1=poll1->userdata1
m2=poll1->userdata2
m3=poll1->comment
m4=cond->test_desc
m5=cond->test_dur
m6=source->material
m7=cond->pre_condi
m8=cond->oper_par
m9=cond->vent_rate
m10=cond->temp_deg_c
m11=cond->rel_humid
m12=sample->sampl_type
m13=sample->sampl_rate
m14=sample->sampl_dur
m15=sample->sampl_size
m16=sample->sampl_meth
m17=sample->analysis
m18=poll1->measuremnt
m19=poll1->user1
m20=poll1->user2
store .t. to looop
do while .t.
DO WHILE poll1->biblio_no=go
if all
    if poll1->pollutant <> pollut
    DO KEYUP2
    select 1   &&->poll1
    goto ptc1  &&->initialized in recfind
    return
    endif
else
     do while source->material <> materl
    skip
    if poll1->pollutant <> pollut
        DO KEYUP2
        select 1   &&->poll1
        goto ptc1  &&->initialized in recfind
        return
    endif
     enddo
endif
SET COLOR TO &rdwh
@ 24,0 CLEAR TO 24,79
@ 24,0 SAY chr(17)+chr(196)+chr(217)
set color to &rdwh
@ 24,15 SAY "PgDn"
@ 24,40 say "PgUp"
@ 24,65 say "R"
set color to &blwh
@ 24,4 say "MORE"
@ 24,20 say "New Material"
@ 24,45 say "Prev Material"
@ 24,67 say "Return"
DO CASE
CASE WINDOW=1    &&.or.looop
top=11
if poll1->user1 <> ' '
    top=10
endif
if poll1->user2 <> ' '
    top=9
```

```
endif
@ top,16 clear to 15,69
SET COLOR TO &blwh
@ 6,17 SAY "OTHER DATA:"
SET COLOR TO &blwh
@ 7,16 to 11,48
@ 8,17 say 'User defined data fields:'
SET COLOR TO &BWHIBL
*change color if new data
    if m1 <> poll1->userdata1 .or. m19 <> poll1->user1
        set color to &ylbl
    endif
*say data here
    @ 9,17 say poll1->user1
    @ ROW(),COL()+1 SAY poll1->userdata1
    set color to &BWHIBL
*change color if new data
    if m2 <> poll1->userdata2 .or. m20<>poll1->user2
        set color to &ylbl
    endif
*say color here
    @ 10,17 say poll1->user2
    @ ROW(),COL()+1 SAY poll1->userdata2
    set color to &blwh
@ 15,0 to 18,79
@ 14,1 say 'COMMENTS: '
*set color if new data
if m3 <> poll1->comment
    set color to &YLwh
endif
@ 16,1 say substr(poll1->comment,1,77)+'-'
@ 17,1 say substr(poll1->comment,78,150)
r=top
set color to &blwh
*endif
CASE WINDOW=2&&.or.loocp

*******************************************************
*window 2
set color to &blwh
*box
set color to &blwh
@ 7,11 say "TEST DATA:"
SET COLOR TO &blwh
@ 8,10 to 21,75
set color to &blwh
@ 9,11 clear to 20,74
*fill
set color to &blwh
@ 9,11 say 'Measured Emission: '
if m18 <> poll1->measuremnt
set color to &ylwh
else
set color to &bkwh
endif
store poll1->measuremnt to pt1
store '  '+poll1->units to pt2
@ ROW(),COL() say pt1
@ row(),col()+1 say pt2
set color to &blwh
@ 10,11 say 'Description: '
if m4 <> cond->test_desc
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->test_desc
set color to &blwh
@ ROW()+1,11 SAY 'Duration: '
if m5 <> cond->test_dur
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->test_dur
set color to &blwh
@ row()+1,11 to row()+1,74
@ row()+1,11 say 'Material:'
if m6 <> source->material
set color to &ylwh
else
set color to &bkwh
endif
@ row(),col() say substr(source->material,1,55)
set color to &blwh
@ ROW()+1,11 SAY 'Matl. Pre-conditioning: '
if m7 <> cond->pre_condi
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->pre_condi
```

```
set color to &blwh
@ ROW()+1,11 SAY 'Operating Parameters: '
if m8 <> cond->oper_par
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->oper_par
set color to &blwh
@ ROW()+1,11 SAY 'Room/Chamber Volume: '
@ ROW()+1,11 SAY 'Ventilation Rate: '
if m9 <> cond->vent_rate
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->vent_rate
set color to &blwh
*@ ROW(),COL() say cond->chamb_vol
set color to &blwh
@ row()+1,11 to row()+1,74
@ ROW()+1,11 SAY 'TEMP (C'+chr(248)+'): '
if m10 <> cond->temp_deg_c
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say TRIM(COND->temp_deg_c)+' '
set color to &blwh
@ ROW()+1,11 say 'Humidity (% RH): '
if m11 <> cond->rel_humid
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say cond->rel_humid
set color to &blwh
*@ row()+2,1 say recno()
store recno() to recd
CASE WINDOW=3&&.or.looop

**********************************
*WINDOW 3
*background
set color to &blwh
@ 8,5 say "SAMPLING DATA: "
set color to &blwh
set color to &blwh
*box
@ 9,4 to 22,77
set color to &blwh
@ 10,5 clear to 21,76
*fill
set color to &blwh
@ 10,5 SAY 'INSTRUMENT: '
@ row()+1,5 to row()+1,76
@ row()+1,5 say 'SAMPLE- '
@ row()+1,5 say '  Type   : '
set color to &blwh
if m12 <> sample->sampl_type
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say sample->sampl_type
set color to &blwh
@ row()+1,5 SAY '  Rate   : '
set color to &blwh
if m13 <> sample->sampl_rate
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say sample->sampl_rate
set color to &blwh
@ ROW()+1,5 SAY '  Duration: '
set color to &blwh
if m14 <> sample->sampl_dur
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say sample->sampl_dur
set color to &blwh
@ ROW()+1,5 SAY '  Size   : '
set color to &blwh
if m15 <> sample->sampl_size
set color to &ylwh
else
set color to &bkwh
endif
@ ROW(),COL() say sample->sampl_size
```

```
set color to &blwh
@ row()+1,5 to row()+1,76 *
@ ROW()+1,5 SAY 'Sample Method: '
set color to &blwh
if m16 <> sample->sampl_meth
set color to &ylwh
else
set color to &bkwh
endif
@ ROW()+1,5 say sample->sampl_meth
set color to &blwh
@ ROW()+1,5 SAY 'Analytic Method: '
set color to &blwh
if m17 <> sample->analysis
set color to &ylwh
else
set color to &bkwh
endif
@ ROW()+1,5 SAY  sample->analysis
endcase
looop=.f.
ct=0
SET COLOR TO &bkwh
@ last,0 clear to last,79
if m18 <> poll1->measuremnt
     set color to &ylwh
else
     set color to &bkwh
endif
store poll1->measuremnt to pt1
store ' '+poll1->units to pt2
@ last,1 say pt1
@ last,col()+1 say pt2
set color to &bkwh
@ last,col()+1 say trim(upper(poll1->pollutant))
*background
set color to &blwh
point=.t.
do while point
     point=.f.
     i=0
do while i=0
     do while i=0
          i=inkey()
     enddo
     if i=35
          i=0
          do help with 1
          endif
     enddo
     do case
     case i=113.or.i=81      &&Qq
               close all
               return to master
     case i=114.or.i=82      &&Rr
          set color to &blwh
          clear
          frow=5
          lrow=19
          DO KEYUP2
          select 1   &&->poll1
          goto ptc1   &&->initialized in recfind
          return
     case i=3.or.i=18
     m1=poll1->userdata1
     m2=poll1->userdata2
     m3=poll1->comment
     m4=cond->test_desc
     m5=cond->test_dur
     m6=source->material
     m7=cond->pre_condi
     m8=cond->oper_par
     m9=cond->vent_rate
     m10=cond->temp_deg_c
     m11=cond->rel_humid
     m12=sample->sampl_type
     m13=sample->sampl_rate
     m14=sample->sampl_dur
     m15=sample->sampl_size
     m16=sample->sampl_meth
     m17=sample->analysis
     m18=poll1->measuremnt
     looop=.t.
          do case
          case i=3
               skip
               if poll1->pollutant <> pollut
                    DO KEYUP2
                    select 1   &&->poll1
                    goto ptc1   &&->initialized in recfind
                    return
               endif
```

```
        exit
    case i=18
    skip-1
    if all
        if poll1->pollutant <> pollut
            DO KEYUP2
            select 1   &&->poll1
            goto ptc1   &&->initialized in recfind
            return
        endif
    else
            do while source->material <> materl
                skip-1
                if poll1->pollutant <> pollut
                    DO KEYUP2
                    select 1   &&->poll1
                    goto ptc1   &&->initialized in recfind
                    return
                endif
            enddo
    endif
    EXIT
    endcase
otherwise
    set color to &blwh
    @ last+2,0 clear to 24,79
    window=window+1
    if window=4
        window=1
    endif
endcase
ENDDO
ENDDO
enddo
*End ofDetail

*************************************************************
*                  DISPMENU                 *
*************************************************************
PROCEDURE DISPMENU
* dispmenu
* allows user discretion in display of biblio inform.
@ 9,0 clear to 24,79
@ 8,9 say "This Reference:"
@ 9,8 to 19,33
@ 8,44 say "Other References:"
@ 9,43 to 19,68
set color to &whbl
@ 10,10 say ' 1 '
@ 12,10 say ' 2 '
@ 14,10 say ' 3 '
@ 16,10 say ' 4 '
@ 18,10 say ' 5 '
@ 10,45 say chr(24)
@ 12,45 say chr(25)
@ 14,45 say 'Alt '+chr(24)
@ 16,45 say 'Alt '+chr(25)
@ 18,45 say 'Alt L'
set color to &blwh
@ 10,14 say 'Source Description'
@ 12,14 say 'Test Conditions'
@ 14,14 say 'Test Method'
@ 16,14 say 'Pollutant Results'
@ 18,14 say 'Exit to Main Menu'
@ 10,48 say 'Previous Reference'
@ 12,48 say 'Next Reference'
@ 14,52 say 'First Reference'
@ 16,52 say 'Last Reference'
@ 18,52 say 'Logic'
return
*End ofdispmenu

*************************************************************
*                  ENTER                 *
*************************************************************
PROCEDURE ENTER
SET COLOR TO &ylcy.,&bkwh
set talk off
* ENTER.PRG    -> ALLOWS ENTRY OF IASE DATA
* DON'T CHANGE THESE:
USE ALTERNAT.DBF
SET FORMAT TO entmp.doc
set delimiters on
set delimiters to chr(177)
* TO MAKE BELL RING, "SET BELL ON"
SET BELL OFF
* DON'T CHANGE THESE
do while .t.
CLEAR
set color to &bwhbl
@ 10,0 clear to 10,79
@ 10,10 say "Do you want information carried forward record to record?"
```

```
set color to &ylcy
@ 12,35 say "(Y/N)"
@ 23,0 say ''
WAIT '' TO B
IF B $ 'yY'

    clear

    @ 10,17 say "Start with a new blank record?  (Y/N)"
    set color to &bwhbl
    @ 12,5 say '( "N" here will start entries with data from last entered screen )'
    @ 23,0 say ''
    wait '' to d
    if d$'Yy'
    set color to &bwhbl
    @ 10,0 clear to 10,79
    @ 10,0 say "The next entry screen will be blank.  When you have entered data"
    @ 11,0 say "for this initial screen, press <Ctrl><End> and the information contained in it"
    @ 12,0 say "will be carried forward.  Then, reenter only the information that is different."
    @ 14,0 say "When you wish a new blank screen, press <Ctrl><End> again."
    @ 23,0 say ''
    wait
else
    set color to &blwh5
    @ 12,0 clear to 12,79
    @ 12,10 say "The next record is a new record with data copied from last entry."
    set color to &bwhbl
    @ 23,0 say ''
    wait
endif
set color to w+/b,&bkwh
if d$'Nn'
    set carry on
else
    set carry off
endif
APPEND
if B$"Yy"
    set carry on
else
    set carry off
endif
clear
@ 10,17 say "Continue adding more records? (Y/N)"
@ 23,0 say ''
wait '' to c
if c$'Yy'
    @ 12,1 say "The following record is a new record, but with old information copied to it."
    @ 13,1 say "You may enter new data in any field."
    set color to &blwh5
    @ 15,0 clear to 15,79
    @ 15,10 say "YOU MUST FOLLOW THE DATA ENTRY CONVENTIONS IN THE USER
MANUAL!!"
    set color to &ylcy
    @ 23,2 SAY "(More complete instructions will be in final version.)"
    @ 23,1 SAY ''
    WAIT
    append
    EXIT
else
     set format to
     return
endif
endif
enddo
*End ofenter

*****************************************************************
*                    EPALTER                    *
*****************************************************************
PROCEDURE EPALTER
*epalter.prg
set color to &blwh
@ 21,3 clear to 21,48
store ' ' to ch
store chr(169)+'    '+chr(170) to bracktop2
store chr(192)+'    '+chr(217) to brackbot2
set color to &blwh
@ 9,40 clear to 17,73
@ 9,40 to 17,73
@ 9,41 say ' Change Menu '
set color to &blwh
do case
case h3 <> ' '
@ 10,42 say 'A - Change  A  value '
@ 12,42 say 'B - Change  B  value '
@ 14,42 say 'C - Change  C  value '
@ 16,42 say 'D - Reset Defaults '
set color to &blwh
@ 10,53 say ' A '
@ 12,53 say ' B '
@ 14,53 say ' C '
```

```
case h2 <> ' '
@ 10,42 say 'A - Change  A  value '
@ 12,42 say 'B - Change  B  value '
@ 14,42 say 'C - Add a   C  value '
@ 16,42 say 'D - Reset Defaults '
set color to &blwh
@ 10,53 say ' A '
@ 12,53 say ' B '
@ 14,53 say ' C '
otherwise
@ 10,42 say 'A - Change  A  value '
@ 12,42 say 'B - Add a   B  value '
@ 16,42 say 'D - Reset Defaults '
set color to &blwh
@ 10,53 say ' A '
@ 12,53 say ' B '
endcase
set color to &blwh.,&blwh
set bell off
@ 21,4 say 'Type Letter:  '
DO READHELP WITH 1,CH,21,17
set bell on
store upper(ch) to ch
        @ 3,38 CLEAR TO 13,73
            @ 3,38 to 13,73 double
IF CH <> 'N'.or.ch <> 'C'.or.ch <> 'A'.or.ch <> 'B'
    LOGIC='user'
else
    logic='default'
if ch='D'
    return
endif
ENDIF
DO CASE
    case ch='N'
        redo = 'Y'
        pass=4
    case ch='A'
        pass=1
        store .t. to reentry
*store ch to modif
        do passmenu
    case ch='B'
        store .t. to reentry
*store ch to modif
        pass=2
        do passmenu
    case ch='C'
        store .t. to reentry
*store ch to modif
        pass=3
        do passmenu
    case ch='P'
***Display bracket options
            logic='user'
            @ 13,37 SAY 'BRACKET CHANGE MENU'
            SET COLOR TO  &blwh
            @ 5,45 SAY ' L '
            @ 8,45 SAY ' R '
            @ 11,45 SAY ' N '

            SET COLOR TO &blwh
        @ 5,50 say ' A '
        @ 5,55 SAY ' B '
        @ 5,60 SAY ' C '
        @ 8,50 say ' A '
        @ 8,55 SAY ' B '
        @ 8,60 SAY ' C '
        @ 11,50 say ' A '
        @ 11,55 SAY ' B '
        @ 11,60 SAY ' C '
set color to &blwh
            @ 5,50 say '['
            @ 5,57 say ']'
            @ 8,55 say '['
            @ 8,62 say ']'
    set color to &flashblwh
    @ 24,1 say chr(16)
    set color to  &blwh
    @ 24,3 say 'TYPE LETTER CORRESPONDING TO BRACKET TYPE DESIRED'
    @ 21,2 clear to 21,49
    store ' ' to bracket
DO READHELP WITH 2,BRACKET,24,70

    do case
        case bracket$'Rr'
            bracket='RR'
        case bracket$'Ll'
            bracket='LL'
        case bracket$'Nn'
            bracket='NN'
    endcase
```

```
clear
DO BRACKALT
return
    case ch='R'
        @ 3,38 SAY 'Change Right Boolean Operator to:'
    set color to &blwh
        @ 5,45 say 'A'
        @ 7,45 say 'O'
        @ 9,45 say 'N'
set color to &blwh
    @ 5,46 say 'ND'
    @ 7,46 SAY 'R'
    @ 9,46 SAY 'OT)'
    @ 12,43 say 'Type Highlighted Letter...'
        wait '' to bool2
    BOOL2=UPPER(BOOL2)
        do case
        case bool2='A'
            bool2='AND'
        case bool2='O'
            bool2='OR '
        case bool2='N'
            bool2='NOT'
        endcase
    @ 3,38 CLEAR TO 23,73
        return
case ch='L'
        @ 3,38 SAY 'Change Left Boolean Operator to:'
    set color to &blwh
        @ 5,45 say 'A'
        @ 7,45 say 'O'
        @ 9,45 say 'N'
set color to &blwh
    @ 5,46 say 'ND'
    @ 7,46 SAY 'R'
    @ 9,46 SAY 'OT)'
    @ 12,43 say 'Type Highlighted Letter...'
        wait '' to bool1
    BOOL1=UPPER(BOOL1)
        do case
        case bool1='A'
            bool1='AND'
        case bool1='O'
            bool1='OR '
        case bool1='N'
            bool1='NOT'
        endcase
        @ 3,38 CLEAR TO 23,73
        return
ENDCASE
return
*End ofepalter

*****************************************************************
*                    EPFILTER                    *
*****************************************************************
PROCEDURE EPFILTER
*// epfilter:  calls 'LOGIC.prg' to set up default
*          relationships between chosen variables.
*          Allows user to make changes in default
*          settings, or initiates search upon user
*          approval.
*
* called from routing.prg
DO WHILE .T.
    public noshow
    store .f. to noshow
    do LOGIC
    store .t. to noshow
    do LOGIC2
enddo
*End ofepfilter

*****************************************************************
*                    EPFLET                    *
*****************************************************************
PROCEDURE EPFLET
* EPFLET (EP FIRST LETTER)
* CALLED FROM ROUTING.PRG
* ALLOWS USER TO REFINE THE POLLUTANT MENU DISPLAYED BY
* SPECIFYING THE FIRST FEW LETTERS OF THE POLLUTANT.
restore additive from mem1.mem
set typeahead to 20
set format to
do while .t.
store ' ' to firstlet
@ 4,18 clear to 15,77
set color to &bkwh
@ 0,0 clear to 17,15
@ 0,0 clear to 2,79
@ 3,16 to 17,78
set color to &bkwh
```

```
@ 2,16 clear to 2,78
@ 2,18 say " Pollutant Selection: "
set color to &bkwh
@ 18,0 clear to 24,79
@ 4,18 say "Enter first letter of pollutant of interest:"
@ 6, 18 say "(or press "+ret+' for complete pollutant list)'
call bell with "1,6000"
set bell off
call cursoff
DO WHILE .t.
    set color to &bwhrd
    @ 4,63 get firstlet picture "!"
    read
    set color to &blwh
IF READKEY()=0
    call savescr with 's1'
    DO HELP WITH 3
    call savescr with 'r1'
    LOOP
ENDIF
exit
enddo
***end help loop
set typeahead to 0
set bell on
call curson
* xstring contains the first letters of all pollutants
if substr(firstlet,1,1) $ xstring
    return
else
call bell with "1,5000"
call bell with "1,3000"
set color to &bkwh
    @ 4,18 clear to 16,77
    @ 8,18 say [No Pollutants with the first letter(s) "]+firstlet-["]
    @ 9,18 say [press any key to continue...]
    wait ''
    loop
endif
enddo
*End ofepflet

*************************************************************
*                IFAUTH                *
*************************************************************
PROCEDURE IFAUTH
```

```
*ifauth:  allows user to input a string of characters
*         to be used in a search of the authors names
*         field.  String may be in any location in the
*         field
*
* called from ROUTING.PRG
set color to &bkwh.,&bkwh
@ 20,1 clear to 22,49
@ 20,51 clear to 22,78
set color to &bkwh
@ 1,0 CLEAR TO 1,79
@ 2,25 say 'AUTHOR SELECTION SCREEN                '
@ 3,1 clear to 18,15
@ 5,18 clear to 15,77
@ 5,18 say [Enter a string of characters representing any author's name.]
@ 6,18 say 'Capitalize only the first letter of a name.'
store .t. to authlook,author
do while .t.
set color to &bkwh
@ 14,18 TO 16,77
@ 15,21 SAY "Enter author's name:"
store '                ' to authname
DO READHELP WITH 4,AUTHNAME,15,42
set color to &bkwh
@ 19,60 say '<      >'
set color to &bkwh
@ 19,62 say 'Message'
@ 20,51 clear to 22,77
set color to &bkwh
@ 21,51 say '         WAIT         '
use biblio
count for trim(authname) $ authors to ct
set color to &bkwh
set function 2 to 'F2;'
set function 6 to 'F6;'
do instruct
set color to &rdwh
@ 20,1 say ' F2 '
@ 21,1 say ' F6 '
set color to &bkwh
@ 20,7 say "to re-enter Author's name"
@ 21,7 say "to abandon author search"
if ct>0
set color to &bwhbl
@ 22,1 say ' F10 '
set color to &bkwh
```

```
@ 22,7 say 'OK, add "'+trim(authname)+'" to <Selections>'
endif
set color to &bkwh
@ 19,60 say '<       >'
set color to &bkwh
@ 19,62 say 'Message'
@ 21,51 clear to 21,78
@ 20,55 say 'Articles found authored'
@ 21,55 say 'by '+trim(authname)+':'
set color to &flashblwh
@ row(),col()+1 say ltrim(str(ct))
set color to &bkwh
accept '' to ch
do case
    case ch='Help'
    helpfrom='auth'
    loop
    case ch='F2'
    do instruct
    loop
    case ch='F6'
    store .f. to author
    do mark
    pass=pass-1
    return
    case ch='F10'
        z=trim(authname)
    do mark
    clear
    return
    case ch=' '
    pass=pass-1
    return
endcase
enddo
return
*End ofifauth

*****************************************************************
*                  IFDATE                  *
*****************************************************************
PROCEDURE IFDATE
*// ifdate:  allows user to input publication date as
*          a search criteria
*
* called from ROUTING.PRG
*
*
call savescr with 's1'
@ 1,0 clear to 1,79
@ 1,30 say 'DATE ENTRY SCREEN'
set bell off
set talk off
@ 3,1 clear to 18,15
do while .t.
    set color to &bkwh
    @ 5,18 say 'Enter any year date such as 1983 or 1985, etc., to'
    @ 6,18 say 'limit the search to articles published only in that year.'
    @ 20,1 clear to 22,49
if year(date())<1987
    @ 4,18 clear to 17,77
    @ 12,0 say 'System date is incorrect.  Error checking requires that'
    @ 13,0 say 'the correct system date be present.'
    ! date
    clear
    loop
endif
set color to &bkwh
@ 8,18 clear to 16,77
@ 8,18 TO 10,55
store '    ' to dateyear
@ 9,20 say "Enter year: "
***help loop
DO WHILE .t.
set color to &bkwh,,&bwhbl
@ 9,31 get dateyear picture '####'
read
IF READKEY()=0
    call savescr with 's1'
    DO HELP WITH 12
    call savescr with 'r1'
    LOOP
ENDIF
exit
enddo
***end help loop
set color to
if val(dateyear)=0
    clear
    pass=pass-1
    return
endif
```

```
if asc(dateyear) <> 70
    if (val(dateyear)<1900.or.val(dateyear)>year(date()))
    call bell with "3,4000"
    set color to &rdwh
    @ 12,18 say 'Not a valid date'
    @ 13,18 say 'Press any key to Re-enter... '
    wait ''
    loop
    endif
endif
set color to &flashblwh
@ 20,51 say '        WAIT          '
use biblio
count for dateyear $ date to ct
SET COLOR TO &bkwh
@ 20,51 CLEAR TO 20,78
IF CT=0
    @ 20,51 SAY 'NO'
ELSE
    set color to &flashblwh
    @ 20,51 say ' '+ltrim(str(ct))+' '
ENDIF
    set color to &bkwh
    @ 20,col() say ' Matching Records'
set function 2 to 'F2;'
set function 10 to 'F10;'
@ 20,1 clear to 20,49
@ 20,1 say ' F1 '
set color to &bwhbl
@ 22,1 say ' F10 '
set color to &bkwh
@ 20,7  say 'Abandon date, enter another'
@ 22,7  say 'Incorporate Date into search parameters'
set color to &bkwh
@ 17,0 say ''
accept '' to ch
set color to &bkwh
@ 7,45 clear to 7,70
set color to &bkwh
do case
    case ch='Help'
    helpfrom='date'
    loop
    case ch='F2'
    @ 14,11 clear to 16,60
    loop
    case ch='F10'
z=dateyear
date=.t.
do mark
date=.f.
    return
endcase
enddo
set bell on
clear
return
*End ofifdate

*****************************************************************
*                    IFCAT                    *
*****************************************************************
PROCEDURE IFCAT
*// IFCAT; lists pollutants contained in catlist.dbf
set typeahead to 2
USE CATLIST
GOTO TOP
set echo off
STORE 4 TO R
STORE 1 TO C
DO WHILE .T.
set color to &bwhbk
CLEAR
@ 0,0 to 22,79
set color to &bkwh
@ 1,1 clear to 21,78
set color to &bwhbk
@ 2,1 to 2,78 double
@ 3,39 to 21,39
set color to &bkwh
@ 1,1 clear to 1,78
@ 1,10 say ' Category '
DO WHILE .T.
set color to &bkwh
    @ 3,1 clear to 21,78
    if recno() <> 1
    set color to &ylwh
    @ 3,70 say chr(24)+' PgUp '+chr(24)
    set color to &bkwh
    endif
DO WHILE .NOT. EOF()
    SOUR=trim(CATEGORY)
```

```
* DISPLAY SOURCE MATERIALS IN STATUS BAR
    set color to &ylwh
    @ R,C SAY LTRIM(STR(RECNO()))
    set color to &bkwh
    @ row(),3 say ': '+CATEGORY
    R=R+1
    IF R=22
        R=4
        C=40
    ENDIF
    SKIP
    if recno()<reccount()
    IF R=21.AND.C=40
        @ 20,65 say chr(218)
        @ 20,66 to 20,78
        @ 21,65 say chr(179)
        set color to &ylwh
        @ 21,70 say chr(25)+' PgDn '+chr(25)
        set color to &bkwh
        EXIT
    ENDIF
    endif
ENDDO
set color to &bkwh
@ 23,0 clear to 24,79
set color to &ylwh.,&bwhrd
    @ 23,0 say '  Enter the number of the source category of interest:    then '+ret
    store '  ' to x
***help loop
DO WHILE .t.
    @ 23,55 get x picture '9999'
read
IF READKEY()=0
    call savescr with 's1'
    DO HELP WITH 11
    call savescr with 'r1'
    LOOP
ENDIF
exit
enddo
***end help loop
do case
    case readkey()=16
        store .t. to blank
        pass=pass-1
        return
    case readkey()=272

    case val(x)>reccount().or.readkey()=2
        goto 1
        r=3
        c=1
        loop
    case x='Help'
    store 'MATERIAL' to helpfrom
    y=recno()
    do epahelp
    case readkey()=6
    if recno()-38<1
        goto 1
    else
        goto recno()-38
    endif
    c=1
    r=3
    loop
    case readkey()=7
    if recno()>=reccount()
        goto 1
    endif
        r=3
        c=1
        loop
    case x='  '
    store .t. to blank
    pass=pass-1
    return
     endcase
    goto VAL(X)
* reset from string search for author,date,or reference
    store .f. to dat,author,ref
    do mark
    return
*   @ 22,67 clear to 23,78
*   R=3
*   C=1
*   pass=pass-1
    enddo
    enddo
*   return
*End ofifcat
```

```
*****************************************************************
*                    IFGRP                    *
*****************************************************************
PROCEDURE IFGRP
*// Ifgrp; lists category MATERIAL contained in grplist.dbf
store .f. to blank
set typeahead to 2
USE grplist
GOTO TOP
set echo off
STORE 3 TO R
STORE 1 TO C
set color to &bkwh
@ 0,0 CLEAR to 24,79
@ 0,0 to 22,79
@ 2,1 to 2,78 double
set color to &bkwh
@ 1,1 clear to 1,78
set color to &bkwh
@ 1,10 say ' Group Sources '
DO WHILE .T.
set color to &bkwh
    @ 3,1 clear to 21,78
    if recno() <> 1
    set color to &ylwh
    @ 3,70 say chr(24)+' PgUp '+chr(24)
    set color to &bkwh
    endif
DO WHILE .NOT. EOF()
    grp=TRIM(group)
    * DISPLAY Group types IN STATUS BAR

    set color to &ylwh
    @ R,C SAY LTRIM(STR(RECNO()))
    SET COLOR TO &bkwh
    @ R,3 SAY ': '+grp
    R=R+1
    SKIP
    if recno()<reccount()
    IF R>=22
        set color to &ylwh
        @ 21,70 say chr(25)+' PgDn '+chr(25)
        set color to &bkwh
        EXIT
    ENDIF
    endif
ENDDO
@ 23,0 clear to 24,79
set color to &bkwh,,&bkwh
@ 23,2 say 'Enter the number of the Source Group of interest:    then press'+ret
store '  ' to x
STORE .F. TO ENTERED
*DO READHELP WITH 7,X,23,52
DO WHILE .not. entered
    @ 23,51 get X
    read
    IF READKEY()=0
        call savescr with 's1'
        DO HELP WITH helpscm
        call savescr with 'r1'
        LOOP
    Else
        entered=.t.
        EXIT
    endif
enddo
do case
    case readkey()=16
        WAIT 'XX READKEY = 16, RETURN'
        store .t. to blank
        pass=pass-1
        return
    case readkey()=272
    case val(x)>reccount().or.readkey()=2
        WAIT 'XX READKEY=2 OR VAL(X)>RECCOUNT'
        goto 1
        r=3
        c=1
        loop
    case readkey()=6
    WAIT 'XX READKEY=6'
    if recno()-38<1
        goto 1
    else
        goto recno()-38
    endif
    c=1
    r=3
    loop
    case readkey()=7
    WAIT 'XX READKEY=7'
```

```
    if recno()>=reccount()
        goto 1
    endif
        r=3
        c=1
        loop
    case x=' '
    WAIT 'XX X IS BLANK, PASS=PASS-1,RETURNING'
    store .t. to blank
    pass=pass-1
    return
     endcase
    goto val(x)
* reset from string search for author,date,or reference
    store .f. to dat,author,ref
    do mark
*   pass=pass-1
    return
ENDDO
return
*End of ifgrp

*************************************************************
*                    IFPOL                  *
*************************************************************
PROCEDURE IFPOL
*// IFPOL; lists contents of database containing unique
*         listings of the chosen search area for listing
*         as a menu screen and assigns selections to
*         memvars: r1,r2,r3, and h1,h2, or h3
*
*z=field name
*p=field label
*&Z=field contents
*
* called from ROUTING
SET TYPEAHEAD TO 20
set color to &bkwh
@ 1,0 CLEAR to 22,79
* DRAW BOX
set color to &bwhbk
@ 0,0 to 22,79
@ 2,1 to 2,78 double
@ 3,38 to 21,38
set color to &ylwh
@ 1,1 clear to 1,78
set color to &bwhbk
@ 1,15 say ' Pollutant '
@ 23,0 clear to 24,79
call savescr with "s4"
* STORE FIRST AND LAST POLLUTANT'S RECORD ## TO BEGIN, END
USE &db
GOTO TOP
IF firstlet=' '
    STORE RECCOUNT() TO END
    STORE 1 TO BEGIN
else
    SET EXACT OFF
    STORE upper(trim(FIRSTLET)) TO FIRSTLET
* FIND FIRST POLLUTANT
    LOCATE FOR POLLUTANT=firstlet
* IF NO POLLUTANTS, DISPLAY MESSAGE, ALLOW RE-ENTRY
    IF RECNO()>RECCOUNT()
        set color to &bwhtrd
        call bell with "2,4000"
        @ 5,2 clear to 8,77
        @ 6,15 SAY 'NO POLLUTANTS FOUND WITH "'+FIRSTLET+'" AS THE FIRST LETTERS'
        PASS=PASS-1
        @ 7,26  say 'Press any key to continue...'
        wait ''
        RETURN
    ENDIF
* FIND LAST POLLUTANT WITH SAME FIRST LETTER
    STORE RECNO() TO BEGIN
        DO WHILE POLLUTANT=FIRSTLET
            SKIP
        ENDDO
    STORE RECNO() TO END
    GOTO BEGIN
ENDIF
set echo off
STORE 4 TO R
STORE 1 TO C
pol=' '
DO WHILE .T.
* RE-DRAW SAVED BOX
call savescr with "r4"
    if recno() <> begin
    set color to &ylwh
    @ 3,70 say chr(24)+' PgUp '+chr(24)
    set color to &bkwh
    endif
```

```
DO WHILE RECNO()<END
    show= trim(&z)
    set color to &ylwh
    @ R,C SAY LTRIM(STR(RECNO()))
    set color to &blwh
    @ row(),col() say ':  '+show
    R=R+1
    IF R=22
        R=4
        C=40
    ENDIF
    SKIP
if recno()<END
     IF R=21.AND.C=40
    set color to &ylwh
        @ 21,70 say chr(25)+' PgDn '+chr(25)
        set color to &bkwh
    EXIT
     ENDIF
endif
ENDDO
*DISPLAY INSTRUCTIONS AT BOTTOM OF SCREEN
set color to &bkwh,,&bkwh
@ 23,0 clear to 24,79
set typeahead to 1
@ 23,0 say "  Enter the number of the pollutant of interest:    then "+rea
* ACCEPT USER INPUT FOR DESIRED POLLUTANT
        store '    ' to pol
***help loop
DO WHILE .t.
        @ row(),48 get pol picture '9999'
read
IF READKEY()=0
    call savescr with 's1'
    DO HELP WITH 5
    call savescr with 'r1'
    LOOP
ENDIF
exit
enddo
***end help loop
do case
        case readkey()=15
            store .t. to blank,nochange
            pass=pass-1
            return
        case readkey()=272 &&Return, data changed

        case val(pol)>reccount().or.readkey()=2.or.pol='0'
            goto begin
            r=3
            c=1
            loop
        case readkey()=7.or.readkey()=5
* USER WANTS TO MOVE FORWARD A SCREEN
* LAST POLLUTANT ALREADY SHOWING?
            IF RECNO()=END
            GOTO BEGIN
            R=4
            C=1
* IF LAST POLLUTANT NOT SHOWING
            ELSE
            skip
            ENDIF
    case readkey()=6.or.readkey()=4
* USER WANTS TO BACK UP A SCREEN
            do case
* MORE THAN ONE SCREEN OF POLLUTANTS AVAILABLE?
            case END-BEGIN>70
* LESS THAN ONE FULL SCREEN DOWN?
                    if recno()-70<BEGIN
                        goto BEGIN+70
                endif
* MORE THAN ONE FULL SCREEN DOWN? BACKUP 70.
                    GOTO RECNO()-70
* START DISPLAY AT POSITION
                    r=4
                    C=1
                    LOOP
* LESS THAN ONE SCREEN OF POLLUTANTS AVAILABLE?
        case END-BEGIN<70
                    goto BEGIN
                    r=4
                    c=1
                    loop
    endcase
endcase
    pol=val(pol)
* IF INPUT NUMBER IS ACCEPTABLE
    if POL>=1.AND.POL<=RECCOUNT()
    GOTO POL
    h=&Z
```

```
* reset Z's if string search previously chosen,
* disable if not chosen: string search for author, date,
* or reference in mark.prg
    store .f. to dat,author,ref
    do mark  && store string inputs to variables
    return
    endif

    @ 22,67 clear to 23,78
    R=4
    C=1
ENDDO
* End of ifpol

*************************************************************
*                  IFREF                    *
*************************************************************
PROCEDURE IFREF
*// IFREF; lists REFERENCES contained in reflist.dbf
store '  ' to xq
store .t. to ref
USE REFLIST
GOTO TOP
set talk off
set echo off
STORE 3 TO R
STORE 1 TO C
public refer
DO WHILE .T.
@ 1,0 CLEAR to 23,79
set color to &bkwh
@ 1,1 clear to 22,78
    set color to &bwhbk
@ 0,0 to 22,79
@ 2,1 to 2,78 double
set color to &ylwh
@ 1,1 clear to 1,78
@ 1,10 say ' Journal '
DO WHILE .NOT. EOF()
    refer=TRIM(JOURNAL)
* DISPLAY JOURNAL IN STATUS BAR
    set color to &bkwh
    @ R,C SAY LTRIM(STR(RECNO()))
    set color to &bkwh
    @ row(),col() say ': '+refer
    R=R+1
    SKIP
if recno()<reccount()
    IF R>=20
    set color to &flashblwh
    @ 20,65 say chr(218)
    @ 20,66 to 20,78
    @ 21,65 say chr(179)
    set color to &bkwh
    @ 21,67 say " MORE..."+RET
    set color to
    EXIT
    ENDIF
endif
ENDDO
set color to &bkwh
@ 24,0 clear to 24,79
    set color to &flashblwh
    @ 24,0 say chr(16)
    set color to &ylwh
@ 24,1 say 'Enter the number of the REFERENCE of interest:    then press'
set color to &bkwh
DO READHELP WITH 9,XQ,24,48
do case
    case val(xq)>reccount()
        store 'Up' to xq
    case xq='Help'
    store 'REF' to helpfrom
    y=recno()
    do epahelp
    case xq="Up"
        goto 1
        r=4
        C=1
        LOOP
    case xq=' '
        pass=pass-1
        return
endcase
xq=VAL(xq)
    if xq>=1.AND.xq<=RECCOUNT()
    GOTO xq
    @ 24,1 CLEAR TO 24,78
    @ 0,0 CLEAR to 23,79
    z=trim(journal)
    z=stuff(z,1,0,"")
    z=stuff(z,len(z)+1,1,"")
```

```
    do mark
    return
    ELSE
    @ 22,67 clear to 23,78
    R=3
    C=1
    endif
ENDDO
*end offfref

*************************************************************
*                    IFMAT                  *
*************************************************************
PROCEDURE IFMAT
*// Ifmat; lists source type contained in matlist.dbf
store .f. to blank
set typeahead to 2
USE Matlist
GOTO TOP
set echo off
STORE 3 TO R
STORE 1 TO C
set color to &bkwh
@ 0,0 CLEAR to 24,79
@ 0,0 to 22,79
@ 2,1 to 2,78 double
set color to &bkwh
@ 1,1 clear to 1,78
set color to &bkwh
@ 1,10 say ' Source Material '
DO WHILE .T.
set color to &bkwh
     @ 3,1 clear to 21,78
     if recno() <> 1
     set color to &ylwh
     @ 3,70 say chr(24)+' PgUp '+chr(24)
     set color to &bkwh
     endif
DO WHILE .NOT. EOF()
     refer=TRIM(MATERIAL)
     * DISPLAY Source types IN STATUS BAR

     set color to &ylwh
     @ R,C SAY LTRIM(STR(RECNO()))
     SET COLOR TO &bkwh
     @ R,3 SAY ': '+refer
     R=R+1
     SKIP
     if recno()<reccount()
     IF R>=22
          set color to &ylwh
          @ 21,70 say chr(25)+' PgDn '+chr(25)
          set color to &bkwh
          EXIT
     ENDIF
     endif
ENDDO
set color to &bkwh.,&bkwh
@ 23,0 clear to 24,79
@ 23,2 say 'Enter the number of the Source Material of interest:    then press'+ret
store '  ' to x
***help loop
DO WHILE .t.
          @ row(),55 get x picture '9999'
read
IF READKEY()=0
     call savescr with 's1'
     DO HELP WITH 5
     call savescr with 'r1'
     LOOP
ENDIF
exit
enddo
***end help loop
do case
     case readkey()=16
          store .t. to blank
          pass=pass-1
          return
     case readkey()=272
     case val(x)>reccount().or.readkey()=2
          goto 1
          r=3
          c=1
          loop
     case readkey()=6
     if recno()-38<1
          goto 1
     else
          goto recno()-38
     endif
     c=1
```

```
    r=3
    loop
    case readkey()=7
    if recno()>=reccount()
        goto 1
    endif
        r=3
        c=1
        loop
    case x=' '
    store .t. to blank
    pass=pass-1
    return
     endcase
    goto val(x)
* reset from string search for author,date,or reference
    store .f. to dat,author,ref
    do mark
    return
ENDDO
return
*end ofiftnat

*****************************************************************
*                  HELPSCRN                  *
*****************************************************************
PROCEDURE HELPSCRN
*// helpscrn
@ 0,0 clear to 24,79
set color to &bkwh
@ 0,15 say 'EXPLANATION OF PARAMETERS THAT MAY BE SEARCHED'
set color to &bkwh
@ 2,1 say 'GROUP:'
@ 5,1 say "CATEGORY:"
@ 8,1 say 'MATERIAL:'
@ 11,1 say 'POLLUTANT:'
@ 14,1 say "AUTHOR:"
@ 17,1 say 'JOURNAL or DATE:'
set color to &bkwh
@ 3,1 say 'Major classes of sources, including structure, water, chemical sources, etc.'
@ 6,1 say "A broader grouping of sources, such as textiles, wood products, etc"
@ 9,1 say 'Individual sources: carpet, stoves, etc. (not necessarily physical materials)'
@ 12,1 say 'Individual pollutants having source materials listed in the database'
@ 15,1 say 'Author of an article used in the database.'
@ 18,1 say 'Publishing journal and published date of articles included in the database.'
set color to &bkwh
@ 1,0 to 23,79 double
SET COLOR TO &bkwh
@ 24,0 clear to 24,79
@ 24,20 SAY ' Press any key to continue...'
@ 23,21 say ''
wait ''
return
*End ofhelpscrn

*****************************************************************
*                  INSTRUCT                  *
*****************************************************************
*>>>>>>>>>>>>>>>>>
PROCEDURE INSTRUCT
    set color to &bkwh
    @ 19,0 clear to 23,79
    @ 19,0 to 23,79 double

    @ 19,50 say chr(203)
    @ 20,50 to 22,50 double
    @ 23,50 say chr(202)
    @ 19,16 say "<          >"
    @ 19,18 say 'Program Options'
    RETURN
*End ofInstruct

*****************************************************************
*                  KEYUP                  *
*****************************************************************
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
PROCEDURE KEYUP
close databases
select a
use poll1
select b
use sample index samp_no
select c
use cond index cond_no
select d
use source index source_no
select e
use biblio index biblio_no
select a
set relation to samp_no into b
select b
set relation to cond_no into c
```

```
select c
set relation to source_no into d
select d
set relation to biblio_no into c
RETURN
*End ofKeyup

*****************************************************************
*                    KEYUP2                    *
*****************************************************************
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
PROCEDURE KEYUP2
CLOSE all
SELECT 1
USE POLL1
SELECT 2
USE SOURCE INDEX SOURCE_NO
SELECT 1
SET RELATION TO SOURCE_NO INTO SOURCE
RETURN
*End ofkeyup2

*****************************************************************
*                    LINK                    *
*****************************************************************
PROCEDURE LINK
*// LINK -> Displays bibliography of each found biblio
*         reference that matched the search criteria

******
PUBLIC FIRSTIME,write
write=.t.
call cursoff
if len(xstr)=0
    return to master
endif
* INITIALIZE VARIABLES AND SET UP ENVIRONMENT
close databases
CLOSE ALL
ctr=1
SET SCOREBOARD OFF
SET STATUS OFF
*SET PROCEDURE TO DISPPROC
PUBLIC LP,number
set color to &bkwh
clear
set typeahead to 20
USE BIBLIO
* DETERMINE THE NUMBER OF RECORDS IN THE STRING
* IMPORTED FROM THE SEARCH MODULE
store INT(len(xstr)/4) to length
store 0 to counter
scment=1
    set color to &rdwh
    @ 11,21 say chr(24)
    @ 13,21 say chr(25)
    @ 15,21 say 'Q'
    @ 17,20 say ' F2'
    set color to &bwhbl
    @ 19,20 say "F10"
    set color to &bkwh
    @ 11,24 say 'Previous matching article'
    @ 13,24 say 'Next matching article'
    @ 15,24 say 'Quit to main menu'
    @ 17,24 say 'Abstract for this article'
    @ 19,24 say 'Data for this article'
** Save the instruction bar
call savescr with 's1'
do while .not. eof()
    call savescr with 'r1'
* LOCATE RECORD IN "XSTR" AND DISPLAY
    set color to &bkwh
    if counter<=length  &&->length= # records
        store (val(substr(xstr,counter*4+1,3))) to point
        if point>0
        goto point
        else
        clear
        return to master
        endif
    else
        call bell with '2,3000'
        counter=1
        loop
    endif
    STORE RECNO() TO R,number
do case
    case scment<1
        scment=1
    case scment>length
        scment=length
endcase
```

```
set color to &bkwh
*DISPLAY BIBLIOGRAPHY IN FORMATTED MANNER
do bibdisp
@ 9,0 clear to 9,79
@ 9,15 say ltrim(str(scment))+' of '+ltrim(str(length))+' Articles matching search criteria.'
    CH=0
do while ch=0
    do while ch=0
        ch=inkey()
    enddo
    if ch=35
    ch=0
    do help with 8
    endif
enddo
* INDEX POINTER IN DATABASE ACCORDING TO ARROW KEYS
set memowidth to 75
    DO WHILE .T.
        set color to &bkwh
        if ch=72.or.ch=68
            store .t. to hlp3
            do helptxt
            loop
        endif
        DO CASE

        case ch=-1
            set color to &bkwh
            clear
            disp off abstract
            wait
            exit
        case ch=5     &&->up arrow
            IF COUNTER=0
                CALL BELL WITH "2,5000"
                set color to &bkwh
                counter=1
                scment=1
            LOOP
            ELSE
                COUNTER=COUNTER-1
                scment=scment-1
                EXIT
            ENDIF
        CASE ch=24        && *****-> DN ARROW
                IF COUNTER<LENGTH-1
                    counter=counter+1
                    scment=scment+1
                    EXIT
                ELSE
                    CALL BELL WITH "2,5000"
                    set color to &bkwh
                    counter=counter-2
                    LOOP
                ENDIF
                LOOP
            CASE ch=-9.or.CH=13   && *****-> F10 or ENTER KEY
                set color to &bkwh
                @ 8,0 clear to 24,79
* DISPLAY THE DATA ASSOCIATED WITH THE RECORD
* POINTED TO
                do keyup2
                public more
                more=.f.
                firstime=.t.
                write=.t.
                do datadisp
                exit
            case chr(ch) $ 'Qq'
                close databases
                store .f. to hlp
                return to master
            OTHERWISE
                EXIT
            ENDCASE
    ENDDO
ENDDO
*End of link

*************************************************************
*                    LOGIC                  *
*************************************************************
PROCEDURE LOGIC
*// LOGIC.prg:   Assigns the default boolean operators
*              Displays the logical relationships
*              Allows user to initiate other rel's.
* CALLED FROM MAKESTRG, EPFILTER
store .f. to reenter
set color to &bkwh
@ 4,17 clear to 16,77

************************************************
```

```
* STORE LABELS OF SEARCH AREAS TO TEMP. VARIABLES
    a=h1
    b=h2
    c=h3

*********************************************************
* CHANGE POSITION OF VARIABLES SO THAT VARIABLES IN THE
* SAME DATABASE ARE GROUPED AS THE 2ND AND 3RD VARIABLES
do case

*********************************************************
* If A = B, BUT NOT C, exchange A and C

    CASE A=B.AND.A <> C.and.C <> ' '
** EXCHANGE LABELS
        HOLD=A
        A=C
        C=HOLD
** EXCHANGE FIELDS
        HOLD=R1
        R1=R3
        R3=HOLD
** EXCHANGE SEARCH STRINGS
        HOLD=Z1
        Z1=Z3
        Z3=HOLD
* If A = C BUT NOT B, exchange A and b
    CASE A=C.AND.A <> B
        HOLD=A
        A=B
        B=HOLD
        HOLD=R1
        R1=R2
        R2=HOLD
        HOLD=Z1
        Z1=Z2
        Z2=HOLD
    ENDCASE

*** determine default values for Boolean operators
*** or, if user assigned, use user values (R or L)
** IF C IS BLANK, 2ND BOOLEAN OPERATOR MUST BE BLANK TOO
if C= ' '
    bool2=' '
** IF B IS BLANK, 1ST BOOLEAN OPERATOR MUST ALSO BE BLANK
    if B=' '
        bool1=' '
** IF NEITHER ARE BLANK...
            else
        if A=B.and.bracket <> 'N'
            BOOL1='OR '
            EVBOOL1='.or.'
        else
            bool1='AND'
            EVBOOL1='.and.'
        endif
    endif
else
    if B=C  &&.and.bracket <> 'N'
        bool2='OR '
        EVBOOL2='.or.'
    else
        bool2='AND'
        EVBOOL2='.and.'
    endif
    if A=B &&.and.bracket <> 'N'
        bool1='OR '
        EVBOOL1='.or.'
    else
        bool1='AND'
        EVBOOL1='.and.'
    endif
endif
* Determine bracket positions
do case
*DEFAULT BRACKET POSITIONS
    case (BOOL2='OR '.AND.BOOL1='AND')
    bracket='R'
    case (BOOL2='AND'.AND.BOOL1='OR ')
    BRACKET='L'
    case (BOOL1=BOOL2)
    BRACKET='N'
*IF USER DECIDES NOT TO USE DEFAULT BRACKET POSITIONS
    case bracket='RR'
    bracket='R'
    case bracket='LL'
    bracket='L'
    case bracket='NN'
    bracket='N'
endcase
set color to &bkwh
@ 19,52 to 19,77 double
```

```
set color to &bkwh
***Clear old brackets (whether it needs it or not)
if noshow
@ 14,32 clear to 14,60
@ 16,32 clear to 16,60
@ 15,43 say ' '
@ 15,58 say ' '
@ 15,47 say ' '
***Draw new brackets
SET COLOR TO &bkwh
do case
case bracket='N'
case bracket='L'
@ 14,32 say bracktop
@ 16,32 say brackbot
@ 15,32 say chr(179)
@ 15,47 say chr(179)
case bracket='R'
@ 14,43 say bracktop
@ 16,43 say brackbot
@ 15,43 say chr(179)
@ 15,58 say chr(179)
endcase
***  Draw A,B,C Labels, and boolean relators
SET COLOR TO &bkwh
if bool1 <> ' '
@ 15,33 say ' A '
endif
if B <> ' '
@ 15,44 SAY ' B '
endif
if C <> ' '
@ 15,55 SAY ' C '
endif
set color to &bkwh
@ 15,39 SAY BOOL1
@ 15,50 SAY BOOL2
endif
    h1=a
    h2=b
    h3=c

**********
return
*End of logic
```

```
*****************************************************************
*                    LOGIC2                    *
*****************************************************************
PROCEDURE LOGIC2
*// LOGIC2   Determines relationships between entries.
*          determines which dbf's to search
*          assigns number of search passes to 'dblook'
* called from epfilter
DO CASE    && which dbf does Z1 belong ?
   case Z1="AUTHORS".OR.Z1="TITLE".OR.;
        Z1="DATE".OR.Z1="REFER"
        STORE 'BIBLIO' TO DBF1  &&,DBF
   case Z1="GROUP".OR.Z1='MATERIAL'.OR. Z1='CATEGORY'
        STORE 'SOURCE' TO DBF1  &&,DBF
   case Z1="POLLUTANT"
        STORE 'POLL1' TO DBF1  &&,DBF
ENDCASE
DO CASE    && which dbf does Z2 belong ?
   case Z2="AUTHORS".OR.Z2="TITLE".OR.;
        Z2="DATE".OR.Z2="REFER"
        STORE 'BIBLIO' TO DBF2  &&,DBF
   case Z2="GROUP".OR.Z2='MATERIAL'.OR. Z2='CATEGORY'
        STORE 'SOURCE' TO DBF2  &&,DBF
   case Z2="POLLUTANT"
        STORE 'POLL1' TO DBF2  &&,DBF
   case z2=' '
        store ' ' TO DBF2
ENDCASE
DO CASE    && which dbf does Z3 belong ?
   case Z3="AUTHORS".OR.Z3="TITLE".OR.;
        Z3="DATE".OR.Z3="REFER"
        STORE 'BIBLIO' TO DBF3  &&,DBF
   case Z3="GROUP".OR.Z3='MATERIAL'.OR. Z3='CATEGORY'
        STORE 'SOURCE' TO DBF3  &&,DBF
   case Z3="POLLUTANT"
        STORE 'POLL1' TO DBF3  &&,DBF
   case z3=' '
        store ' ' TO DBF3
ENDCASE
*DETERMINE RELATIONSHIPS BETWEEN CHOSEN DBF'S
STORE .F. TO ENT1_DB1,ENT2_DB2,ENT2_DB1,ENT3_DB1,ENT3_DB2,ENT3_DB3
DO CASE
   CASE BOOL1=' '
        STORE .T. TO ENT1_DB1
   CASE BOOL1 <> ' '.AND.BOOL2=' '
        do case
```

```
            case dbf1 <> dbf2
                store .t. to ENT2_DB2
            case dbf1=dbf2
                store .t. to ENT2_DB1
        endcase
    CASE BOOL1 <> ' '.AND.BOOL2 <> ' '
        do case
            case dbf1 <> dbf2 .and. dbf2 <> dbf3
                store .t. to ENT3_DB3
            case (dbf1 <> dbf2 .and. dbf2=dbf3).or.(dbf1=dbf2.and.dbf2 <> dbf3)
                store .t. to ENT3_DB2
            case dbf1=dbf2 .and. dbf2=dbf3
                store .t. to ENT3_DB1
        endcase
    ENDCASE
return to master
*End of logic2

*************************************************************
*                  MAINMENU                 *
*************************************************************
PROCEDURE MAINMENU
*// MAINMENU   DISPLAY MAIN MENU
set color to &bkwh
do while .t.
***Do not reset these if user is correcting an entry
store ' ' to bool1,bool2,bool3,r1,r2,r3,z1,z2,z3,h1,h2,h3
* WRITE MENU TO SCREEN
CLEAR
set color to &bkwh
@ 3,28 say ' dMAPS MAIN MENU '
set color to &bkwh
@ 6,7 to 8 ,70
set color to &bkwh
set color to &bwhbl
@ 7,8 SAY '  Quit     Search data     Enter data     Utilities   '
set color to &bkwh
@ 7,20 say ' '
@ 7,36 say ' '
@ 7,53 say ' '
set color to &bwhbl
    @ 7,12 say 'Q'
    @ 7,23 say "S"
    @ 7,40 say "E"
    @ 7,57 say "U"
* DISPLAY SCREEN MENU HELP INFORMATION FOR FIRST MENU
    pass=1
    set color to &bkwh
    @ 20,0 clear to 20,79
    @ 20,15 SAY 'Type the first letter of an appropriate menu item.'
    set color to &bkwh
set color to &bkwh
    @ 12,16 SAY '<Q> = Quit DMAPS, return to DOS'
    @ 13,16 say '<S> = Begin a search of DMAPS'
    @ 14,16 say '<E> = Add information to DMAPS'
    @ 15,16 say '<U> = Utilities menu (Printer set-up, colors, etc.)'
    set color to &bkwh
    @ 0,0 say ''
    call savescr with "S5"
* WAIT FOR USER INPUT FROM MENU
* AND DISPLAY HELP MESSAGE IF TIME EXCEEDS 5 SECONDS
SET TYPEAHEAD TO 20
i=0
do while i=0
    do while i=0
        i=inkey()
    enddo
    if i=35
    i=0
    do help with 8
    endif
enddo
pick=chr(i)

*************************************************************
* DRAW BOX FOR MESSAGE IF NON-WORKING FEATURE OR "QUIT"
if pick$'SsQqeEuU'

*****************************
* NO OTHER OPTIONS WORKING
    else
        @ 10,14 clear to 23,68
        @ 11,18 to 16,58
        if pick$'Bb'
            @ 12,24 say 'This feature not available'
        else
            @ 12,24 say 'Not an appropriate keypress.'
        endif
        @ 15,20 say 'press a correct letter to choose.'
        @ 14,20 say 'Press any key to continue, then'
        wait ''
endif
```

```
**********************************************
* DIRECT PROGRAM PATH UPON USER INPUT
do case
    case pick$"Qq"
        clear all
        close all
        quit

    case pick$"Hh"
        @ 10,20 say 'Help function not available'
        @ 11,20 say 'Press any key to continue...'
        set color to &bkwh
        wait ''
    case pick$"Ss"
        do PASSMENU
        set typeahead to 0
    case pick$"Ee"
        do enter
        set color to &bkwh
    case pick$"Uu"
        do utility

    otherwise
        loop
    endcase
enddo
*End of mainmenu

*************************************************************
*                  MAKESTRG                 *
*************************************************************
PROCEDURE MAKESTRG
*// MAKESTRG:  assigns strings to incorporate into actual
*          search strings, assembles search strings
*          and assigns them to the memvar "evaluation"
*
*  called from epastart
store ' ' to evbool1,evbool2
* DETERMINE BOOLEAN OPERATORS MOST LIKELY, GIVEN ENTRIES
do LOGIC
*DETERMINE HOW MANY SELECTIONS HAVE BEEN ENTERED
set typeahead to 0
IF evbool2 <> ' '          &&if all 3, bool2 will not be blank
    STORE '3' to key
ELSE
    IF evbool1 <> ' '      &&if A&B, bool1 is not blank
        STORE "2" to key
    ELSE
        STORE "1" to key   &&if A only, bool1 is blank
    ENDIF
ENDIF

**********************************************
* DISPLAY USER SELECTIONS
@ 0,0 CLEAR TO 2,79
@ 0,0 clear to 18,15
set color to &bkwh
@ 2,16 clear to 2,78
@ 2,18 say "LOGIC DISPLAY SCREEN"
    DO CASE
        CASE key='1'
                @ 4,17 clear to 12,77
            set color to &bkwh
            @ 5,18 say ' A '
            set color to &bkwh
            @ 5,22 say z1+" = "+r1
        CASE KEY='2'
            @ 4,17 clear to 12,77
            set color to &bkwh
            @ 5,18 say ' A '
            @ 7,18 say ' B '
            set color to &bkwh
            @ 5,22 say z1+" = "+r1
            @ 7,22 say z2+" = "+r2
        CASE KEY='3'
            @ 4,17 clear to 12,77
            set color to &bkwh
            @ 5, 18 say ' A '
            @ 7, 18 say ' B '
            @ 9, 18 say ' C '
            set color to &bkwh
            @ 5, 22 say z1+" = "+r1
            @ 7, 22 say z2+" = "+r2
            @ 9 ,22 say z3+" = "+r3
    ENDCASE
* CREATE SEARCH STRINGS
do case
    case ent1_db1.or.ent2_db1.or.ent3_db1
        do case
            case ent1_db1
                STORE "&z1='&R1'" TO EVALUATE1
```

```
        case ent2_db1
            STORE "&z1='&R1'&EVBOOL1&z2='&R2'" TO EVALUATE1
        otherwise
            STORE "&z1='&R1'&EVBOOL1&z2='&R2'" TO PART1
            STORE "&PART1&EVBOOL2&z3='&R3'" TO EVALUATE1
    endcase
case ent3_db2.or.ent2_db2
    if ent3_db2
        if dbf2=dbf3
            STORE "&z3='&R3'&EVBOOL2&z2='&R2'" TO EVALUATE2
            STORE "&z1='&R1'" TO EVALUATE1
            dbf2=dbf3
        else
            temp=dbf3
            dbf3=dbf1
            dbf1=temp
            store "&z1='&R1'&evbool1&z2='&r2'" to evaluate2
            store "&z3='&r3'" to evaluate1
        endif
    else
            STORE "&z1='&R1'" TO EVALUATE1
            STORE "&z2='&R2'" TO EVALUATE2
    endif
case ent3_db3
            STORE "&z1='&R1'" TO EVALUATE1
            STORE "&z2='&R2'" TO EVALUATE2
            STORE "&z3='&R3'" TO EVALUATE3
endcase
store .f. to dosearch

*****************************************************
* ALLOW USER TO ALTER DEFAULT SETTINGS AND ENTRIES
if .not. dosearch
    set color to &rdwh
    @ 20,2 clear to 22,49
    @ 20,51 clear to 22,78
    @ 21,10 say 'Are selections correct?  (Y/N)'
    @ 22,10 say ' '
    set color to &whwh
    wait '' to x
    set color to &bkwh
    @ 2,16 clear to 2,79
    do case
        case x $ 'Yy'
            store .t. to dosearch
        case x $ 'Nn'
            set typeahead to 20
* BRANCH TO ALTERATION PROGRAM
            do epalter
            set typeahead to 0
    endcase
endif
* CONTINUE IF USER PICKED "Y" ABOVE
if dosearch
    set color to &bkwh
    @ 21,1 clear to 21,49
    @ 20,51 say 'PLEASE WAIT, program active '
* BRANCH TO PROGRAM TO DISPLAY AND TIME THE SEARCHES
    DO TIMER
    set typeahead to 20
    return
endif
* END OF MAKESTRG.PRG

*****************************************************
*                  MARK                 *
*****************************************************
PROCEDURE MARK
**mark
*called from "if's"
if blank
    return
endif
public string
do case
    case author
        string="&z"
        P="&Z"
        Z="AUTHORS"
    case ref
        string="REFER"
    case substr(z,2,1) $ '1'
        string=z
    case dat
        z="DATE"
        string=dateyear
    otherwise
    STRING=lower(TRIM(&Z))   && Z=field name from PASSMENU
    endcase
    do case
        case pass=1
        I11=lower(P)        && P=string to search for
```

```
        R1=string
        Z1=Z
        case pass=2
        H2=lower(P)
        R2=string
        Z2=Z
        case pass=3
        H3=lower(P)
        R3=string
        Z3=Z
    endcase
return
*End of mark

*************************************************************
*                    MONOBAR                    *
*************************************************************
PROCEDURE MONOBAR
parameters frow,lrow,records,beginrng,start
store frow+1 to frow,row
lrow=lrow+1
barpoint=1
set typeahead to 5
chosen=.f.
store 'chr(row)+chr(col)+chr(hi)+chr(wide)+chr(color)' to fill
load cfill
set index to
goto recno()-records
store recno() to rec
store "" to stmg
store 0 to i
store 11 to col
store 1 to hi
store 57 to wide
store 31 to color
do while .t.
call cfill with &fill
point=.t.
do while point
    point=.f.
    i=0
    do while i=0
    do while i=0
        i=inkey()
    enddo
    if i=35
        i=0
        do help with 8
    endif
    enddo
enddo
color=113
do case
    case i=5  &&up
        if row>frow
            call cfill with &fill
            rec=rec-1
            barpoint=barpoint-1
            color=31        &&WHITE/BLUE
                ROW=ROW-1
    else
        row=row+1
        call cfill with &fill
        row=row-1
        color=31
    endif
    case i=24  &&->Dn
        if row<lrow-1
        call cfill with &fill
        row=row+1
        rec=rec+1
        color=31        &&WHITE/BLUE
        barpoint=barpoint+1
    endif
    case i=3    &&PGDN
        more=.f.
        y=pollutant
        write=.t.
        if recno()+records<=reccount()
            skip records
        else
            GOTO BOTTOM
        endif
    records=0
    frow=frow-1
    return

    case i=18.or.i=1  && (PGUP or Home)
    write=.t.
    goto start
    row1=lrow-2
    records=0
```

```
    frow=frow-1
    page=1
    return
    case i=-9.or.i=13  && F10 or Enter
    clear
    do recfind
    DO datadisp
    case i=113.or.i=81  &&Qq

        close all
        return to master
    case i=114.or.i=82  &&Rr
    write=.f.
    store .t. to more
    return
    endcase
i=0
enddo
RETURN
*End Monobar

*************************************************************
*                  PASSMENU                 *
*************************************************************
PROCEDURE PASSMENU
*// passmenu
*displays messages at top of screen, depending
*on which pass through the choice menu the
*         user is in.
* called from mainmenu
Set color to &bkwh
set talk off
do while .t.
do bar
if pass <> 4
     set color to &bkwh
     @ 24,0 clear to 24,79
endif
     set color to &bkwh
if reentry
     STORE .F. TO REENTRY
     @ 5,18 clear to 10,77
     @ 5,18 say ' Re-enter the new or incorrect search criteria...'
     @ 6,18 SAY ' [F10] FOR NO CHANGE'
else
     @ 5,18 clear to 10,77
     do case
     case pass=1
          set color to &bkwh
          @ 5,18 say 'Begin a search by typing the first letter '
          @ 6,18 say 'of a search parameter item listed at left...'
          set color to &bkwh
          @ 8,18 say 'The top four selections search for data items, while the'
          @ 9,18 say 'bottom three selections search bibliographic information.'
          @ 11,18 say 'You may make up to three simultaneous search selections, and'
          @ 12,18 say 'you may select from both groups concurrently.'
     case pass=2
          SET COLOR TO &bkwh
          @ 5,18 clear to 10,77
          @ 5,18 say 'Choose a second search parameter or press [F10] to'
          @ 6,18 say 'approve and find the selection listed below...'
     case pass=3
     set color to &bkwh
          @ 5,18 clear to 10,77
          @ 5,18 say 'Choose a third search parameter or press [F10] to'
          @ 6,18 say 'approve and find the selections listed below...'
     case pass=4.or.reentry
          set color to &bkwh
endcase
endif
if pass>1
     if pass=4
     set color to &bkwh
     @ 3,1 clear to 17,15
     @ 5,18 clear to 10,77
     @ 5,18 say 'Only three selections are allowed.'
     @ 7,18 say 'Press [F10] to find or alter the selections made...'
     endif
     set color to &bwhbl
     @ 22,2 say ' F10 '
     set color to &bkwh
     @ 22,8 say ' Search Database '
endif
@ 0,1 say ''
set color to &bkwh
wait '' to x
set typeahead to 0
set bell off
set color to &bkwh
x=upper(x)
do case
     case x$'Rr'.and.pass=1
```

```
        set typeahead to 20
        return
    CASE x$"hH"
        DO HELPSCRN
        loop
    case asc(x)=247  &&[F10]
        if pass=1
*@ 10,0 clear to 10,79
*@ 10,10 say '
            loop
        endif
    case x$"GCMPAJD"
    otherwise
        do help with 6
        if pass<=1
            loop
        endif
endcase
do ROUTING      && routes according to menu selection
enddo
if reentry
    store .f. to temp
    return
else
    wait
endif
return

*>>>>>>>>>>>>>>>>>>>>
PROCEDURE recfind
flag=1
do while flag<barpoint
    p1=pollutant
    do while pollutant=p1
        skip
    enddo
    flag=flag+1
enddo
public ptc1
store recno() to PTC1
do summary
return

PROCEDURE ROUTING

*// routing
*
* do case routine for main menu screen
**
* Z=FIELD NAME
* P=SCREEN LABEL
* R=STRING TO CHECK Z FOR (ASSIGNED IN EPIF???.PRG'S)
*
* called from passmenu
PUBLIC blank
do while .t.
IF asc(x)=247.OR.X$"HPGMCADJ"
    do case
    case asc(x)=247  &&->"F10"
        do epfilter
    case x='H'
        @ 10,20 say 'Help not available'
        @ 11,20 say 'Press any key to continue...'
        store .t. to na
        pass=pass-1
        if pass=0
            pass=1
        endif
    case x='P'
* PROMPT USER FOR FIRST LETTERS OF POLLUTANT
        do epflet
** (typeahead=0)
*store name of menu list dbf to db
        db='POLLIST'
        store .f. to na
*store appropriate field name in DB
        Z='POLLUTANT'  && FIELD NAME
*store screen label to variable P
        P='Pollutant:'
*access program to display DB (menu)
        do ifpol
        if nochange
        store .f. to blank
        endif
    case x='G'
        db='grplist'
        store .f. to na
        Z='GROUP'
        p='Group Sources:'
        do ifgrp        && does indiv groups.
    case x='M'
        db='matlist'
```

```
        store .f. to na
        Z='MATERIAL'
        p='Source MATERIAL:'
        do ifmat        && does indiv src mats.
    case x='C'
        db='catlist'
        Z='CATEGORY'
        p='Source Category:'
        do ifcat        && does s-mater. classes
    case x='A'
        Z='AUTHORS'
        P='Author(s):'
        do ifauth
        store .f. to na
    case x='D'
        Z='DATE'
        P='Publication Date:'
        do ifdate
        store .f. to na
    case x='J'
        @ 4,18 clear to 16,77
        @ 6,18 say "This feature not yet operational"
        @ 8,18 say 'Press any key to continue...'
        wait ''
        return
*   db='reflist'
*   Z='REFER'
*   P='Journal Name:'
*   do ifref
*   store .f. to na
endcase
else
     pass=pass-1
endif
pass=pass+1
return
enddo
*End of routing

*****************************************************************
*               SEARCH                  *
*****************************************************************
PROCEDURE SEARCH
set typeahead to 0
store evaluate1 to evaluation
store dbf1 to dbf
store 1 to dbpass
* PERFORM SEARCH WITH NON-MASTER STRINGS
DO WHILE .T.
*If the present search is for items from the source.dbf
*then turn on the recmemory to store the found records
*in a string named bigstr.  This string is used to apply
*the "flag" to the pollutant/condition display for pollutants
*that are associated with a chosen group,category, or material
*that is not listed on that screen
if dbf="SOURCE"
    recmemory=.t.
else
    recmemory=.f.
endif
set color to &bkwh
@ 22,51 say '                '
use &dbf
evaluation=upper(evaluation)
store 0 to location,c
store ltrim(str(int(reccount()))) to frac
* IF BIBLIO IS PRESENT DBF, CONDUCT STRING SEARCH
        if dbf="BIBLIO".and.z='AUTHORS'
        locate for string$authors
        store .f. to author
        else
* OTHERWISE SEARCH FOR ASSEMBLES NON-MASTER STRING
        locate for (&evaluation)
        endif
* DISPLAY LOCATION OF PROGRAM ON SCREEN AND APPEND VARIABLE
* XSTR WITH EACH FOUND RECORD'S BIBLIO_NO
public bigstr,bigstr2,bigstr3,bigstr4
STORE ' ' TO BIGSTR,bigstr2,bigstr3
STORE -1 TO OLD
set color to &rdwh
@ 21,1 clear to 21,49
set color to &bkwh
do while found()
    c=c+1
    call bell with "1,10000"
    @ 21,51 say 'Possible match '+ltrim(str(c))+' at: '
    @ 22,51 say ltrim(str(recno()))+' of '+frac
    store biblio_no to x
    store str(recno()) to Y
*NOTE: each 'if' statement below allows 40 finds.
*therefore these three will allow only 120 matches
*before an error statement
```

```
if recmemory.and. len(bigstr)<=240          &&(dbf=source)
        STORE stuff(bigstr,len(bigstr)+1,0,(substr(y,6))-',') TO bigstr
endif
if len(bigstr)>240.and.len(bigstr2)<=240.and.recmemory
        STORE stuff(bigstr2,len(bigstr2)+1,0,(substr(y,6))-',') TO bigstr2
endif
if len(bigstr2)>240.and.len(bigstr3)<=240.and.recmemory
    STORE stuff(bigstr3,len(bigstr3)+1,0,(substr(y,6))-',') TO bigstr3
endif
* MAKE STRING OF BIBLIOGRAPHY NUMBERS THAT MATCH,(w/o replications)
    IF X <> OLD
        STORE stuff(xstr,len(xstr)+1,0,(substr(str(x),8))-',') TO XSTR
        STORE X TO OLD
    ENDIF
*Eliminate biblio_no from further searches. This prevents duplicate
*bibliography numbers from being added to the evaluation string XSTR
IF .NOT. RECMEMORY
        do while biblio_no=x.and.found()
        continue
        enddo
ELSE
    CONTINUE
ENDIF
enddo
*DETERMINE HOW MANY MATCHES FOUND
PUBLIC MATCHED
*if substr(xstr,1,4)=' 0,'
*   xstr=stuff(xstr,1,4,'')
*endif
store len(xstr)/4 to matched
* STORE ASSEMBLED NON-MASTER STRINGS IN VARIABLE FOR DISPLAY
do case
* IF ONE DATABASE SEARCHED
    case dblook=1
        store xstr to timer1
        set typeahead to 20
        return
* IF TWO DATABASES SEARCHED
    case dblook=2
        do case
            case dbpass=1
                store xstr to timer1
* ALLOW FOR UNUSUAL ORDER OF ENTRY BY USER
                if dbf2=dbf1.and.dbf3 <> ' '
                    store dbf3 to dbf
                else
                    store dbf2 to dbf
                endif
                store evaluate2 to evaluation
            case dbpass=2
                store xstr to timer2
                set typeahead to 20
                return
        endcase
* IF THREE DATABASES SEARCHED
    case dblook=3
        do case
            case dbpass=1
*store results of search to timer.var
                store xstr to timer1
*prep dbf for next database
                store dbf2 to dbf
*prep evaluation string for second search
                store evaluate2 to evaluation
* SAME COMMENTS BELOW AS IN DBPASS=1 CASE
            case dbpass=2
                store xstr to timer2
                store dbf3 to dbf
                store evaluate3 to evaluation
            case dbpass=3
                store xstr to timer3
                set typeahead to 20
                return
        endcase
endcase
* INCREMENT PASS COUNTER
dbpass=dbpass+1
* BLANK XSTR IN PREPERATION OF NEXT NON-MASTER STRING SEARCH
store '' to xstr
enddo
set typeahead to 20
*End ofof search

*****************************************************************
*                    SUMMARY                    *
*****************************************************************
procedure summary
STORE .F. TO BOTT,DONTPRNT
store .t. to once
set color to &bkwh
clear
frow=5
```

```
lrow=19
DO KEYUP2
select 1   &&->poll1
goto ptc1   &&->initialized in recfind
do while .t.
IF DONTPRNT
ELSE
SET COLOR TO &bkwh
CLEAR
set color to &bwhbl
@ 1,0 say '<<< Pollutant='+trim(pollutant)+' >>>'
SET COLOR TO &blwh
@ 3,0 clear to 3,79
@ 3,0 say " Rec#  Material                    Measured Emission      Group"
@ 4,0 say ''
set heading off
set color to &blwh
CTR=0
p2=pollutant
b2=biblio_no
store recno() to toprec,OLDTOPREC
DO WHILE pollutant=p2.AND. CTR<15.
     store substr(str(recno()),5,10)+"  "+;
               substr(source->material,1,35)+;
               substr(str(measuremnt),1,10)+' '+units+'    '+;
               substr(source->group,1,10) to line
     @ row()+1,0 say line
     CTR=CTR+1
     SKIP
ENDDO
*draw box according to how many pollutants displayed
set color to &blwh
store row()+1 to rw
@ rw,0 to rw,79
@ rw,42 say chr(193)
@ rw,6 say chr(193)
@ rw,0 say chr(192)
@ rw,79 say chr(217)
@ 4,0 to 4,79
@ 4,0 say chr(218)
@ 4,79 say chr(191)
@ 5,0 to rw-1,0
@ 5,79 to rw-1,79
@ 4,42 say chr(194)
@ 4,6 say chr(194)
@ 5,6 to rw-1,6
@ 5,42 to rw-1,42
records=ctr

******
*user instructions go here
set color to &rdwh
@ 23,1 SAY 'F10'
@ 22,1 say ' F9'
@ 24,1 say "  R"
set color to &bkwh
@ 21,2 to 21,77
@ 23,7 say 'view only data for material similar to highlighted material'
@ 22,7 say 'view data for all materials, starting at highlight'
@ 24,7 say 'Return to previous screen'
endif
set color to &bkwh
if recno()>PTC1+15
     set color to &ylwh
     @ 5,73 say chr(24)+"PgUp"+chr(24)
     set color to &bkwh
endif
if pollutant=p2
     set color to &ylwh
     @ 19,73 say chr(25)+"PgDn"+chr(25)
     set color to &bkwh
endif
set color to &bkwh
i=0
do summbar with rw+1,toprec,i
do while i=0
     do while i=0
          i=inkey()
     enddo
     if i=35
     i=0
     do help with 5
     endif
enddo
*pages around in summary screen*
     do case
          case i=113.or.i=81
                    close all
                    return to master
          case i=1     &&HOME
               goto ptc1
          case i=6     &&END
```

```
            do while pollutant=p2
                skip
            enddo
*skip -15 so last record is not at top of page
            skip-15
            STORE .T. TO BOTT
            do while pollutant <> p2
                skip
            enddo
            @ 19,0
        case i=18   &&pgup
            wait 'skipping up 15 records'
            skip -(15+(row()-6))
            do while  pollutant <> p2
                skip
            ENDdo
goto ptcl
        case i=3   &&PGDN
                ct=1
                do while .not. eof().and.pollutant=p2.and.ct<9
                    ct=ct+1
                    skip
                enddo
                IF EOF()
                    GOTO BOTTOM
                    store .t. to bott
                ENDIF
                DO WHILE pollutant <> p2
                    SKIP-1
                    store .t. to bott
                ENDDO
                ct=0
**jump up 10 or top of pollutant, whichever 1st
                do while pollutant=p2.and.ct<10
                    skip-1
                    ct=ct+1
                enddo
                skip
    case i=5.or.i=24
        do summbar with rw+1,toprec,i
    case i=82.or.i=114
            write=.f.
            call savescr with 'r2'
            return
    case i=-9.or.i=-8  &&F10 or F9
            call savescr with 'x3'
            clear
            public all
            if i=-8
                store .t. to all
            else
                store .f. to all
            endif

            do keyup
            select 1
            do detail
            store .T. to once
            goto OLDTOPREC
            loop
    endcase
store recno() to toprec
enddo
return
*End ofSummary

*****************************************************************
*                    SUMMBAR                    *
*****************************************************************
PROCEDURE summbar
parameters lrow,rec,i
*menu selection from pollutant/emission screen
barpoint=1
lrow=lrow-1
set typeahead to 5
choscn=.f.
store 'chr(row)+chr(col)+chr(hi)+chr(wide)+chr(color)' to fill
load cfill
set index to
store '' to smg
store 6 to frow
if dontpent
    store row to row
else
    store 6 to row
endif
store 8 to col
store 1 to hi
store 35 to wide
*sets first bar to white/red
*cursor color
store 31 to color
```

```
PUBLIC POINTER
do while .t.
call cfill with &fill
POINTER=REC
do case  &&not used on first pass from calling pgm
      case i=24
          if row<lrow
               rowcount=rowcount+1
               rec=rec+1
               pointer=rec
*normal color
               color=113
               i=0
          endif
          loop
     case i=113.or.i=81   &&Qq
               close all
               return to master
     case i=5        &&-up
          if row>frow
               rec=rec-1
               pointer=rec
               color=113
               i=0
          else
     endif
          loop
      endcase
set talk off
i=0
ct=0
if pointer<>rec
@ 20,50 say pointer
@ 20,COL()+1 SAY REC
@ 21,0 say 'WARNING!! Program error... data may not match material ot reference'
endif
point=.t.
do while point
      point=.f.
      i=0
do while i=0
      do while i=0
           i=inkey()
      enddo
      if i=35
      i=0
      do help with 1
      endif
enddo
      if i=72.or.i=68
      store .t. to hlp5
      do helpext
      point=.t.
      loop
      endif
enddo
do case
     case i=24  &&->Dn
          i=0
          if row()<lrow
               pointer=rec+1
          endif
          if row<lrow
                    color=113
               endif
               call cfill with &fill
               row=row+1
               rec=rec+1
               color=31        &&white on red
               barpoint=barpoint+1
      case i=5
          i=0
          if row()>frow
               pointer=rec-1
          endif
          if row>frow
               color=113
               call cfill with &fill
               rec=rec-1
               barpoint=barpoint-1
*cursor
               color=31
               ROW=ROW-1
          else
*normal color
               color=113
               call cfill with &fill
*row=row-1
*cursor color
*color=12
          endif
     case i=113.or.i=81  &&Qq
```

```
            close all
            return to master
    case i=13
    return
    case i=114.or.i=82  &&NnRr
        store .t. to more
        return
    otherwise
        return
endcase
enddo
goto rec
RETURN
*End of Summbar

*************************************************************
*                  TIMER                    *
*************************************************************
PROCEDURE TIMER
*// TIMER:  Time module, calls locate.prg where searching
*         begins, returns here to collect ending timing
*         marks
* called from MAKESTRG
*
store .f. to test
store ' ' to timer1,timer2,timer3
set talk off
pass=1
set exact on
public xstr,matched,s1,s2,s3
store '' to xstr
store 1 to finished
store time() to starttime
**begin searches
do SEARCH
** searches finished
store time() to endtime
store val(substr(starttime,7,2)) to stsec
store val(substr(starttime,4,2)) to stmin
store val(substr(endtime,7,2)) to endsec
store val(substr(endtime,4,2)) to endmin
store .t. to again
if stsec<endsec
    stsec=stsec+60
    stmin=stmin-1
endif
SUBTOTSEC=STSEC-ENDSEC
TOTMIN=STMIN-ENDMIN
MINSECS=60*TOTMIN
TOTSEC=abs(MINSECS+SUBTOTSEC)
store LTRIM(STR(TOTSEC)) to sec
do cmaster
store .t. to display
return to master
*END OF TIMER

*************************************************************
*                  USERPIX                  *
*************************************************************
PROCEDURE USERPIX
*//USERPIX:  displays the users choices at the
*         bottom of the screen
* called from BAR.PRG
if .not. blank
if reentry.or.nochange
    nochange=.f.
    return
endif
do case .
    case pass=2
        set color to &bkwh
        @ 19,58 say "<          >"
        set color to &ylwh
        @ 19,60 say 'Selections'
        set color to &bkwh
        if z="AUTHORS".or.z='DATE'
            @ 20,51 say string
        else
            set color to &bkwh
                @ 20,51 say substr(&z1,1,27)
        endif
    case pass=3
        if z='AUTHORS'.or. z='DATE'
        @ 21,51 say substr(string,1,27)
    else
        set color to &bkwh
        @ 21,51 say substr(&z2,1,27)
    endif
    case pass=4
        if z="AUTHORS".or. z='DATE'
        @ 22,51 say substr(string,1,27)
        else
```

```
            set color to &bkwh
        @ 22,51 say substr(&z3,1,27)
        endif
endcase
set color to &bkwh
else
    store .f. to blank
endif           &&if not blank
return
*End of userplx

*************************************************************
*               UTILITY             *
*************************************************************
PROCEDURE UTILITY
**Utility
set color to &bwhbk
clear
public xmonitor
do while .t.
if iscolor()
    @ 2,5 say "Your computer video signal is color/graphics"
    @ 3,5 say "Are you using a color monitor?    (Y/N)"
    wait '' to xmonitor
else
endif
if xmonitor$'YyNnTtFf'
else
    loop
endif
do case
    case xmonitor$'YyTt'
        set color to w/n,bg/bg
        clear
        @ 2,5 say 'Some color displays do not distinguish between white and bright white.'
        set color to &bwhbk
        @ 4,5 say 'Is there an intensity difference between these two lines?  (Y/N)'
        wait '' to xwhite
        set color to bg/bg
        if xwhite$'YyTt'
            store "whiteplus" to displaytype
        else
            set color to &bkwh
            @ 6,5 say 'Your monitor/video card combination does not distinguish'
            @ 7,5 say 'between white and bright white.'
            store "whiteonly" to displaytype
            wait
        endif
endcase
set color to &bkwh
enddo
return
*End of utility
```

APPENDIX C
dMAPS USER'S MANUAL

dMAPS

a database of Micro-Environmental Air Pollution Sources

James C. Reaves

UNC School of Public Health
Chapel Hill, NC

December 18, 1987

for IBM AT or Compatable Computer

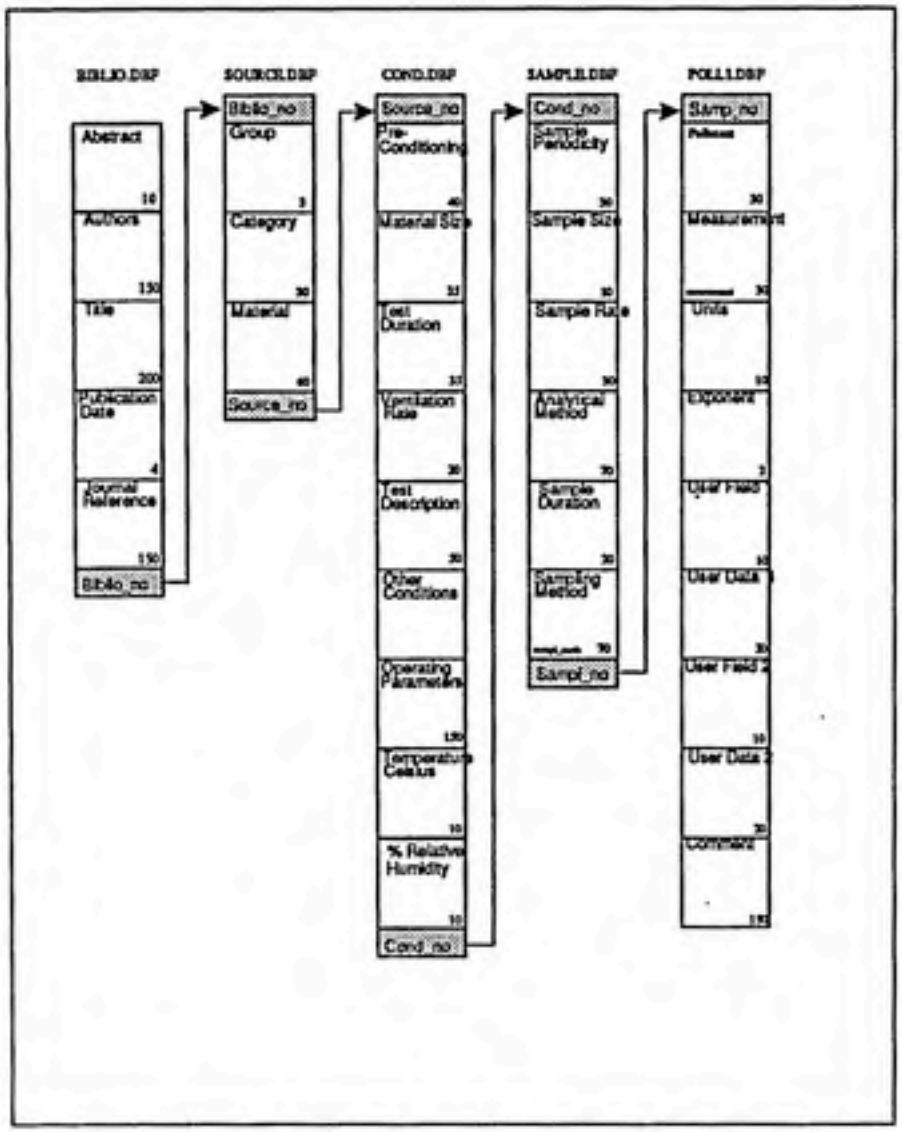


Figure D: Contents of the five dIAPS databases and their linking fields. Each shaded field at the bottom of each database column is unique in that database, but is not unique in the following database. For example, the Cond_no field in COND.DBF will have only one record with a value of 4, while the Cond_no field in the SAMPLE.DBF may contain many records with the value 4.
Numerals are the number of characters allowed in that field.

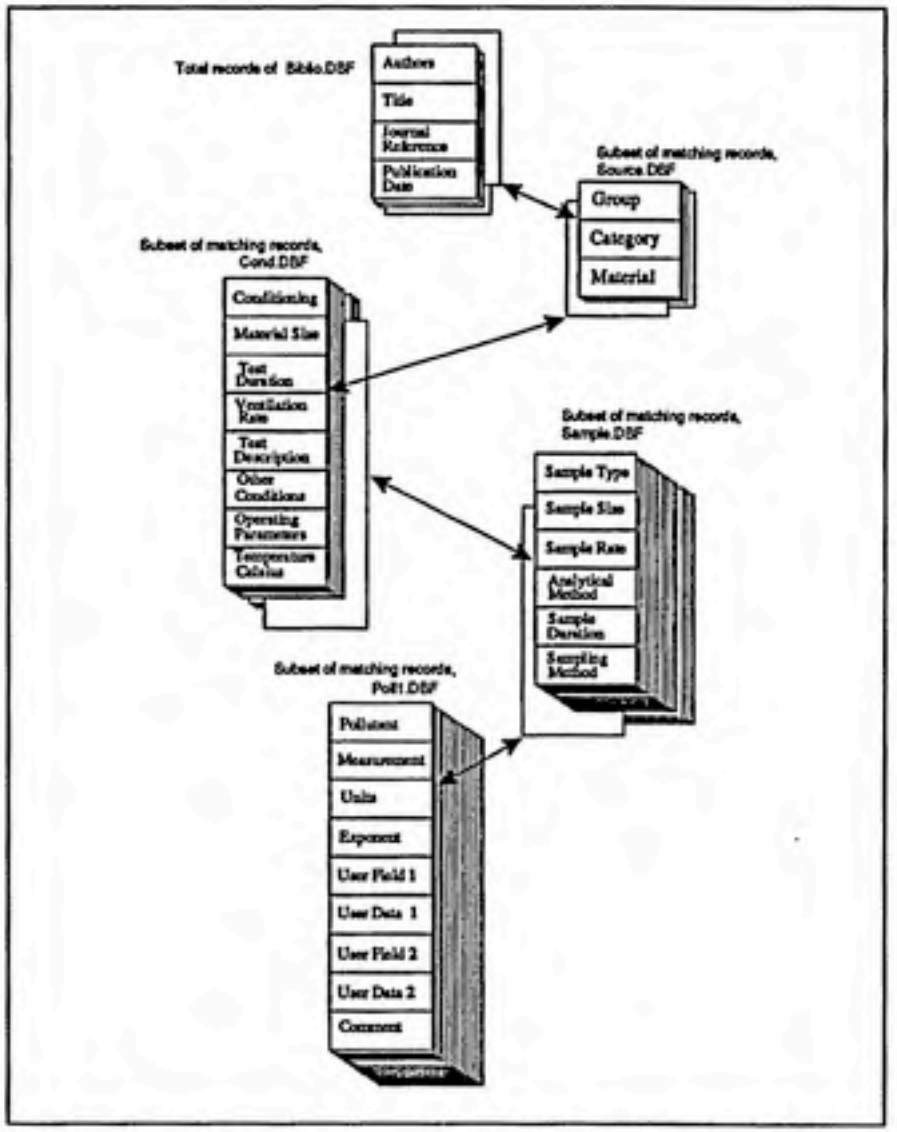


Figure E: Representation of the relationship between databases in dIAPS. Each specific item (record) in each database is associated to related records in the database immediately below it. This structure allows information to be related in data searches from general to specific; or once specific items are selected by the user, upward, from specific items to their gene characteristics.

Samp_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, POLL1.

Cond_no (4)

A field containing the condition number in the previous ranking database with which this record is associated. Many records in this database may share the same cond_no.

POLL1 DATABASE

This database contains information concerning the final test results, including measured emissions for every pollutant in the database. Also included here are the user defined labels and data fields, and specific comments concerning any individual pollutant and its test condition.

**Pollutant (30)*

This is the chemical name of the pollutant that was evaluated under the conditions of the experiment.

Measuremnt (30)

The air concentration of the pollutant emitted by the source material being tested.

Units (10)

The units of concentration of the pollutant measured as reported by the original article. This may be in PPM, Mg/m3, or any other appropriate units. If both CGS and English units are reported, CGS units are preferred.

Exponent (3)

When measured concentration is extremely large or extremely small, there may not be enough room in the measuremnt field to accommodate these numbers. If this is the case the exponent field may be employed to allow meaningful entries in the measurement field.

Comment (150)

This comment field is associated with each pollutant/measurement value and is available for detailed description of how this particular test or condition is unique.

Userdata1 (20)

This is one of two user definable fields, and is called the User Defined Label in the data entry screen. The user may enter any label into this field, and associate any data with it in the User1 field, which is called Data in the entry screen. Any type of information or comment concerning this pollutant and this condition may be entered in these fields.

Userdata2 (20)

Second of two user definable fields.

User1 (10)

The data field associated with the Userdata1 field which contains its label.

User2 (10)

The data field for the Userdata2 label.

Samp_no (4)

A field containing the sample number which associates records in this database with a particular samp_no from the previous ranking database, SAMPLE. Many records in this database may share the same samp_no.

COND DATABASE

Information concerning the experimental conditions are contained in this database. Either material condition or environmental condition factors effecting emissions from are in this database.

Pre_Condi (40)

Any pre-conditioning of the source material before the beginning of the test is entered here.

Test_dur (35)

The test duration is entered here. This is not the same as the sampl_dur field in the sample.dbf, which is the duration of the sampling activity, but is the duration of the entire test under any particular condition.

Test_desc (50)

A short verbal test description, including any information which may be useful to the dMAPS user.

Oper_par (37)

Any operating parameters, environmental or otherwise, which may have had an effect on the outcome of results of the test.

Temp_Deg_C (30)

The temperature in degrees Celsius under which the test was conducted.

Rel_humid (10)

The percent relative humidity under which the test was performed.

Vent_rate (20)

The ventilation rate of the material during the time the measurements were being taken, or during the time of the entire experiment, whichever is more appropriate.

Mat_size (35)

Material size, the surface area or any other measurement identifying the amount of material evaluated.

Other_Cond (150)

Any other operating conditions which may be of use in evaluating the data. Any special conditions of the test, special environmental conditions, or comments about the test are included in this field.

Cond_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, SAMPLE. See Figure D for the ranking of the databases.

Biblio_no (4)

A field containing the bibliography number from the previous ranking database with which the record is associated. Many records in this database may share the same biblio_no.

SAMPLE DATABASE FIELDS

This database holds information concerning sample collection procedures and analytic methods.

Sampl_Per (30)

This is the sample periodicity: Continuous, Periodic, Integrated, etc.

Sampl_Rate (30)

This field lists the sampling rate at which the pollutant was measured, such as liters/minute.

Sampl_Dur (20)

The sample duration, the length of time a sample was taken for evaluation of the emissions from a particular material/condition combination.

Sampl_Meth (70)

This is the method used to collect or measure the pollutant emitted by any particular source material. The sample method includes entries such as charcoal tube, direct reading colorimetric tubes, or impinger collection into distilled water.

Sampl_Size (10)

Sample size, the amount of sample taken for analytical measurement. This is not the statistical sample size, nor is it the size of the material from which pollutants were measured.

Analytic (70)

The analytic method used to determine the concentration of pollutant collected in the sample device. When direct reading instruments are used to determine the pollutant concentration, this field can be the same as the sample method or the method that the direct reading instrument uses to determine concentration.

DATABASE FIELD STRUCTURE AND CONTENTS

The following are descriptions of the data fields contained in the dMAPS program databases. The dMAPS program contains five databases, BIBLIO, SOURCE, COND, SAMPLE, AND POLL1. The numbers enclosed in parentheses after every field name indicate the number of spaces allocated in the database to that field. Field names which are prefaced by an asterisks are fields which may be used for searches of the data.

BIBLIO DATABASE

This database deals with the information concerning the published article in which the data was found. All data in the other databases ares associated with a parent article contained in this database.

Abstract (10)

This is a special field in dBASE III called a memo field. Employed in every BIBLIO record, it allows large descriptive narratives to be associated with each individual record with out taking up large amounts of storage space. In this case the field contains the abstract of the published article associated with the record being viewed. The abstract is physically located in the file BIBLIO.DBT, but is linked to the database via this field.

**Authors (150)*

This field holds the names of the authors of the article. Listed in the order in which they appeared on the original article heading, all authors are listed except when the length would exceed the 150 character maximum length.

Title (200)

The original title of the article, book, or publication as presented in the original publication.

**Date (4)*

The year the article was originally published.

Refer (150)

This field holds information concerning the reference publication from which the article was gathered. As well as the publication, other reference information is included in this field such as reference publication volume number and the pages the article occupies in the publication.

The previous four fields, when appended together by the dMAPS program, form a classical bibliographic reference: Authors, Title, Journal Reference, and Publication Date.

Biblio_no (4)

The number linking this database with the source database, and indirectly with all four other databases. Each bibliography record in the Biblio database receives a unique biblio_no which associates all data throughout the database with its original journal article. See Figure D for the ranking of the databases.

SOURCE DATABASE

This database has information concerning the indoor air pollution source classes.

**Group (15)*

This is the broadest division for indoor air pollution sources and generally includes such classifications as Water, Combustion, Consumer Products, and Structural Materials.

**Category (30)*

This classification is somewhat more specific than Group, but still is related to broad types of indoor air pollution sources. Types of entries in this area are Textiles, Insulating Materials, Plastics, and Applied Coatings.

**Material (60)*

This field is the classification that deals with the most specific sources of indoor air pollutants, and includes specific source materials such as Plywood, Paint, Carpet, and Cigarettes.

Source_no (4)

Unique for every record in this database, this field value associates each record in this database with several associated records in the next ranking database, COND. See Figure D for the ranking of the databases.

Biblio_no (4)

A field containing the bibliography number from which the record is associated. Many records in this database may share the same biblio_no.

If the logical operator is "AND" (A intersect B) the program determines which article ID numbers are common to both strings. This results in a String C which contains ID numbers for articles which have information concerning search criteria A and search criteria B. Comparison for A intersect B is performed in the COMPAND program.

When three different databases are searched, the process is nearly identical. In this case however, when string A is compared to string b the result is stored to a new string A. The string from the third database is compiled, stored to a new string B and then compared to A. When the program has searched through all the appropriate databases, the final stored string contains a list of all articles which contain information about the user's search criteria. Although this sounds complicated, it practice it is quite straight-forward as can be seen in Figure C.

The above searching method speeds operation of the search in two ways. First, skipping records is much faster than evaluating them to determine if they match a search criteria. Secondly, this method allows several databases to be searched and the logical relationships between the databases found without linking them on common key fields. Searches conducted on linked databases are the easiest way to search multiple databases in dBASE III Plus, but the method is very slow since the pointers in each linked database must be updated every time the pointer is moved in any of the databases. In dMAPS this would requires opening and closing of each of the five database search database in turn to move the pointer. Even though it is done automatically, the opening and closing of these databases takes considerable time in large databases.

In summary, in order to make searching of the dMAPS databases as fast as possible a dedicated searching program was developed to search each database separately, compile lists of matching records in each database, and then to logically compare these lists against each other. The result is a list of articles which contain information related in the manner the user specified in his or her search criteria. Although requiring several program modules in the dMAPS program, this method of searching provides an average searching speed sixteen times faster than the standard dBASE III Plus searching method. This considerable improvement is speed means that a search taking 37 seconds using the developed search procedure would take about 10 minutes using the standard dBASE III Plus method.

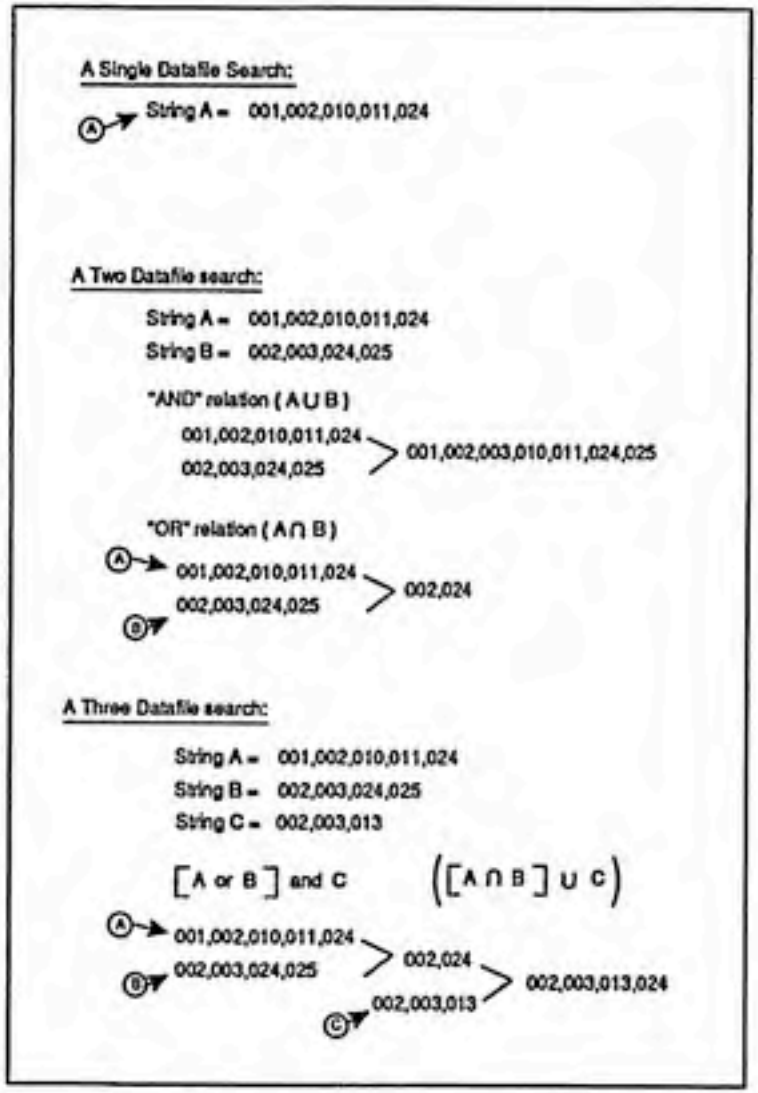


Figure C: The method by which the dMAPS program determines which datafile records meet the user's criteria. The three digit numbers represent the bibliography numbers which have information matching the individual search criteria. The matching articles are listed as strings A, B, and C which are compared to each other according to the logical relationships assigned by the program or the user.

DESCRIPTION OF PROGRAM SEARCH LOGIC

The dMAPS program allows searches of the database on seven key fields. These fields are *Group, Category, Material, Pollutant, Authors, Date of Publication, and Journal title*. The *Group, Category, and Material* fields allow searches for data associated with indoor air pollution sources, while the *Pollutant* field allows searches for specific pollutants. The *Authors, Date, and Journal* fields allow searches for data associated with specific published articles.

The program also allows any combination of these fields to be searched simultaneously. For example, the user may query the program to find all information concerning the emission of the pollutants *formaldehyde* and *hexane* from the source material *carpet*

There are five separate databases in the dMAPS data section, each related to the others through identifying fields (Figure B). Although this type of structure requires a complex data relation formula, it is necessary to use this technique to save disk storage space. A more detailed description of database relationships in dMAPS is given in the dMAPS Structure section.

Although there are five databases in dMAPS, searchable fields are found in only three of them. The *Author, Date, and Journal* fields are located in the BIBLIO database, *Pollutant* is located in the POLL1 database, and *Group, Category, and Material* are in the SOURCE database. Which databases are searched depends upon the user's search criteria. If the user picks only a pollutant, dMAPS will search only the POLL1 database, while if the user picks a date and a pollutant, the program will search the POLL1 database and then the BIBLIO database finding information meeting the user's search criteria. As matching information is found in each database the program assembles a character string identifying which published article is associated with the match. After the searches are complete these strings are then compared according to the logic operators the user has specified.

Single Database Searches

The internal programming algorithms used to search for user queries has been written to provide the speediest search possible. In order to provide this fast search, the program begins searching every record sequentially until a match is found, and then "remembers" the article from which the data came. The program then resumes the search. To speed the search however, the program now ignores all records associated with that article This is useful since once an article is identified as containing information of interest to the user it is redundant and time consuming to repetitively re-identify it. As other matches are found at other locations in the database, the new article ID number is also remembered and subsequently ignored.

At the end of the search, these "remembered" article numbers are used to allow the data display portion of the dMAPS program to quickly jump between sections of the database which contain information concerning the user's needs. Areas which are not associated with the user's search criteria are quickly skipped. For example, if article number ten was the only article which dealt with the user's query, then only the data derived from that article will be accessed.

Multiple Database Searches

If a user specifies two search fields that are in separate databases, the above procedure is carried out with each database. The result is two assembled strings, String A and String B. String A contains the ID numbers of the articles matching the first criterium, and String B containing the ID numbers of the second criterium. The program is then able to evaluate the interaction of these two strings via any logical operator the user has specified. See Figure C.

If the logical operator is "OR", as is the case when the user requests information concerning CARPET or FORMALDEHYDE, (A union B) String A is compared to String B and duplicate ID numbers are deleted. A third string, String C, is then created containing all of the ID numbers from both strings. This operation is performed in the program COMPOR.

```
    case ENT3_DB2.OR.ENT2_DB2
       dblook=2
  endcase
* BEGIN SEARCH SEQUENCE BY ASSEMBLING A SEARCH STRING
  DO makestrg
* START ACTUAL SEARCH WITH MASTER SEARCH STRING RETURNED TO HERE
if display
   CLOSE DATABASES
   do link
endif
enddo
enddo
```

```
*******************************
*       R,PRG
*******************************
**r.prg -----DISPLAYS LOGO SCREEN, INITIALIZES VARIABLES, SETS
**FUNCTION KEYS ---
set color to gr+/bg
set status off
clear
@ 12,30 say 'Loading dMAPS...'
SET ECHO OFF
SET TALK OFF
clear all
close all
set help off
set function 2 to 'call curson;'
set delimiters off
set procedure to &path.dmaps2.PRG
* HOUSEKEEPING
store 'e:\newdata\' to path
do colors
set safety off
save to color all like color*
set safety on
restore from color additive
LOAD CURSON
LOAD CURSOFF
LOAD SAVESCR
LOAD BELL
public localhlp,hlp1,hlp2,hlp3,hlp4,hlp5,help
store .f. to localhlp,hlp1,hlp2,hlp3,hlp4,hlp5,help
set typeahead to 20
set function 3 to 'set color to w+/w;call curson;modi comm '
set talk off
set status off
set scoreboard off
set echo off
public hlp,last,PICK,rectot,rec,author,dat,nochange,choice,TITLEVAR
store .f. to hlp,author,ref,dat,reentry,nochange,blank
public string,r,h,z,x,last,dbf,dbf1,dbf2,dbf3,timer1,timer2,timer3,display
set color to &blwh
clear
* BEGIN DISPLAY OF TITLE PAGE
set color to b/b &&bkwh
@ 3,6 clear to 19,72
SET COLOR TO &bkwh
@ 4,27 say ' d    M    A    P    S  '
set color to &bkwh
* INTERRUPT TITLE AND DO HOUSEKEEPING
store ' ' to
string,Z1,Z2,Z3,vqx,evaluation,authname,firstlet,timer1,timer2,timer3
* BEGIN TITLE AGAIN
@ 4,30 say '  '
@ 4,35 say '  '
@ 4,40 say '  '
```

```
@ 4,45 say '  '
@ 4,50 say ' '
*set color to &bkwh
@ 6 ,11 say ' Database of Micro-environmental Air Pollution Sources'
set color to bg+/b &&bkwh
* MORE INITIALIZATION
store ' ' to ch,bool1,bool2,h1,h2,h3,r1,r2,r3,modif,bracket
store 0 to pass,localpass
store 'default' to logic
set function 10 to 'F10;'
* MORE TITLE
@ 9 ,11 sAY 'The Department of Environmental Sciences and Engineering'
@ 12,11 say '                                for'
@ 10,11 say '  University of North Carolina School of Public Health'
@ 14,11 say '   The United States Environmental Protection Agency'
@ 15,11 say '         Research Triangle Park, North Carolina'
@ 16,11 say '                EPA Project CR813460-01-0'
D10=' F10 '
* MORE INITIALIZATION
set function 5 to 'call curson;'
store chr(17)+chr(196)+chr(217) to ret
store chr(218)+'                '+chr(191) to bracktop
store chr(192)+'                '+chr(217) to brackbot
public evaluate1,evaluate2,evaluate3
set color to &bkwh
@ 21,50 say ' Any key to continue...'
set color to &rdwh
@ 0,0 SAY 'Version: 0.97'
@ 22,1 to 24,79 double
set color to &bkwh
@ 23,4 say chr(17)+chr(17)+" Development Version -- NOT FOR DISTRIBUTION
"+chr(16)+chr(16)
wait ''
public ENT2_DB1,ent1_db1,ent2_db2,ent2_db3,ent3_db1,ent3_db2,ent3_db3
store .f. to display
store .f. to ent1_db1,ent2_db1,ent2_db2,ent3_db1,ent3_db2,ent3_db3
store .t. to again
* show main menu as next screen when starting new
do while again
pass=1
store ' ' to ch,bool1,bool2,h1,h2,h3,r1,r2,r3,modif,bracket
* BEGIN PROGRAM EXECUTION
do mainmenu
store .f. to again
* PROGRAM RETURNS HERE WHEM ALL EVALUATION CRITERIA SPECIFIED
do while .not. again  &&again given .t. value in TIMER
store ' ' to evaluate1,evaluate2,evaluate3
* SET VARIABLE DBLOOK DEPENDING ON LOCATION OF SEARCH RECORDS
public dblook
  do case
    case ENT1_DB1.OR.ENT2_DB1.OR.ENT3_DB1
       dblook=1
    case ENT3_DB3
       dblook=3
```

--Source also increased because Biblio_no changed and source is now chipboard

--Condition increased by 1 because the previous no.'s changed and the test conditions are different in this article

--Sample=13 due to change in numbers above, Sample=14 because CH2O has a different sample/analytical method than that for hexanol and iso-propanol (which have the same).

What would you do for the other brand of chipboard? Should you change the source number since it is a different manufacturing product? Or would it stay the same? The answer is that it would remain the same. No information under the Source section is changing: the group would remain Structure, the category would still be Wood product, and the material Chipboard. If a different brand name or other information was unique to this piece of chipboard the additional information could be entered in the parameter "Other Conditions"

Pollutants tested from chipboard emissions

Formaldehyde	Hexanol	propanol, iso-
Bib.=2	Bib.=2	Bib.=2
Source=3	Source=3	Source=3
Condition=8	Condition=8	Condition=8
Sample=15	Sample=16	Sample=16

--Condition no. increases because new information is placed in "Other Condition", an entry under the Cond section. Remember, any time an entry in any section is changed, the key number in that section must be changed, as well as all key numbers below it in the hierarchy.

The fundamental ideas to remember are to:

- o Obtain list of last Bibliography, Source, Condition, and Sample numbers used to enter data.
- o Assign the initial set of numbers to a new article by increasing the last entered set by one
- o Increase a section number when data within an article has changed
- o Observe the hierarchy of numbering; remember to increase those section numbers beneath the one you change additional comments about format. All existing database information has been entered using the format given in the example found in Data entry. Consistency is important if one wishes to compare different research efforts presented within the database's library. Strict adherence to the sequencing of numbers is paramount. Once the integrity of the Section numbers is compromised, the database looses its search foundation. Thoughtful preparation makes a sound base to work from.

This indoor air pollution database has been designed to allow the user to enter new data into the existing structure. Fundamentally, a researcher can update his own work or that of his colleagues to keep abreast of new information. With the ability to add new data comes the responsibility of entering the information in a manner consistent with the original data entry. Deviations from the structural format may corrupt the integrity of the database. The intention of this section has been is to guide the user through some data entry. Examples have been included to illustrate significant points.

Now let's look at what happens if we continue to input data for the medium fuel rate:

Pollutants tested using heater at medium fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=1	Bib.=1	Bib.=1
Source=1	Source=1	Source=1
Condition=2	Condition=2	Condition=2
Sample=3	Sample=3	Sample=4

–Bibliography is still 1 because article is the same

–Source also stayed the same

–Condition increased by 1 since we're at medium rate instead of low fuel rate

–Sample number increased to 3 because we increased Condition number (remember the hierarchy rule), and Sample=4 for CO because again, the difference in sampling and/or analytical methods.

The same logic would hold true for high fuel rate conditions:

Pollutants tested using heater at high fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=1	Bib.=1	Bib.=1
Source=1	Source=1	Source=1
Condition=3	Condition=3	Condition=3
Sample=5	Sample=5	Sample=6

We will continue this exercise by taking a look at what happens if the author reports emission data from a second source. We will assume that research was also done using the same three test conditions on a natural gas space heater:

Pollutant tested using a NG heater at low fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=1	Bib.=1	Bib.=1
Source=2	Source=2	Source=2
Condition=4	Condition=4	Condition=4
Sample=7	Sample=7	Sample=8

–Biblio_no=1, still the same article

–Source no. is increased by 1 because the source is now a natural gas heater instead of kerosene heater

–Condition number is increased because we changed Source no.

–Sample number also increases by 1 since Source and Condition number changed

Continue the logic by filling in the missing numbers:

Pollutant tested using a NG heater at medium fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=	Bib.=	Bib.=
Source=	Source=	Source=
Condition=	Condition=	Condition=
Sample=	Sample=	Sample=

Pollutant tested using a NG heater at high fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=	Bib.=	Bib.=1
Source=	Source=	Source=2
Condition=	Condition=	Condition=6
Sample=	Sample=	Sample=12

Now that you have mastered number assignment within an article, let's look at what would happen if you wanted to enter data from another article. The second author studied emissions from 2 brands of chipboard.

Pollutants tested from chipboard emissions

Formaldehdye	hexanol	propanol, iso-
Bib.=2	Bib.=2	Bib.=2
Source=3	Source=3	Source=3
Condition=7	Condition=7	Condition=7
Sample=13	Sample=14	Sample=14

–Bibliography number increased by 1 since this is a new article (Remember the hierarchy rule, all numbers beneath must also increase by 1)

Reference	APCF 99(3):233-240(vol.(no.):pages)
Group	Combustion
Category	Stove
Material	Natural gas
Pretest cond	Conditioned 12 days at ambient temp.
Test description	Chamber study, 34 m3
Test duration	24 hours
Operating parameter	4 burners operating at high fuel rate
Temperature Deg C	50
Relative humidity	30 %
Material size	12 cm x 34 cm x 56 cm
Ventilation rate	1.5 ACH
Other conditions	Source is an unvented natural gas stove. Fuel rate 35,000 L/hr at 50% efficiency.
Sample type	Continuous
Sample rate	1 L/hrSample duration 6 hours
Sample method	TEA coated acrylic tube
Analytical method	Modified parasoaniline
Pollutant	Nitrogen dioxide
Emission data	1.0
exponent	10 +7
units	ug/Kg
Comments:	Standard deviation for emission data = 300 +/- 50 ug/Kg, where N=69. Researcher reports NO2 concentration increases with increasing fuel rate.
User 1	(open for user to input data of his own, or put data in other standard units)
User 2	(Second user data field)

Data in the database are related by key numbers associated with the Bibliography, Source, Condition and Sample numbers used. These numbers are essential to the database in that they form the linking variables upon which the search procedures operate. Each unique article is assigned a consecutive unique Bibliography number. the Source, Condition and Sample numbers are also assigned consecutively within an article as any parameters within each of these three fields change. These four numbers (Biblio_no., Source no., Condition no. and Sample no. respectively) form a hierarchy within the database. Once one of these numbers is changed within an article, all numbers beneath in the hierarchy (and also beneath in the data entry screen) are increased by one. This is an important fact to keep in mind while organizing and inputting new data.

Data entry up to this point probably seems almost impossible to do successfully, but fear not, it is not really that complicated.

The first article entered into the database was given Bib=1,

the first source identified was assigned Source=1, the initial test conditions were given Cond=1, and logically, the first sample conditions used to obtain the first emission data reported was assigned Samp=1. As more data was entered from this first article, Source, Condition, and Sample numbers changed to reflect new entries. Bib did not change until a new article was begun, and then Bib became 2.

We will use the past example as an exercise in assigning section numbers.

The author of the first article reported source emissions from an unvented kerosene space heater using three different testing conditions (low, medium, and high fuel rates). The following numbers were assigned during data entry:

Pollutants tested using heater at low fuel rate

Nitrogen dioxide	Nitric oxide	Carbon monoxide
Bib.=1	Bib.=1	Bib.=1
Source=1	Source=1	Source=1
Condition=1	Condition=1	Condition=1
Sample=1	Sample=1	Sample=2

–Bibliography is 1 since this is the first article

–Source is 1 since this is the first source reported (unvented kerosene heater)

–Condition is 1 since it also is the first reported (low fuel rate)

–Sample number changes from 1 to 2 for CO because although not shown here, the article indicated that CO was sampled and/or analyzed in a manner different from NO2 and NO.

Entering and structuring data

by

Linda Mann

After selecting an article for entry into the database, highlight key points expressed within the document. Look at what information will be needed under each section (Bibliography, Source, Condition and Sample) to complete data entry. Color-coded highlighting of each parameter has been very helpful in reducing the time needed to organize and enter data.

One such organizational format is presented below with suggested information to be included within each section.

- General information
  - Description of test
    - -chamber study
    - -room or house study
    - Duration
    - Parameters tested
    - Test results,limitations,
      - and/or recommendations
  - Conditions
    - Pretest conditioning used
    - Conditions such as
      - -temperature
      - -relative humidity
      - -air exchange
      - -fuel type/loading rate
      - -other condition factors
  - Data values
    - Appropriate chart data
    - Data found in summary or
      - other text statements
  - Instrumentation
    - Type of sampling used
      - -continuous
      - -noncontinuous
      - -intermittent
    - Duration
    - Sampling method used
    - Number of samples taken
    - Analytical method used

The key to efficient and effective data entry is organization. Before entering any data, develop a plan of action. Examine the information the researcher presents in the article, and notice his research logic. Researchers may report emission data for a variety of sources under various conditions, and may indicate several different pollutants. Don't let the amount of data overwhelm you. The first article is usually the hardest to enter, but once you learn the ropes data entry becomes almost routine. Much thought has gone into the experiment; a little more time taken to examine the structure of the report can save a lot of time in entering the data.

If for example an author reports emission data for three source materials in the same article, enter all data for each source before starting the next. In a more complex article where five pollutants are reported using three test conditions of single source, enter all five pollutant data for one condition before beginning another test condition. In other words, don't enter all conditions for one pollutant and then repeat the process for the other four. This strategy not only saves time, but as you will see it also helps to keep the numbering sequence manageable (see data input for an explanation of the numbering hierarchy).)

Successful data entry relies on the user inputting appropriate text and data in a manner consistent with the format used initially in the preparation of the database. Deviations from the established guidelines could corrupt the integrity of the database's search capabilities. After that word of warning, the real art of data entry begins. The highlighting and organizational skills used in preparation of data entry are now tested - man is pitted against machine at last!

Begin with the first section, Bibliography, and proceed, referring to the sample data entry screen in this manual as necessary. Not all database parameters will be used in every article. For example, not all researchers report the relative humidity ofthe test conditions. Enter data for all applicable areas. The following illustration is presented as an example of what types of information go into each section. Section numbers, which are necessary for the database to keep track of the relationships between data entires, have been left out. An explanation of their meaning and assigned values is presented in Data Input. All information herein is fictional data, and does not reflect true research data, but is given purely to show entry format:

Author	Smith, J.D., M.D. Ames and C.F. Jones
Title	Indoor Air Pollution: A Research Effort
Date	1990

Data Entry

Data entry for dMAPS is achieved by entering data and bibliographic information from articles in a special module of the program. The information entered in this area is stored in a file that is not incorporated into the full dMAPS datafiles until the user requests it. This method speeds up data entry and makes the data more easily appended or changed.

The data entry module is reached from the main menu screen, while the command to update the dMAPS datafiles with the newly entered information is in the Utilities module. These two actions are separated in order to discourage the frequent updating of the main datafiles for just a few records. Frequent updating is undesirable since it takes time to order and index the newly updated datafiles, and also because it becomes more difficult to correct entry errors once the new data has been incorporated into the main datafiles

As can be seen in Figure A, the data entry screens are divided into sections with each section corresponding to one of the dMAPS datafiles. Although the user must keep track of the structure of the data entered into the program, effort has been made to make make this task as easy as possible. Repetitive typing has been minimized, and the input screens have been organized in such a way that user can most easily determine the datafile structure.

In order to do this the data entry screens have the entry blanks arranged in order of increasing specificity. The bibliographic information comes first since this will not change throughout the article being entered. The next section contains the entry blanks concerning the sources of pollutants covered in the article, while the next section contains information concerning the different conditions under which the sources were evaluated. The final two sections contain information about the sampling method for each of the source/condition relationships mentioned above and, finally, about the evaluated pollutant and associated test results.

This structure makes it relatively easy to enter large amounts of data for single articles. When the user first begins entry for an article all of the entry blanks are empty, and the user fills in all the appropriate blanks. The user then calls up another screen to fill in, but since much of the information will be identical the next screen appears with the new entry blanks filled with the old information from the previous screen. The user may then change only those items that require it. The first screen, completely filled in by the user, and every subsequent screen partially filled in by the user constitute one record each in the dMAPS datafiles.

The arrangement is such that if any item is changed only items in its immediate section or in the sections below it need be updated. For example, if an author conducted research on two pollutants under identical conditions and used identical sampling methods, only the pollutant and results and accompanying units would have to be changed, everything else would be the same. On the other hand, if the user began entering data from an article concerning a different source material, all entry blanks within and beneath the section containing MATERIAL would would be candidates for change. Again, no areas above it would have to be altered.

The hierarchical format of the data entry screens make it almost as fast to enter data for an article dealing with 10 pollutants as for one dealing with only one pollutant. Although it may be possible on future versions of dMAPS, the program is not now able to automatically determine the relationships between data in an article. This task must be done by the user. Again, this is made relatively simple by the hierarchical structure of the data input screen. When any item is changed, the corresponding key number in that section and in all sections below it must be increased by one. This technique relieves the user of most of the task of keeping track of the relationships between article data.

As the user enters data from articles, the data is stored in a temporary datafile called UPDATE.DBF. This file is stored and appended until the user determines that its information should be incorporated into the macin dMAPS datafiles. This makes it easy to stop in the middle of entering data for an article and to resume later. The program either picks up where the user last stopped, or presents a new blank screeen to begin another article. A series of questions at the beginning of the data entry screens determine which entry screen should appear.

dMAPS Structure

Data in the dMAPS programs are contained in five datafiles. Each of these datafiles holds data concerning a particular type of data. For example, the SOURCE.DBF datafile holds data concerning source materials which were evaluated for emissions, while the BIBLIO.DBF datafile holds data concerning the bibliographic information about the publication from which the data have been collected. The information in these five datafiles is related to the others by key fields.

Each datafile contains units called records. In the Biblio datafile each of these records contains several types of information (Authors, Journal, Article Title, and Publication date) about one particular publication. If information has been entered for forty different articles, there will be forty records, each having information about a different article. There is an additional field in each record, normally invisible to the user, which contains a unique identification number for each article, and so is therefore unique for each record.

Information about sources of indoor air pollutants are held in the Source datafile, just as bibliographic information is held in the Biblio datafile. However, each record in the Source datafile also contains a small piece of the Biblio datafile: the ID number of the article in which its data was found. Therefore, any published article listed in the Biblio datafile which deals with several source materials will be linked to only those records in the Source datafile which share its ID number. Conversely, any specific record in the Source datafile can immediately be associated with its parent record in the Biblio datafile due to their common ID number.

This type of structure means that for every record in the Source datafile it is possible to immediately have access to all of the information about its bibliographic origin held in the Biblio datafile. It is this type of ID number linking which will associate every piece of information in all five datafiles with the information about its publication article.

Since the dmaps database contains five datafiles, the relationship between them becomes more complex. Each of these datafiles is related only to the datafile above and below it. The Source datafile for example, is related to the Biblio datafile as described above, but is also related to the Cond datafile in exactly the same way that the Biblio datafile is related to the Source datafile. In other words, each record in any datafile is related to one and only one parent record in the datafile above it. In turn, that parent record is related to only one grandparent record in the datafile above it. This chaining continues until the topmost datafile, Biblio, is reached, which contains the most general information associated with any piece of data. This chaining allows every individual record in any of the datafiles to have constant access to all of the more general information concerning it which has been entered into the other dMAPS datafiles. Figure D is a graphical representation of the logical linking of the five datafiles, while Figure E displays the same relationship in a different manner. Figure E also lists the fields in the datafiles which hold the ID numbers, and indicates how they relate the datafiles to those above and below.

Even though it would be more direct and would result in simpler datafile structures and searching programs to put all the information in one big datafile where all information from one bibliographic article is constantly associated, it would take up far too much storage space to replicate all the bibliographic information for each of the different conditions or pollutants covered in all the articles. It is therefore advantageous to relate the datafiles to each other via common fields so that the information common to many pieces of data need be contained in only a few records, not several thousand.

There are five datafiles in the dMAPS program, each related to the datafile immediately above it in the datafile hierarchy in the same way that the Source datafile is related to the Biblio datafile. This relationship is used in two ways in the dMAPS program. First, this structure allows the dMAPS program to access any piece of data in any of the five datafiles dealing with any specific published article; and secondly, as any specific piece of data in any of the datafiles is accessed, the relationships may be traced upward through all the datafiles, ending in the Biblio datafile with the reference to the originating publication.

BIBLIOGRAPHY

1. Crum, Jane M.(1986) Indoor Air Pollution Source Database. Presented at 79th Annual APCA, Minneapolis, MN, June 22-27, (EPA-600/D-86-082:NTISPB86-191376).

2. Caceres, T, H. Soto, E. Lissi and R.(1983) Cisternas Indoor House Pollution Appliance Emissions and Indoor Ambient Concentrations *Atmospheric Environment*. 17(5):1009-1014

3. Dave, J.M.,(1984) Studies on Emission from Coal Burning Stoves as Used in Eastern India Indoor Air. 4:383-388

4. Fortmann, R.C., J.E. Borrazzo and I.D. Cliff,(1984) Characterization of Parameters Influencing Indoor Pollutant Concentrations Indoor Air. 4:259-264 *Proceedings of the 3rd International Conference on Indoor Air Quality and Climate*

5. Girman. J.R., J.R. Allen, M.G. Apte, V.M. Martin and G.W. Traynor,(1983) Pollutant Emission Rates from Unvented Gas-Fired Space Heaters: A Laboratory Study. *Lawrence Berkeley Laboratory*, University of California LBL-14502, UC-41

6. Girman, J.R., M.G. Apte, G.W. Traynor, J.R. Allen, and C.D. Hollowell,(1982) Pollutant Emission Rates from Indoor Combustion Appliances and Sidestream Cigarette Smoke. *Environment International* 8:213-221

7. Goto, Y. and G.T. Tamura,(1984) Measurement of Combustion Products from a Gas Cooking Stove in a Two-Story House. *Proceedings 77th Annual Meeting of the Air Pollution Control Association* 84-32.5

8. Hawthorne, A.R. and R.B. Gammage,(1982) Formaldehyde Release from Simulated Wall Panels Insulated with Urea Formaldehyde Foam Insulation. *Journal Air Pollution Control Association* 32(11):1126-1131

9. Hodgson, A.T., J.R. Girman, A.S. Newton and A. Winkes,(1983) Emissions of Volatile Organic Compounds from Architectual Materials with Indoor Applications. *University of California Berkely*, CA LBL-17130, EEB-Vent 83-21

10. Knight, C.V. and M.P. Humphreys,(1984) Indoor Air Quality Associated with Conventional and New Technology Wood Heaters in a Weatherized Home. *Proceedings the 77th Annual Meeting of the Air Pollution Control Association* June 24-29, 84-32.6

11. Leaderer, Brian P., (1982) Air Pollutant Emissions from Kerosene Space Heaters. *Science* 218(4577):1113-1115

12. Matsushita, H. and T. Mon,(1984) Nitrogen Dioxide and Nitrosamine Levels in Indoor Air and Sidestream Smoke of Cigarettes. *Indoor Air* 2:335-339

13. Matthews, T.G., A.R. Hawthorne, C.R. Daffron, M.D. Corey, T.J. Reed and J.M. Schrimsher, Formaldehyde Surface Emission Monitor. *Journal of Analytical Chemistry*

14. Matthews, T.G., A.R. Hawthorne, C.R. Daffron, T.J. Reed and M.D. Corey(1981) , Formaldehyde Release from Press-Wood Products. *17th International Washington State University Particleboard/Composite Materials Symposium*

15. Matthews, T.G. R.R. Westley, Determination of Formaldehyde Emission Levels from Ceiling Tiles and Fibrous Glass Insulation Products: Project Report. *U.S. Consumer Product Safety Commission* CPSC-IAG-82-1181

16. Matthews, T.G., T.J. Reed, B.J. Tromberg, C.R. Daffron and A.R. Hawthorne(1984) , Formaldehyde Emissions from Consumer and Construction Products: Potential Impact on Indoor Formaldehyde Concentrations. *Proceedings of the 3rd International Conference on Indoor Air Quality and Climate* 2:115-120.

17. Matthews, T.G., T.J. Reed, C.R. Daffron and A.R. Hawthorne(1984) , Environmental Dependence of Formaldehyde Emissions from Pressed-Wood Products: Experimental Studies and Modeling Proceedings. *18th International Washington State University Particleboard/Composite Materials Symposium*

18. Molhave, L., I. Andersen, G.R. Lundquvist and O. Nielsen(1982) , Gas Emission from Building Materials: Occurance and Hygienic Standard. *Danish Building Research Institute (SBI)* Report No. 137

19. Molhave, L., P. Bisgaard and S. Dueholm(1983) , A Mathematical Model of Indoor Air Pollution due to Formaldehyde from Urea Formaldehyde Glued Particleboards. *Atmospheric Environment* 17(10):2105-2108

20. Moschandreas, D.J., S. Relwani, H.J. O'Neill, J.T. Cole, R.H. Elkins and R.A. Macriss(1985) , Characterization of Emission Sources Into the Indoor Enivronment. Final Report March 1982-1985 *Gas Research Institute* Contract No. 5081-251-0575

21. Myers, G.M.(1983) , Formaldehyde Emission from Particleboard and Plywood Paneling: Measurement, Mechanism, and Product Standards. *Forest Products Journal* 33(5):27-37

22. Nelms, Leonard H., Mark A. Mason and Bruce A. Tichenor(1986), The Effects of Ventilation Rates and Product Loading on Organic Emission Rates from Particleboard *Proceedings. ASHRAE Indoor Air Quality Meeting Atlanta, GA April*

23. Pickrell, J.A., B.V. Mokler, L.C. Griffis, C.H. Hobbs and I. Bathija(1983) , Formaldehyde Release Rate Coefficients from Selected Consumer Products. *Enivronmental Sciences and Technology* 17(12):753-757

24. Pickerell, J.A., Larry C. Griffis, Brian V. Mokler, George H. Kanapilly and C.H. Hobbs(1984) , Foramldehyde Release from Selected Consumer Products: Influence of Chamber Loading, Multiple Products, Relative Humidity, and Temperature. *Environmental Science and Technology* 18(9)682-686

25. Porter, Warren K., Pollutant Emissions from Kerosene Heaters and Unvented Gas Space Heaters. U.S. *Consumer Product Safety Comission*

26. Ritchie, I.M. and L.A. Oatman(1983) Residential Air Pollution from Kerosene Heaters. *J Air Pollution Control Association* 33(9):879-881

27. Singh, J., R. Walcott and C. St. Pierre(1982) , Evaluation of the Relationship Between Formaldehyde Emissions from Particleboard Mobile Home Decking and Hardwood Plywood Wall Paneling.

28. Skiest, E.N (1981) Product Standards: A Means to Control Indoor Ambient Levels. 15th Washington State University International Symposium on Particleboard

29. Suchomel, F.H.(1983) , Gas Appliance Technology Center Annual Report. *Gas Appliance Technology Center, Gas Research Institute Chicago,* IL Contract No. 5082-241-0772

30. Traynor, G.W., D.W. Anthon and C.D. Hollowell(1982) , Technique for Determining Pollutant Emissions from a Gas Fired Range. *Atmospheric Environment* 16(12):2979-2988

31. Traynor, G.W., J.R. Allen, M.G. Apte, A.R. Carruthers, J.F. Dillworth, J.R. Girman, and V.M. Martin(1983) , Indoor Air Pollution Due to Emission from Unvented Gas-Fired Space Heaters: A Controlled Field Study. *Lawrence Berkeley Laboratory University of California,* Berkeley Building and Ventilation and Indoor Air Quality Program Project Report # LBL-15130

32. Traynor, G.W., J.R. Allen, M.G. Apte, J.F. Dillworth, J.R. Girman, C.D. Hollowell and J.F. Koonce(1982) , Indoor Air Pollution from Portable Kerosene-Fired Space Heaters, Wood-Burning Stoves and Wood-Burning Furnaces. *Proceedings Speciality Conference on Residentail Wood and Coal Combustion March 1-2, 1982*

33. Traynor, J.R., M.G. Apte, A.R. Carruthers, J.F. Dillworth, D.T. Grimsrud and L.A. Gundel(1982), Indoor Air Pollution due to Emissions from Wood Burning Stoves. *Proceedings from the 77th Annual Meeting fo the Air Pollution Control Association* San Francisco, CA Junw 24-29, . 84-33.4

34. Traynor, J.R., M.G. Apte, J.F. Dillworth, J.R. Girman, C.D. Hollowell and J.F. Koonce(1982) , Indoor Air Pollution from Portable Kerosene-Fired Space Heaters, Wood-Burning Stoves, and Wood-Burning Furnaces. Presented at the *Speciality Conference on Residential Wood and Coal Combustion March 1-2, 1982* Louisville, KY

35. Tu, K.W. and L.E. Hinchliffe(1983) , A Study of Particulate Emissions from Portable Space Heaters. *American Industrial Hygiene Association Journal* 44(11):857-862

36. Woodring, J.L., T.L. Duffy, J.T. Davis and R.R. Bechtold(1983) , Measurements of Emission Factors of Kerosene Heaters. Argonne National Laboratory Contract with U.S *Department of Energy,* Contract Number W-31-109-ENG-38

37. Yamanaka, Shin'ichi, Hiroshi Hirose and Takada Susumu(1979) , Nitrogen Oxides Emissions from Domestic Kerosene-Fired and Gas-Fired Appliances. *Atmospheric Environment* 13(3):407-412

38. Zawachi, T.S., J.T. Cole, V. Huang, H. Banasiuk and R.A. Macriss(1984) , Efficiency and Emissions Improvement of Gas-Fired Space Heaters Final Report. (Feb.-Aug. 1983) *Institute of Gas Technology* Chicago, IL Contract Number GRI-5182-241-1763