MEDICAL SYSTEMS DIVISION

Department of Pathology

College of Medicine

University of Florida

Gainesville, Florida

(NASA-CR-169018) THE CLINICAL PRACTICE LIBRARY OF MEDICINE (CPLM): AN ON-LINE BIOMEDICAL COMPUTER LIBRARY. SYSTEM DOCUMENTATION Final Report (Florida Univ.) 131 p HC A07/MF A01 CSCI 06E G3/52

N82-26964

Unclas 28084

The Clinical Practice Library of Medicine (CPLM): An On-Line Biomedical Computer Library.

System Documentation

Development partially funded by
NASA Grant #NAG10-0004

Principal Investigator:	Ralph R. Grams, M.D.
Project Manager:	James K. Massey M.E.E.
Project Staff:	Diane Nelson
Consultant:	William Ingram, Ph.D.

The enclosed documentation as of Version 2.0, 06 June 1982 constitutes the final report for NASA Grant #NAG10-0004.

Distribution: Original and one copy to project manager

Two copies to NASA project staff

UPDATE HISTORY

Last Major Update: Version 2.0 06 June 1982

Interim Updates:

Acknowledgements

Software used in this project was run on equipment of the Northeast Regional Data Center (NERDC) of the University of Florida.

TABLE OF CONTENTS

CHAPTER 1
The Clinical Practice Library of Medicine

1.0 Introduction

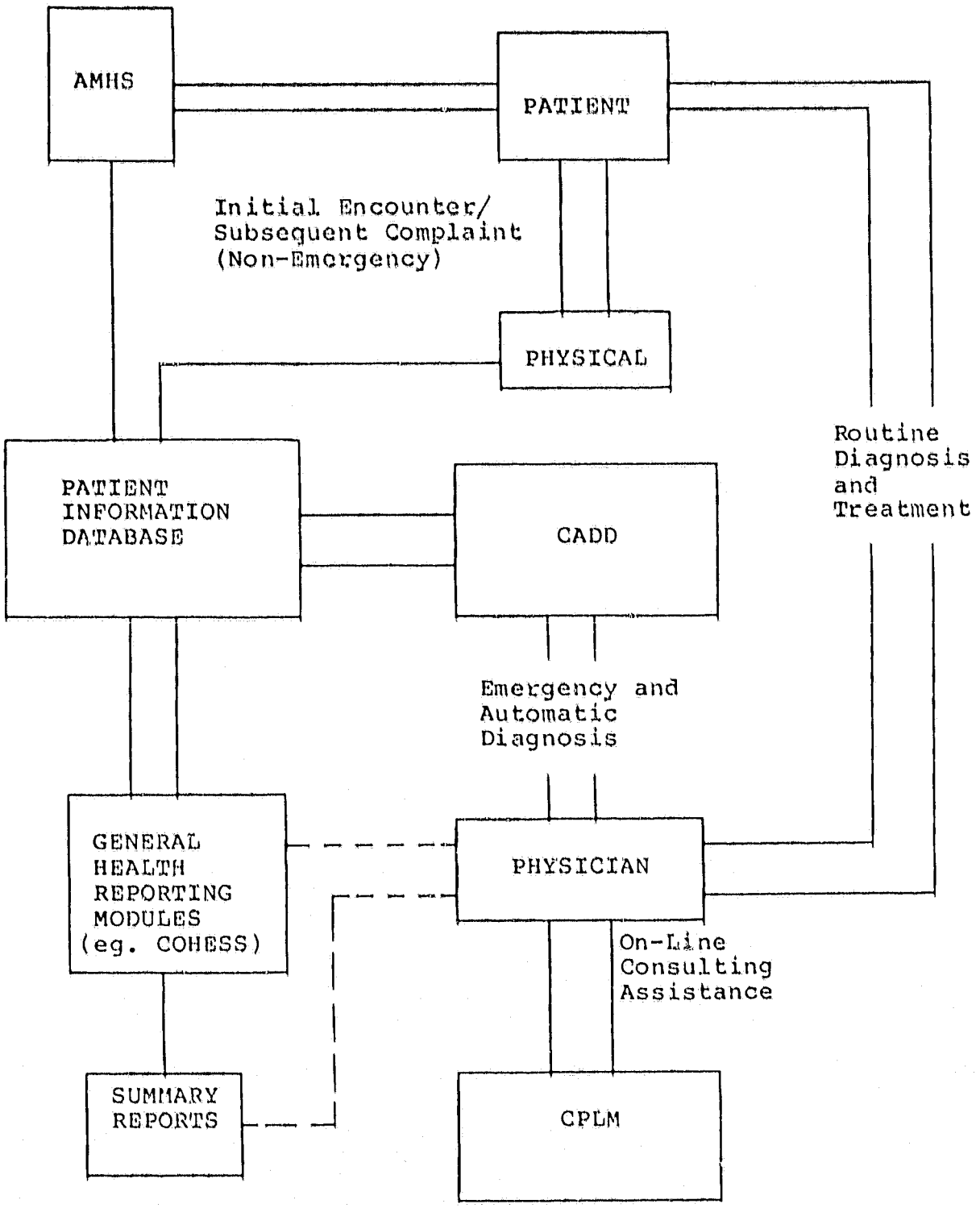
The Clinical Practice Library of Medicine (CPLM) is an investigational project aimed at providing the physician with critical in-depth information similar to that obtained from a medical reference library or consultant. When used in conjunction with the physician's knowledge, the CPLM can provide valuable background information to assist the physician in rapidly reaching a suitable diagnosis so that he may prescribe appropriate treatment.

In order to provide maximum services to his patient population as a whole, and especially to select optimum recovery paths for individual presentations, today's physician is expected to spend a great deal of time reviewing, selecting, reading, and understanding modern medical practice. The tremendous deluge of information with which he is presented makes this process at best a never ending (and highly frustrating) battle; and at worst, results in sub-optimal patient care.[1] Additionally, many emerging technological advances utilizing computers in medicine provide data in such profusion and detail that the physician requires assistance in obtaining the best available information, especially when confronted with a disease with which he is unfamiliar. Figure one illustrates an integrated patient care system which provides direct access for the physician to current online data and the resources to manage and utilize it. The CPLM is an important element of such a system. The roles of the various other elements in the figure have been previously discussed.[2,3]

CPLM in its current form contains a moderately large database of information on clinical pathology laboratory tests and a sampling of infectious disease information. In its maximally expanded form, CPLM would include access to a large and comprehensive base of medical knowledge. CPLM contains facilities to enable this base of knowledge to be rapidly searched to determine all available information as accessed by simple keyword searches (synonyms, etc.). In addition, CPLM is programmed in a high level database language (INQUIRE®) to allow investigation of "most effective presentation" methods.

®Inquire is a registered trademark of INFODATA Incorporated describing their high level database management systems.

Figure 1. Diagram of Kennedy Space Center's envisioned comprehensive computer supported health care system.



1.1 Background

The traditional application of computers as applied to the medical database includes the following areas: Computer Assisted Instruction (CAI)[4-10], automated medical records[11-13], diagnostic information[14-16], and computerized management systems[17-19]. Many articles describe citation and abstracting services based on the primary source document.[20-23] Additional data is available regarding data base compression, file structure, and dictionary type retrieval.[24-26] Although much effort has been invested in the development of bibliographic retrieval systems, it is the combination of skills in the above areas plus the addition of a practical patient care approach that makes the CPLM a unique entity.

In 1972, a detailed design was published postulating the development of an online biomedical library.[27] Since that time, additional effort has been put forth on an internal basis at the University of Florida to coalesce many of the ideas into a working, viable prototype.[28,29] The library was initially structured around a standard protocol and applied to clinical pathology. Since this early development, the software has been improved and drastically modified for speed, cost, and database efficiency. The initial protocols have been evaluated by over twenty medical participants in the library system. Feedback from these participants has resulted in many changes culminating in the present version of the system.

A recent article from the Lister Hill Center at the National Library of Medicine confirms that work has been done intramurally on automating a library system based on a disease profile and is consistent with our original design criteria as established in 1972.[30]

1.2 System Motivation

For today's practicing physician, be he on the space shuttle, in his own office, or at an outpatient clinic, the major share of information that is used for clinical judgement is retained in the memory of the individual physician or nurse. We rely extensively upon this on-board data bank for many of the routine procedures which are conducted. In situations of stress and complexity, it is expected that the physician will make the best approximation, since other alternatives for data retrieval are extremely cumbersome, slow, and unproductive. Practicing physicians know very well that it is physically impossible to maximally utilize our current biomedical library facilities and still maintain an active practice. Problems must be solved at the time they occur and in the framework of the patient's care program if our new knowledge is to be of any utility. Postponing decisions or waiting

for further information from a remote source is not compatible with an active medical practitioner.

In addition, our current base of information is not practice oriented, but is more disease and incident oriented, such that a physician has great difficulty retrieving specific pieces of information relevant to patient care without an extensive library search.

Individual textbooks contain only pieces of the whole database which may be required during the active care of any patient. To fully support the information needs of an active practitioner requires extensive library texts along with numerous journals and a mechanism of search and retrieval. In the past, we have used MEDLARS and now MEDLINE for a search on key words. Here, the physician must pick a small number of common key words and be ready to review anywhere from 1 to 3,000 separate citations in order to find a particular piece of information. Such a mechanism of retrieval is extremely painful and unproductive for those with clinical responsibility.

1.3 System Software Requirements

During the past few years, increasing emphasis has been placed on the development and utilization of methodologies and techniques for designing, implementing, and maintaining a software system.[31-34] The first goal of software design is reliability. Following the attainment of that goal, one strives for maintainability, efficiency, flexibility and generality. The key to software design quality is design clarity. This clarity is reflected in the functional requirements and the degree to which the design specifications are mirrored in the source code.

Design clarity is enhanced by functional modularity of the software system. This is accomplished by decomposing a system into distinct program modules that communicate through well defined interfaces. Each module is then identified with a specific system function. The three design modules of CPLM are: 1) the database scan/builder, 2) the database loader, and 3) the on-line user interface.

The software construction of the CPLM has involved a migration of previous efforts into a new language and then enhancements of the basic package. The software design for CPLM is divided into two main areas:

- Database maintenance and
- Inquiry

Database maintenance: This is a batch oriented system that accepts the textual data as input and performs the following steps:

- Updates the keyword and reference index files
  . creates keyword references automatically based on the outline structure of the document
  . updates the keyword pointer files of the database

- Updates the indexed textual files
  . structures the textual data into database pages
  . builds the keys and pointers that create the logical scheme of the database
  . updates the file
- Report on operations performed
- Backup files to tape

Database Inquiry: This is an interactive system that performs the following functions:

- Receives input commands from the user.
- Interprets these commands into actions that must be performed.
- When requested, it interrogates the database via key words, building up sets of pointers into the textfile. These sets may be combined logically with boolean operators (and, or, not) to create new sets.
- Upon request, it will display portions of the created sets.

The database structure selected for the CPLM is an important consideration. We have already performed considerable research into the optimal design for the CPLM database.

The problem we are addressing involves searching textual data in a conversational, time shared environment. To achieve the necessary response times, it is essential to be able to search the files without following chains or rings embedded in data on electromechanical storage.[35,36] The solution to this problem is to use an inverted file system. Thus, the database as currently structured consists of three files:

- A text file containing the actual data of interest; the medical text information.
- A keyword file that contains all pointers into the text file.
- A reference file that contains all the references for each document in the text file.

The INQUIRE database management system is used to manage all of these elements.

CHAPTER 2
Overview of The System

2.0 System Philosophy

CPLM is a system designed to access a large range of available medical information in an on-line interactive fashion. This is done by using a very high level query type database manager (INQUIRE) and results in a system that provides a broad spectrum of medical textbook data immediately available to the physician. The system is based on the presentation of text in a standardized outline format. Text prepared in the outline form can be scanned by a pre-processor. The result of this scan is a database which is then formatted and loaded by INQUIRE. On-line access is currently provided through IBM TSO facility.

2.1 System Design

The System is designed in three modules:

1) Database scan/builder
2) Database loader
3) On-line inquiry facility

The database scan/builder consists of a PL/1 program which scans data presented in a pre-defined outline format and creates an INQUIRE Standard Input (ISI) Format bibliographic database (See Appendix D for program listing). Appendix A shows the predefined outline for Clinical Pathology Laboratory Information and the Infectious Disease Information.

Text in this format is scanned by the program PARSE to generate the ISI format file. In addition, PARSE reorganizes the input outline to the presentation outline form. Appendix B illustrates the presentation outline form in skeleton format. All items below a major field title on the presentation outline remain in the same order.

The database loader consists of a set of INQUIRE provided utilities discussed in chapter VIII which takes the ISI format input and generates the loaded CPLM database.

The on-line inquiry facility consists of a set of interactive statements written in the INQUIRE macro language. These macros allow the user to obtain desired information by entering requests in a simplified user oriented format.

CHAPTER 3
Special Considerations

3.0 Extent of Knowledge Base

CPLM currently contains a large amount of information on Clinical Laboratory Information with some additional information on Infectious Diseases which has been added for testing purposes. Additional information is continually being added but extensive typing and editing staff will be needed to bring the overall database up to its maximum potential.

3.1 Validity of The database

Since entering the data into CPLM, none of the data has been throughly verified by competent medical authority. The data is, however, accurate in as much as it was extracted from recognized medical text literature. Before final use could be made of this data in a clinical setting some form of medical peer review would have to be performed.

3.2 Dynamic Nature of the Database and On-line Inquiry Facility

Since CPLM is an prototype system, elements of the database and the user interface commands are continually undergoing change. Should questions arise concerning user commands or modifications of the data, please call a member of the Medical Systems Division Staff (904-392-4571).

CHAPTER 4
Operating Instructions

4.0 Initiating Communications

CPLM consists of a set of programs which run at the Northeast Regional Data Center (NERDC) at the University of Florida, Gainesville, FL. In order to operate these programs, the user requires a modem, a computer terminal (either printer or CRT), and a NERDC account number.

4.0.1 Modem/Terminal Requirements

NERDC currently supports dial-up compatibilities with Bell 103 (300 baud) and Bell 212A (1200 baud) Series, asynchronous datasets (modems). Access by the remote user may be made via accoustic modems or via Direct Access Arrangement (DAA) modems. The telephone numbers are:

300 baud	904-392-5311
1200 baud	904-392-4727

The computer terminal must be configured at the appropriate baud rate with the following characteristics:

ADM3-A (300 or 1200 baud)

```
CODE              = ASCII (Upper/lower case optional)
NUMBER DATA BITS  = 7
NUMBER STOP BITS  = 1
PARITY            = Even
HALF DUPLEX
```

TI SILENT 700 (300 baud)

```
HALF DUPLEX
```

DEC WT/78 (1200 baud)

```
CODE              ASCII
NUMBER DATA BITS  = 8     |  7     |  7
NUMBER STOP BITS  = 1     |  2     |  1
PARITY            = NONE  |  NONE  |  MARK
BUFF CONTROL      = NO
OPTIONS           = KH HS KS (will not support printer
                              at 1200 baud)
```

When the telephone connection is completed (carrier present) the first transmitted characters must be:

```
at 300 baud   =  PP<CR><CR>* (P)
at 1200  baud =  p<CR><CR> (p)
```

*NOTE: <CR> is ASCII carriage return (hex 0A)

Following the second <CR> the terminal should respond with:

Enter t for TCP, c for CICS, m for MUSIC, l for APL, o for TSO

If the terminal does not respond in a reasonable time (15 seconds) then hang up and try the connection again. When the above message is received the user will be connected to NERDC and ready to sign on.

4.0.2 NERDC Accounts

Before executing any programs at the NERDC, a user must obtain an account number and password. NERDC requires account numbers to identify users for billing purposes. The password acts as the user's security against unauthorized use of the account. Signon procedures will require the account number (ACCNT#), sequence number (SEQ#), and password (PASSWORD) to be entered.

4.0.3 Starting CPLM

Once the user is connected to NERDC the following steps must be executed in order to enter CPLM:

1) Enter -- ologonACCNT#,SEQ#<CR>

2) The computer will respond with:

   NAC0006 ENTER PASSWORD -
   MMMMMMMM

   Enter -- PASSWORD<CR>

   The computer should respond with several sign-on message lines ending with:

   READY<CR>

   If the computer system is particularly busy, it may take as long as 2 minutes for the READY message to appear. If the response time is too slow, the user may disconnect at this point by merely hanging up the telephone.

3) Enter -- exec A0081909(CPLM)<CR>
   The computer will respond with several INQUIRE activation lines ending with:

   ENTER COMMAND>

The user is now in CPLM and may execute any of the commands described in section 4.1 of this chapter.

Due to the fact that this system runs on a computer in a university environment, on-line response time is quite variable. During certain times of the day response times will generally may deteriorate beyond acceptable limits. Best response times in the daytime will occur prior to 9:00 a.m. and after 9:00 p.m. At other times during the day periodic lulls may occur but these cannot be predicted. Response time gets worse near the end of the school term (final project time) and is considerably better during school breaks.

Several things are being done to remedy this problem. First a double rate premium may be requested which will assure maximum possible response from the NERDC. This is done at sign-on by specifying SID after the SEQ# of the logon line. Additionally the NERDC is in the process of migrating a large part of their interactive processing to a new to a new computer system. This should free up some interactive resources resulting in better response time on throughout the day.

4 Problems with signing on

A schedule of NERDC normal operating times may be found in appendix F. If the user has difficulty signing on during normal operating hours, then the computer system may be down. To confirm this the user may call a status hot-line at 904-392-6775. For all problems not associated with computer down time, please contact a member of the Medical Systems Division staff (904-392-4571).

4.0.5 Signing Off

When the user is through with CPLM he will generally issue the END command (See Section 4.1.5). This will return him to normal TSO (the READY message) where he should type: LOGOFF<CR>. This will disconnect the user and halt accounting charges being generated during normal connections. If for any reason a user cannot sign off in this fashion he should immediately dial NERDC operations (904-392-2291) on another line and ask them to dump TSO terminal number @4Z85#F for SEQ#=1. Failure to follow this procedure can lead to large amounts of charges being generated with no useful computing going on.

4.1 CPLM On-line Inquiry Facility Commands

The CPLM on-line inquiry facility consists of a set of programs that allow user input commands to be interpreted by the INQUIRE macro language interpreter. The basic structure of all CPLM commands is: [command verb] [operands]

where [command verb] is one of the following:

COUNT, DICT, END, FREE, HELP, HISTORY, OUTLINE, PRINT, SEARCH, SET, SHOW, SUMMARIZE, or USER.

The following sections describe in detail the function and syntax of each of these commands and their associate operands.

4.1.1 Command: COUNT

Function: Identifies and counts documents which satisfy a logical condition and/or previous search criteria.

Syntax: COUNT [Set Number] IF [Fieldname] [Relator] [Value] [Boolean] [Repeat]
Words in brackets are user-supplied variables. The entire command must reside on a single line.

Required: [Set Number] or IF [Fieldname] [Relator] [Value] NOTE: a null [Set Number] the account applies to the entire CPLM database.

Operands:

Set Number - the set number of a temporary database created by a previous SEARCH or USER command.

IF - begins a qualifying condition. The full condition must consist of: IF [Fieldname] [Relator] [Value]

Fieldname - a field from the document database.

Relator - indicates field-to-value comparison, as follows:

'GT'	greater than
'LT'	less than
'GE'	greater than or equal to
'LE'	less than or equal to
'EQ'	equals
'NE'	not equal to
'IS'	field begins with specified value
'NOT'	field does not begin with specified value
'CONTAINS'	value is found anywhere in field
'EXCLUDES'	value is not found anywhere in field

Value - one or more values against which field is compared. Multiple values must be separated by commas and enclosed in parentheses. A range may be specified with the 'IS' relator and 'TO' connecting the upper and lower values in the range. The 'SET,COMMAS,ON' option must be in effect when values with embedded blanks are specified.

Boolean - operator used to combine multiple qualifying conditions, as follows:
'AND' Both conditions must be true.
'OR' At least one condition must be true.
'NOT' The first condition must be true and the second false.

Repeat - additional multiple [fieldname] [relator] [value] phrases may be appended with each separated by a [relator]

Examples: COUNT 4
COUNT 6 IF CODE GT 12
COUNT 13 IF CODE IS (40,TO,60)
COUNT 2 IF TITLE EQ VDRL AND MDPART CONTAINS VD
COUNT 7 IF DEFINITN CONTAINS (LIVER,LUNG,WBC)

4.1.2 Command: DICT (NOT CURRENTLY ACTIVE)

Function: Displays a pre-defined dictionary of synonyms for a specified word, and the number of times each word occurs in the document database.

Syntax: DICT <Word>
Words in angle brackets are required user-supplied variables.

Required: <Word>

Operands: Word - a word in a synonym dictionary.

Note: Root search is automatic; that is, all words beginning with the characters specified in WORD will be used.

Examples: DICT VDRL

4.1.3 Command: DISPLAY

Function: Displays words from the index, and the number of times each word occurs in the document database.

Syntax: DISPLAY <Word-list>
Words in angle brackets are required user-supplied variables.

Required: <Word-list>

Operands: Word-list - a single word or multiple words separated by commas, for which a posting count display is desired.

Note: Root search is automatic; that is, all words beginning with the characters specified in WORD-LIST will be displayed.

Examples: DISPLAY LIVER
DISPLAY WBC HEMOGLOBIN

4.1.4 Command: END

Function: Terminates search session.

Syntax: END

Operands: None.

Examples: END

4.1.5 Command: FREE

Function: Deletes Temporary Databases (Sets) created by previous commands.

Syntax: FREE [ALL] or FREE [Set number(s)]
Words in brackets are user supplied variables.

Default: [ALL]

Operands: 'ALL' - Erases all sets.
Set Number(s) - Erases only specified sets.

Examples: FREE
FREE ALL
FREE 1 2 16

4.1.6 Command: HELP

Function: Displays the function, syntax, operands, and examples of a command.

Syntax: HELP [Command] [Segment]
Words inside brackets are user-supplied variables.

Default: All segments of the HELP information for [Command] are displayed.
Optional: [Command], to display information on a particular command.
Optional: [Segment], to display information on a given segment.

Operands: Command - A declaration of one of the following will display HELP information for that command: COUNT, DICT,DISPLAY, END, FREE, HELP, HISTORY, OUTLINE, PRINT, SEARCH, SET, SHOW, SUMMARIZE, USER.

Segment - A particular part of the HELP information for a command may be specified, as follows:
'FUNCTION' or 'F'
'SYNTAX' or 'S'
'OPERANDS' or 'O'
'EXAMPLES' or 'E'

Note: Only one Segment at a time may be specified. No blanks are allowed when specifying the HELP command.

Examples: HELP COUNT
HELP FREE S

4.1.7 Command: HISTORY

Function: Displays set numbers, counts, and search criteria for temporary databases (sets) created by previous commands. Also displays most recently summarized dataset number.

Syntax: HISTORY [Set Number(s)]
Words in brackets are user-supplied variables.

Default: If set number(s) are omitted, information for all sets displayed.

Optional: Set Number(s)

Operands: Set Number(s) - set numbers and search criteria for specified sets will be displayed. No counts are displayed if Set Number operand is specified.

Note: Only posting counts are displayed.

Examples: HISTORY
HISTORY 3 8

4.1.8 Command: OUTLINE

Function: Prints the outline structure of available documents.

Syntax: OUTLINE [name]
Words in brackets are user-supplied variables.

Default: Name = ALL

Operands: Name - must be one of the following literals:

DISEASE - Outline format for infectious disease documents.

LAB - Outline format for Clinical Pathology Laboratory documents

ALL - prints outlines of all active documents in the CPLM database.

Examples: OUTLINE DISEASE

4.1.9 Command: PRINT

Function: Prints documents satisfying a logical condition and/or previous search criteria.

Syntax: PRINT [Format] [Set Number] IF [Fieldname] [Relator] [Value] [Boolean] [Repeat]
Words in brackets are user-supplied variables. The entire command must fit on a single line.

Default: Default format is established at installation time.
Required: <Set Number> or IF <Fieldname> <Relator> <Value>
Optional: Format

Operands: Format - specifies what is to be printed. Formatting options are designed for each system at installation time. Current valid formats for CPLM are:
DEF -- displays document definitions
SYN -- displays document synonyms
All -- displays all of above

Set Number - the set number of a temporary database to be printed.

'IF' - begins a qualifying condition. The full condition must consist of:
IF [Fieldname] [Relator] [Value]

Fieldname - a field from the document database.

Relator - indicates field-to-value comparison, as follows:

'GT'	greater than
'LT'	less than
'GE'	greater than or equal to
'LE'	less than or equal to
'EQ'	equals
'NE'	not equal to
'IS'	field begins with specified value
'NOT'	field does not begin with specified value
'CONTAINS'	value is found anywhere in field

'EXCLUDES' value is not found anywhere in field

Value - one or more values against which field is compared. Multiple values must be separated by commas and enclosed in parentheses. A range may be specified with the 'IS' relator and 'TO' connecting the upper and lower values in the range. The 'SET,COMMAS,ON' option must be in effect when values with embedded blanks are specified.

Boolean - operator used to combine multiple qualifying conditions, as follows:

'AND' Both conditions must be true.

'OR' At least one condition must be true.

'NOT' The first condition must be true and the second false.

Note: A set resulting from a COUNT command may not be printed with HILITE format. A menu of available HILITE specifications will be displayed when 'HELP' is entered after HILITE prompt.

Examples:
```
PRINT 4
PRINT PAGENO 6 IF TESTPART CONTAINS CULTURE
PRINT 14 IF TITLE IS VDRL
PRINT PAGE 2 IF CODE EQ 328 AND SYNONYMS CONTAINS
      THROAT
PRINT 7 IF MDPART CONTAINS (WBC,RBC,VIRAL)
```

4.1.10 Command: SEARCH

Function: Identifies documents based on relationships among words in the documents. Relationships may be specified as follows:

- words adjacent to each other;
- words in the same document, field, or sentence;
- a word within a specified number of words from another;
- word relationships based on 'AND', 'OR', 'NOT' conditions.

Syntax: SEARCH [Field] SYN [Word(s)] [Relator] SYN [Word(s)]...

Words in brackets are optional user-supplied variables.

Multiple word, relator, word combinations are allowed.

Default: Default field(s) are established at installation time.
Required: Word(s)
Optional: Field, SYN, Relator

Operands: Field - delimits field to be searched. A field-group may be designated at installation time to represent several fields under one name.

'SYN' - preceding any word, retrieves all predefined synonyms for that word.

Word(s) - one or more words to be searched. One or more set numbers of previously created temporary databases may be specified in place of Word(s). Use of the set number, once a temporary database has been defined it is more efficient. More than one word/set, separated by commas, indicates an 'OR' condition. A word root followed by '*' indicates a search on all words beginning with the root.

Relator - Combines one word/set with another, as follows:

'AND' Words/sets must occur in the same document

'OR' At least one word/set must occur in the document. 'OR' may be specified or implied.

'NOT' Words/sets following 'NOT' must not occur in same document as those preceding 'NOT'.

Above relators may be mixed in the same command.

'ADJ' Words/sets must be adjacent and must occur in the direction indicated. Noise words (as, of, and, by, etc.) are not searchable, but do count in determining distance between search words.

'SEN' Words/sets must occur in the same sentence, according to installation definition of a sentence. 'SEN' is allowed only when

Sentence Proximity has been implemented in the particular system.

W n — Words/sets must occur within n, an integer, words of each other. If Sentence Proximity has been implemented in the particular system, words/sets must also occur in the same sentence; otherwise, in the same document. Words/sets may occur on either side of each other, except when +n specifies direction.

All relators may be combined with 'OR' in the same command.

Note: The resulting temporary database contains the total number of occurrences of all words satisfying the command criteria; it does not contain the number of documents satisfying the criteria.

Examples:
```
SEARCH LIVER
SEARCH LIVER ADJ FLUKE
SEARCH TITLE BRUCELLOSIS
SEARCH LIVER LUNG LDH
SEARCH SYN VDRL
SEARCH ARTHRO*
SEARCH COMPLEMENT AND FIBRINOGEN
SEARCH WBC NOT BLOOD
SEARCH RBC SEN HEMOGLOBIN
SEARCH BONE W 5 MARROW
SEARCH THROAT W +3 CULTURE
```

4.1.11 Command: SET

Function: Establishes characteristics of the search environment.

Syntax: SET <Characteristic> <Status>
Words in angle brackets are required user-supplied variables.

Defaults: BRIEF OFF, SYN OFF, DOCCNT OFF, COMMAS ON
Required: One Characteristic and one Status

OPTIONS: Characteristic - 'BRIEF' eliminates messages about intermediate temporary databases.

	'SYN'	causes subsequent searches to use the specified word and all of its pre-defined synonyms.
	'DOCCNT'	returns number of documents in addition to number of postings.
	'COMMAS'	in OFF Status, allows spaces instead of commas between words in command strings.
Status -	'ON'	turns on Characteristic operand.
	'OFF'	turns off Characteristic operand.

Note: Only one Characteristic-Status pair may be specified per SET Command, but multiple SET Commands may be declared. A Status will remain in effect until changed.

Examples: SET,BRIEF,ON
SET,DOCCNT,OFF
SET,COMMAS,ON
SET SYN ON

4.1.12 Command: SHOW

Function: Prints a given field selection for items in the most recently SUMMARIZEd temporary dataset. Items are identified by the ITEM value from the SUMMARIZE statement.

Syntax: SHOW [field-list] IN [Item-list]
Words in brackets are optional user supplied variables.

Defaults: Field-list and Item-list initially set to MDPART and 1 respectively but becomes last referenced value after subsequent SHOW commands.

Operands: Field-list - The major field name from the presentation outline. Multiple field names are allowed. Use the OUTLINE command to obtain the valid field names.

Item-list - The item number which identifies a document within the current active temporary database. The Item numbers are obtained by first SUMMARIZing the desired temporary datasets.

Examples: SHOW MDPART IN 1
SHOW TESTPART
SHOW TITLE SYNONYMS DEFINITN IN 1 2 7

4.1.13 Command: SUMMARIZE

Function: Prints documents in the selected temporary dataset. This function also sets the most recently summarized dataset for the SHOW command.

Syntax: SUMMARIZE [n]
Word in brackets is user supplied variable.

Default: n is initially set to 1 but becomes last referenced dataset number after subsequent SUMMARIZE commands.

Operands: n - specifies the temporary dataset number. multiple numbers are not allowed.

Examples: SUMMARIZE 3

4.1.14 Command: USER

Function: Enters User Language mode.

Syntax: USER

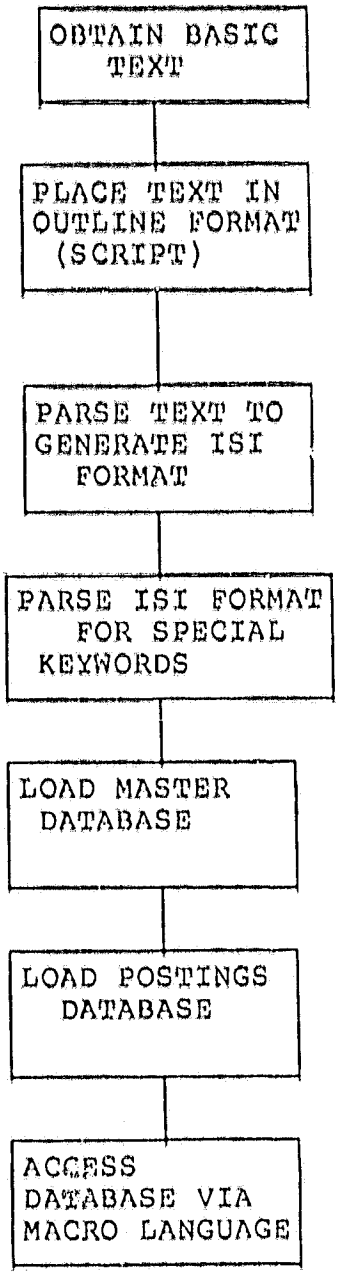
Operands: None.

Note: To return to Proximity Searching from the User Language, enter &BEGIN.

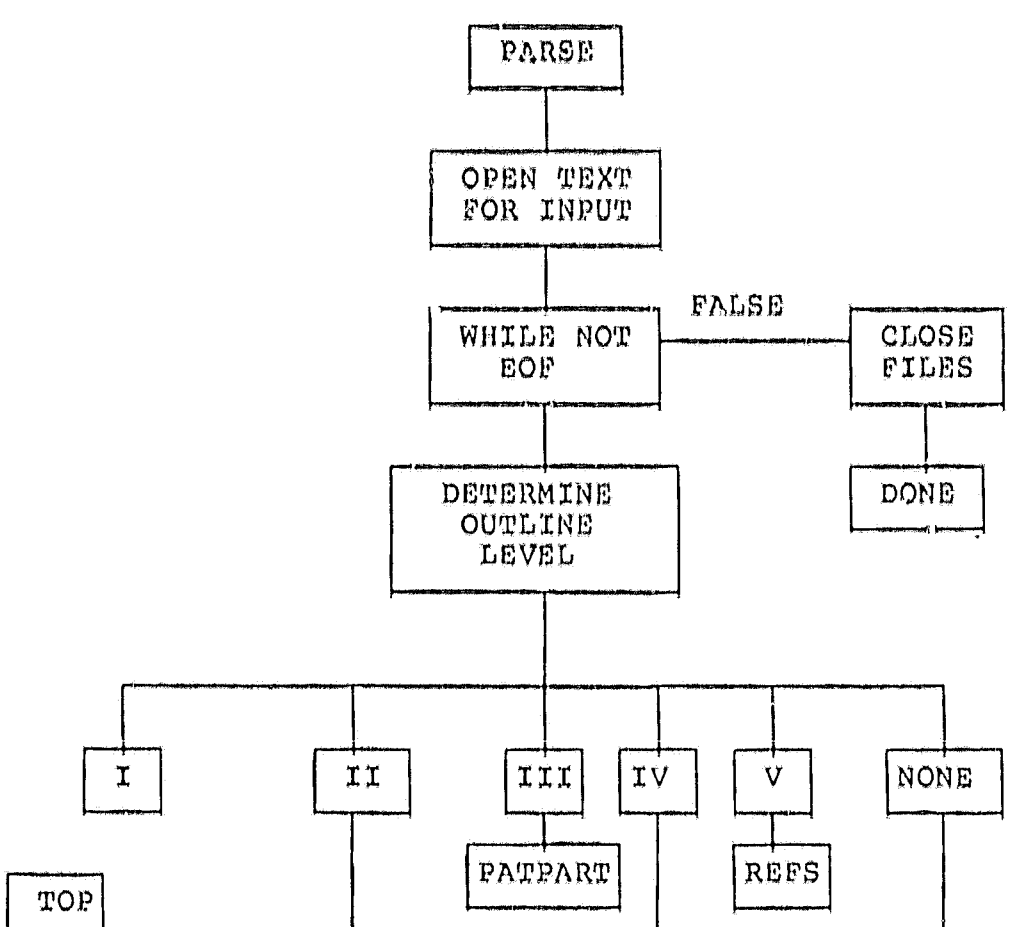
CHAPTER 5
SYSTEM FLOW DIAGRAMS

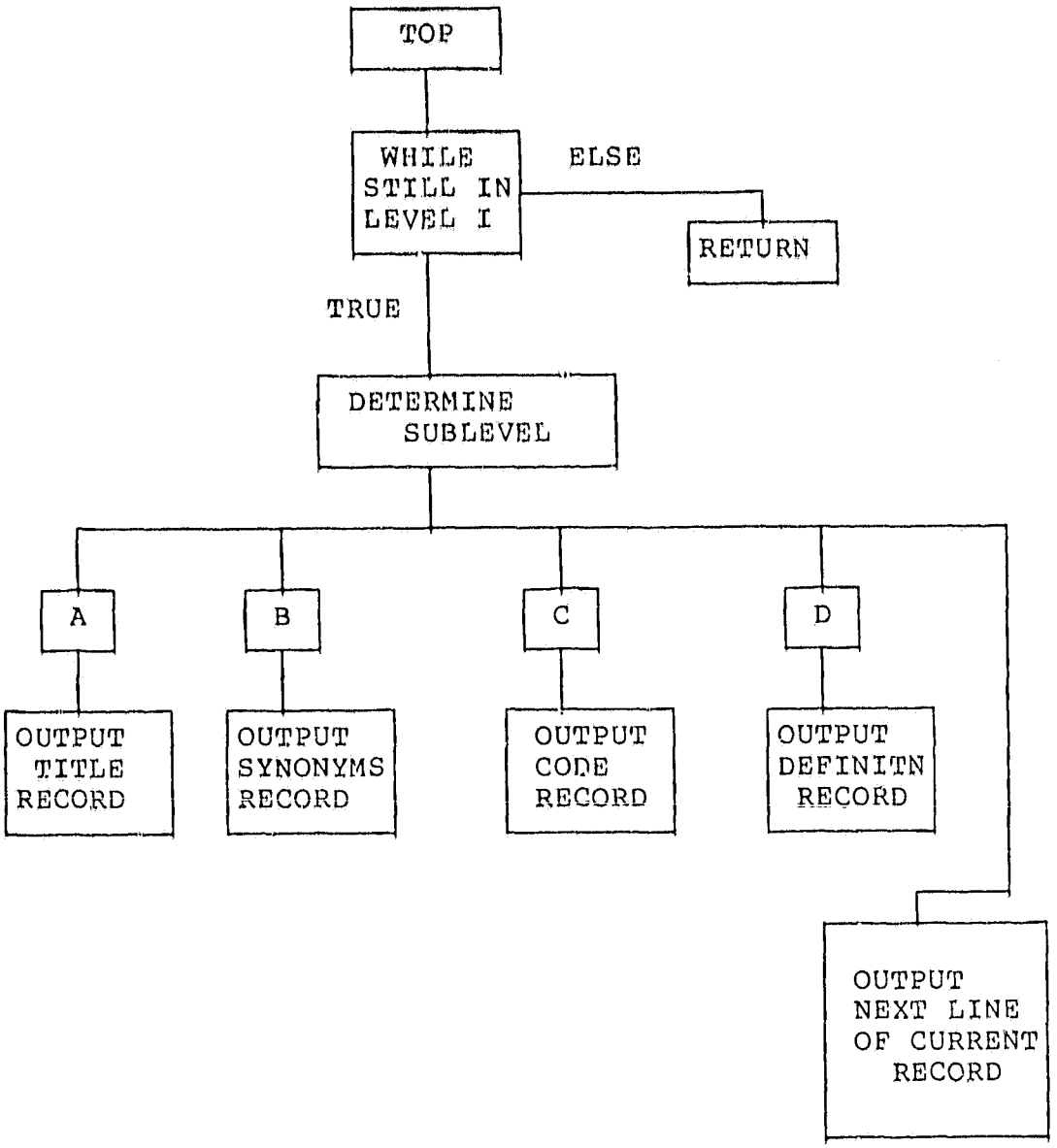
This section contains the system heirarchy diagram as well as basic flow diagrams for the preprocessor.

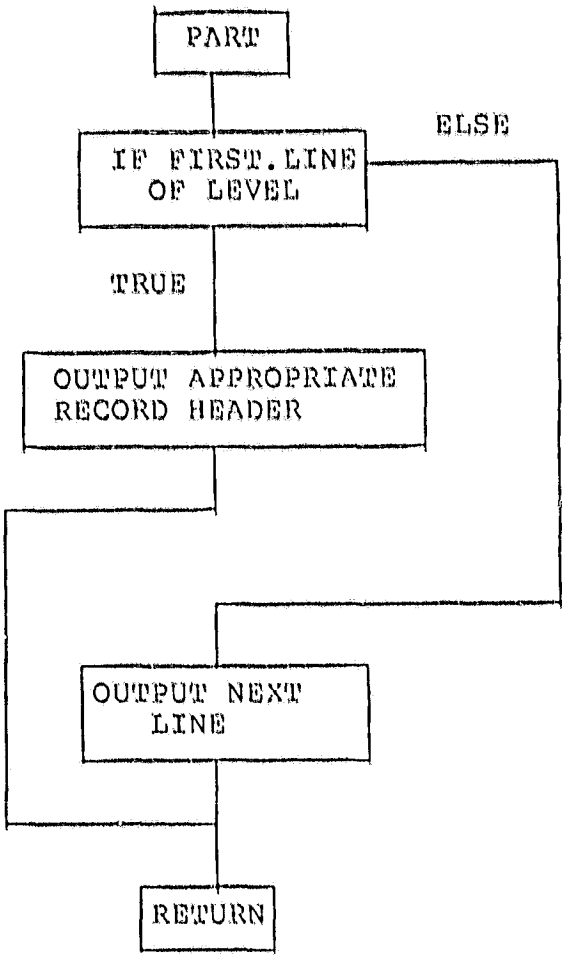
OVERALL SYSTEM FLOW



PARSE FLOW DIAGRAM




TOP
WHILE STILL IN LEVEL I
ELSE
RETURN
TRUE
DETERMINE SUBLEVEL
A
B
C
D
OUTPUT TITLE RECORD
OUTPUT SYNONYMS RECORD
OUTPUT CODE RECORD
OUTPUT DEFINITN RECORD
OUTPUT NEXT LINE OF CURRENT RECORD


PART
IF FIRST.LINE OF LEVEL
ELSE
TRUE
OUTPUT APPROPRIATE RECORD HEADER
OUTPUT NEXT LINE
RETURN

CHAPTER 6
SYSTEM DATABASE DESCRIPTIONS

6.0 The various elements of the CPLM system databases are described in this chapter.

6.1 INPUT DATABASE

The input database consists mainly of four datasets. These datasets are:

1) Pre-formatted text input. This data is in upper and lower case and has inbedded SCRIPT commands for formatting and left/right justification. The basic structure required is the defined outline format as shown in Appendix A. The dataset is named in the form: UF.A0081909.CPLMTEXT.TEXT.(Qualifier) and is RECFM=FB, LRECL=80.

2) Formatted text input. This data is that which was processed from D above through the SCRIPT facility in order to set paragraph justification, etc. The .TEXT dataset step may be skipped if appropriate formatting is done by an alternate means (eg. stand alone word processor) as long as this dataset is generated. The dataset name is of the form: UF.A0081909.CPLMTEXT.TEXT.SCRIPT (Qualifier and has the same DCB characteristics as 1).

3) The basic ISI format is created vis the PARSE utility which operates on the .SCRIPT dataset. The output is a file of records that can be processed by INQUIRE text utilities. The dataset name is of the form: UF.A0081909.CPLMTEXT.ISIFMT.TEXT and has the same 80 column DCB previously described. This dataset is generally temporary in nature, lasting only as long as it is needed for INQKTEXT.

4) The program INQTEXT operates on the .ISIFMT dataset to generate a new dataset containing specifically marked keywords it has extracted from selected fields. The dataset name is of the form: UF.A0081909.CPLMTEXT.ISIFMT.TEXT.WKEYS. This dataset is temporary until it can be used by INQLOAD and INQPOST to load the biblio and posting datasets.

6.2 THE PROGRAM RELATED DATASETS

There are several datasets that are necessary for creating; maintaining, and running the CPLM system. These are:

1) UF.G0081909.PANVALET - this is an online PANVALET library that is maintained for storage of program source and JCL for backup and archiving purposes.

All the source (excepting macros) and JCL for all operational jobs is maintained in this library.

2) UF.A0081909.CPLMTEXT.LINKLIB - this is an online library of compiled routines that are accessed and linked as necessary by INQUIRE. It is also the residence of the output of the INQASML job. It is linked to during startup of the CPLM macros.

3) U.A0081909.CPLMTEXT.PROXMAC - this is the PDS library containing all the macros executed in the CPLM User Interface.

4) U.A0081909.CLIST - this is a PDS containing the member CPLM (among others) which invokes the CPLM user interface when a TSO user executes the command:
EXEC A0081909(CPLM)<CR>

5) There are also a number of transient datasets maintained for editing and debugging in the interactive program space of NERDC. These datasets (files) generally only have use to individual programmers.

6.3 CPLM Working Datasets

There are eight permanent datasets comprising the CPLM working datasets. In addition, during transactions with the user numerous transient datasets are created and deleted. This section only discusses the eight permanent ones which are:

1) UF.A0081909.CPLMTEXT.DATA - this is the loaded text dataset.

2) UF.A0081909.CPLMTEXT.INDEX - this dataset contains the unique binary keys flagged by INQKEXT. A pointer in this field points to the end of a chain of records in the .SEARCH file

3) UF.A0081909.CPLMTEXT.SEARCH- this dataset contains a linked list of pointers for all occurrences of the key to its physical address in the .DATA dataset.

4) UF.A0081909.CPLMTEXT.SEARCHOV- this dataset contains room to hold records for which insufficient room is available in the .SEARCH dataset.

5) UF.A0081909.CPLMPOST.INDEX- this dataset contains the unique keys extracted using the proximity searching utilities of INQPOST. Its function is similar to the CPLMTEXT.INDEX dataset except it applies only to the .CPLMPOST chain and keys.

6) UF.A0081909.CPLMPOST.SEARCH- see 3)

7) UF.A0081909.CPLMPOST.SEARCHOV- see 4)

8) UF.A0081909.CPLMPOST.DATA- this dataset is pointed to by the .CPLMPOST.SEARCH and it in turn contains posting entires that point to the .CPLMTEXT.DATA entry.

6.4 ARCHIVE DATASETS

Only one dataset is backed up in multiple copies at this time as all the others can be reconstructed rather quickly and inexpensively. This one backup dataset is the formatted text input and is named:

UF.A0081909.CPLMTEXT.TEXT.BACKUP. This dataset is maintained on tape.

CHAPTER 7
GENERATING THE TEXT FOR THE DATABASE

7.0 Introduction

Generating the text for CPLM is a multiple step iterative process. The first step is organizing the data. All text to be input to CPLM must be organized in the same five level outline format listed in appendix B. The major level headings cannot be changed though the sublevels may be changed as necessary. For consisting sake all data in a particular area should follow the same outline format. Missing items at the sublevel should be left blank or coded as NONE.

Once the basic outline format is prepared then the text must be prepared. The following sections describe how to prepare text for the five major levels thus far implemented in CPLM.

7.1 Procedure Identification

The Procedure Identification section contains four specific sections of information. As indicated by the level title this information is identity related.

The TITLE section should contain a concise description of the material to be contained in the new document. This field will be frequently displayed in CPLM responses so keep the word-count down to a minimum. Be sure to contain major key terms if possible, especially if they are descriptive (eg. Brucella Culture). The maximum length of any title currently allowed is 94 letters, spaces, and punctuation (characters).

The SYNONYMS section should contain a list of known synonyms for the title or main subject of the document. Currently up to 165 characters are allowed in this section. Separation by single spaces is sufficient. Currently no cross correlation is being done in the CPLM user interface but this is under consideration. By allowing the author to specify the range of synonyms a great deal of possibly erroneous information can be better controlled.

The CODE section was originally designed to contain procedure identification codes but is being migrated to contain information on the document status (eg. Reviewed date, etc.). Thirty-five characters can be used in this section.

The last section in level I is the DEFINITN section. Here a concise definition of the subject should be listed. For some subjects this will be a straight

dictionary type entry while for others it may serve as an abstract section. Up to 2000 characters (about 27 lines) can be contained in this field.

7.2 Physician Procedure Description

This section should contain information relevant to the physician such as Diagnosis, Occurrence, Physiology, etc. The sublevels of this and all subsequent major levels may be organized as necessitated by the subject matter. This is currently the largest section with 13,400 characters (about 186 lines) allowed.

7.3 Patient Related Description

This section should contain information directly related to the patient. Such items as treatment, protocol, susceptibility, or communicability period are recorded here. Eight thousand characters (about 100 lines) may be used here.

7.4 Test Related Description

This section was originally named consistent with the clinical laboratory in mind and contained test specific information. Similar material can be found in most other areas of medicine though the specific items displayed here will need to be chosen carefully. In the case of the Infectious Diseases documents, this section contains information on control and prevention. Eight thousand characters may be used here.

7.5 References

This section contains the original references for the material in the document as ell as references for further reading. Up to 900 characters (about 12 lines) may be used in this section.

CHAPTER 8
GENERATING THE WORKING DATABASE

The steps to be followed in loading the database are:

1) take the source file and edit it using SCRIPT.
   infile is CPLMTEXT.TEXT
   outfile is CPLMTEXT.TEXT.SCRIPT
   Job is INQSCRPT

2) Parse the source to ISI format
   infile is CPLMTEXT.TEXT.SCRIPT
   outfile is CPLMTEST.ISIFMT.TEXT
   Job is INQPARSE

3) Perform keyword extraction.
   infile is CPLMTEXT.ISIFMT.TEXT
   outfile is CPLMTEXT.ISIFMT.WKEYS
   Job is INQKEXT

4) Load the database
   infile is CPLMTEXT.ISIFMT.WKEYS
   outfiles are CPLMTEXT.KDUSFIL (Keyword list)
   and CPLMTEXT.DATA
   The sort field file is not kept or catalogued but the value is:
   Sort Fields = (5,044,A), Format=CH,size 00000361

5) Sort the unsorted keyword list
   infile is CPLMTEXT.KDUSFIL
   outfile is CPLMTEXT.KDSFIL
   Job is INQSORT

6) Generate the biblio database index
   infile is CPLMTEXT.KDSFIL
   outfiles are CPLMTEXT.INXFILU
   CPLMTEXT.SRUSFIL
   Job is INQLOAD

7) Sort the biblio search file keys
   infile is CPLMTEXT.SRUSFIL
   outfile is CPLMTEXT.SRSRFIL
   Job is INQLOAD

8) Load the Search and Overflow files
   infile is CPLMTEXT.SRSFIL
   outfiles are CPLMTEXT.SEARCH
   CPLMTEXT.SEARCHOV
   Job is INQLOAD

9) Load the Index file
   infile is CPLMTEXT.INXFILU
   outfile is CPLMTEXT.INDEX
   Job is INQLOAD

10) Extract all the non-noise keywords
from the biblio database
infile is CPLMTEXT.DATA
outfile is CPLMPOST.SFLDFIL
Job is INQPOST
Step is POST1

11) Sort the extracted keys
infile is CPLMPOST.PROXFIL
outfile is CPLMPOST.SORTOUT
Job is INQPOST
Step is POST2

12) Load the posting data file
outfile is CPLMPOST.DATA
Job is INQPOST
Step is POST4

13) Dummy the posting Index, Search, & Overflow files
Job is INQPOST
Step is POST6

CHAPTER 9
SUMMARY OF ERROR MESSAGES AND CODES

9.0 Introduction

Errors in this system occur in one of two major steps:

1) at database load time
2) at User language time

9.1 Database Load Errors

The errors associated with database loading are primarily those associated with mainframe IBM JCL. Wrong units, media failure, and inadequate size are some common errors which can be overcome only by programmer intervention. In addition, file specification errors, file size, and FDT overflow are INQUIRE related errors that are covered in more detail in the INQUIRE Messages and Codes Manual.

9.2 User Interface Errors

The errors that can occur at the User Interface include operator errors due to syntax, spelling, etc., or program size limitations. All known operator errors generate screen prompts to all interactive correction. Problems having to do with memory allocation should be reported to the Medical Systems Division staff with a complete history leading to the problem. This will allow the programmming staff to make corrections. There are no user fixable errors other than those prompted by the Interface language.

CHAPTER 10
INDEX

User: 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 27, 29, 33, 34, 35, 45, 47, 68, 71, 72, 73, 74, 75, 78

User Interface: 4, 7, 27, 29, 33, 34

WT/78: 8, 34

REFERENCES

1. Moser, Robert H., "Reflections on access and intellectual triage in CME", pages 4,5,17.

2. Grams, Ralph R., and Massey, James K., On-Line Automated History & Physical Feasibility Study, Preliminary Report on NASA Grant NAG10-0004, 4 December, 1981.

3. Ben-Bassat, M., et al., "Pattern-Based Interactive Diagnosis of Multiple Disorders: The MEDAS System", IEEE Transactions on Pattern Analysis & Machine Intelligence, VOL PAMI-2, #2, March 1980.

4. Smith, S.C. Educational Uses of the Plato Computer Systems, Science, V. 192, pp. 344-52, April 1976.

5. Vanselow, N.A. Computer Assisted Instruction in Pharmacology at the University of Arizona Medical School, Ariz. Med, 32(8) 633-4, Aug. 75.

6. Hoffer, E.P., Barnett, G., Farquhar, B.B., Prather, P.A., Computer-aided Instruction in Medicine. Annual Review Biophy. Bioeng., No. 4, pp. 103-118, 1975.

7. Valish, A.U., Boyd, V.J. The Role of Computer Assisted Instruction in Continuing Education of Registered Nurses: An Experimental Study. Journal of Continuing Educ., Nurs., No. 6, pp. 13-32, Jan.-Feb. 1975.

8. Hoffer, E.P. Computer Aided Instruction in Community Hospital Emergency Departments: A Pilot Project. Journal of Medical Education, No. 50, pp. 84-86, January 1975.

9. Meyer, J.H. An Evaluation of Computer Assisted Teaching in Physiology. Journal of Medical Education, No. 49, pp. 295-297, March 1974.

10. Weinberg, A.D. CAI at the Ohio State University College of Medicine. Comput. Biol. Med., No. 3, pp. 293-8, October 1973.

11. Mellner, C. The Computerized Problem - Oriented Medical Record at Kurdinska Hospital - Format Function Users Acceptance and Patient Attitude to Questionnaire. Methos Inform Medicine, Vol. 15, November 1976.

12. Weed, L.L. Your Health Care and How to Manage It. Essex, Essex Junction, VT. 1976.

13. Luff, C.A. Computer Code for the Recording of Patient Problems. Comput. Biomed. Research, No. 8, pp.267-78, June 1975.

14. Okubo, R.S. Natural Language Storage and Retrieval of Medical Diagnostic Information: Experience at UCLA Hospital and Clinics. Computer Programs in Biomedicine, No. 15, pp. 105-130, 1975.

15. Ressler, N. Computer Assisted Diagnosis by a Model Free System of Direct Data Analysis. Perspectives in Biology and Medicine, pp. 101-117, 1975.

16. Feinstein, A.R. An Analysis of Diagnostic Reasoning. No. 3, The Construction of Clinical Algorithms. NLN Publ., pp. 1-11, 1975.

17. Levine, G.M. An Introduction of Management Information System in Community Health Agencies. NLN Publ., pp. 1-11, 1975.

18. Barlett, M.H. Patients Receive Current Concise Health Information by Telephone Hospitals, No. 50, pp. 79-80, 82, February 1976.

19. Wood, M. A Systems Approach to Patient Care, Curriculum and Research in Family Practice. J. Med. Ed., No. 50, pp. 1106-12, December 1975.

20. Summit, R.K. Lockheed Experience in Processing Large, Data Bases for its Commercial Informational Retrieval Service. J. Chem. Inf. Comput. Science, No. 15, pp. 48-51, February 1975.

21. Cwodra, C.A., SPC Experiences with large data bases. J. Chem. Inf. Comput. Science 15: pp. 48-57, Feb. 1975.

22. Schaap, A.L. Computer Retrieval of Articles on the Therapy of Poisonings. Clin. Toxicol., No. 8, pp. 301-310, 1975.

23. Weitzel, R. Medline Services to the Developing Countries. Bull Med. Lib. Assoc., No. 64, pp. 32-35, January 1976.

24. Heaps, H.S. Data Compression of Large Document Data Bases. J. Chem. Inf. Comp. Science, Vol. 15, pp. 32-39, February 1975.

25. Thiel, L.H. Program Design for Retrospective Search on Large Data Bases. Informal Stor. Retrieval, Vol. 8, pp. 1-20, February 1972.

26. Ein-Dor, Phillip, Comparative Efficiency of Two Dictionary Structures for Document Retrieval, Infor. J., Vol. 12, pp. 87-108, February 1974.

27. Grams, R.R. Problem Solving, System Analysis and Medicine. Charles C. Thomas, Springfield, Ill., 1972.

28. Litzkow, L.R. A Real Time Library of Medicine for Practicing Physicians. Journal of Medical Systems, Vol. 1, No. 3, pp. 299-306, 1977.

29. Grams, R.R., Progress Towards a Second Generation Laboratory Information System (LIS). Journal of Medical Systems, Vol. 1, No. 3, pp. 263-274, 1977.

30. Schoolman, H.M., Bernstein, L.M. Computer Use in Diagnosis, Prognosis, and Therapy. Science, Vol. 200, No. 4344, pp. 926-930, 1978.

31. YGH, Raymond T. (Ed). 1977. Current Trends in Programming Methodology: Software Specification and Design Vol. I. Prentice-Hall, Englewood Cliffs, New Jersey.

32. Tausworthe, Robert C. 1977. Standard Development of Computer Software, Prentice-Hall, Englewood Cliffs, New Jersey.

33. Brinch Hanson, Per. 1977. The Architecture of Concurrent Programs, Prentice-Hall, Englewood Cliffs, New Jersey.

34. Brinch Hanson, Per. 1973. Operating System Principles. Prentice-Hall, Englewood Cliffs, New Jersey.

35. Martin, James, Computer Data-Base Organization, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

36. Wiederhold, Gio. Database Design. McGraw-Hill, New York, 1977.

APPENDIX A
Listings of Documents Currently in CPLM

On the following pages are listed the titles of the documents currently in the CPLM database. These documents fall into one of two general categories: Clinical Pathology Laboratory or Infectious Diseases. The Clinical Pathology Laboratory documents describe lab tests procedures and results. Each Infectious Disease document describes a particular disease (or class of diseases), the characteristics of the disease, and curative measures (see appendix B).

Clinical Pathology Laboratory

Name

Actinomyces Culture
Alkaline Phosphatase
Antimitochondrial Antibody
Anti-Nuclear Antibody
Antismooth Muscle Antibody
Australia Antigen
Beta-Strep Screen
Billirubin
Biopsy Culture
Blood Culture
Bronchial Washing Culture
Brucella Culture
BSP
BUN
Calcium
Carbon Dioxide
C Diphtheriae Culture
Creatinine
C'3 Complement
C'4 Complement
C-Reactive Protein
Cryoglobulin
Cryofibrinogen
Differential
Direct Coombs
EBV Antibody Titre
Eosinophil Count
ESR
Fecal Fat
Fecal Muscle Fiber
Fetal HGB%
FTA-ABS Serum
Glucose
Gonorrhea Culture
Hematocrit
Hemoglobin
Hemoglobin Electrophoresis
Joint Fluid Culture
Lactic Dehydrogenase
Large Light Chains
Listeria Culture
Mono Spot Test
Occult Blood
Peretoneal Abscess Culture
Pertussis Culture
Pinworm Slide
Platelet Count
PT -- Prothrombin Time
PTT -- Partial Thromboplastin Time
RBC Count

Reticulocyte Count
Rheumatoid Factor
Rubella Antibody
Serum Protein Electropheresis
Serum Glutamic Pyruvate Transaminase
SGOT -- Serum Glutamic Oxaloacetic Transaminase
Sickle Screen
Sodium (Na+)
Stool Culture
Stool O & P
TGT
Throat Culture
Thrombin time
Total Protein
Urine Culture
Urinalysis
Urine Creatinine (24 Hr)
VDRL-Serum
WBC
Wound Culture

Infectious Diseases

Name

Amebiasis
Ancylostomiasis
Arthropod-Borne Viral Arthritis
Arthropod-Borne Viral Encephalitides
Brucellosis
Cat-Scratch Disease
Chickenpox -- Herpes Zoster
Colorado Tick Fever and other Tick-Borne Fevers
Crimean Hemmorrhagic Fever
Hemmorrhagic Fever
Omsk Hemmorrhagic Fever and Kyasanur Forest Disease
Primary Amebic Menengoencephalitis
Russion Spring Summer Encephalitis
Sandfly Fever
Viral Arthropod-Borne Hemmorrhagic Fever
Viral Fevers

APPENDIX B
Outline of Input Format

The outline format is used to input data into the CPLM database. The text preprocessor (see appendix E) takes data in the outline format shown on the following pages and translates it into the format recognized by the database management system, INQUIRE (see appendix C).

I. Procedure Identification
   A. Name
   B. Synonyms
   C. Code #s
   D. Definition
II. Physician Procedure Description
   A. Risks and Contraindications
   B. Normal Limits
   C. Physiology
      1. Normal
      2. Abnormal
      3. Algorithms
   D. Elevations
      1. Diagnosis (grade to level of probability)
      2. False positives
      3. Interpretation
   E. Depressions
      1. Diagnosis (grade to level of probability)
      2. False negatives
      3. Interpretations
   F. Variability
      1. Within Day
      2. Between Day
   G. Cost Factor
      1. Technical
      2. Professional
   H. Processing Time
      1. Routine
      2. Stat
III. Patient Related Description
   A. Patient Processing Instructions
      1. STAT
      2. Routine
   B. Procedure Description for Patient
IV. Test Related Description
   A. Specimen
      1. Volume
      2. Type
      3. Handling Instructions
   B. Processing Procedure
   C. Testing Procedures
      1. Primary Method
         a. Equipment Required
         b. Errors
         c. Technical Requirements (personnel, time, and control)
      2. Alternate Methods
         a. Equipment Required
         b. Errors
         c. Technical Requirements (personnel, time, and control)
V. References

Infectious Diseases

I. Procedure Identification
   - A. Name
   - B. Synonyms
   - C. Code #s
   - D. Definition

II. Physician Procedure Description
   - A. Diagnosis
   - B. Occurance
   - C. Infectious Agent

III. Patient Related Description
   - A. Resevoir
   - B. Mode of Transmission
   - C. Incubation Period
   - D. Period of Communicability
   - E. Susceptibility and Resistance

IV. Test Related Description
   - A. Methods of Control
   - B. Preventative Measures
   - C. Control of Patient, Contacts, and Immediate Environment
   - D. Epidemic Measures
   - E. International Measures

V. References

I. Procedure identification
   A. C'4 COMPLEMENT
   B. SYNONYMS - beta 1e
   C. CODE - 277
   D. DEFINITION - measurement of levels of the 4th element of complement

II. physician procedure description
   A. RISKS - none
      CONTRAINDICATIONS - none
   B. NORMAL LIMITS - 20-40
   C. PHYSIOLOGY

The complement system is a complex cascade of factors with various immunologic roles: hemolysis, cytotoxicity, chemotaxis, immune adherence, anaphylaxis, viral neutralization, opsonization, kinin like activity, aggregation and increased affinity of antigen-antibody complexes. There are two routes of activation: the classical and the alternate (or properdin) pathway. The classical pathway, involving both c'3 and c'4 is generally activated by antigen-antibody complexes. The alternate pathway, involving c'3 but not c'4, is activated by certain fungal, and bacterial antigens as well as some antigen antibody complexes. Complexes with c'4 are responsible for viral neutralization while c'3 split products are active in chemotaxis, anaphylaxis, opsonization, and immune adherence. Deficiency of c'4 leads to a lupus-like syndrome, c'3 deficiency is marked by repeated pyogenic infection. The complement system is important in the pathogenesis of a number of diseases: hereditary angioedema is associated with c' esterase deficiency; paroxysmal cold hemoglobinuria is the result of rbc lysis by complement; rbc's in paroxysmal nocturnal hemoglobinuria are especially sensitive to c' lysis; c' has been implicated in platelet destruction in itp; c' serum levels decrease in active sle nephritis, post-strep glomerular nephritis, serum sickness. Although serum levels do not change in ra, synovial fluid c' levels are very low; c' may increase lysis or clearance of transfused rbc's, and in malarial paroxysms, hepatitis with arthritis and severe liver disease (decreased production), c' levels may also be low.

C'4 is indicative of classical pathway activation and is usually accompanied by decreased c'3. Decreased c'4 may also be a result of liver disease or very rarely, genetic deficiency. Decreased c'3 and c'4 is indicative of alternate pathway. Activation or of decreased production. Levels of c'3 and c'4 tend to return to normal with resolution of the underlying problem, and fall again with reactivation providing a means of prognostication. Measurement of c'3 and c'4 is by radial immuno-diffusion, not functional activity.

   D. ELEVATIONS - none
   E. DEPRESSIONS
      1. Diagnoses (decreased c'4 and c'3)
         A. Lupus nephritis
         B. Cns lupus (csf levels 0
         C. Early post-strep glomerular nephritis
         D. Rheumatoid arthritis ra (joint fluid)
         E. Goodpasture's nephritis
         F. Liver disease
         G. Autoimmune hemolytic anemia
         H. Post-burn (1-3 weeks)
         I. Malarial paroxysm

J. Genetic decrease (very rare)
2. False - technical error
3. Interpretations - further tests by dx
A. Ana, rf, biopsy
B. Ana, rf, biopsy
C. Aso, hx, biopsy
D. Rf, joint fluid examination
E. Ana, rf, biopsy
F. Liver function tests, biopsy, australia antigen
G. Dag, urine urobilinogen, hemogram
H. Physical exam
I. Examination for plasmodium, hx and pe
J. Pedigree, by elimination

I. Procedure identification
A. C'3 COMPLEMENT
B. SYNONYMS - beta 1c
C. CODE - 276
D. DEFINITION - measurement of levels of the 3rd element of complement

II. physician procedure description
A. RISKS - none
CONTRAINDICATIONS - none
B. NORMAL LIMITS - 60-140 (20-40)
C. PHYSIOLOGY

The complement system is a complex cascade of factors with various immunologic roles: hemolysis, cytotoxicity chemotaxis, immune adherence, anaphylaxis, viral neutralization, opsonization, kinin like activity, aggregation and increased affinity of antigen-antibody complexes. There are two routes of activation: the classical and the alternate (or properdin) pathway. The classical pathway, involving both c'3 and c'4, is generally activated by antigen-antibody complexes. The alternate pathway, involving c'3 but not c'4, is activated by certain fungal and bacterial antigens as well as some antigen antibody complexes. Complexes with c'4 are responsible for virul neutralization while c'3 split products are active in chemotaxis, anaphylaxis, opsonization, and immune adherence. Deficiency of c'4 leads to a lupus-like syndrome, c'3 deficiency is marked by repeated pyogenic infection.

The complement system is important in the pathogenesis of a number of diseases: hereditary angioedema is associated with c' esterase deficiency; paroxysmal cold hemoglobinuria is the result of rbc lysis by complement; rbc's in paroxysmal nocturnal hemoglobinuria are especially sensitive to c' lysis; c' has been implicated in platelet destruction in itp; c' serum levels decrease in active sle nephritis, post-strep glomerular nephritis, serum sickness. Although serum levels do not change in ra, synovial fluid c' levels are very low; c' may increase lysis or clearance of transfused rbc's, and in malarial paroxysms, hepatitis with arthritis and severe liver disease (decreased production), c'4 levels may also be low.

C'4 is indicative of classical pathway activation and is usually accompanied by decreased c'3. Decreased c'4 may also be a result of liver disease or very rarely, genetic deficiency. Decreased c'3 and decreased c'4 is indicative of alternate pathway activation or of decreased production. Levels of c'3 and c'4 tend to return to normal with resolution of the underlying problem, and fall again with

APPENDIX C
Outline of CPLM Database Format

This format is used by INQUIRE to store and retrieve the documents in the database. The fieldnames on the left (in all caps), are used in the SHOW command (see appendix H). This enables the user to look at only a particular section of the document instead of displaying the entire document from the beginning.

The comments in parentheses: "(formerly I.A Name)", etc., refer to the corresponding heading in the input format outline (see appendix B).

TITLE (formerly I.A Name)
SYNONYMS (formerly I.B Synonyms)
CODE (formerly I.C Code)
DEFINITN (formerly I.D Definition)
MDPART (formerly II. Physician Procedure Description)

- A. Risks and Contraindications
- B. Normal Limits
- C. Physiology
  1. Normal
  2. Abnormal
  3. Algorithms
- D. Elevations
  1. Diagnosis (grade to level of probability)
  2. False positives
  3. Interpretation
- E. Depressions
  1. Diagnosis (grade to level of probability)
  2. False negatives
  3. Interpretations
- F. Variability
  1. Within Day
  2. Between Day
- G. Cost Factor
  1. Technical
  2. Professional
- H. Processing Time
  1. Routine
  2. Stat

PATPART (formerly III. Patient Related Description)

- A. Patient Processing Instructions
  1. STAT
  2. Routine
- B. Procedure Description for Patient

TESTPART (formerly IV. Test Related Description)

- A. Specimen
  1. Volume
  2. Type
  3. Handling Instructions
- B. Processing Procedure
- C. Testing Procedures
  1. Primary Method
     a. Equipment Required
     b. Errors
     c. Technical Requirements (personnel, time, and control)
  2. Alternate Methods
     a. Equipment Required
     b. Errors
     c. Technical Requirements (personnel, time, and control)

REFS (formerly V. References)

APPENDIX D
INQUIRE LOAD UTILITIES

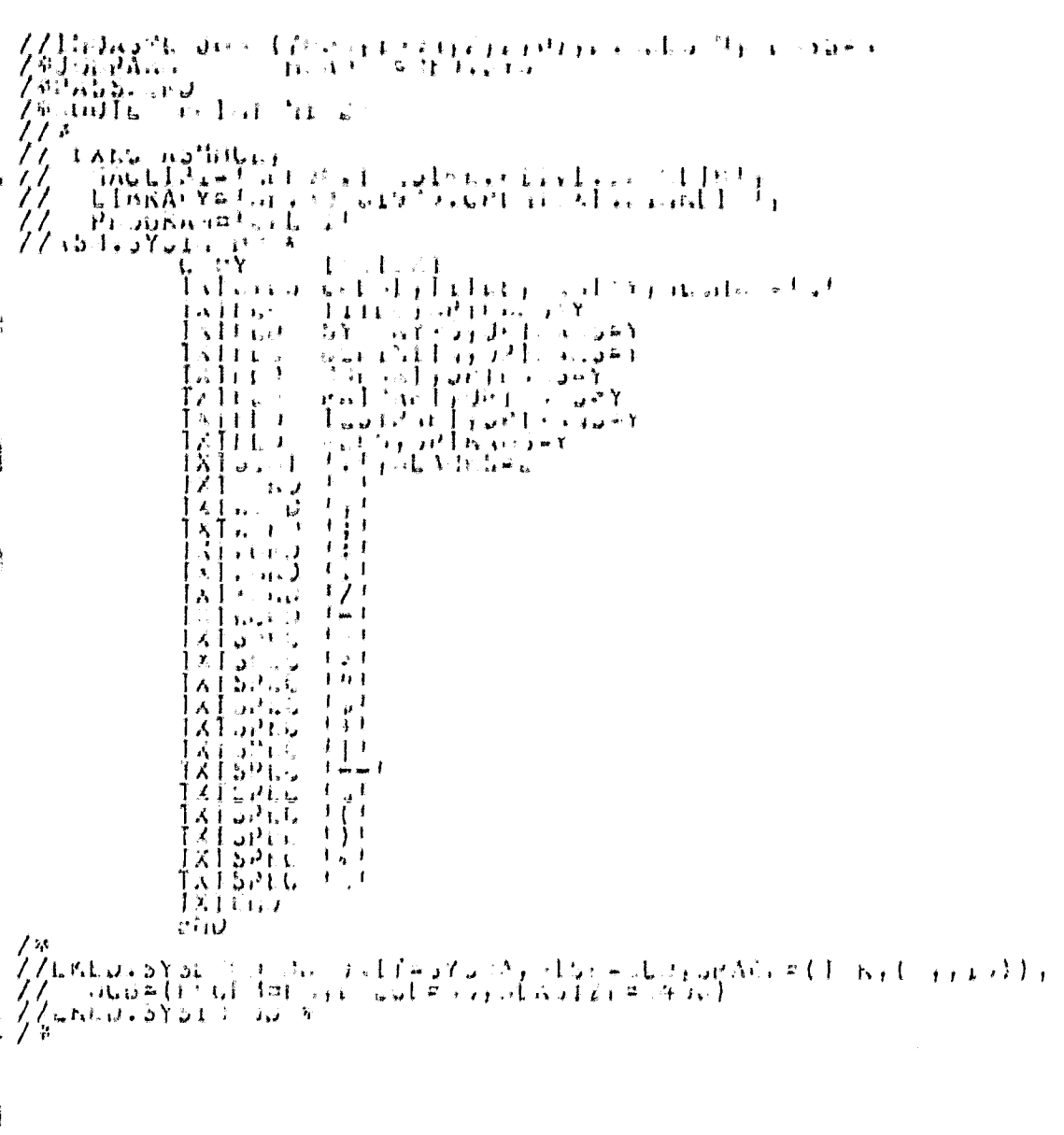
This appendix contains the necessary JCL for loading the INQUIRE CPLM database. The specific jobs and their functions are:

1. INQASML - This job builds the internal structure to be used in future INQUIRE User Language Transactions. INQASML also defines word and sentence delimeters as well as other special delimeters.

2. INQKEXT - This job parses the initial ISI format input stream to extract keywords from the SYNONYMS field. The output of this job is later used to load the actual database.

3. INQLOAD - This job loads the basic CPLM database. At this point searches can be made using the FIND statement only for those keywords parsed by INQKEXT. This is the basic INQUIRE single database method of operation.

4. INQPOST - This multi-step job processes the input text to generate the postings database which when coupled to the master (or biblio) database will allow true generalized text searching capability. This coupling is done in the INQUIRE Macro facility.



ORIGINAL PAGE IS
OF POOR QUALITY

```
//INDKEXT JOB  ([illegible]),[illegible]
/*JOBPARM      [illegible]=[illegible].SYS
/*PASSWORD
/*ROUTE  PRINT [illegible]
//*
//KEYBLD EXEC  [illegible]
//STEPLIB  DD  [illegible]LIVE.LINK[illegible]
//SYSPRINT DD  SYSOUT=A
//SYSOUT   DD  SYSOUT=A
//ABOFILE  DD  SYSOUT=A
//SYSUDUMP DD  SYSOUT=A
//SYS010   DD  [illegible]
//SYS011   DD  [illegible]
//             [illegible],BLKSIZE=[illegible]),UNIT=SYSDA,
//             SPACE=(TRK,(10,5))
//SYS015   DD  *
CONTROLS [illegible] TITLE
FIELD=SYNONYMS
/*
//SYS020     DD *
SYNONYMS
BOTH
AND
FOR
NOT
OR
/*
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FILE: LOAD
//LOADJOB JOB (70),'J04,,,5)',),NOTIFY=&SYSUID
/*JOBPARM
/*PASSWORD
/*ROUTE PRINT NODE1
//*
//          EXEC PGM=IEFBR14
//*
//*    NORMALLY COMMENTED ACTIVE OUT IF FAILS TO ACTIVATE JOB
//*
//DD1         DD DSN=SYS1.ADDS1900.CPLMTEXT.DATA,DISP=(OLD,DELETE)
//DD2         DD DSN=SYS1.ADDS1900.CPLMTEXT.INDEX,DISP=(OLD,DELETE)
//DD3         DD DSN=SYS1.ADDS1900.CPLMTEXT.OVERFLOW,DISP=(OLD,DELETE)
//DD4         DD DSN=SYS1.ADDS1900.CPLMTEXT.SYSUTIL,DISP=(OLD,DELETE)
//*DD5        DD DSN=SYS1.ADDS1900.CPLMTEXT.LOCK,DISP=(OLD,DELETE),
//*             UNIT=SYSDA,VOL=SER=USER00
//*
//LOAD      EXEC PGM=IEFBR14
//INDEX       DD DSN=SYS1.ADDS1900.CPLMTEXT.INDEX,DISP=(,CATLG,KEEP),
//              UNIT=SYSDA,VOL=SER=USER04,SPACE=(CYL,(1,1)),
//              DCB=(DSORG=IS,RECFM=FB,LRECL=12,BLKSIZE=1200,
//              KEYLEN=8,OPTCD=YLEI,CYLOFL=1,NCYLNK=100)
//*
//          EXEC IMPLOAD,
//              ALLVOL=USER00,
//              DSNAME='SYS1.ADDS1900.CPLMTEXT',
//              MASTER='CPLMTI',
//              KEYLEN=100,
//              LRECL=110,
//              INXLEN=120,
//              DATSPC=20,
//              DATSIZE=256K,
//              ERRDISP=DELETE,
//              OKDISP=PASS,
//              DATPARM='IS4=400',
//              SRT=YES,
//              SRTSPC=4,
//              SRTSIZE=37,
//              SRTUSIZE=37,
//              SECPARM='RECVFIL',
//              SORTPGM=SYNCSORT
//DAT.ISIOJ DD DSN=SYS1.ADDS1900.CPLMTEXT.ISIFMT.WKEYS,DISP=(OLD,DELETE)
//DAT.SYSIN DD *
TITLE     VP        94
SYNONYMS  V        145
CODE      V         15
OLFINITN  V       2000
MDPART    V      13400
PATPART   V       8000
TESTPART  V       8000
RLFS      V        900
END
/*
//ILD.INXFIL DD DSN=SYS1.ADDS1900.CPLMILEXT.INDEX,DISP=(OLD,PASS),
//              UNIT=SYSDA,VOL=SER=USER04
```

ORIGINAL PAGE IS
OF POOR QUALITY

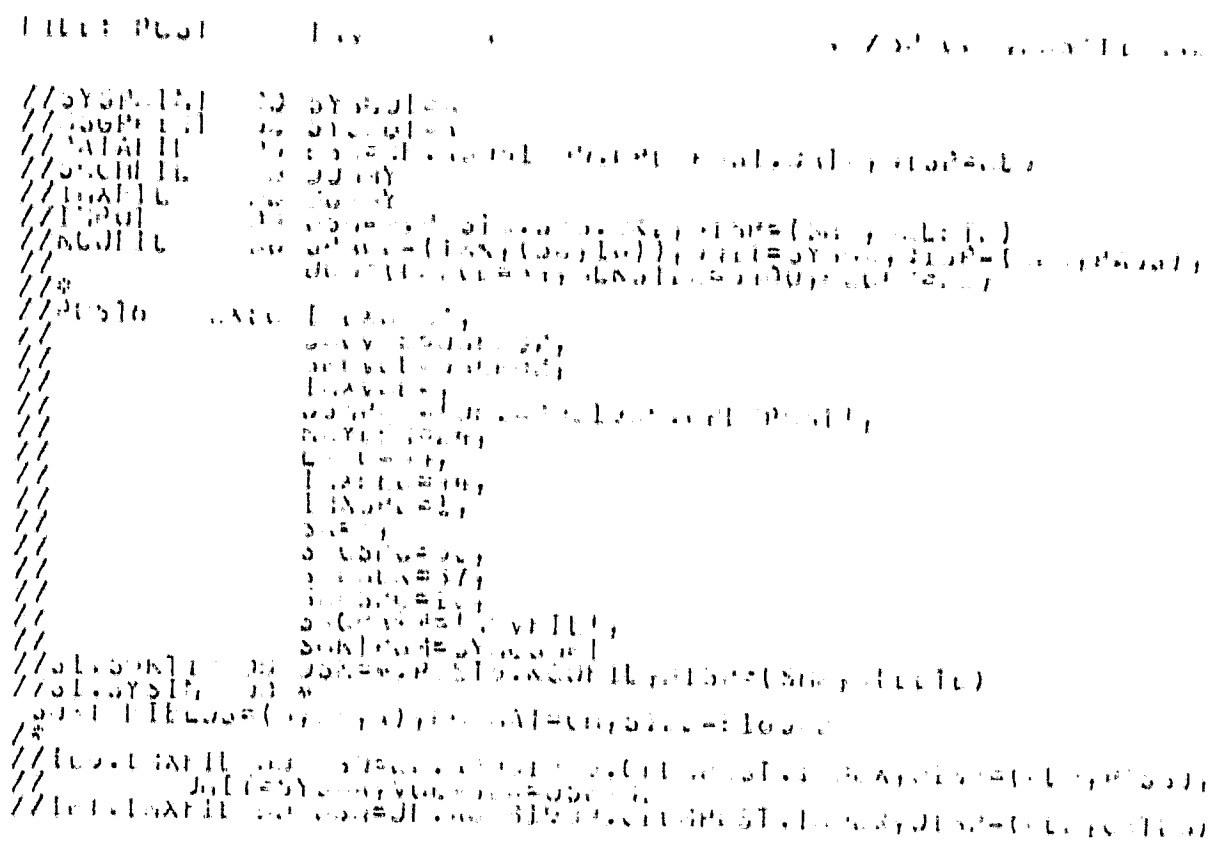
```
FILE: POST      DMN          A                    V1/SP CONVERSATIONAL MONITOR SYST
//DMNPOST  JOB (7000,1904,50,20),'NELSON',CLASS=A,
//             REGION=512K
//*JOBPARM     COPIES=1,LINDEX=ALL.SYS
//*PASSWORD
//*ROUTE PRINT   NLK25
//*
//POSTO    EXEC PGM=IEFBR14
//DD1          DD DSN=UT.A000L2007.CPLIMPOST.DATA,DISP=(OLD,DELETE)
//DD2          DD DSN=UT.A000L1909.CPLIMPOST.INQUIRE,DISP=(OLD,DELETE)
//*DD3           DD DSN=UT.A000L2007.CPLIMPOST.STATCHG,DISP=(OLD,DELETE)
//*DD4           DD DSN=UT.A000L2007.CPLIMPOST.SLAVCHGV,DISP=(OLD,DELETE)
//*DD5           DD DSN=UT.A000L2007.CPLIMPOST.INDEX,DISP=(OLD,DELETE),
//*              UNIT=SYSDA,VOL=SER=DSK002
//*
//DRAB     EXEC PGM=IEFBR14
//INDEX        DD DSN=UT.A000L2007.CPLIMPOST.INDEX,DISP=(NEW,PASS),
//             UNIT=SYSDA,VOL=SER=DSK002,SPACE=(CYL,(1,1)),
//             DCB=(DSORG=IS,RECFM=FB,LRECL=44,BLKSIZE=440),
//             KEY=20,OPTCD=YLRT,CYLOFL=1,NCYLEI=2)
//*
//POSTI    EXEC PGM=IXIEXT,
//             PARM='/DATA=CPLMI,KEYS=5,KEY=1,35,PRINT,ACCESS='
//STEPLIB      DD DSN=CRDC.INQUIRE.LIVE.LINKLIB,DISP=SHR
//             DD DSN=UT.A000L1909.CPLIMEXT.LINKLIB,DISP=SHR
//SYSPRINT     DD SYSOUT=A
//ABENDAID     DD SYSOUT=A
//DATAFILE     DD DSN=UT.A000L1909.CPLIMTEXT.DATA,DISP=SHR
//PRINTFIL     DD UNIT=SYSDA,DISP=(NEW,PASS,DELETE),
//             SPACE=(TRK,(30,5)),
//             DCB=(RECFM=FB,LRECL=30,BLKSIZE=3000)
//SFLOWFIL     DD UNIT=SYSDA,DISP=(NEW,PASS,DELETE),SPACE=(TRK,1),
//             DCB=(RECFM=F,BLKSIZE=80)
//SYSIN        DD *
&
A
ABOUT
ABOVE
ACROSS
AFTER
AGAIN
AGE
AGES
ALONG
ALSO
ALTHOUGH
AMONG
AN
AND
ANY
ARE
AS
AT
B
BE
BEFORE
```

IS
IT
ITS
J
K
KEEP
KN
KNEW
KNOW
L
LIKE
LL
M
MANY
MAY
MORE
MOST
N
NAME
NEARER
NEITHER
NEXT
NINE
NOR
NOT
O
OBTAIN
OBTAINED
OBTAINING
OF
OFF
OFTEN
ON
ONE
ONLY
ONTO
OR
OTHER
OUT
OVER
P
PAST
Q
R
S
SEVEN
SINCE
SIX
SO
SOME
STILL
SUCH
T
TAKE
TAKEN

```
TAKES
THAN
THAT
THE
THEIR
THEM
THEMSELVES
THEN
THERE
THEREAFTER
THEREBY
THEREFORE
THESE
THEY
THIS
THOSE
THREE
THROUGH
TO
TOO
TOWARD
TWO
U
UNCLEAR
UNDER
UNDERTAKEN
UNTIL
UP
UPON
USE
USUALLY
V
W
WAS
WELL
WERE
WHEN
WHERE
WHEREAS
WHETHER
WHICH
WHILE
WILL
WITH
X
Y
YOU
YOUR
Z
/*
//*
//POST2   EXEC PGM=SYNCSORT
//SYSOUT     DD SYSOUT=*
//SORTLIB    DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01   DD UNIT=SYSDA,SPACE=(CYL,(10,5)),DISP=(NEW,DELETE)
```

```
FILE: PL/I
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSIN     DD DSN=&&PLIDIN,DISP=(MOD,PASS)
//SORTIN    DD DSN=&&SORTIN,DISP=(MOD,PASS)
//SORTOUT   DD UNIT=SYSDA,DISP=(,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
//             SPACE=(TRK,(30,5))
//*
//POSTO    EXEC PGM=POSTOK
//OUT1      DD DSN=&&POSTOK,DISP=(NEW,PASS)
//OUT2      DD DSN=&&POSTOK(INDEX),DISP=(NEW,PASS),
//             SPACE=(TRK,(5,5)),DCB=(BLKSIZE=3600)
//SYSIN     DD *
DATA_DSI;
   INFILE OUT1;
   FILE OUT2;
   INPUT @1 KEY $20. @1 RECORD $80.;
   IF KEY=LAST THEN PUT RECORD $80.;
   CLOSE DEL 1;
*PROC FILES;
*   TABLES KEY/
*   SUPPLIN  SUPPLIN;
*PROC SORT DATA=OUT; BY OFFSET RECORD;
*PROC PRINT DATA=OUT;
*   TITLE1 'ROUTINE PRINTOUT OF SEARCH KEY AND FILE';
*   TITLE2 'TITLE2 OF THE PROGRAM';
/*
//*
//POSTO    EXEC PGM=POSTOK,
//             PARM='LIBRARY',
//             DSNPRE=SYS1.CPLMPOST,
//             KEYLEN=20,
//             LRECL=80,
//             BLKSIZE=6400,
//             OUTPUT='OUT2',
//             DATA=2500
//DATA.SYSIN DD *
KEYWORD   SOURCE
END
/*
//DATA.SYSIN DD *
KEYWORD   VAR   CHAR
POSITION  N     1
NAME      N     2
POSITION  F     3          ABOVE      F STRING
ITEM      N     4          SPOSITION 1
FIELDNO   N     1          SPOSITION 3
SENTNO    N     2          SPOSITION 5
WORDNO    N     2          SPOSITION 7
END
/*
//*
//POSTO    EXEC PGM=IXITP,
//             PARM='/MOD=L,O=CPLM.2,OK',
//STEPLIB   DD DSN=SYS1.CPLMPOST.LOADLIB,DISP=SHR
```

ORIGINAL PAGE IS
OF POOR QUALITY



APPENDIX E
PL/1 Listing of Text Preprocessor

SOURCE LISTING

```
STMT LEV NT

   1      0 !PARSE:  PROCEDURE OPTIONS (MAIN);
            !/*****************************************************************/
            !/*                                                               */
            !/*  Title:  PARSE                                                */
            !/*                                                               */
            !/*  Purpose:  to format text in ISI format for use with the      */
            !/*            INQUIRE database system.                           */
            !/*                                                               */
            !/*  Written by:  Diane Nelson, Medical Systems, UF               */
            !/*                                                               */
            !/*  Date:  03Dec81              Version:  1.0                    */
            !/*                                                               */
            !/*  Input:  free text in 80 byte fixed length records,           */
            !/*          stream input read from the file identified           */
            !/*          by the ddname TEXT.                                  */
            !/*                                                               */
            !/*  Output:  the text in ISI format with end of line             */
            !/*           characters inserted at the end of each 80 byte      */
            !/*           input record.                                       */
            !/*                                                               */
            !/*  Internal procedures used:                                    */
            !/*     CARDIN  -  get another 80 byte input record and set       */
            !/*                the variables ZONE1 and ZONE2.                 */
            !/*     TOP     -  process the top section of each item read in.  */
            !/*     ITEM    -  process the rest of each item read in.         */
            !/*     LINEOUT -  print the output records with end of line      */
            !/*                characters.                                    */
            !/*     ERROR   -  print the input record if an error occurs.     */
            !/*                                                               */
            !/*****************************************************************/
```

```
STMT LEV NT

   2   1  0 :  DECLARE  /*  files  */
             :     TEXT FILE STREAM INPUT,
             :     ISI FILE STREAM OUTPUT,
             :     MSG FILE PRINT;
             :
   3   1  0 :  DECLARE  /*  character variables  */
             :     CARD       CHAR (80)       INIT (''),
             :     EOL        CHAR (1)        INIT ('@'),
             :     LASTCHAR   CHAR (1)        INIT (''),
             :     LETTER     CHAR (1)        INIT (''),
             :     LINE       CHAR (80) VAR   INIT (''),
             :     MDPART     CHAR (8)  VAR   INIT ('MDPART'),
             :     PATPART    CHAR (8)  VAR   INIT ('PATPART'),
             :     REFS       CHAR (8)  VAR   INIT ('REFS'),
             :     TESTPART   CHAR (8)  VAR   INIT ('TESTPART'),
             :     ZONE1      CHAR (4)        INIT (''),
             :     ZONE2      CHAR (1)        INIT ('');
             :
   4   1  0 :  DECLARE  /*  numeric integer variables  */
             :     I          FIXED           INIT (0),
             :     LEN        FIXED           INIT (0),
             :     PTR        FIXED           INIT (0);
             :
   5   1  0 :  DECLARE  /*  boolean constants  */
             :     TRUE       BIT (1)         INIT ('1'B),
             :     FALSE      BIT (1)         INIT ('0'B);
             :
   6   1  0 :  DECLARE  /*  boolean variables  */
             :     EOF        BIT (1)         INIT ('0'B),
             :     FIRSTMSG   BIT (1)         INIT ('1'B),
             :     FIRSTREC   BIT (1)         INIT ('1'B);
             :
   7   1  0 :  DECLARE  /*  builtin functions  */
             :     LENGTH     BUILTIN,
             :     SUBSTR     BUILTIN;
```

```
STMT LEV NT

                    /*  main procedure:  PARSE  */
   8   1  0         ON ENDFILE (TEXT) EOF = TRUE;

   9   1  0         OPEN FILE (TEXT) INPUT;
  10   1  0         OPEN FILE (ISI) OUTPUT;
  11   1  0         OPEN FILE (MSG) PRINT;

  12   1  0         CALL CARDIN;
  13   1  0         DO WHILE (^EOF);
                       /*  delete blank lines at the beginning of the file  */
  14   1  1            DO WHILE (CARD = ' ' & FIRSTREC & ^EOF);
  15   1  2               CALL CARDIN;
  16   1  2            END;

  17   1  1            IF ZONE1 = ' ' THEN DO;
  18   1  2               LINE = SUBSTR (CARD,5);
  19   1  2               CALL LINEOUT;
  20   1  2            END;

  21   1  1            ELSE DO;  /*  ZONE1 is not blank  */
  22   1  2               SELECT (ZONE1);
  23   1  3                  WHEN ('I.  ') CALL TOP;
  24   1  3                  WHEN ('II. ') CALL ITEM (MDPART);
  25   1  3                  WHEN ('III.') CALL ITEM (PATPART);
  26   1  3                  WHEN ('IV. ') CALL ITEM (TESTPART);
  27   1  3                  WHEN ('V.  ') CALL ITEM (REFS);
  28   1  3                  OTHERWISE CALL ERROR;
  29   1  3               END;
  30   1  2            END;
                       /*  if ZONE1 = 'II. ', the next record has already been read  */
  31   1  1            IF ZONE1 ^= 'II. ' THEN CALL CARDIN;
  32   1  1         END;

                    /*  write out the END statement for the last item processed  */
  33   1  0         PUT FILE (ISI) EDIT ('END') (COL(1),A);

  34   1  0         CLOSE FILE (TEXT);
  35   1  0         CLOSE FILE (ISI);
  36   1  0         CLOSE FILE (MSG);
```

```
STMT LEV NT

  37   1  0  CARDIN:  PROCEDURE;
              /*****************************************************************/
              /*                                                               */
              /*  Title:  CARDIN                                               */
              /*                                                               */
              /*  Purpose:  read the next 80 byte input record and set the     */
              /*            character variables ZONE1 and ZONE2 using the      */
              /*            SUBSTR builtin function.                           */
              /*                                                               */
              /*  Input:  fixed 80 byte records from the file TEXT.            */
              /*                                                               */
              /*****************************************************************/

  38   2  0     GET FILE (TEXT) EDIT (CARD) (A(80));
  39   2  0     ZONE1 = SUBSTR (CARD,1,4);
  40   2  0     ZONE2 = SUBSTR (CARD,5,1);

  41   2  0  END CARDIN;
```

```
STMT LEV NT

  42   1  0 TOP:  PROCEDURE;
                 /*****************************************************************/
                 /*                                                               */
                 /*   Title:  TOP                                                 */
                 /*                                                               */
                 /*   Purpose:  to process section I of each item.                */
                 /*                                                               */
                 /*   Procedures used:  CARDIN, LINEOUT                           */
                 /*                                                               */
                 /*****************************************************************/

                    /*  output 'END', except if this is the first record  */
  43   2  0       IF FIRSTREC THEN FIRSTREC = FALSE;
  44   2  0       ELSE PUT FILE (ISI) EDIT ('END') (COL(1),A);

                    /*  label the TITLE field  */
  45   2  0       PUT FILE (ISI) EDIT ('TITLE') (COL(1),A);

                    /*  get the title and print it  */
  46   2  0       GET FILE (TEXT) EDIT (CARD) (A(80));
  47   2  0       LINE = SUBSTR (CARD,9);
  48   2  0       PTR = 10;
  49   2  0       CALL LINEOUT;

                    /*  read another card  */
  50   2  0       CALL CARDIN;

                    /*  process section I  */
  51   2  0       DO WHILE (ZONE1 = ' ' & ^EOF);
  52   2  1          IF ZONE1 ^= ' ' THEN DO;
  53   2  2             SELECT (ZONE2);
  54   2  3                WHEN ('B') PUT FILE (ISI) EDIT ('SYNONYMS') (COL(1),A);
  55   2  3                WHEN ('C') PUT FILE (ISI) EDIT ('CODE') (COL(1),A);
  56   2  3                WHEN ('D') PUT FILE (ISI) EDIT ('DEFINITN') (COL(1),A);
  57   2  3                OTHERWISE CALL ERROR;
  58   2  3             END;

                       /*  change the output pointer value          */
                       /*  if this is the beginning of a new field  */
  59   2  2          PTR = 10;
  60   2  2       END;
  61   2  1       LINE = SUBSTR (CARD,9);

                    /*  print the line, then read a new card  */
  62   2  1       CALL LINEOUT;
  63   2  1       CALL CARDIN;
  64   2  1    END;

  65   2  0 END TOP;
```

```
STMT LEV NT

  66   1  0  ITEM:  PROCEDURE (PART);
              /*****************************************************************/
              /*                                                               */
              /*   Title:  ITEM                                                */
              /*                                                               */
              /*   Purpose:  to process sections II through V of each item     */
              /*             read in.                                          */
              /*                                                               */
              /*   Procedures used:  CARDIN, LINEOUT                           */
              /*                                                               */
              /*****************************************************************/

  67   2  0     DECLARE  /*  parameters  */
                   PART      CHAR (*)  VAR;

                 /*  output the fieldname  */
  68   2  0     PUT FILE (ISI) EDIT (PART) (COL(1),A);

                 /*  read the next card to get to the beginning of the text,  */
                 /*  then print it                                            */
  69   2  0     CALL CARDIN;
  70   2  0     LINE = SUBSTR (CARD,5);
  71   2  0     PTR = 10;
  72   2  0     CALL LINEOUT;

  73   2  0  END ITEM;
```

```
STMT LEV NT

  74   1  0 LINEOUT:  PROCEDURE;
                /*****************************************************************/
                /*                                                               */
                /*   Title:  LINEOUT                                             */
                /*                                                               */
                /*   Purpose:  to output a file containing the reformatted text. */
                /*                                                               */
                /*   Output:  the output file (ISI) contains fieldnames in       */
                /*            columns 1-8, a blank in column 9, and text in      */
                /*            columns 9-72.  An end of line character follows    */
                /*            each line of input text.  The word 'END' in        */
                /*            columns 1-3 terminates each item in the file.      */
                /*                                                               */
                /*****************************************************************/

  75   2  0    LEN = LENGTH (LINE);
  76   2  0    LASTCHAR = SUBSTR (LINE,LEN,1);
                /*  strip off trailing blanks  */
                /*  ignor blank lines  */
  77   2  0    DO WHILE ((LASTCHAR = ' ') & (LINE ^= ' '));
  78   2  1       LEN = LEN - 1;
  79   2  1       LASTCHAR = SUBSTR (LINE,LEN,1);
  80   2  1    END;

                /*  insert a blank at the beginning of the line  */
  81   2  0    PTR = PTR + 1;

                /*  print the line  */
  82   2  0    DO I = 1 TO LEN;
  83   2  1       LETTER = SUBSTR (LINE,I,1);
  84   2  1       IF PTR = 73 THEN PTR = 10;
  85   2  1       IF PTR = 74 THEN PTR = 11;
  86   2  1       PUT FILE (ISI) EDIT (LETTER) (COL(PTR),A);
  87   2  1       PTR = PTR + 1;
  88   2  1    END;

                /*  output the end of line (EOL) character  */
  89   2  0    IF PTR = 73 THEN PTR = 10;
  90   2  0    PUT FILE (ISI) EDIT (EOL) (COL(PTR),A);
  91   2  0    PTR = PTR + 1;

  92   2  0 END LINEOUT;
```

```
STMT LEV NT

 93   1  0  ERROR:  PROCEDURE;
             /*****************************************************************/
             /*                                                               */
             /*   Title:  ERROR                                               */
             /*                                                               */
             /*   Purpose:  to print all records that are in error.           */
             /*                                                               */
             /*   Output:  to the file MSG.  A header is printed followed     */
             /*            by the error records as they were read in.         */
             /*                                                               */
             /*****************************************************************/

             /*  print a heading if this is the first record in error,  */
             /*  otherwise, just print the record.                      */
 94   2  0     IF FIRSTMSG THEN DO;
 95   2  1        PUT FILE (MSG) EDIT ('The following records are in error:')
                    (COL(1),A);
 96   2  1        FIRSTMSG = FALSE;
 97   2  1     END;
 98   2  0     PUT FILE (MSG) EDIT (CARD) (COL(1),A);

 99   2  0  END ERROR;


100   1  0  END PARSE;
```

APPENDIX F
INQUIRE Macros Used by CPLM

This appendix contains information used by programmers in documenting and maintaining the INQUIRE CPLM Macros. The material is currently organized as:

1. Macro List with Command References

2. Built-in INQUIRE Macros

3. Macro structures for selected commands

   A. COUNT N
   B. COUNT 1 IF MDPART IS HEART
   C. DISPLAY LUNG
   D. PRINT N
   E. HISTORY
   F. HISTORY 1 2
   G. FREE 3
   H. SEARCH LUNG 2 5 FLUKE
   I. SEARCH LUNG AND LIVE

4. Programmer notes on changes to INQUIRE CPLM Macros

5. CPLM Macros

MACRO LIST WITH COMMAND REFERENCES

$FORM	print
$FUNC	
$F1	
$F2	print
$OPERS	
$SYN	
ADJ	search
ALLSYN	search
AND	search
ANDOR	count
ANDORCHK	
ATEQ	count
BEGIN	begin
BEX	display
BLOOP	display
BPROMPT	display
BRWS	display
CHF	print
CHKHL	print
CHL	print
CHS	print
CMDSTART	begin
CNTEM	count
COUNTDCS	print search
DBNERR	print
DEFLD	search
DELTDBW	print search count
DICT	dict
DICTOP	dict
DICTOP1	dict
DSPL	history
DSPLOOP	history
EQCHK	print count
FALSE	print search count
FORMCHK	print
FREE	free
FREECUR	search
FREEIT	free
FULLDSP	history
FULLFREE	free
FVCOPCHK	print count
FVVAL	print count
GET	OFFPRES
GETFORM	
GETSYN	search
HCOUN	help
HCOUND	help
HCOUNE	help
HCOUNF	help
HCOUNO	help

```
HCOUNS     help
HDICT      help
HDICTD     help
HDICTE     help
HDICTF     help
HDICTN     help
HDICTO     help
HDICTS     help
HDISP      help
HDISPD     help
HDISPE     help
HDISPF     help
HDISPN     help
HDISPO     help
HDISPS     help
HELP       help
HELPGO     help
HEND       help
HFREE      help
HFREED     help
HFREEE     help
HFREEF     help
HFREEO     help
HFREES     help
HHELP      help
HHELPD     help
HHELPE     help
HHELPF     help
HHELPN     help
HHELPO     help
HHELPS     help
HHIST      help
HHISTD     help
HHISTE     help
HHISTF     help
HHISTN     help
HHISTO     help
HHISTS     help
HL         print
HLCONT     print
HLHELP     print
HLT        print
HOLDLOOP   history
HOULT      help
HOULTD     help
HOULTE     help
HOULTF     help
HOULTO     help
HOULTS     help
HPRIN      help
HPRIND     help
HPRINE     help
HPRINF     help
```

HPRINN	help
HPRINO	help
HPRINS	help
HSEAR	help
HSEARD	help
HSEARE	help
HSEARF	help
HSEARN	help
HSEARO	help
HSEARS	help
HSET	help
HSETD	help
HSETE	help
HSETF	help
HSETN	help
HSETO	help
HSETS	help
HSHOW	help
HSHOWD	help
HSHOWE	help
HSHOWF	help
HSHOWO	help
HSHOWS	help
HSUMM	help
HSUMMD	help
HSUMME	help
HSUMMF	help
HSUMMO	help
HSUMMS	help
HUSER	help
IFANDNOT	could be used
IFFLD	search
IFKNOWN	used w/OFFPRES
INQ	INQ
INSCOMMA	all
INSPARM	count
IOLOOP	count
IOLOOP2	count
KEYFCHK	print count
MERGE	search
NEWSRCH	search
NEXTWORD	search
NKWORD	count
NODBCHK	history
NXTFLD	count
NXTWORD	count
OFFPRES	ZERO RECORDS
OPCHK	search
OPER	search
OPERROR	search
OPER2	search
OPER3	search
OPGO	search

OPNOT	search
OUTL	outline
OUTLD	outline
OUTLL	outline
P	print
POPCHK	search
PREDEF	DISPDEF reference not found
PRSFVC	print count
RELEV	&CURLEV
REST	&RESTART
RETAND	search
SCAN	print
SCANCNT	count
SCAN21	print
SETBRIEF	set
SETFIND	print
SETFORM	print
SETOFF	print
SETOP	set
SETDB	print count
SETSCAN	print
SHOW	show
SHOW1	show
SHOW2	show
SHOW3	show
SHOW4	show
SRCH1	search
SS	search
SSETFIND	count
SSETSCAN	count
STARTUP	begin
SUMMARY	summarize
SUMMARY1	summarize
TDB	search
TDBCHK	search
TEMPSTART	begin
TOCHK	count
TOP	all
TOP2	all
UPCP	count
UPHOLD	print
W	search
WHOLD	history
WSET	search

```
+1
ACCNO
ANDIF(value1, operator, value2)
ASK(question)
ATTN
CALL(&PARM)
CURITEM(n)
DATE
DATE(MDY)
DATE(DMY)
DBNAME
DBNAME(n)
ELSE
EXIT
EXIT(n)
GOTO
HOLDCNT(n)
HOLDDB(n)
HOLDINDX(name)
HOLDNAME
HOLDTYPE(n)
IF(text)
IF(value1, operator, value2)
IFITEMS(n)
INBUFF
INDEX(text1, text2)
ITEMSFND
LASTMAC
LASTSAVE
LENGTH(text)
MEND
NONMAC
NOPRINT
NULL
NUMDB
NUMHOLD
ORIF(value1, operator, value2)
OSJOB
OSSTEP
PADL(text, len)
PADR(text, len)
PAGE
PARM
PARM(n)
PARMSET(text)
READ(prompt)
RESTART
RESTART(macro name)
RETURN
RETURN(text)
SOURCE
```

```
STARTUP
STOPEXIT
SUBSTR(text, start, length)
THEN
THEN(text)
TIME
TODAY
TPSOURCE
TRIM(text)
TSOUSER
UPCASE(text)
WRITE(text)
```

Macro structure for "COUNT, n "

I. TOP
II. TOP2
III. CNTEM
  A. DOCDB
  B. PARM
  C. SETDB
    1. CURWORD
    2. +1
    3. PARM
IV. SCANCNT
  A. DBN
Program Restarts

CNTEM is the "home base" or controling macro for the command COUNT. CNTEM, nested in the 2nd level, processes the example command 'COUNT,n' in the following manner:

&DBN is initialized to &DOCDB (CPLM1). If &PARM(2) is not 'IF', SETDB sets &DBN to the value n in &PARM(2). Since no further operands exists, level 2 control branches to SCANCNT. SCANCNT executes the statement:

SCAN IN &DBN, COUNT, HOLD UNCCNT. For this example &DBN is replaced by n. The temporary data set 'UNCCNT' may not be printed so it is deleted to save space. Control at level 2 branches to TOP.

Important Operational Macros

CNTEM	level 2.	Controls &SETDB, Initializes &DBN to &DOCDB, and checks for boolean condition.
SETDB	level 3 (in CNTEM).	Sets &DBN to &PARM(2).
SCANCNT	level 2.	Executes the SCAN and COUNT statement, and deletes the temporary data set created by that statement.

Macro Structure for "COUNT,1,IF,MDPART,IS,HEART"

```
I.    CNTEM
      A.  DOCDB
      B.  PARM
      C.  SETDB
            1.  CURWORD
            2.  +1
            3.  PARM
      D.  +1
      E.  PARM
      F.  +1
      G.  PARM
II.   SSETSCAN
      A.  PRSFVC
            1.  FVCOPCHK
      A.    PRSFVC (EXIT)
            2.  TEMP
            3.  CURFLD
            4.  CURWORD
            5.  +1
            6.  PARM
            7.  FVVAL
                  a)  +1
                  b)  PARM
                  c)  TEMP2
                  d)  CURVAL
            8.  FVVAL (by recursion)
                  a)  +1
                  b)  PARM
                  c)  REST
      A.    PRSFVC (EXIT)
            9.  TEMP
           10.  TEMP2
           11.  +1
           12.  PARM
           13.  REST
      B.  DBN
      C.  TEMP
*Program Restart*
```

Initial processing of a COUNT statement with a boolean phrase is the same as processing an unconditional COUNT. CNTEM (level 2) sets &DBN to the desired data set in SETDB (level 3), then checks for an IF statement. In this example, IF was found, and control at level 2 branched to SSETSCAN. SSETSCAN processes the boolean phrase and makes the query. Within SSETSCAN, PRSFVC processes the field value comparison and places it into &TEMP. The routine is recursive for multiple comparisons. The boolean phrase consists of a fieldname, an operator, and a field variable value or target value. Within PRSFVC, FVVAL (level 4) accumulates the target values into &TEMP2. FVVAL is also recursive to handle multiple target values. When all boolean

strings have been accumulated into &TEMP, SSETSCAN executes the statement: SCAN IN &DBN &TEMP, COUNT, HOLD CONCNT.

For this example, the statement reads:

SCAN IN 1 MDPART IS (HEART), COUNT, HOLD CONCNT.

The temporary data set CONCNT, used for the conditional count, cannot be printed so it is deleted before the program is restarted.

* Important Operational Macros *

CNTEM	level 2.	Determines data set in which counting occurs, Checks for IF statement.
SETDB	level 3 (in CNTEM).	Sets &DBN according to &PARM(2).
SSETSCAN	level 2.	Executes a SCAN statement containing a boolean phrase.
PRSFVC	level 3 (in SSETSCAN).	Sets &TEMP to the INQUIRE form of the given boolean condition. (Recursive)
FVCOPCHK	level 4 (in PRSFVC).	Checks validity of the field value comparison operator.
FVVAL	level 4 (in PRSFVC).	Sets &TEMP2 to a collection of the field variable values within parenteses. (Recursive)

DISPLAY,LUNG

```
I.   BRWS
II.  BLOOP
       A.   +1
       B.   TEMP
       C.   PARM
III. BLOOP
       A.   +1
       B.   TEMP
       C.   THEN
       D.   PARM
       E.   REST
IV.  BEX
       1.   TEMP

*RESTART PROGRAM*

FIND IN INDEX LUNG*, TAB, BREAK ON KEYWORD TOTAL OF POSTCNT
KEYWORD 12, TITLE (NUMBER OF POSTINGS) 1 ('' ''KEYWORD)12.
```

PRINT, n

```
I.   P
       A.  DOCDB
       B.  $FORM
       C.  PARM
       D.  SETOFF*
       E.  PARM
       F.  SETDB
             1.  CURWORD
             2.  +1
             3.  PARM
       G.  CALL (SCAN)
             1.  DBN
       H.  HLCONT
       I.  HL
             1.  HLT
                   a)   RETURN          ┐
                          1)  PARM      ├─ HILITE COMMANDS
                                        ┘
       J.  HLCONT

*RESTART PROGRAM*
```

HISTORY

I. DSPL
II. FULLDSP
    A. NODBCHK
    B. HOLDLOOP
        1. +1
        2. PARM
            a) WHOLD
                1) I
                2) TEMP
                    a. TDBn

PROGRAM RESTART

HISTORY,1,2

```
I.   DSPL
II.  DSPLOOP
     A.  +1
     B.  PARM
     C.  TEMP
         1.  TDB1
III. DSPLOOP
     A.  +1
     B.  PARM
     C.  TEMP
         1.  TDB2
IV.  DSPLOOP
     A.  +1
     B.  PARM
     C.  REST

*PROGRAM RESTART*
```

```
I.   FREE
II.  FREEIT
     A.  +1
     B.  PARM
     C.  PARM
III. FREEIT
     A.  +1
     B.  PARM
     C.  REST

*RESTART PROGRAM*
```

SEARCH,LUNG,W,5,FLUKE

```
I.    TOP
II.   TOP2
III.  SRCH1
      A.  IFFLD
IV.   NEXTWORD
      A.  +1
      B.  PARM
      C.  OPCHK
      D.  FALSE
      E.  TDBCHK
          1.  +1
      F.  CURWORD
      G.  INDFLDS
      H.  DOCDB
      I.  IFITEMS
      J.  THEN
          1.  HOLDNAME
          2.  HOLDNAME
      K.  DEFTDBW
          1.  +1
          2.  NHOLD
          3.  PARM
          4.  NHOLD
          5.  TEMP
              a)  TDBn
      L.  HOLDNAME
      M.  PREVHOLD
      N.  HOLDNAME
V.    NEXTWORD
      A.  +1
      B.  PARM
      C.  OPCHK
VI.   OPER
VII.  OPER2
      A.  WSET
          1.  +1
          2.  PARM
      B.  OPNAME
      C.  PREVHOLD
      D.  RETAND
          1.  CURWORD
VIII. NEXTWORD
      A.  +1
      B.  PARM
      C.  OPCHK
      D.  FALSE
      E.  TDBCHK
          1.  +1
      F.  CURWORD
```

```
      G.  INDFLDS
      H.  DOLDB
      I.  IFITEMS
      J.  THEN
          1.  HOLDNAME
          2.  HOLDNAME
      K.  DEFTDBW (See IV K)
      L.  HOLDNAME
      M.  PREVHOLD
      N.  HOLDNAME
IX.   NEXTWORD
      A.  +1
      B.  PARM
X.    OPER
XI.   OPER3
      A.  OPGO
      B.  W
          1.  WORDS
          2.  OPNAME
          3.  PREVHOLD
          4.  DEFTDBW (See IV K)
      C.  NEWSRCH
          1.  COUNTDCS
              a)  HOLDNAME

*PROGRAM RESTART*
```

SEARCH,LUNG,AND,LIVER

```
I.   SRCH1
     A.  IFFLD
II.  NEXTWORD
     A.  +1
     B.  PARM
     C.  OPCHK
     D.  FALSE
     E.  TDBCHK
         1.  +1
     F.  CURWORD
     G.  INDFLDS
     H.  DOCDB
     I.  IFITEMS
     J.  THEN
         1.  HOLDNAME
         2.  HOLDNAME
     K.  DEFTDBW
     L.  HOLDNAME
     M.  PREVHOLD
     N.  HOLDNAME
III. NEXTWORD
     A.  +1
     B.  PARM
     C.  OPCHK
IV.  OPER
V.   OPER2
     A.  OPNAME
     B.  PREVHOLD
     C.  RETAND
         1.  CURWORD
VI.  NEXTWORD
     A.  +1
     B.  PARM
     C.  OPCHK
     D.  FALSE
     E.  TDBCHK
         1.  +1
     F.  CURWORD
     G.  INDFLDS
     H.  DOCDB
     I.  IFITEMS
     J.  THEN
         1.  HOLDNAME
         2.  HOLDNAME
     K.  DEFTDBW
     L.  HOLDNAME
     M.  PREVHOLD
     N.  HOLDNAME
```

```
VII. NEXTWORD
     A.  +1
     B.  PARM
VIII.OPER
IX.  OPER3
     A.  OPGO
     B.  AND
         1.  OPNAME
         2.  PREVHOLD
         3.  DEFTDBW
     C.  NEWSRCH
         1.  COUNTDCS
             a)  HOLDNAME

*RESTART PROGRAM*
```

NOTES ON CHANGES TO INQUIRE CPLM MACROS

PRINT

The maximum number of lines printed by a HILITE command is now set by the OPTION PGDEPTH 22 statement, found in &STARTUP.

HILITE COMMANDS

HILITE commands have been renamed and one has been added. See &HLHELP.

$F1, $F2, FORMCHK, CHL, CHF, CHS

The macros apparently have no pertinence to the CPLM1 database.

other questionable macros are
PREDEF
$OPERS
IFKNOWN, GET, OFFPRES
OFFPRES has been deleted

HELP

HELP is now menu driven as well as automatic. No segment may be specified when the command is menu driven. See &HELP. The HELP macros are now broken up so the user may finish reading the screen before more text appears. The user is prompted to "PRESS <CR> TO CONTINUE, <BREAK> TO QUIT."

COMMAS

The COMMAS option is set to OFF in &STARTUP. &INSCOMMA is invoked from &TOP to replace blanks with commas. If the user wishes to use embedded blanks, COMMAS must be set to ON using the command: 'SET COMMAS ON'.

DOCCNT

The DOCCNT option is not set to OFF in &STARTUP. The last data set created by each SEARCH command is automatically counted. The temporary data set DOCCNT used to count the data base is automatically deleted.

COUNT

Temporary data sets created by COUNT statements cannot be printed so they are deleted before the user is prompted for another command. UNCCNT is used for an unconditional

COUNT while CONCNT is used when a boolean condition is specified. See &SCANCNT and &SSETSCAN

KEYFCHK

KEYFCHK has been deleted. It was not relevant to the keyfields of the CPLM1 database. The invoking statement &KEYFCHK(CURFLD) located in &CNTEM has been removed. The three macro statements following &GOTO(SSETSCAN) in &CNTEM seem to have no value in the absence of &KEYFCHK. Current unknown if &SSETFIND or &EQCHK are invoked from any other macro.

SEARCH

'&DEFTDBW(&CURWORD)' has been moved to come after the HOLD MERGE statement in &NEXTWORD. The display from a SEARCH command now reads:

TEMPORARY DATABASE n CONTAINS x POSTINGS FROM CPLM1
SET n - name

TEMPORARY DATABASE DOCCNT CONTAINS y ITEMS FROM CPLM1
for all SEARCH queries.

```
$FORM
          &IF(TLx1)&EXIT(-1)                                              00010010
          &IF(OUTLINE)&EXIT(-1)                                           00020020
          &IF(FULL)&EXIT(-1)                                              00030030
          &IF(###)                                                        00040040
4-RECORDS
$FUNC
          &IF(FUNCTIONS)&EXIT(-1)                                         00010
          &IF(FUNC)&EXIT(-1)                                              00020
          &IF(F)                                                          00030
3-RECORDS
$F1
          , TAB TITLE (NL) / DEFINITN (NL) / SYNONYMS (NL) /
            CODE (NL) / NOPART (NL)
2-RECORDS
$F2
          ,TAB CAMPNO MAKE 12 MODEL 38 YEAR 64,TITLE (RECALL NUMBER) C/CAMPNO
          (  MAKE) L/MAKE (  MODEL) L/MODEL (YEAR) L/YEAR
          ,SORT YEAR
          &EXIT(-2)                                                       22.
4-RECORDS
$OPERS
          &IF(OPERANDS)&EXIT(-1)                                          00010
          &IF(OPER)&EXIT(-1)                                              00020
          &IF(O)                                                          00030
3-RECORDS
$SYN
          &IF(SYNTAX)&EXIT(-1)                                            00010
          &IF(S)                                                          00020
2-RECORDS
ADJ
          OPTION MSGLIM 1.                                                31.
          HOLD MATCH WITHIN +1 WORD &OPNAME &PREVHOLD AS UNIQUE.          32.
          OPTION *.                                                       33.
          &DEFTOAR(ADJACENT WORDS - &OPNAME &PREVHOLD)                    34.
          &GOTO(NEXTSRCH)                                                 35.
5-RECORDS
ALLSYN
          OPTION MSGLIM 5.
          FIND IN INDEX &OURKWRD,COUNT,HOLD POSTING &INDFLDS AS SYNT FROM
          &OUCOb .
          &IFITEMS(1)&ELSE&WRITE(&OURKWORD)&ELSE(OPTION *.)
          &ELSE&WRITE(KEYWORD NOT FOUND IN TEXT)&ELSE&RESTART(TOP)
          MACRO DEFINE TI SYNS.
          FIND IN SYN &OURKWRD.* AND IN INDEX,COUNT,HOLD POSTING &INDFLDS AS
          SYNS FROM &OUCOb.
          &IFITEMS(1)&ELSE(MACRO DEFINE TI.)
          OPTION *.
          HOLD MERGE &TI SYNT AS UNIQUE FREE.
          OPTION *.
          &DEFTOAR(&SMSG &OURKWORD)
          MACRO DEFINE SMSG.
          MACRO DEFINE CONHOLD &HOLDNAME.
          MACRO DEFINE PREVHOLD &PREVHOLD &HOLDNAME.
          &GOTO(NEXTWORD)
17-RECORDS
AND
          OPTION MSGLIM 1.                                                55.
          HOLD MATCH SAME DOCUMENT &OPNAME &PREVHOLD AS UNIQUE.           56.
          OPTION *.                                                       57.
          &DEFTOAR(SAME DOCUMENT - &OPNAME &PREVHOLD)                     58.
          &GOTO(NEXTSRCH)                                                 59.
5-RECORDS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ANDOR
          &IF(&AND)&EXIT(-1)                                              61.
          &IF(&OR) &EXIT(-1)                                              62.
          &IF(&NOT)&EXIT(-1)                                              63.
          &IF(&*&)                                                        64.
4-RECORDS

ANDORCHK
          &IF(&AND)&THEN &MACRO DEFINE TEMP &TEMP AND.)&THEN&EXIT(-1)     65.
          &IF(&OR)&THEN &MACRO DEFINE TEMP &TEMP OR .)&THEN &EXIT(-1)     66.
          &IF(&*&)                                                        67.
3-RECORDS

ATEQ
          MACRO DEFINE GROUP &CURKWORD.                                   70.
          MACRO DEFINE CURPARM &+1(&CURPARM).                             71.
          MACRO DEFINE CURMACRO &PARM(&CURPARM).                          72.
          MACRO DEFINE CP 1.                                              73.
          &IGROUP(&CURMACRO)                                              74.
          MACRO DEFINE CURPARM &+1(&CURPARM).                             75.
          MACRO DEFINE CURMACRO &PARM(&CURPARM)&ELSE&REST&ELSE&EXIT(-1).  76.
          &GOTO(NXTKWORD)                                                 77.
8-RECORDS

BEGIN
          &RELEV                                                          79.
          &RESTART(BEGIN)                                                 80.
2-RECORDS

BEX
          FIND IN INDEX &TEMP,
          IAS POSTING KEYWORD 15,
          TITLE (NUMBER OF POSTINGS) 3 (* * * * KEYWORD) 15.
          &RESTART(TOP)
4-RECORDS

BLOOP
          MACRO DEFINE CURPARM &+1(&CURPARM).                             812.
          MACRO DEFINE TEMP &TEMP &IF(&TEMP,NE,) &THEN(OR)                813.
           &PARM(&CURPARM).* &ELSE&REST&ELSE&GOTO(BEX) .                  814.
          &GOTO(BLOOP)                                                    815.
4-RECORDS

BPROMPT
          MACRO DEFINE TEMP &READ(ENTER KEYWORD).* &IF()&THEN&REST        86.
          &THEN&RESTART(TOP) . &GOTO(BEX)                                 87.
                                                                          88.
3-RECORDS

BRWS
          &IF(&PARM(2),EQ,)&GOTO(BPROMPT)                                 90.
          MACRO DEFINE CURPARM 1. MACRO DEFINE TEMP.                      91.
          &GOTO(BLOOP)                                                    92.
3-RECORDS

CHK
          MACRO DEFINE CMD FIND.                                          94.
          &GOTO(CHKL)                                                     95.
2-RECORDS

CHKHL
          &IF(&PUNCH,NE,FULL)&EXIT(-1)
          &IF(&DSN,E.,&DOCDSN)&EXIT(-1)
          MACRO DEFINE CURCMD HELLO.
3-RECORDS
```

```
CHL
        &&TMP1 IF &CMPT &TEMP,&COUNT,HOLD UNIQUE.                       00000010
        &IFFILE&S(1) &ELSE &GOTO(UPHOLD)                                00000020
        &DEFTDB&(&SEARCH)                                               00000030
        MACRO DEFINE CURHOLD &HOLDNAME.                                 00000040
        OPTION MSGLIM 1.                                                00000050
        HOLD MATCH SAME DOCUMENT &DSN &HOLDNAME AS UNIQUE.              00000060
        &DEFTDB&(SAME DOCUMENT - &DSN &CURHOLD )                        00000070
        OPTION *.                                                       00000080
        &IF(&DOCCNT,EQ,0) &GOTO(DOCS                                    00000090
        FILE IF &FILENAME ,TAB 'RECALL: ' CAMPNO + SKIP .               00000100
        &WRITE( )                                                       00000110
        &GOTO(RECCNT)                                                   00000120
12-RECORDS

CMS
        MACRO DEFINE CMND SCAN.                                         114.
        &GOTO(CHL)                                                      115.
2-RECORDS

CMDSTART
        START.
1-RECORD

CNTEM
        MACRO DEFINE CURPARM 2.                                         00000010
        MACRO DEFINE DSN &HOLDS.                                        00000020
        MACRO DEFINE CURHOLD &PARM(&CURPARM).                           00000030
        &IF(&CURHOLD,NE,)&DEFTDB(&PARM)                                 00000040
        &IF(&CURPARM,EQ,)&GOTO(SCANCNT)                                 00000050
        MACRO DEFINE CURPARM &+1(&CURPARM).                             00000060
        MACRO DEFINE CURHOLD &PARM(&CURPARM).                           00000070
        MACRO DEFINE CURPARM &+1(&CURPARM).                             00000080
        MACRO DEFINE CURHOLD &PARM(&CURPARM).                           00000090
        &GOTO(SSETSCAN)                                                 00000101
        &EQCHK(&CURHOLD) &GOTO(SSETFIND)                                00000110
        &GOTO(SSETSCAN)                                                 00000120
12-RECORDS

COUNTDOCS
        OPTION MSGLIM 1.                                                130.
        HOLD MATCH &HOLDNAME AS DOCCNT, FREE DOCCNT.                    131.
        OPTION *.                                                       133.
3-RECORDS

DBNERR
        &WRITE(FILE OF FULL DATABASE INVALID)                           135.
        &WRITE(SUBSET MUST BE SPECIFIED)                                136.
        &RESTART(TOP)                                                   137.
3-RECORDS

DFFLD
        MACRO DEFINE INQFLDS (
        &IF(SUMMARY)&THEN(1
        &IF(SELECTION)&THEN(1 2)
        &IF(ALLTEXT)&THEN(1 2)
        ).
        MACRO DEFINE CURPARM &+1(&CURPARM).
6-RECORDS

DEFTDB
        MACRO DEFINE HOLD &+1(&NHOLD).
        MACRO DEFINE TDB&HOLD &WRITE(SET &HOLDNAME - &PARM).
        MACRO DEFINE TEMP &TDB&HOLD.
        &TEMP
4-RECORDS
```

```
DICT
                OPTION NOSOURCE;                                        149.
                FIND IN SYN &COMMAND.+ AND IN INDEX,                    150.
                BREAK BEFORE TITLE SKIP                                 151.
                 BEFORE KEYWORD COMPUTE XX FORMATTED 7(0)               152.
                ON KEYWORD XX KEYWORD                                   153.
                ON TITLE IN INDEX COMPUTE XX FORMATTED 7(XX + POSITION), 154.
                TITLE (NUMBER OF POSITIONS) R/XX 1' ' ' ' KEYWORD       155.
                L/KEYWORD.                                              156.
                &IF&TITLE(1,&LENGTH&TITLE(&COMMAND)                     157.
                &ELSE&WRITE(NOT FOUND IN DICTIONARY)                    158.
                OPTION *.                                               158.1
11-RECORDS

DICTOP
                MACRO DEFINE COMOP 1.                                   160.
                &GOTO(DICTOP1)                                          161.
2-RECORDS

DICTOP1
                MACRO DEFINE COMOP &+1(&COMOP).                         818.
                MACRO DEFINE COMMAND &PARM(&COMOP) . &ELSE&RESTART(TOP) 819.
                &DICT                                                   820.
                &GOTO(DICTOP1)                                          821.
4-RECORDS

DSPL
                &IF(&PARM(2),EQ,)&GOTO(FULLDSP)                         163.
                MACRO DEFINE COMPARM 1.                                 164.
                &GOTO(DSPLOOP)                                          165.
3-RECORDS

DSPLOOP
                MACRO DEFINE COMPARM &+1(&COMPARM).                     167.
                MACRO DEFINE TEMP &STORE&PARM(&COMPARM) &ELSE&RESTART(TOP) . 168.
                &TEMP &GOTO(DSPLOOP)                                    169.
3-RECORDS

EOCHK
                &IF(&OTHEXIT(-1)                                        171.
                &IF(&SLATE(-1)                                          172.
                &IF(...)                                                173.
3-RECORDS

FALSE
                &IF(&)
1-RECORD

FORMCHK
                &IF(&LIST)&RETURN(,LIST)                                00020010
                &IF(&FULL)&RETURN(&FULL)                                00030020
                &IF(&TITLE)&RETURN(&TF2)                                00040030
                &ASK(INVALID FORMAT - REENTER) &GOTO(FORMCHK)           00050040
4-RECORDS

FREE
                &IF(&PARM(2),EQ,)&GOTO(FULLFREE)                        164.
                &IF(&PARM(2),EQ,ALL)&GOTO(FULLFREE)                     165.
                MACRO DEFINE COMPARM 1.                                 166.
                &GOTO(FREE1)                                            167.
4-RECORDS

FREECOM
                &GOTO(&EXTERNAL)                                        169.
1-RECORD
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FREEIT
          MACRO DEFINE CURPARM &+1(&CURPARM).
          HOLD FREE &PARM(&CURPARM)&ELSE&RESTART(TOP) .
          MACRO DEFINE TOO&PARM(&CURPARM).
          &GOTO(FREEIT)
4-RECORDS

FULLDSP
          MACRO DEFINE I 0.                                              196.
          &MODECHK(SYNHOLD) &RESTART(TOP)                                197.
          SYNCLOSUP(&SYNHOLD)                                            198.
          &IF(&QUIET,EQ,ON)&RESTART(TOP)                                 199.
          HOLD DISPLAY.                                                  200.
          &RESTART(TOP)                                                  201.
6-RECORDS

FULLFREE
          HOLD FREE ALL.                                                 203.
          MACRO DEFINE SYNHOLD 0.                                        204.
          &RESTART(TOP)                                                  205.
3-RECORDS

EVALOPCHK
          &IF(EQ)&EXIT(-1)                                               207.
          &IF(LS)&EXIT(-1)                                               208.
          &IF(CONTAINS)&EXIT(-1)                                         209.
          &IF(GT)&EXIT(-1)                                               210.
          &IF(LT)&EXIT(-1)                                               211.
          &IF(LE)&EXIT(-1)                                               212.
          &IF(GE)&EXIT(-1)                                               213.
          &IF(EXCLUDES)&EXIT(-1)                                         214.
          &IF(NOT)&EXIT(-1)                                              215.
          &IF(NE)&EXIT(-1)                                               216.
          &WRITE(&PARM)                                                  217.
          &WRITE(IS AN INVALID COMPARISON OPERATOR)                      218.
          &RESTART(TOP)                                                  219.
13-RECORDS

EVVAL
          MACRO DEFINE CP &+1(&CP).                                      221.
          MACRO DEFINE CURVAL &PARM(&CP)&ELSE&REST&ELSE&EXIT(-1).        222.
          &IF(&CURVAL,EQ,OR)&GOTO(EVVAL)                                 223.
          MACRO DEFINE TEMP2 &TEMP2'&CURVAL. &GOTO(EVVAL)                224.
4-RECORDS

GET
          FIND IN INDEX &PARM(1),COUNT,HOLD POSTING AS &PARM(2) FROM     226.
          &RECDD.                                                        227.
2-RECORDS

GETFORM
          , &FORMCHK(&FORMAT) . &RESTART(TOP)                            229.
1-RECORD

GETSYN
          &WRITE(SYNONYM DATABASE NOT AVAILABLE)                         231.
          &RESTART(TOP)                                                  232.
          MACRO DEFINE CURPARM &+1(&CURPARM).                            233.
          MACRO DEFINE CURWORD &PARM(&CURPARM).                          234.
          MACRO DEFINE STOP SYN -.                                       235.
          &GOTO(ALLSYN)                                                  236.
6-RECORDS

RECORD
          &WRITE( )                                                      00000010
          &WRITE(COMMAND:  COUNT                                         00000020
          &IF(&PARM(3),EQ,) &THEN &RECORD &THEN &RECORDS &THEN &RECORD   00000030
          &THEN &READ(PLEASE <CR>TO CONTINUE, <BREAK> TO QUIT)           00000040
          &THEN &RECORD &THEN &RECORD                                    00000050
          &IF(&PARM(3),IS,#) &THEN &RECORD &THEN &WRITE( ) &THEN &RESTART(TOP)  00000060
          &IF(&PARM(3),IS,S) &THEN &RECORDS &THEN &RECORD &THEN &WRITE( )  00000070
          &THEN &RESTART(TOP)                                            00000080
          &IF(&PARM(3),IS,0) &THEN &RECORD &THEN &RECORD &THEN &WRITE( )  00000090
          &THEN &RESTART(TOP)                                            00000100
          &IF(&PARM(3),IS,L) &GOTO(RECORD)                               00000110
11-RECORDS
```


C-2

```
HCOUND
        &WRITE( )
        &WRITE(          REQUIRED: SET NUMBER, OR 'IF',FIELDNAME,RELATOR,VALUE
2-RECORDS
HCOUNE
        &WRITE( )                                                              00000010
        &WRITE(EXAMPLES: COUNT 4                                               00000020
        &WRITE(          COUNT 6 IF SERNO GT 12                                00000030
        &WRITE(          COUNT IS IF SCORE IS (90,70,60)                       00000040
        &WRITE(          COUNT 2 IF COURSE EQ HIST AND LEVEL LE 4              00000050
        &WRITE(          COUNT 7 IF DOCNAME CONTAINS (MAP,CHART,ATLAS)         00000060
        &WRITE( )                                                              00000070
        &FSTART(TOP)                                                           00000080
8-RECORDS
HCOUNF
        &WRITE( )
        &WRITE(FUNCTION: IDENTIFIES AND COUNTS DOCUMENTS WHICH SATISFY A
        &WRITE(          LOGICAL CONDITION AND/OR PREVIOUS SEARCH CRITERIA.
3-RECORDS
HCOUNO
        &WRITE( )                                                              00000010
        &WRITE(OPERANDS:                                                       00000020
        &WRITE(  SET NUMBER -THE SET NUMBER OF A TEMPORARY D-ATABASE CREATED BY 00000030
        &WRITE(              A PREVIOUS COMMAND.                               00000040
        &WRITE(  'IF'       -BEGINS A QUALIFYING CONDITION.  THE FULL CONDITION 00000050
        &WRITE(              MUST CONSIST OF 'IF',FIELDNAME,RELATOR,VALUE.     00000060
        &WRITE(  FIELDNAME  -A FIELD FROM THE DOCUMENT DATABASE.               00000070
        &WRITE(  RELATOR    -INDICATES FIELD-TO-VALUE COMPARISON, AS FOLLOWS:  00000080
        &WRITE(              'GT'  GREATER THAN                                00000090
        &WRITE(              'LT'  LESS THAN                                   00000100
        &WRITE(              'GE'  GREATER THAN OR EQUAL TO                    00000110
        &WRITE(              'LE'  LESS THAN OR EQUAL TO                       00000120
        &WRITE(              'EQ'  EQUALS                                      00000130
        &WRITE(              'NE'  NOT EQUAL TO                                00000140
        &WRITE(              'IS'  FIELD BEGINS WITH SPECIFIED VALUE           00000150
        &WRITE(              'ISN' FIELD DOES NOT BEGIN WITH SPECIFIED VALUE   00000160
        &WRITE(              'CONTAINS' VALUE IS FOUND ANYWHERE IN FIELD       00000170
        &WRITE(              'EXCLUDES' VALUE IS NOT FOUND ANYWHERE IN FIELD   00000180
        &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)                         00000190
        &WRITE(  VALUE      -ONE OR MORE VALUES AGAINST WHICH FIELD IS         00000200
        &WRITE(              COMPARED.  MULTIPLE VALUES MUST BE SEPARATED BY   00000210
        &WRITE(              COMMAS AND ENCLOSED IN PARENTHESES.  A RANGE MAY  00000220
        &WRITE(              BE SPECIFIED WITH THE 'IS' RELATOR AND 'TO'       00000230
        &WRITE(              CONNECTING THE UPPER AND LOWER VALUES IN THE      00000240
        &WRITE(              RANGE.  THE 'SET,COMMAS,ON' OPTION MUST BE IN     00000250
        &WRITE(              EFFECT WHEN VALUES WITH EMBEDDED BLANKS ARE       00000260
        &WRITE(              SPECIFIED.                                        00000270
        &WRITE(  BOOLEAN    -OPERATOR USED TO COMBINE MULTIPLE QUALIFYING      00000280
        &WRITE(              CONDITIONS, AS FOLLOWS:                           00000290
        &WRITE(              'AND'  BOTH CONDITIONS MUST BE TRUE.              00000300
        &WRITE(              'OR'   AT LEAST ONE CONDITION MUST BE TRUE.       00000310
        &WRITE(              'NOT'  THE FIRST CONDITION MUST BE TRUE AND THE   00000320
        &WRITE(                     SECOND FALSE.                              00000330
33-RECORDS
HCOUNS
        &WRITE( )                                                              00000010
        &WRITE(SYNTAX:  'COUNT',SET NUMBER,'IF',FIELDNAME,RELATOR,VALUE,       00000020
        &WRITE(           BOOLEAN...                                           00000030
        &WRITE(         WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,     00000040
        &WRITE(         WITHOUT THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE 00000050
        &WRITE(         USER-SUPPLIED VARIABLES.                               00000060
6-RECORDS
```

```
HDICT
          &WRITE( )
          &WRITE(COMMAND:  DICT)
          &IF(&PARM(3),EQ,) &THEN &HDICTF &THEN &HDICTS &THEN &HDICTD
          &THEN &HDICTD &THEN &HDICTN &THEN &HDICTE
          &IF(&PARM(3),IS,F) &THEN &HDICTF &THEN &HDICTN &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,S) &THEN &HDICTS &THEN &HDICTD &THEN &HDICTN
          &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,D) &THEN &HDICTD &THEN &HDICTD &THEN &HDICTN
          &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,E) &GOTO(HDICTE)
11-RECORDS
HDICTD
          &WRITE( )
          &WRITE(          REQUIRED: WORD
2-RECORDS
HDICTE
          &WRITE( )                                                    00000010
          &WRITE(EXAMPLES: DICT COMPUTER)                              00000020
          &WRITE( )                                                    00000030
          &RESTART(TOP)                                                00000040
4-RECORDS
HDICTF
          &WRITE( )
          &WRITE(FUNCTION: DISPLAYS A PRE-DEFINED DICTIONARY OF SYNONYMS FOR A
          &WRITE(          SPECIFIED WORD, AND THE NUMBER OF TIMES EACH WORD
          &WRITE(          OCCURS IN THE DOCUMENT DATABASE.
4-RECORDS
HDICTN
          &WRITE( )
          &WRITE(NOTE:     ROOT SEARCH IS AUTOMATIC; THAT IS, ALL WORDS BEGINNING
          &WRITE(          WITH THE CHARACTERS SPECIFIED IN WORD WILL BE USED.
3-RECORDS
HDICTD
          &WRITE( )
          &WRITE(OPERANDS: )
          &WRITE(  WORD - A WORD IN A SYNONYM DICTIONARY.
3-RECORDS
HDICTS
          &WRITE( )
          &WRITE(SYNTAX:   'DICT',&WORD
          &WRITE(          WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
          &WRITE(          MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
          &WRITE(          USER-SUPPLIED VARIABLES.
5-RECORDS
HDISP
          &WRITE( )
          &WRITE(COMMAND:  DISPLAY)
          &IF(&PARM(3),EQ,) &THEN &HDISPF &THEN &HDISPS &THEN &HDISPD
          &THEN &HDISPD &THEN &HDISPN &THEN &HDISPL
          &IF(&PARM(3),IS,F) &THEN &HDISPF &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,S) &THEN &HDISPS &THEN &HDISPD &THEN &HDISPN
          &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,D) &THEN &HDISPD &THEN &HDISPD &THEN &HDISPN
          &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,E) &GOTO(HDISPE)
10-RECORDS
```

```
HDISPD
        &WRITE( )
        &WRITE(          REQUIRED: WORD-LIST
2-RECORDS
HDISPE
        &WRITE( )                                                       00000010
        &WRITE(EXAMPLES: DISPLAY MARK                                   00000020
        &WRITE(          DISPLAY MANAG ACCOUNT                          00000030
        &WRITE( )                                                       00000040
        &RESTART(TOP)                                                   00000050
5-RECORDS
HDISPF
        &WRITE( )
        &WRITE(FUNCTION: DISPLAYS WORDS FROM THE INDEX, AND THE NUMBER OF TIMES
        &WRITE(          EACH WORD OCCURS IN THE DOCUMENT DATABASE.
3-RECORDS
HDISPN
        &WRITE( )
        &WRITE(NOTE:     ROOT SEARCH IS AUTOMATIC; THAT IS, ALL WORDS BEGINNING
        &WRITE(          WITH THE CHARACTERS SPECIFIED IN WORD-LIST WILL BE
        &WRITE(          DISPLAYED.
4-RECORDS
HDISPO
        &WRITE( )
        &WRITE(OPERANDS:
        &WRITE(  WORD-LIST  -A SINGLE WORD OR MULTIPLE WORDS SEPARATED BY
        &WRITE(              COMMAS, FOR WHICH A POSTING COUNT DISPLAY IS
        &WRITE(              DESIRED.
5-RECORDS
HDISPS
        &WRITE( )
        &WRITE(SYNTAX:   'DISPLAY',WORD-LIST
        &WRITE(          WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
        &WRITE(          MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
        &WRITE(          USER-SUPPLIED VARIABLES.
5-RECORDS
HELP
        &IF(&PARM(2),EQ,)&ELSE&GOTO(HELP00)
        &WRITE( )
        &WRITE(THE COMMANDS AVAILABLE ARE:
        &WRITE(                          COUNT
        &WRITE(                          DICT
        &WRITE(                          DISPLAY
        &WRITE(                          END
        &WRITE(                          FREE
        &WRITE(                          HELP
        &WRITE(                          HISTORY
        &WRITE(                          OUTLINE
        &WRITE(                          PRINT
        &WRITE(                          SEARCH
        &WRITE(                          SET
        &WRITE(                          SHOW
        &WRITE(                          SUMMARIZE
        &WRITE(                          USER
        &WRITE( )
        &WRITE( )
        &WRITE(TO DISPLAY DETAILED INFORMATION ON THESE COMMANDS,
        &READ(TYPE THE COMMAND NAME)
        &IF(&PARM,EQ,) &GOTO(TOP)
        &IF(&PARM,EQ,COUNT) &THEN &PARMSET(HELP,COUNT)
        &IF(&PARM,EQ,DICT) &THEN &PARMSET(HELP,DICT)
        &IF(&PARM,EQ,DISPLAY) &THEN &PARMSET(HELP,DISPLAY)
        &IF(&PARM,EQ,END) &THEN &PARMSET(HELP,END)
        &IF(&PARM,EQ,FREE) &THEN &PARMSET(HELP,FREE)
        &IF(&PARM,EQ,HELP) &THEN &PARMSET(HELP,HELP)
        &IF(&PARM,EQ,HISTORY) &THEN &PARMSET(HELP,HISTORY)
        &IF(&PARM,EQ,OUTLINE) &THEN &PARMSET(HELP,OUTLINE)
        &IF(&PARM,EQ,PRINT) &THEN &PARMSET(HELP,PRINT)
        &IF(&PARM,EQ,SEARCH) &THEN &PARMSET(HELP,SEARCH)
        &IF(&PARM,EQ,SET) &THEN &PARMSET(HELP,SET)
        &IF(&PARM,EQ,SHOW) &THEN &PARMSET(HELP,SHOW)
        &IF(&PARM,EQ,SUMMARIZE) &THEN &PARMSET(HELP,SUMMARIZE)
```

```
          &IF(&PARM,EQ,USER) &THEN &PARMSET(HELP,USER)
          &GOTO(HELPGO)
37-RECORDS
HELPGO
          &IF(&PARM(2),EQ,SEARCH)&GOTO(HSEARCH)
          &IF(&PARM(2),EQ,PRINT)&GOTO(HPRINT)
          &IF(&PARM(2),EQ,COUNT)&GOTO(HCOUNT)
          &IF(&PARM(2),EQ,DISP)&GOTO(HDISP)
          &IF(&PARM(2),EQ,SET)&GOTO(HSET)
          &IF(&PARM(2),EQ,FREE)&GOTO(HFREE)
          &IF(&PARM(2),EQ,HELP)&GOTO(HHELP)
          &IF(&PARM(2),IS,HISTORY)&GOTO(HHIST)
          &IF(&PARM(2),EQ,DICT)&GOTO(HDICT)
          &IF(&PARM(2),EQ,USER)&GOTO(HUSER)
          &IF(&PARM(2),EQ,END)&GOTO(HEND)
          &IF(&PARM(2),IS,SUM)&GOTO(HSUM)
          &IF(&PARM(2),EQ,SHOW)&GOTO(HSHOW)
          &IF(&PARM(2),IS,OUTL)&GOTO(HOUTL)
          &WRITE(INVALID COMMAND.)
          &WRITE(VALID COMMANDS ARE: COUNT, DICT, DISPLAY, END, FREE, HELP,
          &WRITE(                    HISTORY, OUTLINE, PRINT, SEARCH, SET,
          &WRITE(                    SHOW, SUMMARIZE, USER.
          &WRITE( )
          &RESTART(TOP)
20-RECORDS
HEND
          &WRITE( )
          &WRITE(COMMAND:  END
          &WRITE( )
          &WRITE(FUNCTION: TERMINATES SEARCH SESSION.
          &WRITE( )
          &WRITE(SYNTAX:   END
          &WRITE( )
          &WRITE(OPERANDS: NONE.
          &WRITE( )
          &RESTART(TOP)
10-RECORDS
HFREE
          &WRITE( )
          &WRITE(COMMAND:  FREE
          &IF(&PARM(3),EQ,) &THEN &HFREEF &THEN &HFREES &THEN &HFREED
          &THEN &HFREED &ELSE &HFREE
          &IF(&PARM(3),IS,F) &THEN &HFREEF &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,S) &THEN &HFREES &THEN &HFREED &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,D) &THEN &HFREED &THEN &HFREED &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,L) &GOTO(HFREEE)
10-RECORDS
HFREED
          &WRITE( )
          &WRITE(          DEFAULT: 'ALL'
2-RECORDS
HFREEE
          &WRITE( )                                                    00060010
          &WRITE(EXAMPLES: FREE                                        00060020
          &WRITE(          FREE ALL                                    00060030
          &WRITE(          FREE 1 2 16                                 00060040
          &WRITE( )                                                    00060050
          &RESTART(TOP)                                                00060060
6-RECORDS
```

```
.HFREE
          &WRITE( )
          &WRITE(FUNCTION: DELETES TEMPORARY DATABASES (SETS) CREATED BY PREVIOUS
          &WRITE(          COMMANDS.
3-RECORDS

.HFREED
          &WRITE( )
          &WRITE(OPERANDS:
          &WRITE(  'ALL'           - ERASES ALL SETS.
          &WRITE(  SET NUMBER(S) - ERASES ONLY SPECIFIED SETS.
4-RECORDS

.HFREES
          &WRITE( )
          &WRITE(SYNTAX:   'FREE','ALL'/SET NUMBER(S)
          &WRITE(          WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
          &WRITE(          WORDS IN APOSTROPHES. WORDS NOT IN APOSTROPHES ARE
          &WRITE(          USER-SUPPLIED VARIABLES.
5-RECORDS

.HHELP                                                                    00000010
          &WRITE( )                                                       00000020
          &WRITE(COMMAND:  HELP                                           00000030
          &IF(&PARM(3),EQ,) &THEN &HHELPP &THEN &HHELPS &THEN &HHELPD     00000040
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)            00000050
          &THEN &HHELPD &THEN &HHELPA &THEN &HHELPE                       00000060
          &IF(&PARM(3),IS,P) &THEN &HHELPP &THEN &WRITE( ) &THEN &RESTART(TOP)  00000070
          &IF(&PARM(3),IS,S) &THEN &HHELPS &THEN &HHELPD &THEN &HHELPN    00000080
          &THEN &WRITE( ) &THEN &RESTART(TOP)                             00000090
          &IF(&PARM(3),IS,D) &THEN &HHELPD &THEN &HHELPD &THEN &HHELPN    00000100
          &THEN &WRITE( ) &THEN &RESTART(TOP)                             00000110
          &IF(&PARM(3),IS,E) &GOTO(HHELPE)                                00000120
11-RECORDS

.HHELPD
          &WRITE( )
          &WRITE(          DEFAULT:  ALL SEGMENTS OF THE HELP INFORMATION ON A
          &WRITE(                    COMMAND ARE DISPLAYED.
          &WRITE(          REQUIRED: COMMAND, TO DISPLAY INFORMATION ON A
          &WRITE(                    PARTICULAR COMMAND.
          &WRITE(          OPTIONAL: SEGMENT
5-RECORDS

.HHELPE
          &WRITE( )                                                       00000610
          &WRITE(EXAMPLES: HELP COUNT                                     00000620
          &WRITE(          HELP FREE S                                    00000630
          &WRITE( )                                                       00000640
          &RESTART(TOP)                                                   00000650
5-RECORDS

.HHELPH
          &WRITE( )
          &WRITE(FUNCTION: DISPLAYS THE FUNCTION, SYNTAX, OPERANDS, AND
          &WRITE(          EXAMPLES OF A COMMAND.
3-RECORDS

.HHELPN
          &WRITE( )
          &WRITE(NOTE:     ONLY ONE SEGMENT AT A TIME MAY BE SPECIFIED.
          &WRITE(          NO BLANKS ARE ALLOWED WHEN SPECIFYING THE HELP
          &WRITE(          COMMAND.
4-RECORDS
```

```
.HELPU
          &WRITE )
          &WRITE(OPERANDS:
          &WRITE   COMMAND - A DECLARATION OF ONE OF THE FOLLOWING WILL DISPLAY
          &WRITE             HELP INFORMATION FOR THAT COMMAND: COUNT, DICT,
          &WRITE             DISPLAY, END, FILE, HELP, HISTORY, PRINT, SEARCH,
          &WRITE             SET, USER.
          &WRITE   SEGMENT - A PARTICULAR PART OF THE HELP INFORMATION FOR A
          &WRITE             COMMAND MAY BE SPECIFIED, AS FOLLOWS:
          &WRITE             'FUNCTION' OR 'F'
          &WRITE             'SYNTAX' OR 'S'
          &WRITE             'OPERANDS' OR 'O'
          &WRITE             'EXAMPLES' OR 'E'
12-RECORDS

.HELPS
          &WRITE )                                                        00000010
          &WRITE(SYNTAX:   'HELP' COMMAND SEGMENT                         00000020
          &WRITE           WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,  00000030
          &WRITE           MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE 00000040
          &WRITE           USER-SUPPLIED VARIABLES.                       00000050
5-RECORDS

.HIST
          &WRITE )
          &WRITE(COMMAND:  HISTORY
          &IF(&PARM(3),EQ,) &THEN &WRITE &THEN &WRITE &THEN &WRITE
          &THEN &WRITE &THEN &WRITE &THEN &WRITE
          &IF(&PARM(3),LE,1) &THEN &WRITE &THEN &WRITE ) &THEN &RESTART(TOP)
          &IF(&PARM(3),LE,5) &THEN &WRITE &THEN &WRITE )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),LE,8) &THEN &WRITE &THEN &WRITE &THEN &WRITE
          &THEN &WRITE ) &THEN &RESTART(TOP)
          &IF(&PARM(3),LE,8) &GOTO(WRITE)
10-RECORDS

.HISTD
          &WRITE )
          &WRITE           DEFAULT:  IF SET NUMBER(S) ARE OMITTED, INFORMATION
          &WRITE                     FOR ALL SETS IS DISPLAYED.
          &WRITE           OPTIONAL: SET NUMBER(S)
4-RECORDS

.HISTE
          &WRITE )                                                        00000010
          &WRITE(EXAMPLES: HISTORY                                        00000020
          &WRITE           HISTORY 3 8                                    00000030
          &WRITE )                                                        00000040
          &RESTART(TOP)                                                   00000050
5-RECORDS

.HISTF
          &WRITE )
          &WRITE(FUNCTION: DISPLAYS SET NUMBERS, COUNTS, AND SEARCH CRITERIA FOR
          &WRITE           TEMPORARY DATABASES (SETS) CREATED BY PREVIOUS
          &WRITE           COMMANDS.
4-RECORDS

.HISTN
          &WRITE )
          &WRITE(NOTE:     IF SET RESULTED FROM COUNT COMMAND, ITEM COUNTS ARE
          &WRITE           DISPLAYED; OTHERWISE, POSTING COUNTS ARE DISPLAYED.
3-RECORDS

.HISTO
          &WRITE )
          &WRITE(OPERANDS:
          &WRITE   SET NUMBER(S) - SET NUMBERS AND SEARCH CRITERIA FOR SPECIFIED
          &WRITE                   SETS WILL BE DISPLAYED.  NO COUNTS ARE
          &WRITE                   DISPLAYED IF SET NUMBER OPERAND IS SPECIFIED.
5-RECORDS
```

```
.HHISTS
          &WRITE( )
          &WRITE(SYNTAX:    'HISTORY',SET NUMBER(S)
          &WRITE            WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN.
          &WRITE            WORDS IN APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
          &WRITE            USER-SUPPLIED VARIABLES.
5-RECORDS

.HL
          .HELITE &DAT &HELP(&PARM) .                                    00060010
          &GOTO(.HLCONT)                                                 00060020
2-RECORDS

.HLCONT
          &ASK(ENTER .HELITE COMMAND)                                    00060030
          &IF(.HELP)&GOTO(.HHELP)                                        00060040
          &IF()&GOTO(.HL)                                                00060050
          &IF(END)&RESTART(TOP)                                          00060060
          &IF(QUIT)&RESTART(TOP)                                         00060070
          &IF(PG) &GOTO(.HL)                                             00060080
          &IF(BACK)&GOTO(.HL)                                            00060090
          &IF(NX)&GOTO(.HL)                                              00060100
          &IF(FIRST)&GOTO(.HL)                                           00060110
          &IF(TAB)&GOTO(.HL)                                             00060120
          &IF(&IN) &GOTO(&IN)                                            00060130
          &WRITE(INVALID .HELITE COMMAND - ENTER HELP FOR HELP)          00060140
          &GOTO(.HLCONT)                                                 00060150
13-RECORDS

.HHELP
          &WRITE(.HELITE COMMANDS ARE:)                                  00060160
          &WRITE(  NX       CONTINUE WITH NEXT POSTING)                  00060170
          &WRITE(  <CR>     NEXT LINES OF CURRENT PAGE)                  00060180
          &WRITE(  BACK     PREVIOUS LINES OF CURRENT PAGE)              00060190
          &WRITE(  PG       PRINT FROM NEXT PAGE IN SUBSET)              00060200
          &WRITE(  END      RETURN TO COMMAND MODE)                      00060210
          &WRITE(  FIRST    PRINT FIRST LINES OF CURRENT PAGE)           00060220
          &WRITE(  ORIG     PRINT ORIGINAL POSTING)                      00060230
          &GOTO(.HLCONT)                                                 00060240
9-RECORDS

.HLT
          &IF()&RETURN(+1)                                               00060250
          &IF(PG)&RETURN(NX)                                             00060260
          &IF(NX)&RETURN(.HL)                                            00060270
          &IF(FIRST)&RETURN(FIRST)                                       00060280
          &IF(BACK)&RETURN(-1)                                           00060290
          &PARM                                                          00060300
6-RECORDS

.HHOLDLOOP
          MACRO DEFINE I &+1(&1).
          &PARM
          &IF(&I,GE,&HOLD&NAME)&EXIT(-1)
          &GOTO(.HHOLDLOOP)
4-RECORDS

.HOUTL
          &WRITE( )
          &WRITE(COMMANDS:  OUTLINE
          &IF(&PARM(3),EQ,) &THEN &HOUTLF &THEN &HOUTES &THEN &HOUTED
          &THEN &IF(&PRESS(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT))
          &THEN &HOUTED &THEN .HOUTLE
          &IF(&PARM(3),IS,F) &THEN &HOUTLF &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,S) &THEN &HOUTES &THEN &HOUTED &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,D) &THEN &HOUTED &THEN &HOUTED &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),IS,E) &GOTO(.HOUTLE)
11-RECORDS
```

```
.MODULE
          &WRITE( )
          &WRITE(              DEFAULT: NAME = 'ALL'
2-RECORDS
.MODULE
          &WRITE( )
          &WRITE(EXAMPLES: OUTLINE DISEASE
          &WRITE(          OUTLINE ALL
          &WRITE(          OUTLINE
          &WRITE( )
          &RESTART(TOP)
6-RECORDS
.MODULE
          &WRITE( )
          &WRITE(FUNCTION: PRINTS THE OUTLINE STRUCTURE OF AVAILABLE DOCUMENTS.
2-RECORDS
.MODULE
          &WRITE( )
          &WRITE(OPERANDS:
          &WRITE(  NAME          - MUST BE ONE OF THE FOLLOWING LITERALS:
          &WRITE( )
          &WRITE(                  'DISEASE' - OUTLINE FORMAT FOR INFECTIOUS
          &WRITE(                  DISEASE DOCUMENTS.
          &WRITE( )
          &WRITE(                  'LAB' - OUTLINE FORMAT FOR CLINICAL
          &WRITE(                  PATHOLOGY LABORATORY DOCUMENTS.
          &WRITE( )
          &WRITE(                  'ALL' - PRINTS OUTLINES OF ALL ACTIVE
          &WRITE(                  DOCUMENTS IN THE CPLM DATABASE.
12-RECORDS
.MODULE
          &WRITE( )
          &WRITE(SYNTAX:  'OUTLINE' NAME
          &WRITE(         WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
          &WRITE(         MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
          &WRITE(         USER-SUPPLIED VARIABLES.
5-RECORDS
.MPRIN                                                                    00000010
          &WRITE( )                                                       00000020
          &WRITE(COMMAND:  PRINT                                          00000030
          &IF(&PARM(3),EQ,) &THEN &MPRINF &THEN &MPAINS &THEN &MPAIND     00000040
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)            00000050
          &THEN &MPAIND &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)  00000060
          &THEN &MPAIND &THEN &MPRINC                                     00000070
          &IF(&PARM(3),IS,F) &THEN &MPRINF &THEN &WRITE( ) &THEN &RESTART(TOP)  00000080
          &IF(&PARM(3),IS,S) &THEN &MPAINS &THEN &MPAIND &THEN &WRITE( )  00000090
          &THEN &RESTART(TOP)                                             00000100
          &IF(&PARM(3),IS,D) &THEN &MPAIND &THEN &MPRIND &THEN &MPAIN     00000110
          &THEN &WRITE( ) &THEN &RESTART(TOP)                             00000120
          &IF(&PARM(3),IS,E) &OUTDO(MPRINE)
12-RECORDS
.MPRIND
          &WRITE( )
          &WRITE(              DEFAULT:  DEFAULT FORMAT IS ESTABLISHED AT
          &WRITE(                        INSTALLATION TIME.
          &WRITE(              REQUIRED: SET NUMBER, OR 'IF',FIELDNAME,RELATOR,VALUE
          &WRITE(              OPTIONAL: FORMAT
5-RECORDS
```

```
.HPRINT
        &WRITE( )
        &WRITE(EXAMPLES: PRINT 9                                                00060010
        &WRITE           PRINT PAGENO 6 IF SERNO GT 12                          00060020
        &WRITE           PRINT 14 IF SCORE IS (40,TO,60)                        00060030
        &WRITE           PRINT PAGE 2 IF COURSE EQ HIST AND LEVEL LE 9           00060040
        &WRITE           PRINT 7 IF DOCNAME CONTAINS (MAP,CHART,ATLAS)          00060050
        &WRITE( )                                                               00060060
        &RESTART(HOP)                                                           00060070
3-RECORDS

.HPRINT
        &WRITE( )
        &WRITE(FUNCTION: PRINTS DOCUMENTS SATISFYING A LOGICAL CONDITION AND/OR
        &WRITE           PREVIOUS SEARCH CRITERIA.
3-RECORDS

.HPRINT
        &WRITE( )
        &WRITE(NOTE:     A SET RESULTING FROM A COUNT COMMAND MAY NOT BE
        &WRITE           PRINTED WITH HILITE FORMAT.
        &WRITE           A MENU OF AVAILABLE HILITE SPECIFICATIONS WILL BE
        &WRITE           DISPLAYED WHEN 'HELP' IS ENTERED AFTER HILITE PROMPT.
3-RECORDS

.HPRINT
        &WRITE( )
        &WRITE(OPERANDS:
        &WRITE  FORMAT      - SPECIFIES WHAT IS TO BE PRINTED.  FORMATTING
        &WRITE                OPTIONS ARE DESIGNED FOR EACH SYSTEM AT
        &WRITE                INSTALLATION TIME.
        &WRITE  SET NUMBER  - THE SET NUMBER OF A TEMPORARY DATABASE TO BE
        &WRITE                PRINTED.
        &WRITE  'IF'        - BEGINS A QUALIFYING CONDITION.  THE FULL CONDITION
        &WRITE                MUST CONSIST OF 'IF',FIELDNAME,RELATOR,VALUE.
        &WRITE  FIELDNAME   - A FIELD FROM THE DOCUMENT DATABASE.
        &WRITE  RELATOR     - INDICATES FIELD-TO-VALUE COMPARISON, AS FOLLOWS:
        &WRITE                'GT'  GREATER THAN
        &WRITE                'LT'  LESS THAN
        &WRITE                'GE'  GREATER THAN OR EQUAL TO
        &WRITE                'LE'  LESS THAN OR EQUAL TO
        &WRITE                'EQ'  EQUALS
        &WRITE                'NE'  NOT EQUAL TO
        &WRITE                'IS'  FIELD BEGINS WITH SPECIFIED VALUE
        &WRITE                'NOT' FIELD DOES NOT BEGIN WITH SPECIFIED VALUE
        &WRITE                'CONTAINS' VALUE IS FOUND ANYWHERE IN FIELD
        &WRITE                'EXCLUDES' VALUE IS NOT FOUND ANYWHERE IN FIELD
        &READ(PRESS <CR> TO CONTINUE, <CONTROL-Z> TO QUIT)
        &WRITE  VALUE       - ONE OR MORE VALUES AGAINST WHICH FIELD IS
        &WRITE                COMPARED.  MULTIPLE VALUES MUST BE SEPARATED BY
        &WRITE                COMMAS AND ENCLOSED IN PARENTHESES.
        &WRITE                A RANGE MAY BE SPECIFIED WITH THE 'IS' RELATOR
        &WRITE                AND 'TO' CONNECTING THE UPPER AND LOWER VALUES IN
        &WRITE                THE RANGE.  THE 'SET,COMMAS,ON' OPTION MUST BE IN
        &WRITE                EFFECT WHEN VALUES WITH ENCLOSED BLANKS ARE
        &WRITE                SPECIFIED.
        &WRITE  BOOLEAN     - OPERATOR USED TO COMBINE MULTIPLE QUALIFYING
        &WRITE                CONDITIONS, AS FOLLOWS:
        &WRITE                'AND'  BOTH CONDITIONS MUST BE TRUE.
        &WRITE                'OR'   AT LEAST ONE CONDITION MUST BE TRUE.
        &WRITE                'NOT'  THE FIRST CONDITION MUST BE TRUE AND THE
        &WRITE                       SECOND FALSE.
36-RECORDS

.HPRINT
        &WRITE( )
        &WRITE(SYNTAX:   'PRINT',FORMAT,SET NUMBER,'IF',FIELDNAME,RELATOR,
        &WRITE             VALUE,BOOLEAN...
        &WRITE           WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
        &WRITE           WORDS IN APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
        &WRITE           USER-SUPPLIED VARIABLES.
6-RECORDS
```

```
HSEARA
          &WRITE( )                                                          00000010
          &WRITE(COMMAND:  SEARCH                                            00000020
          &IF(&PARM(1),EQ,? &THEN &HSEARF &THEN &HSEARS &THEN &HSEARD        00000030
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)               00000040
          &THEN &HSEARD                                                      00000050
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)               00000060
          &THEN &HSEARH &THEN &HSEARE                                        00000070
          &IF(&PARM(1),LE,1 ) &THEN &HSEARF &THEN &WRITE( )                  00000080
          &THEN &RESTART(TOP)                                                00000090
          &IF(&PARM(3),LE,5) &THEN &HSEARS &THEN &HSEARD &THEN &HSEARH       00000100
          &THEN &WRITE( ) &THEN &RESTART(TOP)                                00000110
          &IF(&PARM(3),LE,6) &THEN &HSEARE &THEN &HSEARD &THEN &HSEARH       00000120
          &THEN &WRITE( ) &THEN &RESTART(TOP)                                00000130
          &IF(&PARM(3),LE,6) &GOTO(HSEARE)                                   00000140
14-RECORDS

HSEARD
          &WRITE( )
          &WRITE(          DEFAULT:  DEFAULT FIELD(S) ARE ESTABLISHED AT
          &WRITE(                    INSTALLATION TIME.
          &WRITE(          REQUIRED: WORD(S)
          &WRITE(          OPTIONAL: FIELD
          &WRITE(                    'SYN'
          &WRITE(                    RELATOR
7-RECORDS

HSEARE
          &WRITE( )                                                          00000010
          &WRITE(EXAMPLES: SEARCH POLLUTION                                  00000020
          &WRITE(          SEARCH WATER ADJ POLLUTION                        00000030
          &WRITE(          SEARCH TITLE POLLUTION                            00000040
          &WRITE(          SEARCH POLLUTION CONTAMINATION                    00000050
          &WRITE(          SEARCH SYN POLLUTION                              00000060
          &WRITE(          SEARCH POLLUT*                                    00000070
          &WRITE(          SEARCH POLLUTION AND INDUSTRY                     00000080
          &WRITE(          SEARCH * NOT AIR                                  00000090
          &WRITE(          SEARCH AIR SEN POLLUTION                          00000100
          &WRITE(          SEARCH WATER W 5 POLLUTION                        00000110
          &WRITE(          SEARCH POLLUTION W *3 AIR                         00000120
          &WRITE( )                                                          00000130
          &RESTART(TOP)                                                      00000140
14-RECORDS

HSEARF
          &WRITE( )
          &WRITE(FUNCTION: IDENTIFIES DOCUMENTS BASED ON RELATIONSHIPS AMONG
          &WRITE(          WORDS IN THE DOCUMENTS.  RELATIONSHIPS MAY BE
          &WRITE(          SPECIFIED AS FOLLOWS:
          &WRITE(          - WORDS ADJACENT TO EACH OTHER;
          &WRITE(          - WORDS IN THE SAME DOCUMENT, FIELD, OR SENTENCE;
          &WRITE(          - A WORD WITHIN A SPECIFIED NUMBER OF WORDS FROM
          &WRITE(            ANOTHER;
          &WRITE(          - WORD RELATIONSHIPS BASED ON "AND", "OR", "NOT"
          &WRITE(            CONDITIONS.
10-RECORDS

HSEARH
          &WRITE( )
          &WRITE(NOTE:     THE RESULTING TEMPORARY DATABASE CONTAINS THE TOTAL
          &WRITE(          NUMBER OF OCCURRENCES OF ALL WORDS SATISFYING THE
          &WRITE(          COMMAND CRITERIA.  IT DOES NOT CONTAIN THE NUMBER OF
          &WRITE(          DOCUMENTS SATISFYING THE CRITERIA.
5-RECORDS
```

```
HSEARC
          &WRITE( )                                                              00000010
          &WRITE(KEYWORDS:                                                       00000020
          &WRITE(   FIELD    - DELIMITS FIELDS TO BE SEARCHED.  A FIELD-GROUP MAY 00000030
          &WRITE(              BE DESIGNATED AT INSTALLATION TIME TO REPRESENT   00000040
          &WRITE(              SEVERAL FIELDS UNDER ONE NAME.                    00000050
          &WRITE(   'SYN'    - PRECEDING ANY WORD, RETRIEVES ALL PRE-DEFINED     00000060
          &WRITE(              SYNONYMS FOR THAT WORD.                           00000070
          &WRITE(   WORD(S)  - ONE OR MORE WORDS TO BE SEARCHED.  ONE OR MORE SET 00000080
          &WRITE(              NUMBERS OF PREVIOUSLY CREATED TEMPORARY DATABASES 00000090
          &WRITE(              MAY BE SPECIFIED IN PLACE OF WORD(S).  USE OF THE 00000100
          &WRITE(              SET NUMBER, ONCE A TEMPORARY DATABASE HAS BEEN    00000110
          &WRITE(              CREATED, IS MORE EFFICIENT.  MORE THAN ONE        00000120
          &WRITE(              WORD/SET, SEPARATED BY COMMAS, INDICATES AN 'OR'  00000130
          &WRITE(               CONDITION.  A WORD MUST FOLLOWED BY '*' INDICATES A 00000140
          &WRITE(              SEARCH ON ALL WORDS BEGINNING WITH THE ROOT.       00000150
          &WRITE(   RELATOR  - COMBINES ONE WORD/SET WITH ANOTHER, AS FOLLOWS:  00000160
          &WRITE(              'AND'  WORDS/SETS MUST OCCUR IN THE SAME DOCUMENT. 00000170
          &WRITE(              'OR'   AT LEAST ONE WORD/SET MUST OCCUR IN THE    00000180
          &WRITE(                     DOCUMENT.  'OR' MAY BE SPECIFIED OR IMPLIED. 00000190
          &WRITE(              'NOT'  WORDS/SETS FOLLOWING 'NOT' MUST NOT OCCUR  00000200
          &WRITE(                      IN SAME DOCUMENT AS THOSE PRECEDING 'NOT'. 00000210
          &WRITE(              ABOVE RELATORS MAY BE MIXED IN THE SAME COMMAND.  00000220
          &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)                         00000230
          &WRITE(              'ADJ'  WORDS/SETS MUST BE ADJACENT AND MUST OCCUR 00000240
          &WRITE(                     IN THE DIRECTION INDICATED.  NOISE WORDS  00000250
          &WRITE(                     (AS, OF, AND, BY, ETC.) ARE NOT SEARCHABLE, 00000260
          &WRITE(                     BUT DO COUNT IN DETERMINING DISTANCE BETWEEN 00000270
          &WRITE(                     SEARCH WORDS.                              00000280
          &WRITE(              'SEN'  WORDS/SETS MUST OCCUR IN THE SAME SENTENCE, 00000290
          &WRITE(                     ACCORDING TO INSTALLATION DEFINITION OF A  00000300
          &WRITE(                     SENTENCE.  'SEN' IS ALLOWED ONLY WHEN     00000310
          &WRITE(                     SENTENCE PROXIMITY HAS BEEN IMPLEMENTED IN 00000320
          &WRITE(                     THE PARTICULAR SYSTEM.                     00000330
          &WRITE(              'N,'N  WORDS/SETS MUST OCCUR WITHIN N, 'N INTEGER, 00000340
          &WRITE(                     WORDS OF EACH OTHER.  IF SENTENCE PROXIMITY 00000350
          &WRITE(                     HAS BEEN IMPLEMENTED IN THE PARTICULAR    00000360
          &WRITE(                     SYSTEM, WORDS/SETS MUST ALSO OCCUR IN THE 00000370
          &WRITE(                     SAME SENTENCE; OTHERWISE, IN THE SAME     00000380
          &WRITE(                     DOCUMENT.  WORDS/SETS MAY OCCUR ON EITHER 00000390
          &WRITE(                     SIDE OF EACH OTHER, EXCEPT WHEN 'N SPECIFIES 00000400
          &WRITE(                     DIRECTION.                                 00000410
          &WRITE(              ALL RELATORS MAY BE COMBINED WITH 'OR' IN THE SAME 00000420
          &WRITE(              COMMAND.                                           00000430
43-RECORDS

HSEARS
          &WRITE( )
          &WRITE(SYNTAX:   'SEARCH',FIELD,'SYN',WORD(S),RELATOR,'SYN',WORD(S)...
          &WRITE(          WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
          &WRITE(          NEED NOT BE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
          &WRITE(          USER-SUPPLIED VARIABLES.
          &WRITE(          MULTIPLE WORD,RELATOR,WORD COMBINATIONS ARE ALLOWED.
6-RECORDS

HSET
          &WRITE( )                                                              00000010
          &WRITE(COMMAND:  SET                                                   00000020
          &IF(&PARM(1),EQ,) &THEN &HSETP &THEN &HSETS &THEN &HSETD               00000030
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT) &THEN &HSETD      00000040
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)                   00000050
          &THEN &HSETP &THEN &HSETE                                              00000060
          &IF(&PARM(3),I,) &THEN &HSETP &THEN &WRITE( ) &THEN &RESTART(TOP)      00000070
          &IF(&PARM(3),I,) &THEN &HSETS &THEN &HSETD &THEN &HSETA                00000080
          &THEN &WRITE( ) &THEN &RESTART(TOP)                                    00000090
          &IF(&PARM(3),I,) &THEN &HSETD &THEN &HSETD &THEN &HSETA                00000100
          &THEN &WRITE( ) &THEN &RESTART(TOP)                                    00000110
          &IF(&PARM(3),I,) &GOTO(HSETE)                                          00000120
12-RECORDS
```

```
HSHOWD
        &WRITE( )
        &WRITE(          DEFAULT: FIELDNAME   = ASPART
        &WRITE(                   ITEM NUMBER = 1
        &WRITE(                     IF THIS IS THE FIRST INVOCATION OF
        &WRITE(                     THE SHOW COMMAND.
        &WRITE(                   OR THE MOST RECENT FIELDNAME AND ITEM NUMBER,
        &WRITE(                     IF THERE HAS BEEN A PREVIOUS INVOCATION
        &WRITE(                     OF THE SHOW COMMAND.
J-RECORDS

HSHOWL
        &WRITE( )
        &WRITE(EXAMPLES: SHOW
        &WRITE(          SHOW ASPART IN 1
        &WRITE(          SHOW TESTPART
        &WRITE(          SHOW TITLE SYNONYMS DEFINITN IN 1 2 7
        &WRITE( )
        &RESTART(TOP)
7-RECORDS

HSHOWI
        &WRITE( )
        &WRITE(FUNCTION: PRINTS THE CONTENTS OF THE SPECIFIED FIELD FOR
        &WRITE(          ITEMS IN THE MOST RECENTLY SUMMARIZED TEMPORARY
        &WRITE(          DATASET.  ITEMS ARE IDENTIFIED BY THE ITEM VALUE
        &WRITE(          LISTED BY THE SUMMARIZE COMMAND.
5-RECORDS

HSHOWO
        &WRITE( )
        &WRITE(OPERANDS:
        &WRITE(  FIELDNAME    - THE FIELDNAME ASSOCIATED WITH EACH MAJOR
        &WRITE(                 SECTION OF THE PRESENTATION OUTLINE.
        &WRITE(                 MULTIPLE FIELDNAMES ARE ALLOWED.  VALID
        &WRITE(                 FIELDNAMES ARE:  TITLE, SYNONYMS, CODE,
        &WRITE(                 DEFINITN, ASPART, PATPART, TESTPART, REFS.
        &WRITE(                 USE THE OUTLINE COMMAND TO PRINT THE
        &WRITE(                 PRESENTATION OUTLINE AND THE CORRESPONDING
        &WRITE(                 FIELDNAMES.
        &WRITE(  ITEM NUMBER  - THE ITEM NUMBER WHICH IDENTIFIES A
        &WRITE(                 DOCUMENT WITHIN THE CURRENT ACTIVE
        &WRITE(                 TEMPORARY DATABASE.  MULTIPLE ITEM
        &WRITE(                 NUMBERS ARE ALLOWED.  THE ITEM NUMBERS
        &WRITE(                 ARE OBTAINED BY FIRST SUMMARIZING THE
        &WRITE(                 DESIRED TEMPORARY DATASETS.
10-RECORDS

HSYNS
        &WRITE( )
        &WRITE(SYNTAX:  'SHOW' FIELDNAME(S) 'IN' ITEM NUMBER(S)
        &WRITE(         WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
        &WRITE(         EVEN THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
        &WRITE(         USER-SUPPLIED VARIABLES.
5-RECORDS

HSUMM
        &WRITE( )
        &WRITE(COMMAND:  SUMMARIZE
        &IF(&PARM(1),EQ,) &THEN &HSUMM &THEN &HSUMO &THEN &HSUMD
        &THEN &READ(PRESS CR> TO CONTINUE, <BREAK> TO QUIT)
        &IF(&LASTIN,EQ,BREAK) &THEN &HSUMF
        &IF(&PARM(1),EQ,*) &THEN &HSUMF &THEN &WRITE( ) &THEN &RESTART(TOP)
        &IF(&PARM(2),EQ,*) &THEN &HSUMO &THEN &HSUMD &THEN &WRITE( )
        &THEN &RESTART(TOP)
        &IF(&PARM(3),EQ,*) &THEN &HSUMD &THEN &HSUMD &THEN &WRITE( )
        &THEN &RESTART(TOP)
        &IF(&PARM(4),EQ,*) &ACT.&HSUMME)
11-RECORDS
```

```
.HSET3
          &WRITE( )
          &WRITE(                DEFAULTS: BRIEF OFF, SYN OFF, DOCENT OFF, COMMAS ON
          &WRITE(                REQUIRED: ONE CHARACTERISTIC AND ONE STATUS
3-RECORDS

.HSET4                                                                                  00000010
          &WRITE( )                                                                     00000020
          &WRITE(EXAMPLES: SET,BRIEF,ON                                                 00000030
          &WRITE(          SET,DOCENT,OFF                                               00000040
          &WRITE( )                                                                     00000050
          &WRITE(          SET,COMMAS,ON                                                00000060
          &WRITE(          SET SYN ON                                                   00000070
          &WRITE( )                                                                     00000080
          &RESTART(TOP)                                                                 00000090
8-RECORDS

.HSET5
          &WRITE( )
          &WRITE(FUNCTION: ESTABLISHES CHARACTERISTICS OF THE SEARCH ENVIRONMENT.
2-RECORDS

.HSET6
          &WRITE( )
          &WRITE(NOTE:     ONLY ONE CHARACTERISTIC-STATUS PAIR MAY BE SPECIFIED
          &WRITE(          PER SET COMMAND, BUT MULTIPLE SET COMMANDS MAY BE
          &WRITE(          DECLARED.
          &WRITE(          A STATUS WILL REMAIN IN EFFECT UNTIL CHANGED.
5-RECORDS

.HSET7                                                                                  00000010
          &WRITE( )                                                                     00000020
          &WRITE(OPTIONS:                                                               00000030
          &WRITE(   CHARACTERISTIC - 'BRIEF'   ELIMINATES MESSAGES ABOUT                00000040
          &WRITE(                              INTERMEDIATE TEMPORARY DATABASES.        00000050
          &WRITE(                    'SYN'     CAUSES SUBSEQUENT SEARCHES TO USE       00000060
          &WRITE(                              THE SPECIFIED WORD AND ALL OF ITS       00000070
          &WRITE(                              PRE-DEFINED SYNONYMS.                   00000080
          &WRITE(                    'DOCENT'  RETURNS NUMBER OF DOCUMENTS IN          00000090
          &WRITE(                              ADDITION TO NUMBER OF POSTINGS.         00000100
          &WRITE(                    'COMMAS'  IN OFF STATUS, ALLOWS SPACES            00000110
          &WRITE(                              INSTEAD OF COMMAS BETWEEN WORDS IN      00000120
          &WRITE(                              COMMAND STRINGS.                        00000130
          &WRITE(   STATUS -         'ON'      TURNS ON CHARACTERISTIC OPERAND.        00000140
          &WRITE(                    'OFF'     TURNS OFF CHARACTERISTIC OPERAND.       00000150
14-RECORDS

.HSET8
          &WRITE( )
          &WRITE(SYNTAX:   'SET',CHARACTERISTIC,STATUS
          &WRITE(          WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
          &WRITE(          MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
          &WRITE(          USER-SUPPLIED VARIABLES.
5-RECORDS

.HSHOW
          &WRITE( )
          &WRITE(COMMAND:  SHOW
          &IF(&PARM(3),EQ,) &THEN &HSHOWF &THEN &HSHOWS &THEN &HSHOWD
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT)
          &THEN &HSHOWF &THEN &HSHOWF
          &IF(&PARM(3),EQ,F) &THEN &HSHOWF &THEN &WRITE( ) &THEN &RESTART(TOP)
          &IF(&PARM(3),EQ,S) &THEN &HSHOWS &THEN &HSHOWD &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),EQ,D) &THEN &HSHOWD &THEN &HSHOWD &THEN &WRITE( )
          &THEN &RESTART(TOP)
          &IF(&PARM(3),EQ,F) &GOTO(HSHOWF)
11-RECORDS
```

```
HSUMMD
        &WRITE )
        &WRITE          DEFAULT: SET NUMBER = 1
        &WRITE                   IF THIS IS THE FIRST INVOCATION OF
        &WRITE                   THE SUMMARIZE COMMAND.
        &WRITE                OR THE MOST RECENT SET NUMBER,
        &WRITE                   IF THERE HAS BEEN A PREVIOUS INVOCATION
        &WRITE                   OF THE SUMMARIZE COMMAND.
7-RECORDS

HSUMME
        &WRITE )
        &WRITE&ABBREV: SUMMARIZE
        &WRITE          SUMMARIZE )
        &WRITE )
        &RESTART&UP)
5-RECORDS

HSUMMF
        &WRITE )
        &WRITE&FUNCTION: PRINTS THE TITLES OF THE DOCUMENTS IN THE SELECTED
        &WRITE          TEMPORARY DATABASE.  THIS COMMAND ALSO SETS THE MOST
        &WRITE          RECENTLY SUMMARIZED DATASET FOR THE SHOW COMMAND.
4-RECORDS

HSUMMO
        &WRITE )
        &WRITE&OPERANDS:
        &WRITE   SET NUMBER  - SPECIFIES THE TEMPORARY DATASET NUMBER.
        &WRITE                 MULTIPLE NUMBERS ARE NOT ALLOWED.
4-RECORDS

HSUMMS
        &WRITE )
        &WRITE&SYNTAX:   'SUMMARIZE' SET NUMBER
        &WRITE           WORDS IN APOSTROPHES ARE ENTERED EXACTLY AS SHOWN,
        &WRITE           MINUS THE APOSTROPHES.  WORDS NOT IN APOSTROPHES ARE
        &WRITE           USER-SUPPLIED VARIABLES.
5-RECORDS

HUSER
        &WRITE )
        &WRITE&COMMAND:  USER
        &WRITE )
        &WRITE&FUNCTION: ENTERS USER LANGUAGE MODE.
        &WRITE )
        &WRITE&SYNTAX:   USER
        &WRITE )
        &WRITE&OPERANDS: NONE.
        &WRITE )
        &WRITE&NOTE:     TO RETURN TO PROXIMITY SEARCHING FOR THE USER
        &WRITE           LANGUAGE, ENTER &QUIT.
        &WRITE )
        &RESTART&UP)
13-RECORDS

FRACOUT
        &IF(&SYS)1&LEXIT-1)                                          517.
        &IF(&SYS)2&LEXIT-1)                                          518.
        &IF(&SYS)3)                                                  519.
3-RECORDS

FFILE
        &IF(&SUMMARY)&LEXIT-1)
        &IF(&SUBUTIL)&LEXIT-1)
        &IF(&FILEFLAT)&LEXIT-1)
        &IF(&SYS)
4-RECORDS
```

```
IFANDOR
         &IF(&UPPRES) &EXIT(-1)                                          523.
         &IF(...)                                                        524.
2-RECORDS

INC
         &PARM(2) .                                                      526.
         &&&START(&&OP)                                                  527.
2-RECORDS

INSCOMMA
         MACRO DEFINE TEMP &PARM.
         OPTION ASOLIT 5.
         MACRO EDIT TEMP ' ' ' ',EDIT TEMP ', ' ',',EDIT TEMP ' ' ',',
         OPTION 5.
         &PARSE(&IF W) &&&FUNCTION(2)
5-RECORDS

INSPARM
         MACRO DEFINE TEMP &TEMP '&&UNFLD.=&&UNVAL' &PARM.               529.
1-RECORD

IOLOOP
         MACRO DEFINE CONVAL &PARM(&CP).                                 531.
         MACRO DEFINE CP &+1(&CP).                                       532.
         &&CALL(IOLOOP2)                                                 533.
3-RECORDS

IOLOOP2
         MACRO DEFINE CONWORK &PARM(&CP).                                535.
         &TOKEN(&CONWORD)                                                536.
         MACRO DEFINE CONVAL &PARM(&CP).                                 537.
         MACRO DEFINE CP &+1(&CP). &&CALL(.OLOOP2)                       538.
4-RECORDS

KEYFCHK
0-RECORDS

MERGE
         MISS MERGE &PREVMOD AS UNIQUE.                                  00000615
         &OPTION(&MERGE - &PREVMOD)                                      00000620
         MACRO DEFINE PREVMOD &MODNAME.                                  00000630
         &NEWSRCH                                                        00000640
4-RECORDS

MERGIT
0-RECORDS

NEWSRCH
         &IF(&UNCONT,EQ,NF)&RESTART(TOP)
         &&UNFLDS
         &RESTART(TOP)
3-RECORDS

NEXTWORD
         &PARSE( )
         MACRO DEFINE CURPARM &+1(&CURPARM).
         MACRO DEFINE CONWORD &PARM(&CURPARM).
         &IF(&CONWORD,EQ,)&CALL(OPER)
         &IF(&CONWORD,EQ,OR)&CALL(NEXTWORD)
         &PARM(&CONWORD)
         &CALL(OPER)
         &IF(&CONWORD,EQ,...)  &CALL(JOB)
         &IF(&PARM(1),EQ,$)&CALL(&PARM)
         &IF(&PARM(1),EQ,*)&CALL(ALL$)
         OPTION ASSIGN $.
         FIND IN INDEX &CONWORD ,
           ,&CONWORD,&MOD) POSTING &MODELS AS UNIQUE
         PRINT &MODELS.
         &IF(&PARM(1)&EQ)&&BEGINNING(&CONWORD)
         OPTION *.
         &&BEGINNING(KEYWORD NOT FOUND IN INDEX)&&EGOTO(UPMODS)
         &IF(&MODELS,EQ,0)&TYPE(NO MODELS MERGE &MODNAME AS &MODNAME.)
         &GOTO(&CONWORD)
         MACRO DEFINE &MODNAME.
         MACRO DEFINE PREVMOD &PREVMOD &MODNAME.
         &GOTO(.EXIT)
22-RECORDS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
NXWORD      MACRO DEFINE FOR TEST STEP DIF(&CURWORD,EQ,NOT)&THEN(AND) &CURWORD.
            &GOTO(NXTFLD)
2-RECORDS
INDSCHK
            &IF(D)&RETURN                                                    559.
            &IF(I)&RETURN                                                    557.
            &IF(F)                                                           558.
3-RECORDS
NXTFLD
            MACRO DEFINE COMPARE &IF(&CURPARM).                              560.
            MACRO DEFINE CMPTWO &PARM(&CURPARM).                             561.
            MACRO DEFINE COMPARE &IF(&CURPARM).                              562.
            MACRO DEFINE COMPARE &PARM(&CURPARM).                            563.
            &RETURN(&CURFLD) &GOTO(PRSTVC)                                   564.
            &IF(&MAX(&CURWORD)) &GOTO(&NIL.)                                 565.
            &GOTO(&PRSTVC)
7-RECORDS
NXTKWRD
            &WHILE(&CURWORD)&GOTO(&KWDFND)                                   570.
            &WRITE(&CURWORD   IS AN INVALID LOGICAL OPERATOR)                571.
            &RESTART(TOP)                                                    572.
3-RECORDS
OPPRES
0-RECORDS
OPCHK
            &IF(&PREVHOLD,E,,)&GOTO(FALSE)
            &IF(AND)&EXIT(-1)
            &IF(&)&EXIT(-1)
            &IF(&AND)&EXIT(-1)
            &IF(&EQ)&EXIT(-1)
            &IF(NOT)&EXIT(-1)
            &IF(&+&)
7-RECORDS
OPER
            &IF(&PREVHOLD,NE,&CURHOLD)&MERGE
            &IF(&PREVOP,EQ,)&GOTO(OPER2)
            &IF(&CURWORD,EQ,)&GOTO(OPER3)
            &PUSH(&CURWORD)&GOTO(OPERMORE)
            &GOTO(OPER2)
5-RECORDS
OPERROR
            &WRITE(&CURWORD)                                                 604.
            &WRITE(IS AN INVALID OPERATOR IN THIS USAGE)                     605.
            &GOTO(NEWSRCH)                                                   606.
3-RECORDS
OPER2
            &IF(&CURWORD,EQ,)&GOTO(NEWSRCH)
            &IF(&CURWORD,EQ,=)&GOTO(EPARM)
            MACRO DEFINE OPNAME &OPNAME &PREVHOLD &IF(&CURWORD,E,,NOT)&CURWORD.
            MACRO DEFINE PREVOP &RETURN(&CURWORD).
            MACRO DEFINE PREVHOLD.
            &GOTO(NEXTWORD)
6-RECORDS
OPER3
            &OPNAME(&PREVOP)
1-RECORDS
```

```
OPGO
          &IF(&GOTO(=OUTL=))
          &IF(&OBJ)&GOTO(ADJ)
          &IF(*=0)&GOTO(ANO)
          &IF(&ENT&GOTO(S))
          &IF(M)&GOTO(M)
5-RECORDS
OPNOT
          &IF=&NOT(&PREVIOUS) &GOTO(FALSE)                              etc.
          &GOTO(OPERROR)                                                &IF.
2-RECORDS
OUTL
          &WRITE( )
          &IF(&PARM(2),EQ,) &THEN &OUTLL
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT.)
          &THEN &OUTLO &THEN &RESTART(TOP)
          &IF(&PARM(2),EQ,LAB) &THEN &OUTLL &THEN &RESTART(TOP)
          &IF(&PARM(2),EQ,DIS) &THEN &OUTLO &THEN &RESTART(TOP)
          &IF(&PARM(2),EQ,ALL) &THEN &OUTLL
          &THEN &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT.)
          &THEN &OUTLO &THEN &RESTART(TOP)
          &WRITE(INVALID OPERAND.)
          &WRITE(VALID OPERANDS ARE:  LAB, DISEASE, ALL.)
          &RESTART(TOP)
12-RECORDS
OUTLO
          &WRITE( )
          &WRITE(THE ITEMS DESCRIBING INFECTIOUS DISEASES FOLLOW THIS
          &WRITE(GENERAL OUTLINE.  THE FIELDNAMES CORRESPONDING TO EACH
          &WRITE(SECTION OF THE OUTLINE ARE SHOWN ON THE RIGHT.
          &WRITE( )
          &WRITE( )
          &WRITE(I.    TITLE                                        | TITLE   |
          &WRITE(II.   SYNONYMS                                     | SYNONYMS |
          &WRITE(III.  DEFINITIONS                                  | DEFINITN |
          &WRITE(IV.   PHYSICIAN PROCEDURE DESCRIPTION              | ADPART   |
          &WRITE(      A.  DIAGNOSIS
          &WRITE(      B.  OCCURENCE
          &WRITE(      C.  INFECTIOUS AGENTS
          &WRITE(V.    PATIENT RELATED DESCRIPTION                  | PATPART  |
          &WRITE(      A.  RESERVOIR
          &WRITE(      B.  MODES OF TRANSMISSION
          &WRITE(      C.  INCUBATION PERIOD
          &WRITE(      D.  PERIOD OF COMMUNICABILITY
          &WRITE(      E.  SUSCEPTIBILITY AND RESISTANCE
          &WRITE( )
          &READ(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT.)
          &WRITE( )
          &WRITE( )
          &WRITE(VI.   CONTROL RELATED DESCRIPTION                  | RESPART  |
          &WRITE(      A.  METHODS OF CONTROL
          &WRITE(      B.  PREVENTIVE MEASURES
          &WRITE(      C.  CONTROL OF PATIENT, CONTACTS, AND
          &WRITE(          THE IMMEDIATE ENVIRONMENT
          &WRITE(      D.  EPIDEMIC MEASURES
          &WRITE(      E.  INTERNATIONAL MEASURES
          &WRITE( )
31-RECORDS
OUTLL
          &WRITE( )
          &WRITE(THE ITEMS DESCRIBING LAB TESTS FOLLOW THIS GENERAL OUTLINE.
          &WRITE(THE FIELDNAMES ASSOCIATED WITH EACH SECTION OF THE OUTLINE
          &WRITE(ARE SHOWN AT RIGHT BETWEEN THE VERTICAL BARS.
          &WRITE( )
          &WRITE( )
          &WRITE(I.    TITLE                                        | TITLE   |
          &WRITE(II.   SYNONYMS                                     | SYNONYMS |
          &WRITE(III.  CODE                                         | CODE    |
          &WRITE(IV.   DEFINITIONS                                  | DEFINITN |
          &WRITE(V.    PHYSICIAN PROCEDURE DESCRIPTION              | ADPART   |
          &WRITE(      A.  RISKS AND CONTRAINDICATIONS
          &WRITE(      B.  NORMAL LIMITS
          &WRITE(      C.  PHYSIOLOGY
          &WRITE(      D.  ELEVATIONS
```

```
&WRITE(        E.  DEPRESSIONS
&WRITE(        F.  VARIABILITY
&WRITE(        G.  COST FACTOR
&WRITE(        H.  PROCESSING TIME
&WRITE( )
&WRITE(PRESS <CR> TO CONTINUE, <BREAK> TO QUIT.)
&WRITE( )
&WRITE( )
&WRITE(VI.     PATIENT RELATED DESCRIPTION                    | PATPART |
&WRITE(        A.  PATIENT PROCESSING INSTRUCTIONS
&WRITE(        B.  PROCEDURE DESCRIPTION FOR PATIENTS
&WRITE(VII.    TEST RELATED DESCRIPTION                       | TESTPART |
&WRITE(        A.  SPECIMEN
&WRITE(        B.  PROCESSING PROCEDURE
&WRITE(        C.  TESTING PROCEDURE
&WRITE(VIII. REFERENCES                                       | REFS     |
&WRITE( )
&WRITE( )
33-RECORDS
P
                %DCL DEFINE CURPARM Z.                                          00000010
                %MACRO DEFINE CURCMD PATH.                                      00000020
                %MACRO DEFINE DOS CURCMD.                                       00000030
                %MACRO DEFINE FORMAT FULL.                                      00000040
                &DEFINE(&PARM(2)) &SET(&CURCMD(&PARM)                           00000050
                %MACRO DEFINE CURCMD &PARM(&CURPARM).                           00000060
                &SET(&CURCMD(&CURPARM)                                          00000070
                %MACRO DEFINE CURPARM &PARM(&CURPARM).                          00000080
                &IF(&CURCMD,NE,)&CALL(&SCAN)                                    00000090
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            00000100
                %MACRO DEFINE CURFLD &*1(&CURPARM).                             00000110
                &CHAIN                                                          00000120
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            00000130
                %MACRO DEFINE CURCMD &PARM(&CURPARM).                           00000140
                &KEYFORM(&CURFLD) &GOTO(&SETSCAN)                               00000150
                &LOCATE(&CURCMD) &GOTO(&SETFIND)                                00000160
                &GOTO(&SETSCAN)                                                 00000170
18-RECORDS
PUPCHK
                &IF(&OUT)&GOTO(&PAOUT)
                &IF(&PREVCP)
                &ELSE&RETURN &IF(&*2)
3-RECORDS
PREDEF
                &IF(&PREDEF)&GOTO(&ISPDEF)                                      642.
                &IF&CURCMD(&PARM(2))&GOTO(&PARM(2))                             643.
                &WRITE(&CURCMD PREDEFINITION - )                                644.
                &WRITE(&PARM(2))                                                645.
                &TESTART(1,*)                                                   646.
5-RECORDS
PRSRVC
                &PRVCMD(&CURCMD)                                                647.
                %MACRO DEFINE TEMP &TEMP &CURFLD &CURCMD.                       648.
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            649.
                %MACRO DEFINE CP 0.  %MACRO DEFINE CURCMD &PARM(&CURPARM).      650.
                %MACRO DEFINE TEMP2.                                            651.
                &PRVCMD(&CURCMD)                                                652.
                %MACRO DEFINE TEMP &TEMP (&TEMP2).                              653.
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            654.
                %MACRO DEFINE CURCP &PARM(&CURPARM)&SETS-&CURCMD&SETSCAN(-1) .   655.
                %MACRO DEFINE TEMP &TEMP &IF(&CURCMD,EQ,&CURCMD,&SETFIND)&CURCMD.  656.
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            657.
                %MACRO DEFINE CURFLD &PARM(&CURPARM).                           658.
                %MACRO DEFINE CURPARM &*1(&CURPARM).                            659.
                %MACRO DEFINE CURCMD &PARM(&CURPARM).                           660.
                &GOTO(PRSRVC)                                                   661.
15-RECORDS
```

```
RECCV
 1-RECORD    &GOTO(&GOBACKLEV)                                          666.
RESI
 1-RECORD    &RESTART                                                   666.
RETAKD
 1-RECORD    &IF(&OUT).RETURN.(&AND) &GOBACKRD                          666.
SCAN
             &IF(&OBSN.EQ.&OBSCAN).OUT.(DISPLAY)                        00000010
             &IF(&FORMAT.NE.FULL)&GOTO(SCAN21)                          00000020
             WRITE IN &OBSN  .                                          00000030
             &WRITE( )                                                  00000040
             &GOTO(RECCNT)                                              00000050
 5-RECORDS
SCANCNT
             OPTIONS MSGLEVEL=I.                                        00000010
             SCAN IN &OBSN ,&COUNT,HOLD=&RECCNT.                        00000020
             OPTIONS=.                                                  00000030
             HOLD=ALL &RECCNT.                                          00000040
             &MSGLEV(ABOVE COMMAND FREES DATABASES CREATED DURING COUNT) 00000050
             &RESTART(TOP)                                              00000060
 6-RECORDS
SCAN21
             SCAN IN &OBSN, &FORMCHK(&FORMAT) . &RESTART(TOP)           666.
 1-RECORD
SETBRIEF
             &IF(ON) &THEN(OPTIONS MSGLEVEL=E,DEFAULT.)                 663.
             &IF(OFF) &THEN(OPTIONS MSGLEVEL=I,DEFAULT.)                664.
 2-RECORDS
SETOB
             MACRO DEFINE JOB &COMMAND. MACRO DEFINE COMPARM &+1(&COMPARM).  666.
             MACRO DEFINE COMMAND &PARM1(&COMPARM).                     667.
 2-RECORDS
SETFIND
              MACRO DEFINE TEMP.                                        173.
             &ATL.(&PARM)                                               174.
             &IF(&ITEMNO.E.,WRITE)&GOTO(END)                            175.
             FIND IN &OBSN &TEMP &FORMCHK(&FORMAT) .                    176.
             &IF(&ITEMNO.GT.1)&THEN(&WRITE(  )                          177.
             &RESTART(TOP)                                              178.
 6-RECORDS
SETFORM
             MACRO DEFINE FORMAT &PARM1(2). MACRO DEFINE COMPARM &+1(&COMPARM).  670.
 1-RECORD
SETPP
             &IF(&PPLINE).EQ.&EXIT(-1)                                  00000010
             MACRO DEFINE COMPARM &+1(&COMPARM).                        00000020
             &MSG(THIS MACRO NOT USED IN STANDARD SYSTEM)               00000030
 3-RECORDS
SETUP
             &IF(&PARM(2),EQ,STANDARD                                   701.
             &WRITE(SYSTEM DATABASE NOT AVAILABLE)&THEN&RESTART(TOP)    702.
             MACRO DEFINE &PARM(2) &PARM(3) .                           703.
             &IF(&PARM(2),EQ,BRIEF)&SETBRIEF(&PARM(3))                  704.
             &RESTART(TOP)                                              705.
 5-RECORDS
```

```
SETSCAN
          MACRO DEFINE ITEM.                                          707.
          &PASS(&PARM)                                                706.
          &IF(&CUR-CMD,EQ,,WHILE)&GOTO(CHS)                           709.
          SCAN I. &OUT &TEMP &UNMATCH(&FORMAT) .                      710.
          &IF(&LAST)(1)&ELSE&WHILE(ITEMS RETRIEVED  0)                711.
          &RESTART(TOP)                                               712.
6-RECORDS

SHOW
          MACRO DEFINE &PARM ..
          MACRO DEFINE CURWORD &PARM(&CURPARM).
          &IF(&CURWORD,EQ,)&GOTO(SHOW4)
          &IF(&CURWORD,EQ,IN)&GOTO(SHOW2)
          MACRO DEFINE FIELDLST &CURWORD (NL) /.
          &GOTO(SHOW1)
6-RECORDS

SHOW1
          MACRO DEFINE CURPARM &+1(&CURPARM).
          MACRO DEFINE CURWORD &PARM(&CURPARM).
          &IF(&CURWORD,EQ,IN)&GOTO(SHOW2)
          &IF(&CURWORD,EQ,)&GOTO(SHOW4)
          MACRO DEFINE FIELDLST &FIELDLST  &CURWORD (NL) / .
          &GOTO(SHOW1)
6-RECORDS

SHOW2
          MACRO DEFINE CURPARM &+1(&CURPARM).
          MACRO DEFINE CURWORD &PARM(&CURPARM).
          &IF(&CURWORD,EQ,)&GOTO(SHOW4)
          MACRO DEFINE ITMLST &ITMLST  &CURWORD .
          &GOTO(SHOW3)
5-RECORDS

SHOW3
          MACRO DEFINE CURPARM &+1(&CURPARM).
          MACRO DEFINE CURWORD &PARM(&CURPARM).
          &IF(&CURWORD,EQ,)&GOTO(SHOW4)
          MACRO DEFINE ITMLST &ITMLST  &CURWORD .
          &GOTO(SHOW3)
5-RECORDS

SHOW4
          FIND IN &SAMPOB ITEMNO = ( &ITMLST ),                       00000015
               &TITLE;                                                00000020
               HEADER 'CPLM ITEM DISPLAY FOR SET &SAMPOB'             00000030
                      'ITEMS: &ITMLST FIELDS: &FIELDLST',             00000040
               TAG EJECT 'ITEM:  ' ITEMNO /                           00000050
                         'TITLE: ' / TITLE (NL) / / /                 00000060
                    &FIELDLST .                                       00000070
          &RESTART(TOP)                                               00000080
8-RECORDS

SHOW1
          MACRO DEFINE CURPARM 1. MACRO DEFINE &PNAME.
          MACRO DEFINE PREVOP. MACRO DEFINE PREVWORD.
          MACRO DEFINE FIELDS.
          &IFFIELD(&PARM(2))&GOTO(&PARM(2))
          &GOTO(PLAIN-WORD)
5-RECORDS

SS
          OPTION ADDITI.                                              726.
          RULE WITH SAME SENTENCE &OPNAME &PREVWORD AS UNIQUE.        727.
          OPTI. *.                                                    728.
          &DEFJOB(SAME SENTENCE - &OPNAME &PREVWORD)                  729.
          &OPTION(NEASHOW)
5-RECORDS
```

```
SSETFIND
          MACRO DEFINE TEMP.                                                    726.
          &AREA(&PARM)                                                          727.
          OPTION MSGLIM 1.                                                      728.
          FIND IN &DDT &TEMP,COUNT,HOLD UNIQUE &DEFDBK(&SEARCH)                 729.
          OPTION *.                                                             730.
          &RESTART(TOP)                                                         731.
6-RECORDS

SSETSCAN
          MACRO DEFINE TEMP.                                                    00000610
          &PARM(&PARM)                                                          00000620
          OPTION MSGLIM 1.                                                      00000630
          SCAN IN &DDT &TEMP,COUNT,HOLD CURRENT.                                00000640
          OPTION *.                                                             00000650
          HOLD FILE CURRENT.                                                    00000655
          &RESTART(TOP).                                                        00000660
7-RECORDS

STARTUP
          MACRO DEFINE DDTDB CPLM1.                                             00000610
          OPEN FILE CPLM1, FILE CPLM2 AS INDEX INDEX(&DDTDB).                   00000620
          &SELL(,FILE SYN,CORRECT SYN TO INDEX BY SYNONYM KEYWORD.)             00000630
          &MESSAGE ABOVE IS FOR THE SYN FEATURE - NOT IMPLEMENTED HERE)         00000640
          &WRITE(&HELLO-)                                                       00000650
          &WRITE( )                                                             00000660
          &WRITE(&RECORDS TO PROXIMITY SEARCHING IN THE CPLM1 DATABASE.)        00000670
          OPTION PSEARCH 20,NOTIMES,TASKIP 0,POOLMIN 22,RECMAX 8,CLEAR1.       00000680
          ISD SKIPOUT OPEN.                                                     00000684
          ISD SKIPOUT TERMTYPE ADM3A.                                           00000688
          MACRO DEFINE SMAXP 1.                                                 00000695
          MACRO DEFINE FLOAT ADVANT.                                            00000696
          MACRO DEFINE TIMEST 1.                                                00000697
          MACRO DEFINE CURLEV TOP.                                              00000698
          MACRO DEFINE WITH &&OUTDATBEGIN)     .                                00000700
          MACRO DEFINE NMOLD 0.                                                 00000710
          MACRO DEFINE BRIEF OFF.                                               00000720
          MACRO DEFINE CLEAR1 OFF.                                              00000730
          MACRO DEFINE SYN OFF.                                                 00000740
          MACRO DEFINE SIZE.                                                    00000750
          ALLOCATE MAST 24, PAPI 600, MAXR 100, IPCI 30, MAIN 5000.             00000760
          MACRO LOAD BEGIN TOP RELEV DEFDBK NEXTWORD                            00000770
          SPOOL SETFORM SELECT SETDB CHANL                                      00000780
          KEYCHK ECHK TOCHK TOD OPCHK CPER                                      00000790
          SHOW SHOW1 SHOW2 SHOW3 SHOW4                                          00000792
          INSCHMA TOP2.                                                         00000200
          &GOTO(BEGIN)                                                          00000210
27-RECORDS

SUMMRY
          MACRO DEFINE COMMAND &PARM(2).                                        00000010
          &IF(&SUBWORD3,EQ,)&GOTO(SUMMRY1)                                      00000020
          MACRO DEFINE SMRY08 &COMMAND.                                         00000030
          &GOTO(SUMMRY1)                                                        00000040
4-RECORDS

SUMMRY1
          SCAN IN &SMRY08, HEADER 'CPLM SEARCH SUMMARY FOR SET &SMRY08',        00000610
            TAB ITEMID TITLE (NL).                                              00000620
          &WRITE( )                                                             00000625
          &RESTART(TOP)                                                         00000630
4-RECORDS

TOP
          MACRO DEFINE COMMAND &COMMAND.                                        00000010
          MACRO DEFINE PREVIOUS &PREVIOUS &COMMAND.                             00000020
          &GOTO(TEXTWORD)
3-RECORDS
```

```
TOCHK
          &*1(&PARM(1))&THEN&GOTO
          &THEN &RETURN
          &IF(###)
3-RECORDS

TEMPSTAT
          ALLOCATE MAST 24,MAPI 600,MAXM 100,HCI 50,MATA 4000.                    748.
1-RECORD

THCHK
          &IF(&I)&THEN&DOPCP(&I)&THENELEXIT(-1)                                   753.
          &IF(&J)&THEN&DOPCP(&J)&THENELEXIT(-1)                                   751.
          MACRO DEFINE TEMP &TEMP                                                 752.
          '&COMFLD.=&CONVAL&IF(&CONOP,EQ,IS)&THEN(*)' &IF()                       753.
          &ELSE(&A).    &THENELEXIT(-2)                                           754.
5-RECORDS

TOP
          &ASKLETTER CONTINUE)
          &IF(&SCR)&EXIT(0)
          &IF(&CONMAS,EQ,OFF)&GOTO(INSCONMA)
          &GOTO(TOP2)
4-RECORDS

TOP2
          &IF(&PARM(1),EQ,PRINT)&GOTO(P)
          &IF(&PARM(1),EQ,SEARCH)&GOTO(SPCHL)
          &IF(&PARM(1),EQ,COUNT)&GOTO(ENTER)
          &IF(&PARM(1),EQ,HISTORY)&GOTO(DSPL)
          &IF(&PARM(1),EQ,FREE)&GOTO(FREE)
          &IF(&PARM(1),EQ,HELP)&GOTO(HELP)
          &IF(&PARM(1),EQ,DICT)&GOTO(DICTP)
          &IF(&PARM(1),EQ,DISPLAY)&GOTO(DRWS)
          &IF(END)&THENTYPE OPTION WASN'T E. CAUSE SYSTEM.)
          &IF(&PARM(1),EQ,IN)&GOTO(IN)
          &IF(&PARM(1),EQ,SETUP)&GOTO(SETUP)
          &IF(&PARM(1),IS,OUT)&GOTO(OUTF)
          &IF(&PARM(1),IS,SUM)&GOTO(SUMARY)
          &IF(&PARM(1),EQ,SHOW)&GOTO(SHOW)
          &IF()&GOTO(TOP)
          &WRITE(INVALID COMMAND - USE HELP COMMAND)
          &GOTO(TOP)
17-RECORDS

UPCP
          MACRO DEFINE CP &*1(&CP).                                               756.
          &IF(TO) &GOTO([*&PARM])                                                 757.
          &IF(&N(&CONOP)&GOTO(INSPARM)                                            758.
          MACRO DEFINE TEMP &TEMP '&COMFLD.=&CONVAL.*' OR.                        759.
4-RECORDS

UPHOLD
          MACRO DEFINE NHOLD &*1(&WHOLD).
          MACRO DEFINE TOSCRHOLD.
          &RESTART(TOP)
3-RECORDS

4
          OPTION DOUBLE 1.                                                        765.
          HOLD MATCH WITHIN &.WORDS WORDS &COMNAME &PREVHOLD AS UNIQUE.           766.
          OPTION 0.                                                               767.
          &OFFLINE(WITHIN &WORDS - &COMNAME &PREVHOLD)                            768.
          &GOTO(TOSPECH)                                                          769.
5-RECORDS
```

```
ANALYS
        MACRO DEFINE TEMP VARIABLE.                                  111.
        &TEMP                                                        112.
Z-RECORDS
MSET
        MACRO DEFINE COMPARE &FILEUNIPARM1.                          114.
        MACRO DEFINE RECORD &PARMFILEUNIPARM2.                       115.
Z-RECORDS
```

- -

```
        SOURCE LIBRARY DATA SET: LRECL=80,BLKSIZE=3200,
        DSNAME=U.ABCDEFG.CPLMDOC.PRJ.ASC,VOL=SER=USER04
```

```
  201   MEMBERS DEFINED IN SOURCE LIBRARY.
    0   ALIASES DEFINED IN SOURCE LIBRARY.
   10   DIRECTORY BLOCKS ALLOCATED.
   10   DIRECTORY BLOCKS USED.
  201   MEMBERS SELECTED.
 1274   RECORDS READ FROM SOURCE LIBRARY.
```

APPENDIX G

NORTHEAST REGIONAL DATA CENTER
NORMAL SYSTEM OPERATING HOURS

8:00 a.m.	-	3	a.m.	Monday - Thursday
8:00 a.m.	-	5	a.m.	Friday
12 noon	-	3	a.m.	Saturday
4:00 p.m.	-	3	a.m.	Sunday

CICS is available from 8 a.m. - 8:30 p.m. Mon. - Fri.
Remote-batch printing is available until 3:45 a.m.
Sat. - Thurs. and 5:45 a.m. on Fridays.

The card reader in the SSRB lobby will be closed at 3 a.m. on Sat - Thurs and 5 a.m. on Fridays. Output will continue to be filed until 3:30 a.m. (5:30 a.m. on Fridays). Any output not filed by this time may be picked up at 8 a.m. (12 noon on Saturdays and 4 p.m. on Sundays).

If computing services are not available during the times listed above, call the System Status Hotline (904-392-6775) for an explanation.

HOLIDAYS AND DOWNTIME

NERDC Administrative offices are closed on the following holidays: New Years's Day, Memorial Day, July 4th, Labor Day, Veteran's Day, Thanksgiving Day and the day after, and Christmas Day and the day after. Limited computing services will be available from 8 a.m. - 4 p.m. A skeleton operations staff will be on duty and tape and disk mounts will be made only at the operator's discretion as resources are available. Normal services will be available from 4 p.m. - 3 a.m. On some holidays, unassisted services may be offered. Unassisted services mean that no tape or disk setups will be performed; no unit record I/O will be done at the local site, the lobby will be closed, output will not be filed, tapes will not be accepted or delivered; no services will be guaranteed -- if we go down, go home. Schedules for these and other holidays and required downtime will be announced, when possible, in /Update, the Memo System, and in .TIMES. The proposed holiday schedule for 1932 is in memo number 82141.0001.

APPENDIX H

A USER INTRODUCTION TO CPLM

CPLM is an online system which allows retrieval of specific information from a large textual database. Access to CPLM is obtained through a computer terminal connected to the computer system at the Northeast Regional Data Center (NERDC) of the University of Florida. Instructions for gaining access to CPLM through a remote terminal is contained in chapter four of this documentation.

THE CPLM LANGUAGE

The CPLM language consists of a set of user selected command verbs (eg. DISPLAY) followed by one or more key words or phrases. The general syntax is:

<COMMAND VERB> <OPTIONAL PHRASE> <CR>*

The available command verbs are fully defined in chapter four of this documentation. The user should take the time to read about each of these command verbs at this time if he is not already familiar with them.

THE CPLM DATABASE

The database which CPLM will search is composed of two major categories of data (or datasets) and is named CPLM1 (see Figure 1J).

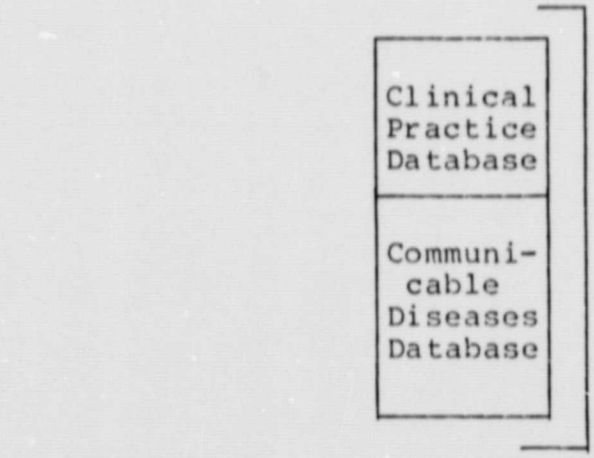


Figure 1J: The current organization of the CPLM1 Database.

* <CR> is ASCII carriage return.

The first of these is the Clinical Practice Database and the second is the Communicable Diseases Database. The current documents (or subjects) currently contained in these two databases is listed in Appendix E and F. Only information about these subjects is currently available. The data itself was extracted from numerous sources by permission for use specifically in this implementation. Much of the data has not been validated by a competent medical authority and some typographical errors may be found. An attempt has been made to keep all spellings consistent with the Dorland's Illustrated Medical Dictionary 25th edition and the authors would appreciate being notified of any discrepancies discovered by the user.

Once a document has been selected for display the specific information organization will be in one of two presentation outline forms. Appendix B and D describe the presentation form for the Clinical Practice Documents and the Communicable Disease Documents respectively. Note that the major headings (listed in all capital letters) correspond to section names within a document that may be specified in the <OPTIONAL PHRASE> portion of a command. These are typically used to restrict the range of a document that is displayed or searched. The section describing how to find documents meeting multiple requirements is described below. The user may review these organizations online with the OUTLINE command.

EXAMPLE H1

COMMAND: OUTLINE

ENTER COMMAND>outline disease

THE ITEMS DESCRIBING INFECTIOUS DISEASES FOLLOW THIS GENERAL OUTLINE. THE FIELDNAMES CORRESPONDING TO EACH SECTION OF THE OUTLINE ARE SHOWN ON THE RIGHT.

I.	TITLE	TITLE
II.	SYNONYMS	SYNONYMS
III.	DEFINITIONS	DEFINITN
IV.	PHYSICIAN PROCEDURE DESCRIPTION	MDPART
	< etc. >	

CPLM TEMPORARY DATASETS

Whenever CPLM performs any search or count operation it causes the creation of one or more temporary datasets which are labeled by a set number and a content label. Each temporary dataset contains only documents with references matching the content label. Future references to these temporary datasets by set number are allowed and frequently can reduce typing and search times significantly.

CPLM COMMAND MODES

CPLM expects commands to be entered when it is one of three modes. These modes are: 1) Command Mode,
2) Subcommand Mode, and
3) User Language Mode.

Command Mode is the most common mode and is recognized by the prompt:

ENTER COMMAND>

which will appear at the left margin of the screen on the last line. All of the commands described under major subheadings in chapter four may be entered from Command Mode.

Subcommand Mode is reached after execution of certain commands in Command Mode (eg: PRINT). Subcommand Mode is recognized by a prompt such as:

ENTER HILITE COMMAND>

Which will vary according to the Command executed to enter Subcommand Mode.

In either Command Mode or Subcommand Mode, typing the command verb HELP will provide information on valid commands that may be entered. In either case pressing the <BREAK> key will return the user to Command Mode.

User Language Mode is reached after entering the command verb USER while in command mode. Once in User Mode, only Inquire User Language commands will be recognized. It is recommended that users not utilize this mode without a thorough knowledge of INQUIRE®. Users may return to Command Mode by entering the command &BEGIN or by pressing the <BREAK> key. No HELP functions are available in User Language Mode.

USER HELP FACILITY

In either Command Mode or Subcommand Mode the user may request assistance by entering the command verb HELP. In Command Mode the general command syntax is:

HELP [COMMAND] [SEGMENT] <CR>

COMMAND is an optional element designating a specific command. All of the commands in CPLM have helps available. Each help displays the same information contained in chapter four for the command.

SEGMENT is an optional element that allows specification of subsections of the entire help command. Valid segments include FUNCTION, SYNTAX, OPERANDS, and EXAMPLES.

The FUNCTION segment describes the functions performed by the command. SYNTAX displays the command syntax and default values. OPERANDS describes the operands used in the SYNTAX segment. EXAMPLES gives a few examples of the command syntax, with various optional operands. INQUIRE is a registered trademark of Infodata Systems Inc., Rochester, New York.

FINDING INFORMATION ABOUT A SPECIFIC WORD

If the user knows specifically what subject he is searching for and that subject can be described in one word the retrieval is quite straightforward. For example, to retrieve all documents containing information about Brucellosis the command would be:

SEARCH BRUCELLOSIS <CR>

The command verb in this case is SEARCH and BRUCELLOSIS is the optional phrase. CPLM does not pay any attention to case during command (or subcommand) entry. CPLM will construct a temporary dataset containing all of the documents with any occurrence (called a posting) of the word Brucellosis. This dataset will be identified in the future by references to this set number. Each set will be comprised of sets of documents or items (subject headings or document names are listed in Appendix G and H). The number of items in the last set created will be counted and displayed as in example H1.

EXAMPLE H2

```
COMMAND: SEARCH

ENTER COMMAND>search brucellosis


TEMPORARY DATABASE 1          CONTAINS      9 POSTINGS FROM
                                            CPLM1
SET 1 - BRUCELLOSIS


TEMPORARY DATABASE DOCCNT     CONTAINS      7 ITEMS FROM
                                            CPLM1
```

The number of documents contained in the temporary dataset can also be determined by use of the <COUNT> command. For example if the Brucellosis temporary dataset

was numbered one, then the command in example H2 would count the number of documents present.

EXAMPLE H3

```
COMMAND: COUNT

ENTER COMMAND>count 1

TEMPORARY DATABASE UNCCNT   CONTAINS          7 ITEMS FROM
                                               CPLM1
```

In order to obtain the name and reference number for selected documents in a temporary dataset the <SUMMARIZE> command is used. Entering <SUMMARIZE dn> where dn is the temporary dataset number will display the document and its internal reference number (referred to as the Document Number).

EXAMPLE H4

```
COMMAND: SUMMARIZE

ENTER COMMAND>summarize 1
                    CPLM SEARCH SUMMARY FOR SET 1

          ITEM    TITLE
            91    COLORADO TICK FEVER AND OTHER TICK-BORNE
                  FEVERS.
            76    BRUCELLOSIS
            59    BLOOD CULTURE
            45    WBC
            44    DIFFERENTIAL
            29    EOSINOPHIL COUNT
            11    BRUCELLA CULTURE
```

When the user has obtained the document number, then selected sections of the document may be viewed by using the <SHOW> command. Syntax for the SHOW command is:

SHOW [Fieldname] IN [sn]<CR>

Fieldname is a major heading obtained from the outline format of the database (see appendix B and D); and sn is a temporary dataset number.[1]

EXAMPLE H5

```
COMMAND: SHOW

ENTER COMMAND>show mdpart in 11
                        CPLM ITEM DISPLAY FOR SET 1
                      ITEMS: 11 FIELD: MDPART(NL) /
ITEM:                 11
TITLE:
BIOPSY CULTURE

A.  RISKS AND CONTRAINDICATIONS - THOSE ASSOCIATED WITH
    STERNAL MARROW ASPIRATION (BLEEDING, PAIN, INFECTION,
    PENETRATION OF STERNUM WITH LACERATION OF VESSELS OR
    PNEUMOTHORAX).
B.  NORMAL LIMITS - NOT NORMALLY PRESENT
C.  PHYSIOLOGY
      < etc >
```

If a specific keyword spelling is not known then the Command <DISPLAY> may be used to list all similarly spelled keywords. This is particularly important as CPLM currently handles plurals, contractions, possessives, etc. as independent entities.

EXAMPLE H6

```
COMMAND:  DISPLAY

ENTER COMMAND>display liver

 NUMBER
   OF
POSTINGS    KEYWORD
     1      LIVERS
    80      LIVER

ENTER COMMAND>display wbc hemoglobin

 NUMBER
   OF
POSTINGS    KEYWORD
     1      WBC'S
    10      WBC
    14      HEMOGLOBINURIA
     2      HEMOGLOBINS
     2      HEMOGLOBINOPATHY
     7      HEMOGLOBINOPATHIES
     1      HEMOGLOBINOMETRY
     1      HEMOGLOBINEMIA
    63      HEMOGLOBIN
```

UTILIZING MORE RESTRICTIVE SEARCH CATEGORIES

If the required search criteria involves more than one keyword then the SEARCH syntax is greatly expanded as:

SEARCH [KEYWORD] [RELATOR] [KEYWORD]...[RELATOR] [KEYWORD] <CR>

where KEYWORD is the Keyword desired and RELATOR is an element from the set:

AND
OR
NOT
ADJ
SEN
W n

AND, OR, and NOT are Boolean relators signifying the set combinations illustrated in figure 2J. Listing consecutive keywords without an explicit relation implies connective AND.

ADJ means that the words must be adjacent to each other within the document while SEN means that they must occur within the same sentence. The W n construct is satisfied when the two words are within 'n' words of each other.

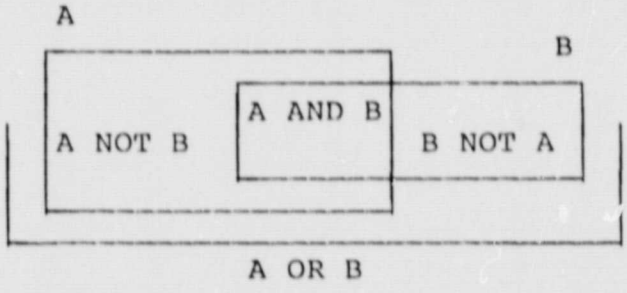


Figure 2J: Venn Diagram illustrating the action of the Boolean Relators AND, OR, and NOT in a CPLM SEARCH optional phrase.

For example, in searching for the occurrence of Sickle Cell Trait within a document then:

SEARCH SICKLE OR CELL <CR>

would result in only documents containing either the words SICKLE or CELL. While the Command:

SEARCH SICKLE AND CELL <CR>

would result in all documents containing both words. A more restrictive search satisfying the desired criteria would be:

SEARCH SICKLE ADJ CELL <CR>

which would result in only those documents containing the word SICKLE adjacent to the word CELL (in either order).

Of course, multiple search criteria may be included in a single SEARCH command.

In cases involving more than one search keyword, each keyword will result in the creation of a temporary dataset containing all documents in which the specified keyword is found. Each of these temporary datasets is identified by a set number and each set number may be used in subsequent searches. Note in the following example that four sets are created, one for each keyword and one for the resultant after application of the implied relator, AND.

EXAMPLE H7

COMMAND: SEARCH

```
ENTER COMMAND>search liver lung ldh

TEMPORARY DATABASE 1        CONTAINS        78 POSTINGS FROM
CPLM1
SET 1 - LIVER

TEMPORARY DATABASE 2        CONTAINS        6 POSTINGS
                                            FROM CPLM1
SET 2 - LUNG

TEMPORARY DATABASE 3        CONTAINS        15 POSTINGS
                                            FROM CPLM1
SET 3 - LDH

TEMPORARY DATABASE 4        CONTAINS        99 POSTINGS
                                            FROM CPLM1
SET 4 - MERGE - 1 2 3

TEMPORARY DATABASE DOCCNT   CONTAINS        37 ITEMS FROM
                                            CPLM1
```

If the user requires a reminder of the temporary datasets has created then he should use the <HISTORY> command as:

EXAMPLE H8

COMMAND: HISTORY

```
ENTER COMMAND>history
SET 1 - LIVER
SET 2 - LUNG
SET 3 - LDH
SET 4 - MERGE - 1 2 3


TEMPORARY     ITEMS  FROM
DATABASE      HELD   DATABASE

1                78  CPLM1
2                 6  CPLM1
3                15  CPLM1
4                99  CPLM1
```

which also results in an item count for each temporary dataset.