NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
Department of Pathology
College of Medicine
University of Florida
Gainesville, Florida

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>Overview of The System</td>
<td>6</td>
</tr>
<tr>
<td>III.</td>
<td>Special Considerations</td>
<td>7</td>
</tr>
<tr>
<td>IV.</td>
<td>Operating Instructions</td>
<td>8</td>
</tr>
<tr>
<td>V.</td>
<td>System Flow Diagrams</td>
<td>21</td>
</tr>
<tr>
<td>VI.</td>
<td>System Database Descriptions</td>
<td>26</td>
</tr>
<tr>
<td>VII.</td>
<td>Generating the Text for the Database</td>
<td>29</td>
</tr>
<tr>
<td>VIII.</td>
<td>Generating the Working Database</td>
<td>31</td>
</tr>
<tr>
<td>IX.</td>
<td>Summary of Error Messages and Codes</td>
<td>33</td>
</tr>
<tr>
<td>X.</td>
<td>Index</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>36</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Listings of documents currently in CPLM Clinical Pathology Laboratory Section</td>
<td>A2</td>
</tr>
<tr>
<td></td>
<td>Infectious Diseases Section</td>
<td>A4</td>
</tr>
<tr>
<td></td>
<td>Example Text</td>
<td>A5</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Outline of input format for Clinical Pathology Laboratory Section</td>
<td>B2</td>
</tr>
<tr>
<td></td>
<td>Infectious Diseases Section</td>
<td>B3</td>
</tr>
<tr>
<td></td>
<td>Example Text</td>
<td>B4</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Outline of CPLM database format for Clinical Pathology Laboratory Section</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>Infectious Diseases Section</td>
<td>C3</td>
</tr>
<tr>
<td>Appendix D</td>
<td>INQUIRE Load Utilities</td>
<td>D2</td>
</tr>
<tr>
<td></td>
<td>INQASML - Set delimiters</td>
<td>D2</td>
</tr>
<tr>
<td></td>
<td>INQKEXT - Extract subset keywords</td>
<td>D3</td>
</tr>
<tr>
<td></td>
<td>INQLoad - Load main text database</td>
<td>D4</td>
</tr>
<tr>
<td></td>
<td>INQPOST - Load postings database for multi-access method</td>
<td>D5</td>
</tr>
<tr>
<td>Appendix E</td>
<td>PL/1 listing of text preprocessor</td>
<td>E1</td>
</tr>
<tr>
<td>Appendix F</td>
<td>INQUIRE macros used by CPLM</td>
<td>F2</td>
</tr>
<tr>
<td></td>
<td>Macro list with command references</td>
<td>F2</td>
</tr>
<tr>
<td></td>
<td>Built in INQUIRE Macros</td>
<td>F6</td>
</tr>
<tr>
<td></td>
<td>Macro structure for selected commands</td>
<td>F8</td>
</tr>
<tr>
<td></td>
<td>Program Notes</td>
<td>F20</td>
</tr>
<tr>
<td></td>
<td>CPLM Macros</td>
<td>F22</td>
</tr>
<tr>
<td>Appendix G</td>
<td>NERDC normal operating schedule</td>
<td>G1</td>
</tr>
<tr>
<td>Appendix H</td>
<td>User tutorial with examples</td>
<td>H1</td>
</tr>
</tbody>
</table>

CPLM Version 2.0

06 June 1982
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 text file. 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.

CPLM Version 2.0
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 thoroughly 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 acoustic modems or via Direct Access Arrangement (DAA) modems. The telephone numbers are:

- 300 baud: 904-392-5313
- 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)**

<table>
<thead>
<tr>
<th>CODE</th>
<th>ASCII (Upper/lower case optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER DATA BITS</td>
<td>7</td>
</tr>
<tr>
<td>NUMBER STOP BITS</td>
<td>1</td>
</tr>
<tr>
<td>PARITY</td>
<td>Even</td>
</tr>
<tr>
<td>HALF DUPLEX</td>
<td></td>
</tr>
</tbody>
</table>

**TI SILENT 700 (300 baud)**

- HALF DUPLEX

**DEC WT/78 (1200 baud)**

<table>
<thead>
<tr>
<th>CODE</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER DATA BITS</td>
<td>8</td>
</tr>
<tr>
<td>NUMBER STOP BITS</td>
<td>1</td>
</tr>
<tr>
<td>PARITY</td>
<td>NONE</td>
</tr>
<tr>
<td>BUFF CONTROL</td>
<td>NO</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>KH HS KS (will not support printer at 1200 baud)</td>
</tr>
</tbody>
</table>

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)

CPLM Version 2.0

- 8 -

06 June 1982
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 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 @4285#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:

CPLM Version 2.0 - 10 - 06 June 1982
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.
Dooloan 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 MDPAI 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'
          Set Number(s)  - Erases all sets.
          - 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.

CPLM Version 2.0       13       06 June 1982
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:

CPLM Version 2.0   - 14 -   06 June 1982
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 pre-defined 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.

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.

CPLM Version 2.0 - 18 - 06 June 1982
'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 dataset. The Item numbers are obtained by first SUMMARIZING the desired temporary datasets.

CPLM Version 2.0 - 19 - 06 June 1982
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 hierarchy diagram as well as basic flow diagrams for the preprocessor.
OVERALL SYSTEM FLOW

1. Obtain basic text
2. Place text in outline format (script)
3. Parse text to generate ISI format
4. Parse ISI format for special keywords
5. Load master database
6. Load postings database
7. Access database via macro language
PART

IF FIRST LINE OF LEVEL

TRUE

OUTPUT APPROPRIATE RECORD HEADER

ELSE

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 (e.g. 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 via 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 INQTEXT.

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

4) UF.A0081909.CLIST - this is a PDS containing the members 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 INQEXT. 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.SEARCH- see 4)

8) UF.A0081909.CPLMPOST.DATA- this dataset is pointed to by the .CPLMPOST.SEARCH and it in turn contains posting entries 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 well 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 INQSCRIPT

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 CPLMTEST.KDUSFIL (Keyword list)
   and CPLMTEST.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.SRSRFIL
   outfiles are CPLMTEST.SEARCH
          CPLMTEST.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.SPLDFIL
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 programming staff to make corrections. There are no user fixable errors other than those prompted by the Interface language.
# CHAPTER 10
## INDEX

ADM3-4: 8

Contains: 1, 5, 7, 21, 12, 15, 16, 18, 27, 28, 29, 30, 34, 47, 49, 69, 72, 74, 75, 78

Count: 11, 12, 13, 14, 16, 17, 29, 34, 39, 40, 49, 50, 52, 53, 56, 57, 58, 68, 69, 72, 74, 75, 79

DEC: 38, 40

DICT: 11, 12, 13, 34, 50

END: 10, 11, 13, 27, 34

Excludes: 11, 16, 34

Free: 10, 11, 13, 14, 34, 36, 49, 50, 63

Help: 11, 13, 14, 16, 34, 50, 51, 52, 68, 73, 74

History: 1, 11, 13, 14, 33, 34, 35, 49, 50, 51, 52, 53, 61, 62, 78, 79

Menu: 16, 34, 68

Modem: 8, 34

NERDC: 1, 8, 9, 10, 27, 34, 70, 71

Outline: 1, 5, 6, 11, 13, 14, 15, 19, 22, 23, 26, 29, 34, 42, 45, 53, 72, 75

Print: 11, 13, 15, 16, 34, 49, 50, 51, 52, 53, 60, 68, 19

Search: 4, 5, 11, 12, 13, 14, 15, 16, 17, 18, 27, 28, 31, 32, 34, 36, 49, 50, 52, 53, 64, 66, 68, 69, 71, 72, 74, 75, 77, 78

Show: 11, 13, 19, 20, 34, 45, 53, 75, 76

Sign-off: 38

Sign-on: 9, 10, 34

Silent 700: 8, 34

Summarize: 11, 13, 14, 19, 20, 34, 53, 75

TI

CPLM Version 2.0 - 34 - 06 June 1982
REFERENCES


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

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actinomyces Culture</td>
</tr>
<tr>
<td>Alkaline Phosphatase</td>
</tr>
<tr>
<td>Antimitochondrial Antibody</td>
</tr>
<tr>
<td>Anti-Nuclear Antibody</td>
</tr>
<tr>
<td>Antismooth Muscle Antibody</td>
</tr>
<tr>
<td>Australia Antigen</td>
</tr>
<tr>
<td>Beta-Strep Screen</td>
</tr>
<tr>
<td>Billirubin</td>
</tr>
<tr>
<td>Biopsy Culture</td>
</tr>
<tr>
<td>Blood Culture</td>
</tr>
<tr>
<td>Bronchial Washing Culture</td>
</tr>
<tr>
<td>Brucella Culture</td>
</tr>
<tr>
<td>BSP</td>
</tr>
<tr>
<td>BUN</td>
</tr>
<tr>
<td>Calcium</td>
</tr>
<tr>
<td>Carbon Dioxide</td>
</tr>
<tr>
<td>C Diphteriae Culture</td>
</tr>
<tr>
<td>Creatinine</td>
</tr>
<tr>
<td>C'3 Complement</td>
</tr>
<tr>
<td>C'4 Complement</td>
</tr>
<tr>
<td>C-Reactive Protein</td>
</tr>
<tr>
<td>Cryoglobulin</td>
</tr>
<tr>
<td>Cryofibrinogen</td>
</tr>
<tr>
<td>Differential</td>
</tr>
<tr>
<td>Direct Coombs</td>
</tr>
<tr>
<td>HBV Antibody Titre</td>
</tr>
<tr>
<td>Eosinophil Count</td>
</tr>
<tr>
<td>ESR</td>
</tr>
<tr>
<td>Fecal Fat</td>
</tr>
<tr>
<td>Fecal Muscle Fiber</td>
</tr>
<tr>
<td>Fetal HGB%</td>
</tr>
<tr>
<td>FTA-ABS Serum</td>
</tr>
<tr>
<td>Glucose</td>
</tr>
<tr>
<td>Gonorrhea Culture</td>
</tr>
<tr>
<td>Hematocrit</td>
</tr>
<tr>
<td>Hemoglobin</td>
</tr>
<tr>
<td>Hemoglobin Electrophoresis</td>
</tr>
<tr>
<td>Joint Fluid Culture</td>
</tr>
<tr>
<td>Lactic Dehydrogenase</td>
</tr>
<tr>
<td>Large Light Chains</td>
</tr>
<tr>
<td>Listeria Culture</td>
</tr>
<tr>
<td>Mono Spot Test</td>
</tr>
<tr>
<td>Occult Blood</td>
</tr>
<tr>
<td>Peritoneal Abscess Culture</td>
</tr>
<tr>
<td>Pertussis Culture</td>
</tr>
<tr>
<td>Pinworm Slide</td>
</tr>
<tr>
<td>Platelet Count</td>
</tr>
<tr>
<td>PT -- Prothrombin Time</td>
</tr>
<tr>
<td>PTT -- Partial Thromboplastin Time</td>
</tr>
<tr>
<td>RBC Count</td>
</tr>
<tr>
<td>CPLM Version 2.0</td>
</tr>
</tbody>
</table>
Reticulocyte Count
Rheumatoid Factor
Rubella Antibody
Serum Protein Electrophoresis
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 Hemorrhagic Fever
Hemorrhagic Fever
Omsk Hemorrhagic Fever and Kyasanur Forest Disease
Primary Amebic Meningoencephalitis
Russian Spring Summer Encephalitis
Sandfly Fever
Viral Arthropod-Borne Hemorrhagic 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 B) 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).
Clinical Pathology Laboratory

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

CPLM Version 2.0 - B2 - 06 June 1982
Infectious Diseases

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

II. Physician Procedure Description
   A. Diagnosis
   B. Occurrence
   C. Infectious Agent

III. Patient Related Description
   A. Reservoir
   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 1
C. CODE - 277
D. DEFINITION - measurement of levels of the 4th element of complement

II. Physician procedure description
A. RISKS - 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

CPLM Version 2.0 - B4 - 06 June 1982
J. Genetic decrease (very rare)
2. False - technical error
3. Interpretations - further tests by dx
   A. Ana, rf, biopsy
   B. Ana, rf, biopsy
   C. Asp, 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

II. 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. PHYSIOLOGY
The complement system is a complex cascade of factors with various immunologic roles: hemolysis, cytotoxicity, chemotaxis, immune adherence, anaphylaxis, viral neutralization, opsonization, kininlike 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 pyrogenic 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).
Clinical Pathology Laboratory

TITLE  (formerly I.A Name)
SYNONYMS  (formerly I.B Synonyms)
CODE  (formerly I.C Code)
DEFINITION  (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)

CPLM Version 2.0 - C2 - 06 June 1982
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 delimiters as well as other special delimiters.

2. INQK4XT - 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 INQK4XT. 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
PL/I OPTIMIZING COMPILER

PARSE: PROCEDURE OPTIONS (MAIN):

SOURCE LISTING

STMT LEV NT

1  01PARSE: 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 dname 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.
  /*
  /********************************************************/
DECLARE /* files */
  TEXT FILE STREAM INPUT,
  ISI FILE STREAM OUTPUT,
  MSG FILE PRINT;

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'),
  ZGNE1 CHAR (4) INIT (''),
  ZGNE2 CHAR (1) INIT ('');

DECLARE /* numeric integer variables */
  I FIXED INIT (0),
  LEN FIXED INIT (0),
  PTR FIXED INIT (0);

DECLARE /* boolean constants */
  TRUE BIT (1) INIT ('1'B),
  FALSE BIT (1) INIT ('0'B);

DECLARE /* boolean variables */
  EGF BIT (1) INIT ('0'B),
  FIRSTMSG BIT (1) INIT ('1'B),
  FIRSTREC BIT (1) INIT ('1'B);

DECLARE /* builtin functions */
  LENGTH BUILTIN,
  SUBSTR BUILTIN;
/* main procedure: PARSE */

ON ENDFILE (TEXT) EOF = TRUE;

OPEN FILE (TEXT) INPUT;
OPEN FILE (ISI) OUTPUT;
OPEN FILE (MSG) PRINT;

CALL CARDIN;

DO WHILE (^EOF);
    /* delete blank lines at the beginning of the file */
    DO WHILE (CARD = '' & FIRSTREC & ^EOF);
    CALL CARDIN;
    END;

    IF ZONE1 = '' THEN DO;
        LINE = SUBSTR (CARD, 5);
        CALL LINEOUT;
    END;

ELSE DO; /* ZONE1 is not blank */
    SELECT (ZONE1);
    WHEN ('I.') CALL TOP;
    WHEN ('II.') CALL ITEM (MDPART);
    WHEN ('III.') CALL ITEM (PATPART);
    WHEN ('IV.') CALL ITEM (TESTPART);
    WHEN ('V.') CALL ITEM (REFS);
    OTHERWISE CALL ERROR;
    END;

    IF ZONE1 ^= 'II.' THEN CALL CARDIN;

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

CLOSE FILE (TEXT);
CLOSE FILE (ISI);
CLOSE FILE (MSG);
PL/I OPTIMIZING COMPILER
PARSE: PROCEDURE OPTIONS (MAIN);

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;

CPLM Version 2.0  - E5 -  06 June 1982
STMT LEV NT

**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(BO));
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 ZONE = ' ' THEN DO;
53 2 2 SELECT (ZONE2);
54 2 3 WHEN ('B') PUT FILE (ISI) EDIT ('SYNONYM') (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 ('DEFINITION') (COL(1), A);
57 2 3 OTHERWISE CALL ERROR;
58 2 1 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 'd a new card */
62 2 1 CALL LINEOUT;
63 2 1 CALL CARDIN;
64 2 1 END;
65 2 0 END TOP;

CPLM Version 2.0  - E6 -  06 June 1982
STMT LEVEL

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 filename */
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;

CPLM Version 2.0

- E7 -

06 June 1982
PL/I OPTIMIZING COMPILER

PARSE: PROCEDURE OPTIONS (MAIN);

STMT LEVEL

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 */
    /* ignore blank lines */
    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;

CPLM Version 2.0 - E8 -
06 June 1982
PL/I OPTIMIZING COMPILER
PARSE: PROCEDURE OPTIONS (MAIN);

STMTLEV NT

93 1 0 "ERROR: PROCEDURE:
//-------------------------------------------------------------------------------
// Title: ERROR
// Purpose: to print all records that are in error.
// Output: to the file MSO. 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 FIRSTMSO THEN DO;
95 2 1 PUT FILE (MSO) EDIT ('The following records are in error: ')
96 2 1 (COL(1),A);  FIRSTMSO = FALSE;
97 2 1 END;
98 2 0 PUT FILE (MSO) EDIT (CARD) (COL(1),A);
99 2 0 END ERROR;

100 1 0 END PARSE;

CPLM Version 2.0 - E9 - 06 June 1982
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 count
ATEQ    count
BEGIN    begin
BEK    display
BLOOP    display
BPROMPT display
BRWS    display
CHF    print
CHKHL    print
CHL    print
CHS    print
CMDDSTART begin
CNTEM    count
COUNTDCS print search
DBNERR    print
DEFLD    search
DELTDBW print search count
DICT     dict
DICTOP    dict
DICTOPL    dict
DSPL     history
DSPLOOP history
EQCHK    print count
FALSE    print search count
FORMCHK print
FREE    free
FRBECUR search
FRBIT    free
FULLDSP history
FULLFREE free
PVCPCHK print count
PVVAL    print count
GET     OFFPRES
GETFORM
GETSYN    search
HCOUN    help
HCOUND    help
HCOUNB    help
HCOUNF    help
HCOUNO    help

CPLM Version 2.0 - F2 - 06 June 1982
HPRINN help
HPRINO help
HPRINS help
HSEAR help
HSEARD help
HSEARE help
HSEARF help
HSEARN help
HSEARO help
HSEARS help
HSET help
HSETD 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
USER help
IFANDNOT could be used
IFPMD search
IFKNOWN used w/OFFPRES
INQ INQ
INSCOMMA all
INSPARM count
IOLOOP count
IOLOOP2 count
KEYFCHK print count
MERGE search
NEWSRCH search
NEXTWORD search
NKNORD count
NODBCHK history
NXTFLD count
NXTWORD count
OFFPRES ZERO RECORDS
OPCHK search
OPBR search
OPERROR search
OPBR2 search
OPBR3 search
OPGO search

CPLM Version 2.0 - F4 - 06 June 1982
Built-in INQUIRE Macros

+1
ACCNO
ANDIF(valuel, 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)
HOLDINDEX(name)
HOLDNAME
POLDTYPE(n)
IF(text)
IF(valuel, operator, value2)
IFITEMS(n)
INBUFF
INDEX(text1, text2)
ITEMSFND
LASTMAC
LASTSAVE
LENGTH(text)
MEND
NONMAC
NOPRINT
NULL
NUMDB
NUMHOLD
ORIF(valuel, 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

CPLM Version 2.0 - P6 - 06 June 1982
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 controlling 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 &DOCD13 (CPLMI). If &PARM(2) is not 'IF', SETDB sets &DBN to the value n in &PARM(2). Since no further operands exist, 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 &DOCD13, 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.

CPLM Version 2.0

06 June 1982
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. BROLL
II. BLOOP
   A. +1
   B. TEMP
   C. PARM
III. BLOOP
   A. +1
   B. TEMP
   C. THEN
   D. PARM
   E. REST
IV. BEX
   I. 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

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*
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*
FREE, 3

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

CPLM Version 2.0 - F16 - 06 June 1982
*PROGRAM RESTART*
SEARCH, LUNG, AND, LIVER

I. SRCH1
   A. IFFLD
II. NEXTWORD
   A. +1
   B. PARM
   C. OPC\:\K
   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. OPC\:\K
IV. OPER
V. OPER2
    A. OPNAME
    B. PREVHOLD
    C. RETAND
       1. CURWORD
VI. NEXTWORD
    A. +1
    B. PARM
    C. OPC\:\K
    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

CPLM Version 2.0 - F18 - 06 June 1982
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 &SSETFIN') 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.
STANDARD ANALYSIS SYSTEM

SMALL

4-RECORDS

NLG

3-RECORDS

SF1

2-RECORDS

 criticize:Inline

criticize:Inline

objection

2-RECORDS

adj

ALLSYN

17-RECORDS

AND

5-RECORDS

CPLM Version 2.0

- F22 -

06 June 1982
STATISTICAL ANALYSIS SYSTEM

11-RECORDS
DICTOP
MACRO DEF E C R U P 1.
GO TO DICTOP1
2-RECORDS
DICTOP1
MACRO DEF E C R U P 2(DICTOP).
MACRO DEF E C R U P 3(DICTOP).
3-RECORDS
DISPL
MACRO DEF E C R U P 4(DISPL).
MACRO DEF E C R U P 5(DISPL).
4-RECORDS
DISPLAY
MACRO DEF E C R U P 6(DISPLAY).
MACRO DEF E C R U P 7(DISPLAY).
5-RECORDS
EOCHR
MACRO DEF E C R U P 8(EOCHR).
MACRO DEF E C R U P 9(EOCHR).
6-RECORDS
FALSE
7-RECORDS
F I N A L
8-RECORDS
FREE
9-RECORDS
FREE1
10-RECORDS
STATISTICAL ANALYSIS SYSTEM

REQUIRED SET NUMBER, OR "IF", FIELDNAME, RELATOR, VALUE

EXAMPLES:
COUNT 4 IF SET 4 LT 12
COUNT 6 IF SET 6 GE 12
COUNT 2 IF SET 2 LT 12 AND SET 2 LE 2
COUNT 7 IF FIELDNAME CONTAINS (MAP, MAP2, ATLAS)
COUNT 11 SPECIAL(15)

FUNCTION: IDENTIFIES AND COUNTS DOCUMENTS WHICH SATISFY A
LOGICAL CONDITION AND/OR PREVIOUS SEARCH CRITERIA.

SET NUMBER - THE SET NUMBER OF A TEMPORARY DATABASE CREATED BY
A PREVIOUS COMMAND.
"IF" - BECOMES 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 - EQUAL
NOT - NOT EQUAL TO
ITSELF MEANS "BEGIN WITH"
CONTAINS VALUE IS FOUND ANYWHERE IN FIELD
"LIKE" MEANS "MATCHES"
SUBSTRING CLIP TO CONTINUE (E g, "BEACH TO WAVE")
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 "AT" RELATOR AND "TO"
COMBINED TO MATCH THE UPPER AND LOWER VALUES IN THE
FIELD. THE "LIKE" OPTION MUST BE IN
SPECIFIED.
BOOLEAN - OPERATOR USED TO COMBINE MULTIPLE QUALIFICATIONS
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.
2-RECORDS

HOTSPD

GAMITE 1
REQUIRED: WORD-LIST

HOTSPF

GAMITE 1
FUNCTION: DISPLAYS WORDS FROM THE INDEX AND THE NUMBER OF TIMES EACH WORD OCCURS IN THE DOCUMENT DATABASE.

HOTSP

GAMITE 1
NOTE: ROWN COUNT IS AUTOMATIC THAT IS, ALL WORDS BEGINNING WITH THE CHARACTERS SPECIFIED IN WORD-LIST WILL BE DISPLAYED.

HOTSP0

GAMITE 1
WORD-LIST - A SINGLE WORD OR MULTIPLE WORDS SEPARATED BY COMMAS, FOR WHICH A POSTING COUNT DISPLAY IS DESIRED.

HOTSPN

GAMITE 1
SYNTAX: 'DISPLAY', WORD-LIST

WORDS IN APOSTROPHE ARE TREATED EXACTLY AS SHOWN.

HOTSPR

GAMITE 1
NON-USER-SUPPLIED VARIABLES.

HELP

HOTSPU

GAMITE 1
THE COMMANDS AVAILABLE ARE:

COUNT

DISPLAY

FILE

HELP

PRINT

SEARCH

SUMMARY

USER

GAMITE 1
DISPLAY DETAIL INFORMATION ON THESE COMMANDS,

CPLM Version 2.0

06 June 1982
STANFORD UNIVERSITY

CPLM Version 2.0

- F35 -

06 June 1982
STATISTICAL ANALYSIS SYSTEM

HSEARCH

GETFIELD - GET FIELD TO BE SEARCHED. A FIELD-GROUP MAY BE DESIGNATED AT INSTALLATION TIME TO REPRESENT SEVERAL FIELDS UNDER THE NAME.

SYN - SEARCH THE USER SPECIFIED FILE(S) FOR THE INDEXED FIELD(S) IN THE FILE(S) LISTED. SEVERAL INDEXED FIELDS MAY BE LISTED IN A SINGLE SYN. THE ORDER OF SEARCH IS THE ORDER OF FIELD LISTING. See HELP FOR MORE DETAILS.

SORT - ONE OR MORE SORTS TO BE PERFORMED. ONE OR MORE SETS OF RECORDS MAY BE SPECIFIED. USE OF THE EXECUTE OPTION IS REQUIRED FOR EACH SET SPECIFIED. SEVERAL SETS MAY BE SPECIFIED TOGETHER AND THE ORDER OF SORTS IS THE ORDER IN WHICH THEY ARE LISTED. See HELP FOR MORE DETAILS.

ALLF - SEARCH ALL FIELDS ASSOCIATED WITH A DATABASE AS FOLLOWS:

FIELD(s) MUST OCCUR IN THE SAME DOCUMENT.

OTHER RELATIONS (IF ANY) MUST OCCUR IN THE SAME COMMAND.

CHECK (PRESS CTRL TO CONTINUE, OR END TO EXIT)

AND - SEARCH THE USER SPECIFIED FIELD(S) IN THE SAME SENTENCE, INCLUDING A RELATIVE PROPORTION INSIDE EACH SENTENCE, IF SENTENCE PROPORTION HAS BEEN IMPLEMENTED IN THE PARTICULAR SYSTEM.

OR - SEARCH THE USER SPECIFIED FIELD(S) IN THE SAME SENTENCE OR ANY SENTENCE CONTIGUOUS TO IT. EACH SENTENCE MUST OCCUR AT LEAST ONCE IN THE SAME DOCUMENT AS THE PRECEDING FIELD.

Priorities are defined by the order in which the commands are received. Commands that are not field searches are executed subsequent to a field search command. See HELP FOR MORE DETAILS.

ALL RELATIONS MAY BE COMBINED WITH "AND" IN THE SAME COMMAND.

CPLM Version 2.0

06 June 1982
STATISTICAL ANALYSIS SYSTEM

19

*ISHTAD

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>FILENAME = ADAPT</td>
</tr>
<tr>
<td></td>
<td>TITLE NUMBER = 1</td>
</tr>
<tr>
<td></td>
<td>IF THIS IS THE FIRST EXECUTION OF</td>
</tr>
<tr>
<td></td>
<td>THE PROGRAM,</td>
</tr>
<tr>
<td></td>
<td>ON THE NEXT EXECUTION OF THIS PROGRAM,</td>
</tr>
<tr>
<td></td>
<td>IF THERE HAS BEEN A PREVIOUS EXECUTION</td>
</tr>
<tr>
<td></td>
<td>OF THE SIMA COMMAND.</td>
</tr>
</tbody>
</table>

J-RECORDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXAMPLE</td>
<td>SIMA:</td>
</tr>
<tr>
<td></td>
<td>SIMA PART IN 1</td>
</tr>
<tr>
<td></td>
<td>SIMA PART 1</td>
</tr>
<tr>
<td></td>
<td>SIMA TITLE SYNONYM DEFINITION IN 1 27</td>
</tr>
</tbody>
</table>

K-RECORDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION</td>
<td>PRINT THE CONTENTS OF THE SPECIFIED FIELD FOR</td>
</tr>
<tr>
<td></td>
<td>TITLE IN THE MOST RECENTLY DEFINED TEMPORARY</td>
</tr>
<tr>
<td></td>
<td>DATASET. ITEMS ARE IDENTIFIED BY THE TITLE VALUE.</td>
</tr>
</tbody>
</table>

S-RECORDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>- THE FIELDNAME ASSOCIATED WITH EACH MAJOR</td>
</tr>
<tr>
<td></td>
<td>SECTION OF THE PRESENTATION COLUMNS.</td>
</tr>
<tr>
<td></td>
<td>MULTIPLE FIELDNAMES ARE ALLOWED. VALID</td>
</tr>
<tr>
<td></td>
<td>FIELDNAMES ARE TITLE SYNONYMS. ITEMS</td>
</tr>
<tr>
<td></td>
<td>DEFINITION, ADAPT, PART, TITLE, STANDARDS, NRS.</td>
</tr>
<tr>
<td></td>
<td>USE THE OUTLINE COMMAND TO PRINT THE</td>
</tr>
<tr>
<td></td>
<td>PRESENTATION OUTLINE AND THE CORRESPONDING</td>
</tr>
<tr>
<td></td>
<td>FIELDNAMES.</td>
</tr>
<tr>
<td>ITEM NUMBER</td>
<td>- THE TITLE NUMBER WHICH IDENTIFIES A</td>
</tr>
<tr>
<td></td>
<td>DOCUMENT WITHIN THE CURRENT ACTIVE</td>
</tr>
<tr>
<td></td>
<td>TEMPORARY DATABASE. MULTIPLE ITEM</td>
</tr>
<tr>
<td></td>
<td>NUMBERS ARE ALLOWED. THE ITEM NUMBERS</td>
</tr>
<tr>
<td></td>
<td>ARE INITIATED BY CALLING THE</td>
</tr>
<tr>
<td></td>
<td>DESIRED TEMPORARY DATABASE.</td>
</tr>
</tbody>
</table>

L-RECORDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXAMPLE</td>
<td>SIMA FIELD-ADAPT (IN ITEM NUMBERS)</td>
</tr>
<tr>
<td></td>
<td>VARS IN APoSTRIPs ARE ENTERED LENTILY AS SHOWN.</td>
</tr>
<tr>
<td></td>
<td>AVOID THE APoSTRIPs. AVOID NOT IN APoSTRIPs ARE</td>
</tr>
<tr>
<td></td>
<td>USER-SUPPLIED VARIABLES.</td>
</tr>
</tbody>
</table>

P-RECORDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>SUBGROUP</td>
</tr>
<tr>
<td></td>
<td>GROUP 1: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 2: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 3: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 4: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 5: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 6: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 7: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
<tr>
<td></td>
<td>GROUP 8: WHICH GROUPING ITEM NUMBER IS GROUPED BY</td>
</tr>
</tbody>
</table>

CPLM Version 2.0  - F38 -  06 June 1982
### 3-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 4-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 5-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 6-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 7-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 8-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 9-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 10-RECORDS

<table>
<thead>
<tr>
<th>RECORD</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
APPENDIX G

NORTHEAST REGIONAL DATA CENTER
NORMAL SYSTEM OPERATING HOURS

<table>
<thead>
<tr>
<th>Time</th>
<th>Service</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:00 a.m.</td>
<td>CICS</td>
<td>Monday - Thursday</td>
</tr>
<tr>
<td>8:00 a.m.</td>
<td>Remote-batch</td>
<td>Monday - 3:45 a.m. Fri.</td>
</tr>
<tr>
<td>12 noon</td>
<td>Card reader</td>
<td>Sat - Thurs 3:30 a.m. Fri.</td>
</tr>
<tr>
<td>4:00 p.m.</td>
<td>Tape mounts</td>
<td>8 a.m. Sat - 4 p.m. Sun.</td>
</tr>
</tbody>
</table>

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 Year'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 1982 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 (e.g. 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).

![Diagram of CPLM Database]

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.

```
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
II. SYNONYMS
III. DEFINITIONS
IV. PHYSICIAN PROCEDURE DESCRIPTION
    < 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 DOCNT 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
```

CPLM Version 2.0 - H4 - 06 June 1982
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 UNCONT 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
```

<table>
<thead>
<tr>
<th>ITEM</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>COLORADO TICK FEVER AND OTHER TICK-BORNE FEVERS.</td>
</tr>
<tr>
<td>76</td>
<td>BRUCELLOSIS</td>
</tr>
<tr>
<td>59</td>
<td>BLOOD CULTURE</td>
</tr>
<tr>
<td>45</td>
<td>WBC</td>
</tr>
<tr>
<td>44</td>
<td>DIFFERENTIAL</td>
</tr>
<tr>
<td>29</td>
<td>EOSINOPHIL COUNT</td>
</tr>
<tr>
<td>11</td>
<td>BRUCELLA CULTURE</td>
</tr>
</tbody>
</table>

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.
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

CPLM Version 2.0 - H6 - 06 June 1982
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.

```
\[
\begin{array}{c}
\text{A} \\
\text{A NOT B} \\
\text{A AND B} \\
\text{B NOT A} \\
\text{A OR B}
\end{array}
\]
```

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:

CPLM Version 2.0
- H7 -
06 June 1982
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 |
| TEMPORARY DATABASE 2 | CONTAINS | 6 POSTINGS FROM CPLM1 |
| TEMPORARY DATABASE 3 | CONTAINS | 15 POSTINGS FROM CPLM1 |
| TEMPORARY DATABASE 4 | CONTAINS | 99 POSTINGS FROM CPLM1 |
| 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:

CPLM Version 2.0 - H8 - 06 June 1982
**COMMAND: HISTORY**

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

<table>
<thead>
<tr>
<th>TEMPORARY DATABASE</th>
<th>ITEMS FROM HELD DATABASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78 CPLM1</td>
</tr>
<tr>
<td>2</td>
<td>6 CPLM1</td>
</tr>
<tr>
<td>3</td>
<td>15 CPLM1</td>
</tr>
<tr>
<td>4</td>
<td>99 CPLM1</td>
</tr>
</tbody>
</table>

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