



# Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY

## Information and Computing Sciences Division

### User's Guide for the ENGNOTE Database System for LBL Engineering Notes

A. Konrad

July 1986

LIBRARY AND DOCUMENTS SECTION  
OCT 17 1988

**For Reference**

**Not to be taken from this room**



PUB-3057  
c.1

## **DISCLAIMER**

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

**User's Guide  
for the ENGNOTE  
Database System  
for LBL Engineering Notes**

**Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098**

**Allan Konrad, X5458  
Office of Computing Resources  
Computing Division**

**Version 1.2  
30 July 1986**

## Contents

I.	Introduction:	1
	Purpose	1
	CMS, SPIRES, the CMS/SPIRES interface	1
	General SPIRES Information	3
II.	Getting started	4
	Logging ON	4
	Logging OFF	6
III.	Organization of the database	7
IV.	Using the ENGNOTE subfile	8
	Description of elements in the ENGNOTE subfile	9
	Displaying records	11
	Searching in SPIRES; Searching the ENGNOTE Subfile	12
	Updating records	14
	Adding new records	16
	Removing records	18
	Changing the key of a record	18
V.	The LPR and LPRCC commands	29
VI.	Making Reports	20
APPENDIX A	Terminal Settings for VT100 for use on UCBCMSA Series/1	
APPENDIX B	Terminal control for IBM PC using YTERM	
APPENDIX C	FORMAT \$PROMPT subcommands	
APPENDIX D	Looking at your CMS files	
APPENDIX E	Documentation	
APPENDIX F	Human Help	
APPENDIX G	Using Xedit	

## I. Introduction:

- .1 Purpose
- .2 CMS, SPIRES, the CMS/SPIRES interface
- .3 General information about SPIRES.

### I.1 Purpose

The ENGNOTE database was implemented at the request of the Mechanical Engineering Department for management of LBL engineering notes. It was converted from an LBL BKY program called ARDIE which was supported on the CDC 7600 in batch mode. The user interface was printed listings. In contrast, the current ENGNOTE database is interactive, though printed listings are easily produceable. ENGNOTE is implemented using the SPIRES database management system, a product of Stanford University, support by LBL on the U.C. Berkeley campus IBM 3090 computer. There is no practical limit to the number of records which may be contained in the database.

### I.2 CMS, SPIRES, the CMS/SPIRES interface

The Stanford Public Information Retrieval System (SPIRES) is a product of Leland Stanford Junior University in Palo Alto, CA. The SPIRES database management system at LBL runs on the UC Berkeley Campus IBM 3090 under the VM/CMS operating system. VM SPIRES consists of three components:

- SPIRES itself (database management system)
- CMS (the operating system that manages the computer)
- SPIRES/CMS interface (maps SPIRES activity onto the CMS environment)

Figure 1 indicates how these components relate to one another. Normally, SPIRES users are not and need not be concerned with the subsystems between themselves and SPIRES. The diagram is provided only to demonstrate context.

Most of the icons are self-explanatory. The purpose of the SERIES/1 is to make the user's ASCII terminal appear as an IBM 3270 terminal to the IBM 3090, and to make the IBM 3090 appear to communicate in ASCII to the user. IBM PC's use special software but provide additional useful capabilities to a standard ASCII terminal.

Section II will describe the commands to move along the path from terminal through the gateways into SPIRES. This generally requires less than 10 seconds and becomes routine.

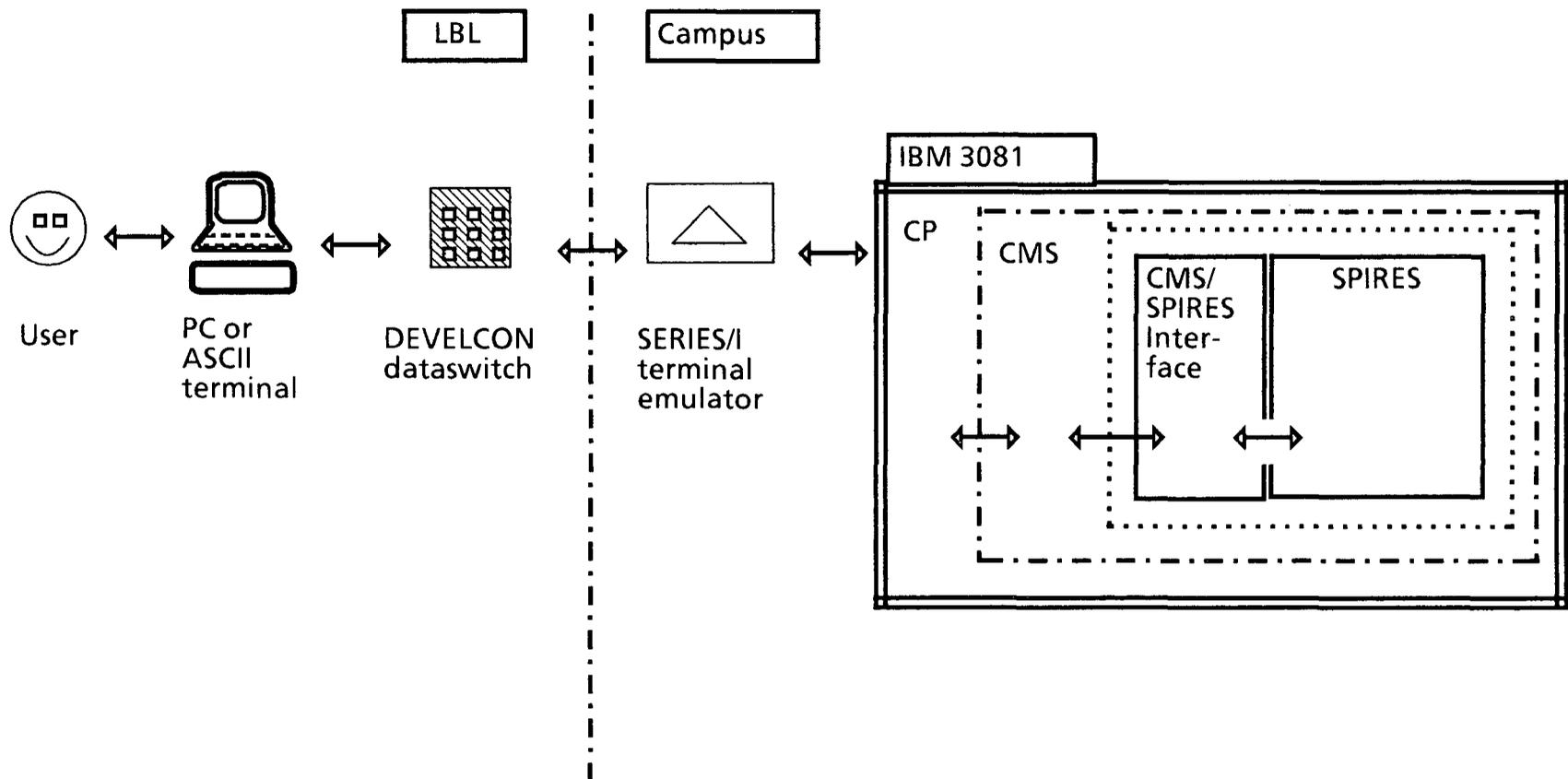


Figure 1. Pathway between user and SPIRES

### I.3 General information about SPIRES.

Information for each engineering note is stored directly into a SPIRES *record*. Each record in the database is comprised of a collection of elements as described below, e.g., author, title, engineering note number. Each record in a SPIRES database has a unique identifier often called *key*. In the ENGINEERING subfile, the key is called ID. The name of the database system, i.e., the file definition, format definitions, and all the physical files, are collectively called ENGNOTE. The "fully qualified name" is ABS20:ENGNOTE. The *logical* name for the SPIRES subfile (database) is ENGINEERING. Further explanation of keys is found in Section IV.1.

For each record, a particular element may be required or optional, singly or multiply occurring, have controlled allowable values, be limited to a particular type of value, and be indexed for ease in searching, etc.

If you are not in SPIRES, the CMS prompt is: **R;**

If you have EXITed SPIRES and you wish to re-enter, enter the command:

**SPIRES**

When logging on, the PROFILE EXEC in CMS and the ENTRY COMMANDS record in SPIRES automatically execute to enter into the ENGINEERING subfile.

The normal SPIRES prompts are as follows:

- ? for UPPER case only
- > for UPPER and lower case
- +? UPPER case in Global For
- +> UPPER and lower case in Global For

All the modifications made to the database during the day (adds, updates, and removes) take effect immediately and are reflected the very next time the record is displayed unless otherwise requested. Indexes are updated immediately in this application.

For most SPIRES commands, only the first three characters need be entered. For example, the FIND command requires only FIN <index> <value>. In this document, commands will be fully spelled out, with the first three letters capitalized; e.g., FIND, SHOW ACTIVE, indicating that only the capitalized characters need be entered.

The "active file" is also a SPIRES concept, and usually refers to the CMS file ACTIVE FILE A. Any CMS filename can be used as the SPIRES active file and is specifiable by the user with the

SET ACTIVE <fn> <ft> <fm> command.

## II. Getting Started.

- .1 Logging ON
- .2 Logging OFF

### II.1 Logging ON using an IBM PC and YTERM. (If using an ASCII terminal, begin with step 6.

1. Turn PC, disk drive, and printer on and wait for the PC to complete booting.
2. Enter the command: **CD \YTERM**
3. Enter the command: **X 7**
4. Enter the command: **T 9600 K**  
System responds: "PLEASE ENTER THE NAME OF THE KEYBOARD TABLE"
5. Enter: **UCBCAD**  
System responds by clearing the screen and displaying DISCONN in the lower left corner
6. Make sure the blue TSB box displays either a green or red light.
7. If red light is illuminated, press the blue button and wait for green light.
8. When green light is illuminated, enter carriage return [CR].

The following dialogue should occur. The system response is in **bold**. The user response in modern font.

9. Request: **ccdb** [CR].
10. System will respond with a bell, and cursor and will jump to next line. Enter carriage return [CR].
11. YALE ASCII TERMINAL COMMUNICATIONS SYSTEM V2.1  
enter terminal type: **YTERM** [CR]. Note: enter **ADM3A** or **VT100** if not using a PC.
12. System will respond with a pseudo-three-dimensional display CFO over the letters VM. Enter another [CR].
13. The screen will clear. Enter:  
**L ABS20** [name of your virtual machine] [CR].
14. ENTER PASSWORD:  
enter your password. It is not a good idea to write your password in this set of instructions. If you write it down, do so elsewhere.

Note: If your previous session ended "abnormally", e.g., by simply pushing the blue button on the TSB box to obtain a red light, you will have to enter, at this point in the logon procedure, the command: IPL CMS and then a [CR]. This should always be done when a paragraph beginning with the word "**RECONNECTED . . .**" appears.

15. Enter yet another [CR]. This causes your PROFILE EXEC to execute. The system will then perform the following tasks automatically:

```
call SPIRES  
SElect ENGINEERING  
SET LENGTH 79  
SET UPLOW (for upper and lower case)
```

Note that no special output format will be set automatically. The display may be left as the SPIRES default format or reset at any time as indicated below.

Note: Henceforth in this document, commands are assumed to be followed by a [CR], except for ESC-sequences and CNTL-sequences.

## II.2 To LOGOFF

If you have one of the SPIRES prompts (-?, +?, ->, +>), enter: EXIT

The system will respond: **Leaving SPIRES.**

Enter: LOG

### III. Organization of the database.

The organization of the ENGNOTE database system is single *goal-record* subfile and five *index-records with subfile access* implemented in one SPIRES logical file, with the normal CMS physical files. Subfile access is provided to the indexes solely for the purpose of making use of the pre-sorted pointers of goal records for report generation. No elements other than key and pointer are presently implemented in subfile-accessed goal-rec record types. No virtual elements or phantom structure pointing to other SPIRES files are used in this application.

#### IV. Using the ENGNOTE Subfile

- 1 Description of elements in the ENGNOTE subfile
- 2 Displaying records
- 3 Searching in SPIRES; Searching the ENGNOTE subfile
- 4 Updating records
- 5 Adding new records
- 6 Removing records
- 7 Changing the key of a record

The purpose of the ENGNOTE subfile is to maintain current descriptive information about each engineering note. Each engineering note is represented in the database by a collection of elements as described below, e.g., author, title, date, etc. Each record has a unique identifier often called *key* or in the ENGNOTE database, simply *ID*. This 4-digit code is assigned by whoever enters the new ENGNOTE record into the database. In SPIRES, record keys are unique. Therefore any attempt to add a record with a key already assigned to another record will be rejected by SPIRES.

For each record, a particular element may be required or optional, singly or multiply occurring, have controlled allowable values, be limited to a particular type of value, or be indexed for ease in searching, etc. The element listing below describes the characteristics of each element.

#### IV.1 Description of elements in the ENGNOTE subfile

<u>Element Name</u>	<u>Required/Opt</u>	<u>Length</u>	<u>Occurrences</u>	<u>Data Type</u>	<u>Indexed</u>
ID (key of the record) <sup>4</sup>	REQUIRED	4 bytes	single	character	Indexed
Title (T)	Optional	Variable	Single	Character	Indexed
ENGNOTE (EN) <sup>1</sup>	Optional	Fixed 5	Single	Character	Indexed
CATCODE (CC)	Optional	Variable	Multiple	Character	Indexed
Author (PA) <sup>5</sup>	Optional	Variable	Multiple	Name	Indexed
Corpauth (CA) <sup>2</sup>	Optional	Variable	Multiple	Character	Indexed
Type <sup>3</sup>	Optional	Variable	Single	Character	Indexed
Date <sup>7</sup>	Optional	(fixed 4)	Single	Date	
DATE.Added <sup>6</sup>	Required (automatic)	Fixed 4	Single	Date	
DATE.UPDATED <sup>6</sup>	Required (automatic)	Fixed 4	Single	Date	
NOTE.STR	Optional	Variable	Multiple	Structure	
NOTE	Optional	Variable	Single	Character	
NOTE-DATE <sup>6</sup>	Automatic	Fixed	Single	Date	

- <sup>1</sup> Stores the engineering note number. It is distinct from the *key* of a record in the database. It is singly occurring, must be exactly 5 character, the first must be a capital M, the following 4 characters must be integer characters.
- <sup>2</sup> For corporate authors such as companies, universities or group names.
- <sup>3</sup> Stores media type and size, e.g., "6 pages".
- <sup>4</sup> Prepend zeros will be added if less than 4 characters are entered. Characters may be numbers or letters.
- <sup>5</sup> Enter a Name in any form, upper or lower case. SPIRES determines the surname.
- <sup>6</sup> Enter Carriage Return for automatically-generated date elements.
- <sup>7</sup> Any form of date is acceptable.

Graphically, the hierarchical nature of a typical ENGNOTE record appears:

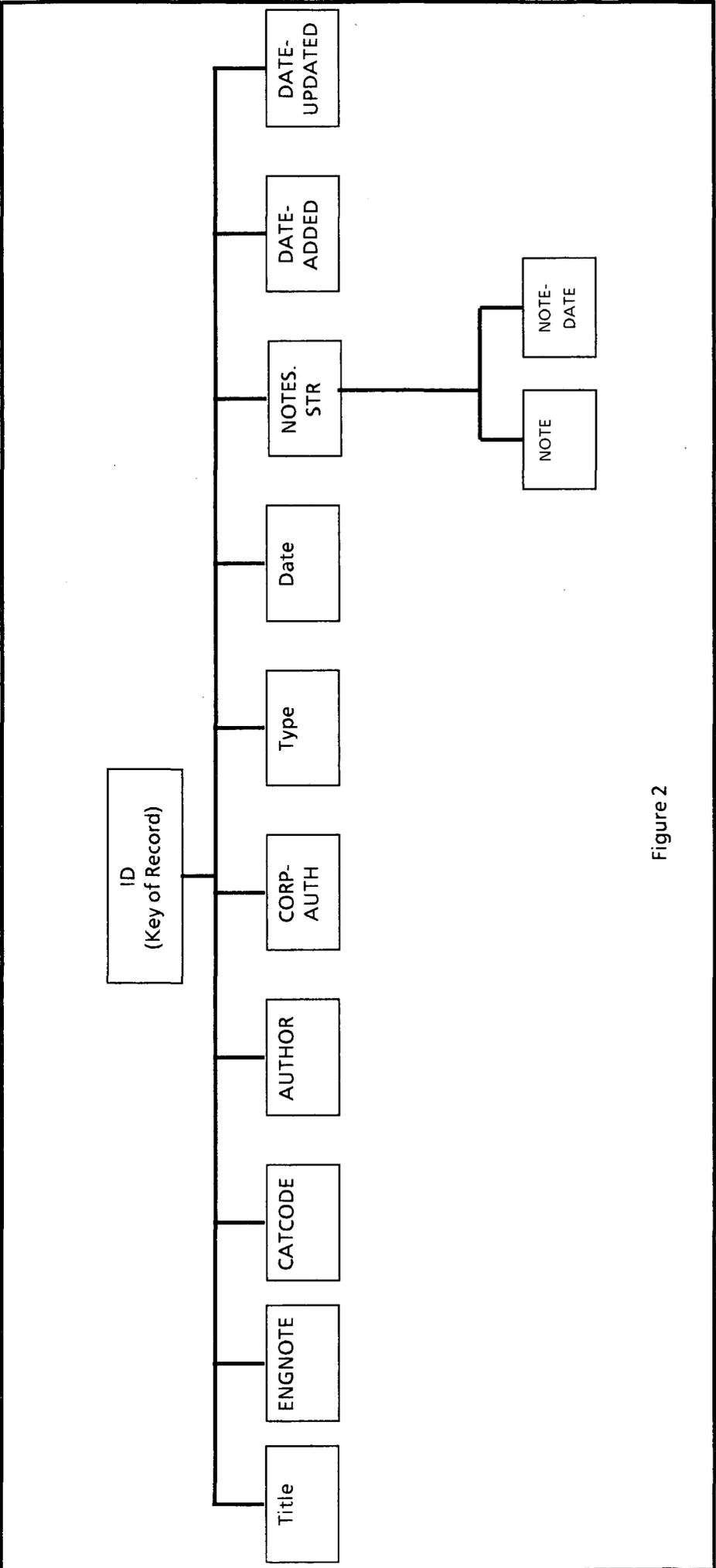


Figure 2

## IV.2 Displaying Records.

To use the ENGNOTE subfile, you must SElect it with the command:

**SElect ENGNOTE**

Whenever you log on or issue the command SPIRES the ENGNOTE subfile will automatically be SElected, it need not be done manually.

If you know the ID for a record that you wish to see, you may use the DISplay command to view it directly:

**DISplay <ID>**

If you do not know the ID for a record which you wish to see, then you must search for it based upon some criteria you do know. Use the FINd command to search indexes for records in this way (Section IV.3) Then, to look at the records which are the result of a FINd command, enter the command TYPE. All of the records in the search result will then be displayed. To display a listing of which indexes are available, enter the command:

**SHOw INDExes**

If you search on a non-indexed element (See Sect IV.3) i.e., using Global For, then use the DISplay <all/first/last/n/ext> command.

Records may be DISplayed in any of three formats:

default SPIRES format  
format ENGNOTEID  
format ENGNOTE

The default SPIRES format is for normal SPIRES usage and displays all information in the record, including notes.

The ENGNOTEID format is for use in printed listings, and is similar to the old ARDIE format. For example:

```
FIND TITLE HELIUM
SET FORMAT ENGNOTEID
TYPE
```

To put the result into a physical file rather than displaying it on the screen, do:

**IN ACT CLN CLR TYPE**

This is useful for printing the results.

Further displaying facilities are described in section VI, Making Reports.

### IV.3 Searching in SPIRES; Searching the ENGNOTE Subfile

You may search for ENGNOTE records based on any element or combination of elements. However, some elements are used as the basis of searching much more often than others. Those elements are *indexed* in the same way as selected keywords are indexed in the back of a book. Rather than searching sequentially through a book to find a particular topic, you find the topic in the index. Associated with its entry is an *address*, usually a page number. SPIRES indices work in much the same way. Indexed elements are listed along with their "addresses". However, you never have to worry about the addresses. You simply enter a FIND command, and SPIRES fetches the addresses and then allows you to display, re-sequence, or update the records as desired.

To see a list of the elements in the subfile, enter the command **SHOW ELEMENTS**.

To see a list of indexes, enter the command **SHOW INDEXES**.

As indicated in the element list above, those elements which are indexed are:

TITLE  
ENGNOTE  
CATCODE  
AUTHOR  
CORPAUTH  
TYPE

The key of a record may also be searched as if it were an indexed element (with the FIND command), which, as implemented in SPIRES, it is, since goal records are stored in order by key.

To search for courses based on any of these elements, use the FIND command, as follows:

**FIND <index name> <relational operator> <value>**

For example, to find all the authors named Jones, enter:

**FIND AUTHOR = JONES**

Then use the TYPE command to see the result (Section IV.2).

If you do not include the relational operator in your search, SPIRES assumes an "equals" operator:

**FIND AUTHOR JONES**

Less commonly-used elements are not indexed; for example, job title. To search for all the engineering notes written after January 21, 1985, enter

**FOR SUBFile WHERE <element> = <value>**

For example,

**FOR SUBF WHERE DATE = JAN 21, 1985**

Then use the DISPLAY command to see the result (Section IV.2).

All indexes make use of the truncation character, the pound-sign (#). For example,

**FIND AUTHOR JON#**

will find engineering notes authored by Jones, Jonstone, Jonybrook, etc.

A complete description of all the searching capabilities in SPIRES is described in the document Searching and Updating listed in Appendix E.

#### IV.4 Updating Records.

There are two methods for updating records in SPIRES: one, by TRANSferring & UPDating, the other by using format \$PROMPT. A complete description of updating records in SPIRES is described in the document Searching and Updating listed in Appendix E. This brief summary provides an overview.

##### Using TRANSfer and UPDate

To update an ENGNOTE record, enter the following commands:

1. Use the FINd and TYPE commands to determine the key of the record you wish to modify.
2. **TRANSfer <ID> CLR**
3. **X ACTIVE FILE** (this enables you to use the editor to modify the file)
4. Edit the record. When all the changes are made, enter the command **FILE** on the command line. This will return you to SPIRES.
5. **UPDate**
6. **DISplay <entry number>** to verify that the record is correct.

When adding or modifying data elements, remember that the format is:

**<DATA ELEMENT> = <value> ;**

Don't forget the semicolon!

Further, if adding a note, insert the following lines:

**NOTE.STR;**  
**NOTE = <text of note, no limit to length> ;**

Don't forget the semicolons! Or, just use **FORMAT \$PROMPT**.

SPIRES validates the data when you update the record. If there are any illegal values, you will receive an error message when you enter the **UPDate** command. If this occurs, return to step 3 and re-edit the record.

Finally, it is always a good idea to retain your source documents after you complete any updating. In four years of running SPIRES at LBL, no data has ever been lost, but users have forgotten why they changed some records.

Note that the current version of the record will always be displayed by SPIRES.

**NOTE:** The key of a record (ID) **cannot** be modified by editing its value and the issuing an **UPDate** command. To change the key of a record, please see Section IV.7.

**NOTE:** Single and double quotation marks are treated specially by SPIRES. *Except* in format \$PROMPT, all quote marks must be doubled, *and* the entire value (between the =-sign and the semicolon) must be in quote marks of the same kind.

##### Using format \$PROMPT

To update an existing ENGNOTE record using \$PROMPT, enter the commands:

```
SEL ENGNOTE (if not already SElected)
SET FORMAT $PROMPT
MERGE <ID>
```

You will be prompted element by element. If an element has an existing value, it will be displayed, and you will be given an opportunity to change it. You are also given the opportunity to add new occurrences of elements and structures. Please see Appendix C for \$PROMPT Subcommands.

#### IV.5 Adding a new record.

A complete description of adding records in SPIRES is described in the document Searching and Updating listed in Appendix E. Below, a very brief summary and sample session provides an adequate overview.

##### **SET FORMAT \$PROMPT**

##### **ADD**

You will be prompted for the value of each element. If an *optional* element should be left blank, enter a carriage return [CR]. Also note that you will be prompted twice for each multiply occurring element. Just enter a [CR] to proceed to the next element. Please see Appendix C for the subcommands used in the \$PROMPT format.

To add several records, simply reissue the ADD command after each previous record is ADDED and DISplayed.

After ADDing a new record, always DISplay <key> to examine it for correctness.

Here's a sample session showing how to add a record (system responses in **bold**):

```
sel engnote                (not necessary if ENGNOTE is already SElected)
set format $prompt
-?
ADD
:ID
115 [CR]  (SPIRES will add a zero: 0115)
:TITLE
Fixing Widgets [CR]
:ENGNOTE
M8123 [CR]
:CATCODE
A1556 [CR]
:CATCODE
[CR]
:AUTHOR
Amanda B. Recondwith [CR]
:AUTHOR
[CR]
:CORPAUTH
Snerdly Physics Group [CR]
:CORPAUTH
[CR]
:TYPE
12 PAGES [CR]
:DATE
12 February 1985 [CR]
:DATE.ADDED
[CR]
:DATE.UPDATED
[CR]
Struc: NOTE.STR(1)
:  NOTE
This is a test note in a test record. [CR]
:  NOTE-DATE
[CR]
```

Struc: NOTE.STR(2)  
: NOTE

[CR]

:DATE.ADDED

[CR]

:DATE.UPDATED

[CR]

-?

dis 115 (or: /dis \$key)

:ID 0115

:Title = Fixing Widgets

:ENGNOTE = M8123

:ENGNOTE = A1556

:AUTHOR = Amanda B. Recondwith

:CORPAUTH = Snerdly Physics Group

:TYPE = 12 Pages

:DATE = 12 February 1985

Struc: NOTE.STR(1)

: NOTE = This is a test note in a test record.

: NOTE-DATE = Tues. May 7, 1985

:DATE.ADDED = Tues. May 7, 1985

:DATE.UPDATED = Tues. May 7, 1985

#### IV.6 Removing Records.

A complete description of REMoving records in SPIRES is described in the document Searching and Updating listed in Appendix E. Below, a very brief summary and sample session provides an overview.

To REMove a record from the ENGNOTE subfile, enter the command:

**REM <ID>**

For example, to REMove record 115, enter:

**REM 115**

#### IV.7 Changing the Key of a Record.

The key of a record (ID) **cannot** be modified by editing its value and then issuing an **UPDate** command. To change the key of a record, enter the following commands:

1. **TRAn**sfer the old ENGNOTE record
2. Edit the ID
3. **ADD**
4. **REMO**ve <key of the old record> (as described in Sect. IV.6 above)

Please ask for human help if you encounter any difficulties (Appendix F).

## V.1 Printing; the LPR and LPRCC commands

Staff may occasionally wish to print files other than standard reports. These may include a file created using the Xedit editor, or created by SPIRES as the result of a FIND or DISPLAY command. (SPIRES usually places search result displays and other output in the CMS file named ACTIVE FILE A or on the CRT or both.) Therefore, it will often be useful to be able to print files directly. There are two EXECs that will send files to the Taleris laser printers on the first floor of Bldg. 50B in the Central Computing Facility machine room area. These EXECs are:

### LPR and LPRCC

The syntax of these commands is:

**LPR** <filename> <filetype> <filemode>

**LPRCC** <filename> <filetype> <filemode>

For example, to print the CMS file, ACTIVE FILE A, enter the command:

**LPR ACTIVE FILE A**

The distinction between the two is that **LPRCC** interprets any characters in the first column of the file (at the left margin) as carriage control (hence the CC; LPR is an acronym for line printer). Generally, users will not insert carriage control characters in a file, and so **LPR** is the appropriate command to use. However, the SPIRES facility **DEFINE TABLE** and **FORMAT \$REPORT** automatically reserve column 1 for carriage control characters, with data beginning in column 2. For files generated by these utilities, **LPRCC** should be used.

A file probably includes carriage control if most of the text begins in column two and column one contains characters such as: 1, 0, and +. For example, it may look something like:

```
1
MARY HAD A LITTLE LAMB
ITS FLEECE WAS WHITE AS SNOW
+           WHITE AS SNOW
0
AND EVERY WHERE THAT MARY WENT
THE LAMB WAS SURE TO GO.
```

A LABEL exec is available to print adhesive labels for mailing and other purposes on a specially configured printer in the central computing facility.

## VI.1 Making Reports

Three general forms of output are provided. First is the "default SPIRES format" where elements are listed, line by line, in the format

<Element name> = <value> ;

For example:

```
ID = 2129;
TITLE = REPORT ON VISIT TO FERMI NATIONAL ACCELERATOR LAB.;
AUTHOR = AVERY, R;
ENGNOTE = M5939;
DATE = JUNE, 1982;
TYPE = 5 PAGES;
```

Additionally, two customized output formats are provided. The first is for displaying and printing records according to their SPIRES ID. This format may be used with normal SPIRES searching commands, e.g.,:

```
-? FIND AUTHOR AVERY
-result 21 records
-? TYPE
```

```
2129 REPORT ON VISIT TO FERMI NATIONAL ACCELERATOR LAB. AVERY, R
Y, R M5939 JUNE, 1982 5 PAGES
```

etc..

The SPIRES Searching Primer may be consulted to tailor search commands and results as described above under Section IV.3, Searching.

Finally, a similar format, ENGNOTE, is provided which places either the author, title, corporate author or ENGNOTE number in the left-hand margin. The command language to produce output in this manner has been cast into the form of single commands which request a range interactively, then produce the report, as follows:

```
-? title
Enter beginning of range:
helium
Enter end of range:
hf
*Your output is in the file: TITLE LISTING A
```

The file TITLE LISTING A may then be displayed using FLIST and ESC-2, or by printing it using the LPR command, described above.

The following commands are provided for the producing reports:

title	for producing ARDIE-like reports by title word
author	for producing ARDIE-like reports by personal author
corpaut	for producing ARDIE-like reports by corporate author
engnote	for producing ARDIE-like reports by engineering note number.
catcode	for producing ARDIE-like reports by category code.

## APPENDIX A

### TERMINAL Settings for VT100 for use on UCBCMSA Series/1

SETUP-B: 0101 1011 0000 1100

No scroll  
Autorepeat  
dark background  
block cursor

Bell on  
key click off  
ANSI  
XON

wrap off  
newline off  
interlace off

parity odd  
parity off  
7 bits  
60 hz

## APPENDIX B

### TERMINAL CONTROL

The SERIES/1 terminal controller commands for the IBM PC are summarized in the document "Using the YTERM Package at UC Berkeley", available from the Computing Services Library on the second floor of Evans Hall on Campus.

Occasionally, the system will not accept characters typed on the keyboard, but rather sound the "bell". To clear this keyboard lock, depress the CONTROL key and, while depressed, enter the letter sequence: RTXQV. This is notated:

#### CNTL-RTXQV

When the system is displaying output on the CRT screen, it will stop after 22 or 23 lines, depending on the kind of terminal. The message **MORE** will be displayed at the lower right. At this point, one has four options:

1. Do nothing. After 50 seconds, the bell will sound. After an additional 10 seconds, the system will clear the screen and display the next page.
2. Enter (large + -key by keypad). This causes the next 23 lines to be displayed immediately.
3. Enter a [CR]. This causes the message in the lower right portion of the screen to change from **MORE** to **HOLDING**. The timer holds, and the screen will not change. Another [CR] causes the message in the lower right to return to **MORE** and the timer is reset.
4. Enter **HT** [CR], then (large + -key by keypad). The **HT** halts typing, preventing the rest of the lines from being displayed. The (large + -key by keypad) then clears the screen .

Several helpful CMS terminal commands are available:

The (@) acts as a CHARDEL (character delete) character.

The (ø) acts as a LINEDEL character (line delete)

The (#) and the (") have been disabled as CMS control characters since they conflict with often-used SPIRES characters.

Series/1 - IBM PC/YTERM control characters (complete list found in 'Using the YTERM Package at UC Berkeley')

<b>CNTL-N</b>	go to next line
keypad left-arrow key	move cursor to the left
keypad right-arrow key	move cursor to the right
keypad up-arrow key	move cursor up
keypad down-arrow key	move cursor down
<b>CNTL-D</b>	deletes a character
<b>CNTL-E</b>	deletes a line
<b>INSERT-key</b>	enter or leave <i>character</i> insert mode

These sequences work in the editor as well as outside the editor.

### Program Function (PF) keys

In some utilities, such as FLIST and Xedit, PF keys are assigned specific functions. When using your IBM PC as a terminal, the PF keys are on the far left-hand side, marked F1 through F10. F11 and F12 are sometimes needed and available by pressing the **NUMLOCK** key for PF11 and **SCROLL LOCK** for PF12. PF11 deletes to the end of line in the FLIST facility. PF3 usually means "quit". PF1 usually calls a CMS help screen. Often a menu of valid PF keys will be displayed in utilities where they are recognized.

At login, your PF5 key is setup to send messages to a disconnected virtual machine at SLACVM called QSPIRES that enable you to have limited use of the HEP database. To use it, enter PF5 and a SPIRES command, such as FIND or DISPLAY. Global For commands are not allowed.

## APPENDIX C

### SPIRES FORMAT \$PROMPT Subcommands

The following commands are recognized by SPIRES when adding new records (or modifying existing records) using SET FORMAT \$PROMPT (formerly SET INPUT FORMAT):

[CR] (carriage return)	Continue to next prompt
//	Puts in a null-length value if legal, otherwise you are reprompted for a legal value.
/N	Skip to the next element of the current structure for input
/S	Skip to the next structure for input (first element of next structure)
/ <b>&lt;value&gt;</b>	Retains leading blanks (blanks in front of the value)
<b>&lt;value&gt; //</b>	Continue value on next line (for long values, e.g., paragraphs)
/E	End input for the current structure, and retain input thus far
/X	Abort input, and do not retain any input

Example of //:                   to enter a null value in a structure without exiting the structure, for example in the TELEPHONE structure:

**STRUCTURE TELEPHONE**

**U.S.AREA.CODE: //**  
**JOB.TITLE President**  
**(other elements)**

This prevents the other elements in the TELEPHONE structure from being skipped merely because there was no value entered for U.S.AREA.CODE.

The full set of subcommands can be found in the SPIRES manual Searching and Updating.

## APPENDIX D

Looking at your CMS files.

The CMS FLIST facility provides a listing of your permanent files and several capabilities to browse, edit, copy, rename, and delete them. To use the FLIST facility, enter the command **FLIST** and your files will be displayed, with the cursor at the top of the list. You may move the cursor up and down to select any file. You may use the **PF** commands on the menu at the bottom to perform various operations, e.g., **PF4** or an **X** will invoke the editor on the selected file, an **PF2** will allow you to browse the file, and **PF8** will allow you to see the next screenful of files on your list if you have more files than can be listed on one screen, and **PF3** will exit FLIST. All the terminal control keys work in FLIST.

There are other file listing facilities besides FLIST. FLIST currently provides the most functionality. For assistance with FLIST, please see Appendix F for human help.

## **APPENDIX E**

### **Documentation**

A complete set of SPIRES documentation is available by issuing the command DOCSPI and following instructions. The most important to use initially are:

1. A Guide to Searching -- A SPIRES Primer.
2. Searching and Updating.
3. Sequential Record Processing: Global FOR Reference Manual.
4. SPIRES Keyterm Index -- An index of all SPIRES terms.

A complete set of CMS documentation is available from the Computer Center library. The following are most likely to be of interest to users of the AWARDS database system.:

1. System Product Editor User's Guide (SC24-5220-1)
2. System Product Editor Command and Macro Reference (SC24-5221-1)

The Computing Services Library on the 2nd floor of Evans Hall on campus can provide documentation about YTERM, including "Using YTERM at U.C. Berkeley".

## **APPENDIX F**

### **Human Help**

For assistance, call:

Allan Konrad

x 5458

## APPENDIX G

### Using Xedit

The following describes use of Xedit with an ADM-3A terminal. For other terminals, please see Appendix B.

(**Note:** If you are using the Xedit editor and SPIRES, be aware that it is helpful to be in the same case mode in the editor as in SPIRES. That is, it is possible to be in SPIRES in upper-and-lower case, while in Xedit in upper only, or vice-versa. The default for the TRAINING system is to be in upper and lower case both in the editor and in SPIRES. If you have problems with case, call for human help (Appendix F).)

Files in the VM/CMS system have three-part names:

filename filetype filemode

usually abbreviated

fn ft fm

The filemode is generally assumed to be A, referring to you "A-disk", 191. This 191 A disk is your private disk.

To edit a file, issue the command

X fn ft

For example, to edit the CMS file ACTIVE FILE A, enter

X ACTIVE FILE A

The document will then appear ready to edit. Case is not significant on this command. You could also enter:

x active file a

If the file ACTIVE FILE did not exist on your A disk, the editor would create a new empty file, with only a top-of-file and a bottom-of-file marker.

Once in the editor, you can:

Use the **DELETE**, **INSERT** and **CNTL-E** keys (see Appendix B)

Use the "cursor" keys to move the cursor around on the screen. On an IBM PC terminal, depress the "arrow keys" on the numerical keypad.

Use the prefix field on the left side of the screen (the five columns of equal signs) to copy, delete or move whole lines or groups of lines.

To save your editing, enter the command **FILE** on the command line. To discard your changes and return the file to its original state, enter the command **QUIT** or **QQ**.

## Often-used Prefix-field Commands.

### D (delete)

To delete one line, place a **d** anywhere in the prefix field to the left of the line you wish to delete. Then hit [CR]. E.g.,

```
==== This is line one
==d== This is line two
==== This is line three
```

results in:

```
==== This is line one
==== This is line three
```

To delete a known number of contiguous lines, enter **d** and the number of lines to be deleted.

```
==== This is line one
==d2= This is line two
==== This is line three
==== This is line four
```

results in:

```
==== This is line one
==== This is line four
```

To delete an *unknown* number of contiguous lines, that is, a "block" of lines enter **dd** on the first line to be deleted and on the last line to be deleted. E.g.,

```
==== This is line one
==dd= This is line two
==== This is line three
dd=== This is line four
==== This is line five
```

results in:

```
==== This is line one
==== This is line five
```

## I (insert)

To insert a new blank line that can be edited, place an **i** in the prefix field on the line which you want the new line to follow. E.g,

```
==== This is line one
==i== This is line two
==== This is line three
==== This is line four
```

results in:

```
==== This is line one
==== This is line two
=====
==== This is line three
==== This is line four
```

The new blank line can now be edited by moving the cursor to anywhere to the right of the prefix field and the first blank column following it.

To insert a specified number of new blank lines that can be edited, place an **i** and the number of blank lines needed in the prefix field on the line which you want the new line to follow. E.g,

```
==== This is line one
==i3= This is line two
==== This is line three
==== This is line four
```

results in:

```
==== This is line one
==== This is line two
=====
=====
=====
==== This is line three
==== This is line four
```

It is also possible to insert lines by entering the command **i** on the command line at the bottom of the screen. This will clear the screen below the column-counter line. You can then enter text and use **CNTL-N** to go to the next line. When you hit a **[CR]**, your text will be shifted up above the column-counter line and the lower part of the screen will be available for more input. Two consecutive **[CR]**'s will return you to normal edit mode.

## C (copy)

To copy one line, place a **c** anywhere in the prefix field to the left of the line you wish to copy and a **p** on the line before which the newly created line should be placed. E.g,

```
==== This is line one
==c== This is line two
====p This is line three
```

results in:

```
==== This is line one
==== This is line two
==== This is line two
==== This is line three
```

the **p** stands for *prior* and instructs the system to put the new copy of the line prior to the line with the **p**. You can use the **f** instead, which means *following*:

```
==== This is line one
==c== This is line two
==== This is line three
===f= This is line four
```

results in:

```
==== This is line one
==== This is line two
==== This is line three
==== This is line four
==== This is line two
```

To copy a known number of contiguous lines, enter **c** and the number of lines to be copied on the first line to be copied, and an **f** or a **p** to mark where the copied lines should be placed:

```
==== This is line one
==c2= This is line two
==== This is line three
===f= This is line four
```

results in:

```
==== This is line one
==== This is line two
==== This is line three
==== This is line four
==== This is line two
==== This is line three
```

To copy a *unknown* number of contiguous lines, that is, a "block" of lines, enter **cc** on the first line to be copied and on the last line to be copied, and an **f** or a **p** to mark where the copies should be placed:

```
==p== This is line one
==cc= This is line two
===== This is line three
cc=== This is line four
===== This is line five
```

results in:

```
===== This is line two
===== This is line three
===== This is line four
===== This is line one
===== This is line two
===== This is line three
===== This is line four
===== This is line five
```

**M** (move)

the move command, **m**, works similarly to copy:

```
===== This is line one
==m== This is line two
====f This is line three
```

results in:

```
===== This is line one
===== This is line three
===== This is line two
```

and,

```
=p=== This is line one
==mm= This is line two
===== This is line three
===mm This is line four
```

results in:

```
===== This is line two
===== This is line three
===== This is line four
===== This is line one
```

Most terminals can only display about 22 lines of text. Therefore, if the file you are editing is longer than 22 lines, not all of them can be displayed simultaneously.

Think of your file as if it were a very tall building. The building is a strange building however, because its floors are numbered from top to bottom rather than from bottom to top! So the first floor is at the top of the building.

Our building has a rather unique elevator. Unquestionably the oddest thing of all is that the elevator doesn't move, the building does! The elevator is fixed, but the building moves up and down, into and out of the ground.

But that's not all! First, its doors are always open, so you can always see out as the building moves up and down in front of you. Furthermore, your elevator is 21 stories high! Stranger yet is that half-way up this tall elevator is a platform on which you stand. Thus, you can see the floor that is level with yourself, the 10 floors lower, and the 10 floors higher.

This peculiar building is like your file and your terminal is like its elevator which provides you with a view of some portion of the building. Imagine standing in the fixed elevator as the building moves up and down in front of you. This is exactly the phenomenon you experience using the editor.

When you first enter the editor, it automatically gives you a view of the top 10 lines of your file. This is like standing in your elevator at the top of the building, with a view of the 10 floors beneath you and 10 stories of thin air above you.

If you wish to look at lower floors of the building, what would you do? You would command the building to shift **up** (which is equivalent to the elevator going down). This is exactly what you do in the editor. The following is a brief summary of the commands that you can use to move around in your file. They are entered on the command line at the bottom of your screen when you're in the editor.

**+5** shifts the file up 5 lines so that your view is the next 5 lines **down**. The "+" is optional. Just a 5 or any number is acceptable.

To adjust your view in the opposite direction, i.e., towards the top of the file, use a minus sign preceding the number of lines you want to shift, e.g., **-20** will display the portion of the file 20 lines above your current position.

The command **top** will go to the top of the file. The command **bot** will go to the bottom of the file.

When a number is preceded with a colon, the editor will go directly to that absolute line number. E.g., **:104** would display lines 93 through 115, with line 104 exactly in the middle of the screen.

To locate a string of characters, enter a slash (/) and the character string to be searched for. It will locate the first instance of that string. If you want to search for later occurrences, continue entering equal signs (=) until you find the occurrence you desire.

Finally, the insert command, **i**, discussed above, is entered from the command line and allows you to insert a virtually infinite number of new lines at that point in the file.

It would not be useful to give every detail of the editor here. See Appendix E for a list of documents which describe how to use the editor. If you need assistance, please see Appendix F for human help.

*LAWRENCE BERKELEY LABORATORY  
TECHNICAL INFORMATION DEPARTMENT  
UNIVERSITY OF CALIFORNIA  
BERKELEY, CALIFORNIA 94720*