

**PRODUCTION OF LIBRARY CATALOG CARDS AND BULLETIN  
USING AN IBM 1620 COMPUTER AND AN IBM 870  
DOCUMENT WRITING SYSTEM**

Donald P. MURRILL: Philip Morris, Incorporated, Richmond, Virginia

*A program is presented which runs on an IBM 1620 Computer and produces punched cards that activate an IBM 870 Document Writing System to type catalog cards in upper- and lower-case characters. Another program produces punched cards which instruct the 870 to type a library accessions bulletin. The programs are written in FORTRAN II and are described in detail. Estimates of costs and production times are included.*

Producing library catalog cards and accessions bulletins with the aid of a computer is not a new idea. Since 1963 several published papers have described projects that have resulted in the production of such cards and bulletins, either as the principal end products or as two products under a total systems concept.

Kilgour (1,2,3), while with the Yale Medical Library, described a project which had been developed jointly with the Harvard and Columbia Medical Libraries under a grant from the National Science Foundation. Six programs were written to produce catalog cards by means of an IBM 1401 computer and either a 1403 line printer or an IBM 870 Document Writer. These programs were of interest to the Philip Morris, Incorporated, Research Center Library because of the upper- and lower-case printing capability.

During the period 1961-1963 the Technical Library of the Bureau of Ships, Department of the Navy, used a 1401 computer to automate the preparation of 3x5 catalog cards and the library accessions bulletin. In

reports concerning Project SHARP (*SH*ips Analysis and Retrieval Project) (4,5) these functions are described as being two of eleven formal outputs of the project. Production of the cards is termed a "by-product" of the accumulated bibliographic data.

In an IBM publication concerning the Administrative Terminal System, Holzbaur and Farris (6) list the production of catalog cards and of the library bulletin as two of approximately seven outputs of a total system using the 1401 computer. IBM began automatic card preparation in 1963. Other publications by IBM also deal with the production of catalog cards and library bulletins with the help of a computer (7,8).

In 1964 Buckland (9) prepared a report for the Council on Library Resources, Inc., in which he described the preparation of catalog data on a tape-punching typewriter. The perforated tape was processed by computer for phototypesetting, tape typewriter, or line printer output.

At the 1967 meeting of the American Documentation Institute (now American Society for Information Science) Cariou (10) discussed the preparation of catalog cards by means of a computer and an IBM Document Writing System. She programmed her computer to count the number of spaces between sentences and to use this count to determine the type of information it could expect next and, thus, what kind of processing it should give that information. The files set up by this program were used with another program to punch cards for the Document Writer.

The computers used with the programs discussed in the foregoing were of the IBM 1400 series or equivalent. The Philip Morris Research Library has access to such computers but only on a limited basis. It has ready access to a 1620 computer. For this reason its programs were written for use with the 1620. The punched card output is processed by the 870 Document Writing System to produce printout in upper- and lower-case characters.

The Philip Morris library contains 5000 books, subscribes to 600 journals, and serves approximately 300 people. It is growing at the rate of approximately 1000 volumes per year.

#### THE IBM 1620 COMPUTER

The 1620 computer for which the programs described in this paper were written has 20,000 positions of core, and typewriter and punched card input. It has typewriter and punched card output and differs from the computers used with the programs mentioned earlier in not having magnetic tape or a line printer. The lack of these devices does not detract from the final output, although that output is not produced as fast with the 870 Document Writer as it would be with a printer. Also, data punched into cards cannot be stored on tape but must be saved, if desired, in the cards. There are many 1620 computers extant, however, and many libraries which might use this machine for the automation of their library functions. Programs which permit the use of the 1620 in the pro-

duction of catalog cards and accessions bulletins might be of help to these libraries.

The programs are written in FORTRAN II, specifically for compilation with a PDQ FORTRAN processor deck, as developed by Maskiell (11) of McGraw-Edison Company. When the program for the Library Catalog Cards is compiled, columns 7-13 of card 25058 in the Fixed Format Subroutine deck must be changed to 4903158. This eliminates the punching of sequence numbers in the computer output cards which are to be run through the 870 system, thus permitting use of the last eight columns for turnup control characters (&) and for the forms feed character (-).

### THE IBM 870 DOCUMENT WRITING SYSTEM

The 870 Document Writing System is a combination of a Control Unit, which is a card punch machine with a control panel, and a typewriter. A complete system could include an auxiliary keyboard, a paper-tape punch, an auxiliary card punch, and a second typewriter; but for an output of library catalog cards and bulletin only the Control Unit and one typewriter are needed.

The punched cards whose contents are to be typed are placed in the hopper of the Control Unit and are passed in a continuous feed under the read head. The control panel interprets the punched characters in the cards and produces the desired alphameric symbols and punctuation marks on the typewriter.

For the production of library catalog cards, continuous 3x5 card forms are put into the typewriter and the punched cards which are output from the computer, as explained in the next section, are passed through the Control Unit. Carriage turnup is controlled by a continuous chain of small beads in which four large beads are equally spaced three-and-a-half inches apart. One rotation of the chain corresponds to the turnup of four 3x5 cards. Before typing begins, one of the large beads is positioned to coincide with the top of a card. A special character in the last card of a unit of punched cards obtained from the computer activates the carriage control, causing turnup to the next large bead and the top of the next card.

Eleven special character exits on the control panel correspond to the special characters on the 836 card-punch keyboard. Any one of the exits can be wired to do a certain job when the special character is encountered in a punched card. It seems logical to have a punched period produce a typed period, a punched comma a typed comma, and a punched slash a typed slash. These three characters, along with the numbers, are lower case on the 866 typewriter. Other special characters on the typewriter, such as parentheses, brackets, colon and semicolon, are obtained by punching the upper-case shift character in the card immediately before the appropriate lower-case character. The left parenthesis, for example, is produced from an upper-case 9, the right parenthesis from an upper-case O. Reference to the typewriter keyboard gives ready knowledge of how

to obtain any desired character (See Table 1). Other special character exits on the control panel are wired to produce lower-case shift, single and multiple upper-case shift, typewriter carriage return, tab control, underlining (obtained from an asterisk punched immediately before the character to be underlined), and forms feed (See Table 2).

Table 1. Special Character Production

Card punch input	Typewriter output
@.	.
@,	,
@/	/
#1	"
#2	l
#3	+
#4	[
#5	]
#6	?
#7	&
#8	,
#9	(
#0	)
#.	:
#,	:
#/	-
*	- (underline)

(@ = lower case shift, # = upper case shift)

Table 2. Control Panel Wiring

- Typewriter 1 ON —> Column zero single
- Card read ON —> Column zero single
- Carriage return —> Card drop out
- Special character exits — Common channel:
- . —> Type-only entry .
- , —> Type-only entry ,
- / —> Type-only entry /
- \* —> Type-only entry \*
- & —> Carriage return
- @ —> Lower case shift
- # —> Single upper case shift
- \$ —> Multiple upper case shift
- % —> Typewriter tab control
- —> Forms feed start 1

(Punched "&" prints as "+", punched "%" as "(", and punched "#" as "=".)

To obtain panel control of the typewriter the star wheels of the Control Unit must be engaged and a program card must be on the drum. For the library programs a blank program card is used.

The same panel that is used to control the printout of the catalog cards is used for the library bulletin. The same manually punched cards are used as input data to both programs, with modifications in the case of the bulletin, as will be noted later.

#### LIBRARY CATALOG CARDS

The data cards containing the bibliographic information on the books being cataloged must be punched with care, but no worksheet need be filled out by the librarian. This saves him a great deal of time and trouble. He must designate the call number, tracings, and other details for the keypunch operator, but he can do this on a plain piece of paper, which also contains a transcription of the title page or of an LC proof slip. He does not need to remember or look up any codes, nor does he need to be concerned with where each letter will go in the punched cards.

The keypuncher must know certain details, of course; e.g., that the author's name always starts in column nine, and that two blank spaces are inserted after a period, one after a comma. She must know the special characters which have been wired in the control panel to produce upper- and lower-case printout on the 866 typewriter. She must remember that the character for multiple upper-case printing will produce capital letters until a different control character is encountered, and she must punch the appropriate control characters where needed. These are details which are quickly learned with use, but because of them only one keypunch operator should be selected to handle the library data and to be responsible for production of the catalog cards.

The 3x5 catalog card will hold seventeen typed lines. It was arbitrarily decided that the tracings, i.e., the subject entries and added entries, would always start on line thirteen. If there are more than twelve lines of bibliographic information before the tracings, the program described here will not punch the thirteenth card, and the computer processing will stop. If there are more than seventeen lines total, the program will not provide for automatic turnup to the necessary continuation card. Catalog cards which fall into these two categories must be typed manually, but only a small percentage of cards need to be prepared in this way.

Computerizing the production of catalog cards enables one set of keypunched cards to produce several sets of computer punched cards and, thus, several catalog cards for each book. The necessity for typing the card manually is eliminated. In the program presented here each punched card represents one printout line on the catalog card. This makes it possible to count the number of lines before the tracings, by counting each punched card as it is read by the computer, and to skip as many lines as is necessary to always start the tracings on line thirteen.

To facilitate explanation in the following discussion certain terms and definitions have been assumed: a "unit" is all of the manually punched cards required for all of the catalog cards for a given book, including main entry cards and tracing cards, the latter being those with headings taken from the subject and added entries; a "set" is all of the manually punched cards required for the main entry card and not including the headings punched separately; the "body" cards are those required for the "body" of the catalog card, down to but not including the tracings. Figure 1 shows how the keypunched cards look.

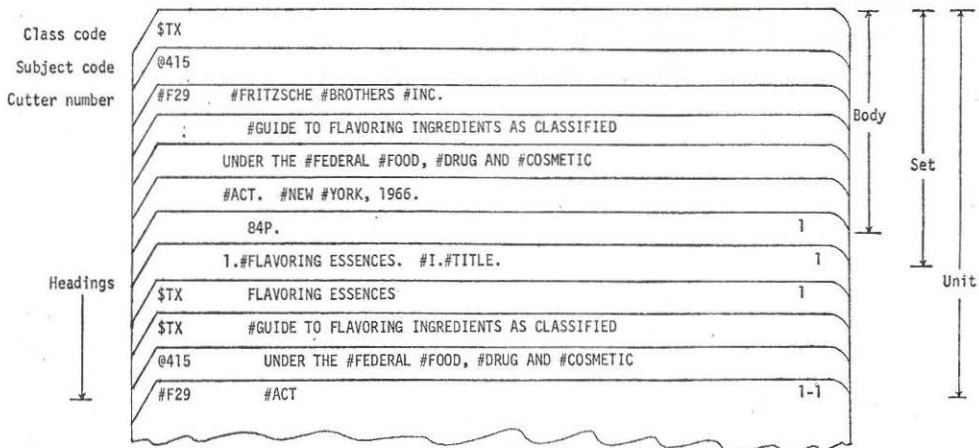


Fig. 1. Keypunched Cards.

Certain controls must be punched into the data cards. A "1" is punched into a field called MON to indicate the last body card; a "1" is punched right-justified into a two-character field called KON to designate the last tracings card, before a repetition of the tracings to serve as headings; and "-1" is punched into the field KON to indicate the last card for a given book. One rule must be followed: if there are no non-spacing characters in a card, e.g., upper- and lower-case shift characters, punching should not extend beyond column 57. For each non-spacing character punching can be extended one column, but must not extend beyond column 65.

In the READ statement of the computer program discussed here sixty-five columns of alphameric data are read and stored in an array. The contents of MON and KON are saved until the next card is read.

```

10 READ 11, M1(I),M2(I),M3(I), (Read statement for all lines
    1M4(I),M5(I),M6(I),M7(I), on main entry cards and all
    2M8(I),M9(I),M10(I),M11(I), lines except headings on trac-
    3M12(I),M13(I),MON,KON      ing cards.)
11 FORMAT(13A5,11X,2I2)
    
```

As long as MON and KON are zero, each card is punched as it is read, and the program returns to the READ statement:

```

15 PUNCH 8, M1(I),M2(I),M3(I), (Punch statement for all lines
    1M4(I),M5(I),M6(I),M7(I),    except last of body of main
    2M8(I),M9(I),M10(I),M11(I),  entry cards.)
    3M12(I),M13(I)
8  FORMAT(13A5)
14 GO TO (32,33),NBC           (NBC is set to "1" at begin-
32 I=I+1                       ning of program.)
    GO TO 10

```

When the last body card is read and MON=1, the program branches to a statement which stores N-1, N being the number of cards read to that point. The program then calculates the difference between thirteen and N and branches to the appropriate statement for punching the last body card and the special characters needed to produce the number of blank lines which will begin the tracings on line thirteen in the printout:

```

23 MAX=13-N
    GO TO (15,19,20,21,22,34), MAX
21 (e.g.) PUNCH 30, M1(I),M2(I), (Sample punch statement for
    1M3(I),M4(I),M5(I),M6(I),    last body line; includes special
    2M7(I),M8(I),M9(I),M10(I),  characters to produce skipped
    3M11(I),M12(I),M13(I)        lines.)
30 FORMAT(13A5,11X,4H&&&&)

```

The computed GO TO contains only six statement numbers to which the program can branch because of the limited memory of the 1620 computer. This means that there must be at least seven cards before the tracings and, if necessary, blank cards must be added to reach this count.

The subject entries and added entries, i.e., the tracings, are read and punched, card by card, after a branch back to the READ statement (10). With the reading of the last tracings card, KON=1. The program branches to a statement which punches the last tracings data and a special character (-) which, during printout, will cause the typewriter to turn up to a new 3x5 card.

This completes the preparation of punched cards for the first main entry card. Most libraries need more than one such card. Additional sets of punched cards are prepared by means of a DO loop and a return to an earlier part of the program:

```

36 DO 6 K=1,NB                 (NB has been previously set to
    6 PUNCH 8,M1(K),M2(K),      one less than the number of
    1M3(K),M4(K),M5(K),M6(K),  body cards.)
    2M7(K),M8(K),M9(K),M10(K), (Punch statement for all except
    3M11(K),M12(K),M13(K)      last body card for all main en-
                                try cards except the first.)

```

MAX is again defined in statement 23 and the last body card is again

punched. Statement 14 sends the program to statements which punch the tracings for the second and subsequent cards:

```

33 NIX=NB+2
    IF(NOT-NIX)1,9,9                (NOT=one less than total
                                     number of cards in set.)

 9 DO 47 JO=NIX,NOT
47 PUNCH 8, M1(JO),M2(JO),        (Punch statement for all but
    1M3(JO),M4(JO),M5(JO),        last line of tracings.)
    2M6(JO),M7(JO),M8(JO),
    3M9(JO),M10(JO),M11(JO),
    4M12(JO),M13(JO)

 1 I=NOT+1
    PUNCH 26, M1(I),M2(I),        (Punch statement for last line
    1M3(I),M4(I),M5(I),M6(I),    of tracings; includes special
    2M7(I),M8(I),M9(I),M10(I),  character to produce type-
    3M11(I),M12(I),M13(I)       writer turnup.)
26 FORMAT (13A5,14X,1H-)
    
```

A count is kept, in a fixed point variable, NCS, of the number of card sets which have been prepared. This variable is now used to determine the next step in the program:

```
GO TO (36,36,53,53,53,53,53,53,etc.),NCS
```

The number of card sets punched for main entry cards is one more than the number of times "36" is inserted in the foregoing computed GO TO statement.

The headings for the tracing cards, as shown in Figure 1, are placed after the card set, thus completing the card unit. A "1" in the MON field controls the processing of one heading at a time. If a heading requires more than one card, the control is punched into the last card. Preparation of the remaining punched cards required for the tracing cards is provided for in the following statements:

```

59 M=1
 2 READ 11,N1,N2,N3,N4,N5,N6,    (Read statement for headings.)
    1N7,N8,N9,N10,N11,N12,N13,
    2MON,KON
    PUNCH 8, N1,N2,N3,N4,N5,N6,  (Punch statement for head-
    1N7,N8,N9,N10,N11,N12,N13  ings.)
    M=M+1
    IF(MON)2,2,51
51 IF(M-4)12,52,52
52 I=3
    PUNCH 13, M2(I),M3(I),        (Punch statement for main en-
    1M4(I),M5(I),M6(I),M7(I),    try on tracing card, drawn from
    2M8(I),M9(I),M10(I),M11(I),  computer memory; class code is
    3M12(I),M13(I)               omitted.)
    
```



A header card is punched not only with the heading but also with the alphabetic class code of the book being cataloged. If there is a second card to the heading, it is punched with the subject code number. A third card, if there is one, contains the Cutter number. In the program statements cited above, provision is made for retrieving from the computer memory any part or parts of the call number that are not read in with the heading. Care is taken to assure that no part of the number that is not needed is retrieved. As headings are generally in capital letters, and the multiple upper-case shift character is used to produce them, it is necessary to precede the subject code number of a second card to a heading with the lower-case shift character. Otherwise, instead of the subject code number's being typed, the upper-case characters for the digits of this number will be typed.

Cards for the remainder of the body of the catalog card, except the last line, are punched by use of a DO loop:

```

M=4
12 DO 7 J=M,NB
  7 PUNCH 8, M1(J),M2(J),M3(J), (Punch statement for remain-
    1M4(J),M5(J),M6(J),M7(J),   der of tracing card through
    2M8(J),M9(J),M10(J),M11(J),  next-to-last body line.)
    3M12(J),M13(J)

```

Before the last body card can be punched, the value of N must be set. Before this can be done, the value of I, subscript for the PUNCH statement, must be tested, which is accomplished in a series of IF statements. I is set to NB+1; then if there are fewer than four cards in a heading:

```

54 IF(I-8)18,42,43
43 IF(I-10)44,45,46
46 IF(I-12)48,49,50
18 N=7
42 N=8
44 N=9
45 N=10
48 N=11
49 N=12

```

After each statement setting N to a specific value, a GO TO statement sends the program back to statement 23.

If the number of cards in a heading is four or more, the value of N is set by the following statements, MAT having been equated earlier to one more than the number of cards in the heading:

```

41 IF(MAT-4)23,27,37
37 IF(MAT-6)38,39,40
27 N=N+1
38 N=N+2
39 N=N+3
40 N=N+4

```

(It is assumed that no heading will be longer than seven lines.)

Again, after each statement setting N to a specific value, a GO TO statement sends the program back to statement 23.

The last card of the unit of manually punched cards contains -1 in the KON field. When this control number is read and tested, the program branches to a DO loop which erases the bibliographic data stored in the computer's memory, then branches to the beginning of the program to statements which set N=0, I=1, NBC=1, and NCS=0. The computer is ready to read the first card of the next unit. Figure 2 shows a printout of a main entry card.

Exact cost figures for the production of catalog cards are not available, but a good estimate would be approximately eighteen cents per card. This cost is high, perhaps, but when the cards issue from the 870 system, they are complete, including call numbers and tracings, items that are missing from the LC cards and that must be typed onto these cards. The saving is in time; the cards produced by the program described here can be ready for filing in the library within a week; delivery of LC cards is, on the average, six months after the order date. The overall cost is reduced somewhat by the fact that the same cards that are punched for the catalog cards are used for the accessions bulletin. They can also be used for bibliographic listings under selected headings.

A listing of the complete catalog card program may be obtained from the author.

TX	
415	
F29	Fritzsche Brothers Inc. Guide to flavoring ingredients as classified under the Federal Food, Drug and Cosmetic Act. New York, 1966. 84p.
	1.Flavoring essences. I.Title.

*Fig. 2. Printout of Main Entry Card.*

## LIBRARY BULLETIN

The data for the *Library Bulletin* program consists of the cards which were punched manually for the catalog card program. The information concerning each book which is to be included in the bulletin is the choice of the individual librarian. At the Philip Morris, Incorporated, Research Center Library none of the data after the publication date is included. Care must be taken that there are at least five cards remaining for each book after unwanted cards have been discarded. The reason for this will become apparent later.

The headings, consisting of the first subject entries in the tracings of the books to be listed, need to be punched. The books are grouped in the bulletin under these headings. Each of the headings is to be in upper-case letters, so the first column of each card must be punched with a dollar sign, the special character wired in the 870 control panel to produce multiple upper-case printout.

As with the catalog card program, certain controls are required for the bulletin program. A "1" must be punched in the field called MON in the last card of each book set. A "1" must be punched in a field called JON in each header card.

A code punched into the cards to facilitate keeping them in the proper order is not necessary for the computer program, but it is certainly desirable. Therefore, a sequence code consisting of eleven digits is in each card: the first five digits designate the subject heading, the next four the author, and the last two the card sequence for each book. Judicious selection of the codes makes it possible to put the cards into proper or near proper order with a card sorter — an especially useful feature for preparing a listing of all the titles under a given heading acquired over a period of several months.

In the bulletin the titles are numbered sequentially beginning with "1", the numbering being controlled with a fixed point variable, NO, which is set to one at the beginning of the computer program.

The data cards are arranged in alphabetic order by author and are placed behind the appropriate header cards, which have also been arranged in alphabetic order. The first card to be read and punched is a header card:

```

READ 4, KA,KB,KC,KD,KE, KF, (Read statement for first head-
1KG,KH,KI,KJ,KK,KL,KM,JON  er card.)
4 FORMAT(13A5,11X,11)
15 PUNCH 3, KA,KB,KC,KD,KE, (Punch statement for header
1KF,KG,KH,KI,KJ,KK,KL,KM  cards; includes special charac-
3 FORMAT(3H&&&, 13A5)      ters to produce skipped lines.)

```

The first three data cards are read in a DO loop. The three parts of the call number are stored in an array, then the main entry, in the third card, is punched, along with the sequence number and the first part of the call number, the alphabetic class code.

DO 6 I=1,3  
 6 READ 1, M(I),N(I),LB,LC,LD, (Read statement for call number and main entry.)  
 1LE,LF,LG,LH,LI,LJ,LK,LL,  
 2LM,MON  
 1 FORMAT(A5,A4,11A5,A1,12X,I1) (Punch statement for sequence number, main entry, and class code; includes characters for skipped line, lower case, and typewriter tab control.)  
 PUNCH 12, NO,LB,LC,LD,LE,  
 1LF,LG,LH,LI,LJ,LK,LL,LM,  
 2M(1),N(1)

12 FORMAT(IH&,IH@,I4,1H.,2X,  
 112A5, 1H%,A5,A4)

The next two cards are read and new cards are punched in another DO loop. The remainder of the call number, the subject code number and the Cutter number, are punched into these two cards. If there are fewer than five cards to be processed in each book set, part of the call number will be lost. Blank cards are added, if necessary, to bring the count to five.

16 DO 7 I=2,3  
 READ 17, KA,KB,KC,KD,KE, (Read statement for fourth and fifth cards of set.)  
 1KF,KG,KH,KI,KJ,KK,KL,MON  
 17 FORMAT(8X,11A5,A2,12X,I1) (Punch statement for second and third lines of bibliographic data and remainder of call number; includes typewriter tab controls.)  
 PUNCH 5, KA,KB,KC,KD,KE,  
 1KF,KG,KH,KI,KJ,KK,KL,M(I),  
 2N(I)  
 5 FORMAT(4X,1H%,12A5,5X,  
 11H%,A5,A4)

7 CONTINUE

The MON field in the fifth card is tested in an IF statement. If this field is zero, indicating that there are more cards in the set, the additional cards are read and new ones are punched in another pair of READ and PUNCH statements. If the MON field is "1", indicating that the processing of the card set is complete, one is added to the sequence number variable, NO, and the next card is read. If this is a header card, as indicated by "1" in the field called JON, the program branches back to statement 15. If, on the other hand, it is the first card of another title under the same heading, the class code is stored in an array, and the second and third cards are read in a DO loop. The main entry, the content of the third card, is then punched, along with the class code and the sequence number.

9 NO=NO+1  
 READ 4, KA,KB,KC,KD,KE,KF, (Read statement, for heading if JON=1, for main entry and class code of new card set if JON = 0.)  
 1KG,KH,KI,KJ,KK,KL,KM,MON

7. IBM Manual No. E20-8094: *Mechanized Library Procedures*, 14.
8. IBM Manual No. E20-0093: *Library Catalog Production - 1401 and 870*.
9. Buckland Lawrence F.: *The Recording of Library of Congress Bibliographical Data in Machine Form*. A report for the Council on Library Resources, Inc. (Washington Council on Library Resources, Inc.: November 1964) (rev. February 1965).
10. Cariou, Mavis: "A Computerized Method of Preparing Catalogue Cards, Using a Simplified Form of Data Input," *Proceedings, American Documentation Institute Annual Meeting*, 4 (October 1967), 186-90.
11. Maskiell, Frank H.: "PDQ FORTRAN (An Interpretive Program for the FORTRAN Language)" (November 1963).