

XIV. Bean's Automatic Tape Manipulator -- A Description, and Operating Instructions

Jeffrey Bean

1. General Description

BATMAN is a generalized updating program for handling BCD card images on tape, in either unblocked form, or in FAP type blocks of sixteen cards per record. It contains facilities for serializing the output in several different ways, monitoring the output by simultaneous printing using fairly sophisticated methods for tape positioning.

It has been constructed with several ideas in mind:

1. Ease of operation. This is seen in the use of control cards with a flexible format, and in the fact that almost nothing is illegal. No matter what happens, the program makes some attempt to continue. Program stops occur only when an end of file is read on the system input tape, and when a tape writing error is detected.
2. Speed. All of the principal I/O is double buffered. Under normal circumstances, the program should be able to operate at the maximum speed of tape motion.
3. Ease of modifications. The program is built up of a set of more or less independent modules which communicate via a set of switches. New control cards are easily defined by adding entries to a pair of expandable tables. Since the program is not intended to be loaded along with any other decks, no attempt has been made to conserve core with the result that the logic of the program is relatively simple to follow.

2. Operating Instructions

The composition of the control cards is more easily appreciated if one understands the logic behind the control card interpreter. This interpreter operates on the basis of certain keywords which may be located anywhere on the card, in any order, delimited either by blanks or by commas. A table is maintained of these keywords, in which every word on the card is looked up. If the word is not found, it is ignored and has no effect on the interpretation of the control card. Thus, the control card may have any number of "padding" words as long as they are not keywords. When a keyword is found, a bit is turned on in a "key". Thus, repetitive occurrences of keywords have no effect beyond the first occurrence. When all the words have been looked up, the "key" itself is looked up in another table from which the address of the appropriate routine is obtained. If the "key" is not found, the control card is ignored and a message is printed out to this effect. Any numbers on the card are taken in order and placed in a last-in-last-out-list. Serial numbers, which may consist of from one to eight digits and alphabetic characters enclosed in parentheses, are right shifted with leading zeroes added (blanks are considered to be zeroes) and put on a separate last-in-last-out-list.

Thus, control cards are defined according to which keywords occur in it. If numbers or serial numbers are required for the interpretation of the card, these are used in the order in which they occur. Anything else is ignored. In the following list of control cards and their meanings, keywords have been underlined, and numbers are represented by 'XX'; serial numbers represented by '(...)'. Optional portions are enclosed in /'s. Reasonable padding words have been added in order to suggest the meaning of the card.

DEFINE LOGICAL TAPE XX AS THE INPUT TAPE

This defines (or redefines) the input tape. Its effect is maintained until the input tape is redefined by a similar control card. If the tape has not yet been used as an input tape, it will be automatically rewound. It is unnecessary to specify whether the input is in blocked or unblocked form (or mixed form). If the tape has been used before, it will be both logically and physically in the same position in which it was left, even in the middle of a blocked record.

DEFINE LOGICAL TAPE XX AS THE BLOCKED
UNBLOCKED OUTPUT TAPE

This defines (or redefines) the output tape. Its effect is maintained until the output tape is redefined by a similar control card. If the tape has not yet been used as an output tape, it will be automatically rewound. If the tape has been used before, it will continue on from the position at which it was left. If the output tape is defined to be logical tape 6, the mode will be automatically set to unblocked regardless of whether it was specified to be blocked or unblocked. Furthermore, carriage control for single spacing will automatically be included in the output.

GO TO FILE NUMBER XX

This causes the input tape to be positioned at the beginning of the file specified, where the files are numbered consecutively starting with 1. If this control card is encountered while the input tape is

positioned in the middle of a file, the remainder of the file will be either deleted or copied depending on whether the last data transmission instruction was a "copying" or "deleting" one, respectively. If no such instruction was executed since the input tape was last positioned at the beginning or end of a file, the remainder of the file will be deleted.

COPY XX FILES
DELETE XX FILES

These instructions cause the number of files specified to be copied or deleted as specified. This does not include the file in which the input tape is now positioned, unless it is positioned at the beginning of the file in which case it is included. Again, the remainder of the current file will be copied or deleted depending on the last data transmission card.

COPY XX CARDS
DELETE XX CARDS

These instructions cause the number of cards specified to be copied or deleted as specified. If an end of file mark is encountered on the input tape while executing the instruction, no more cards will be copied or deleted, and the next control card will be interpreted. If the next control card is also a data transmission instruction, the program will continue on past the file mark. When the file mark is encountered, a file mark will be written on the output tape unless otherwise specified by the appropriate control card.

COPY UP TO (...)
DELETE UP TO (...)

Cards will be copied from the current position of the input tape up to and including the card with matching serialization. If no card with matching serialization exists, the program will continue up to but not including a card with higher serialization (remember: blanks are considered to be zeroes). If the current position of the input tape indicates that the sought for serial number has already been passed, the input tape will be backspaced to the beginning of the file and the search will continue. Again, if an end of file mark is encountered, data transmission will cease and the next control card will be read.

COPY FROM (...) TO (...)
DELETE FROM (...) TO (...)

Essentially, these are shorthand notations for two of the preceding control cards. 'COPY FROM (...) TO (...)' is equivalent to 'DELETE UP TO (...)' followed by 'COPY UP TO (...)', and similarly for 'DELETE FROM (...) TO (...)'. .

COPY THE REST OF THE FILE
DELETE THE REST OF THE FILE

These instructions are obvious.

INSERT THE FOLLOWING CARDS

Cards will be read from the system input tape and copied onto the output tape until a card with '*NEXT' in columns 1 through 5 is found. The latter is not copied onto the output tape.

INSERT THE FOLLOWING CARDS AFTER (...)

This is equivalent to a 'COPY UP TO (...)' followed by the normal insertion instruction.

WRITE AN EOF
WRITE A *STOP

These instructions are obvious (EOF = end of file).

SERIALIZE ON TEXTS, KEYED ON XX COLUMNS, STARTING WITH COLUMN XX

This causes output to be serialized in columns 73-80. Serialization consists of two four digit concatenated numbers, with leading zeroes. The first number represents the text number, while the second number represents the line number within the text. Whenever a card about to be written out contains a '*' in column 1, the contents of the columns specified (maximum number of columns: six) are converted to an integer according to the following algorithm. The first two binary bits of each character are multiplied by 10 to the power $m+n$, where m is the number of columns being converted, and n is the number of characters to the right of the character under consideration. The last four binary bits of the character are multiplied by 10 to the power n . These numbers are summed for each character and become the result.

Examples: in a three-digit field
 123 becomes 0123
 OOA becomes 1001
 OJA becomes 21011
 SJA becomes 321211

This integer is truncated to four digits from the left and is used as the first four columns of the serialization for the remainder of this text. The

last four columns begin with 0000 and are incremented by 10 for each line of text. Thus, a text headed by *TEXT 23D would be serialized 12340000, 12340010, etc.

SERIALIZE STARTING WITH (...) AND INCREMENTED BY XX

This causes output to be serialized in columns 73-80, the first serial number being as specified, and succeeding cards being serialized by numbers incremented by the amount specified.

DO NOT SERIALIZE

This causes columns 73-80 of the output to be blanked out.

COPY THE OLD SERIALIZATION

This causes the serialization on each input card to remain unchanged on output. Note that this may result in incorrectly ordered serialization on the output tape.

/DO NOT/ COPY *STOPS

Whenever a card to be written out contains *STOP in the first five columns, it will be written or not as specified. Normal procedure is to copy the *STOPS. This instruction is useful when concatenating one or more files.

/DO NOT/ COPY EOFs

Whenever an end of file mark is encountered on the input tape, a file mark is written on the output tape, or not, as specified. Normal procedure is to copy the eof's.

PRINT TAPE XX, XX FILES

This causes the tape specified to be printed until the number of file marks specified have been encountered. This includes the file in which the tape is currently positioned. The tape is left in the position where it stopped when encountering the last file mark. It is not returned to the position in which it started. This instruction has no effect on the definition of the normal input and output tapes.

PRINT TAPE XX, XX CARDS

This causes the number of cards specified to be printed from the tape specified unless an end of file mark is encountered first, in which case, the next control card is read. Again, this instruction has no effect on the definition of the normal input and output tapes.

PRINT TAPE XX, FILE XX, XX CARDS

This causes the tape specified to be positioned at the specified file, and then the requisite number of cards printed. The same remarks as for the other PRINT instructions apply.

PRINT TAPE XX, FILE XX, FROM (...) TO (...)

This causes the tape specified to be positioned at the specified file, and then the requisite cards are printed. Operation is similar to the 'COPY FROM (...) TO (...)' instruction. The same remarks apply as for the other PRINT instructions.

NOTE: In the above instructions, several numbers must appear on the control

card. These numbers must appear in the order specified or confusion will result.

/DO NOT/ PRINT SIMULTANEOUSLY WITH THE OUTPUT

This causes (or prevents) each card being written on the output tape to be written also on the system print tape with automatic carriage control for single spacing. Normal procedure is not to print simultaneously. Whenever the output tape is defined to be logical tape 6, the operation of this instruction is suspended, but resumes when the output tape is redefined as something else. This provides automatic protection against double printing.

FINISH

This instruction must be the last one in the deck. It designates that the run has been completed, and causes any input and output tapes which have been used, with logical tape numbers higher than 7 to be rewound.

Notes:

1. In order to rewind an input tape, the following command should be used: `G0 T0 FILE |`.
2. If the output tape is 6, the command `D0 NOT PRINT 0UTPUT` produces an error.
3. The break characters for BATMAN control cards are blank, period and comma, only.