DECUS
PROGRAM LIBRARY

DECUS NO.	8-466E
TITLE	DECTAPE UTILITY PROGRAMS P?S-Ø8-1.1E
AUTHOR	Mario DeNobili Submitted by: Stanley Rabinowitz
COMPANY	Polytechnic Question Society Brooklyn, New York
DATE	March 1, 1971
SOURCE LANGUAGE	PAL III

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

DECTAPE UTILITY PROGRAMS
P?S-Ø8-1.1E

DECUS Program Library Write-up

DECUS NO. 8-466E

DUMP

I. User Manual.

I.1 Introduction

DUMP is a general purpose system which allows the experienced systems programmer to dump blocks from Dectape in both octal and packed ASCII, to transfer blocks from one tape to another, and to make changes to specific words in a given block on a tape. DUMP assumes that the tape is formatted in the standard manner (201_8 words per block). All numbers used in the following will be in base 8. DUMP was written by Rod Dorman.

I.2 Loading Procedure.

DUMP is loaded into core from a tape on which it resides as an RL system by the command to the RL monitor:

```
RUN  DUMP
```

No parameters or source files need be specified. When DUMP is loaded, it will respond with line feeds followed by

```
MODE?
```

I.3 Modes of operation.

There are five modes of operation which may be specified at this point:

(i) BOotstrap
(ii) REwind
(iii) DUmp
(iv) ZAp
(v) TRansfer

The mode must be entered followed by a carriage return (which will not echo). Only the first two letters of the command are significant.

I.3(i) BOotstrap

This command causes the program to bootstrap by branching to location 7600.
AFTER READING BLOCK Ø INTO 76ØØ-7777.

I.3(ii) REwind

This command rewinds the tape. This is useful when changing tapes. Tapes may be changed any time the program asks for the mode.

I.4(iii) DUmp

The purpose of this command is to allow you to dump blocks from the tape onto the teletype in both octal and (packed) ASCII.

The program will ask you for the block number you wish to start dumping from. You respond with the starting block number in octal followed by a carriage return.

(cont.)

The program will then ask you for a number. You respond (in octal) with the number of sequential blocks you wish to dump.

Example: To dump blocks 23, 24, 25, and 26, give 23 as the block number to start at, and 4 as the number of blocks.

Format of the dump.

An example of a dumped block is given in figure 1.

Only the first 200 words in a block are dumped. The 201st word is ignored and cannot be examined or referenced by DUMP. This word is usually of no interest to you and most programs do not use it.

The block is dumped in octal in 20 lines of 10 words each. The lines are numbered on the left from 0000 to 0170 in steps of 10. Since the block corresponds to one page of core, this allows you to compute the relative location (address) of any word from the beginning of the page (block). The first word corresponds to relative address 0000 and the last word corresponds to relative address 0177. To the right of each line is the same 10 words dumped in ASCII format. DUMP assumes in this case that each word represents two ASCII characters packed into a word in the usual format. Zeros are represented by at symbols. The ASCII dump is bordered by asterisks.

Listing Control

While a block is dumping, hitting any character will cause the program to stop dumping that block and go on to the next block (if any are left to be dumped).
While line feeding between listing blocks, hitting any character terminates all further dumping.
The switches on the switch register should all be down during the above operations.
Dumping of a block can be made to commence at a given line number by setting this line number on the switch register, in bits 5, 6, 7, and 8. Thus to start listing at line 140 of a block, the number 140 is set on the switch register.

I.3(iv) ZAp

ZAp is used to make changes to specific words on a tape.

The program will ask you for the block number in which the zap is to be made. You respond with the correct block number (in octal) followed by a carriage return. Before allowing you to make the zap, as a safety measure and a convenience, it will first dump the block in the format explained above.

(continued)

At the conclusion of the dump, the program will ask you for a number. You respond (in octal) with the relative page address (0-177) relative to the beginning of the block of the one word which you wish to change. You then type a carriage return which will echo as a space. If you typed more than 4 octal digits, only the rightmost 4 are significant.

You then type the octal number representing the new contents of the specified word.

The program will then ask you for a number again, and you may modify another location using the same procedure.

You terminate this process by responding with a negative number (e.g. 7777 or 4000) when the program asks you for a number.

At this point the program will make the specified changes on the tape. Write must be enabled. If it is not, the machine will halt; you enable the write, and then press continue.

I.3(v) TRansfer

TRansfer is used to transfer blocks from one tape to another.

The program will ask for the block number from which you wish to start the transfer at. You respond with the desired number followed by a carriage return. The program will then ask for a number. You respond (in octal) with a number from 1 to 33 designating the number of consecutive blocks to be transferred. The program then begins reading blocks into core. When it is ready, it rewinds the tape for you and types CHANGE telling you to change tapes. At this point you mount the tape you wish the transfer to be made to (or keep the same tape mounted if you only wish to juggle around blocks on the same tape). By this time the program has requested a block number. You respond (in octal) with the block number of the block on this newly mounted tape to ~~wish~~ which you wish the data to start writing at. Write must be enabled at this point. When the transferrence is complete, the program will ask for a new mode. If more blocks are to be transferred, you must remount the original tape and begin again. A maximum of 33 blocks can be transferred at one time. If many blocks are to be transferred, it is recommended that they be transferred 30 blocks at a time since this is a nice round number.

I.4 Notes.

The program can be restarted at any time by hitting control bell.
The program is restartable at location 200 which is the starting address.

******DUMP should not be used except by experienced systems programmers.*******

Note: Typing errors can be corrected by hitting alt mode or rub out and then retyping the defective number.

```
RUN DUMP
```

Table 1

```
MODE? DUMP
BLOCK = 12Ø
 NUMBER= 1



BLOCK = Ø12Ø

ØØØØ   Ø454 7422 ØØ1Ø 546Ø 3174 7454 Ø453 7423        *D,<R@H,ØY<<,D+<S*
ØØ1Ø   ØØ1Ø 5454 3217 6464 6457 ØØØ5 ØØ1Ø 5ØØ7        *@H,4ZO444/@E@H(G*
ØØ2Ø   5464 3225 6446 5464 ØØ1Ø ØØØ1 5461 3331        *,4ZU4&44@H@A,1[Y*
ØØ3Ø   7427 1Ø6Ø ØØ1Ø ØØØ2 5455 3245 7424 5464        *<WHØ@H@B,-Z%<T,4*
ØØ4Ø   3244 6425 5464 ØØ12 5Ø11 5464 3252 6445        *Z$4U44@H(I,4Z*4%*
ØØ5Ø   6464 ØØ1Ø 546Ø 3267 7436 Ø453 7417 ØØ1Ø        *44@H,ØZ7<↑D+<O@H*
ØØ6Ø   5466 3266 6466 6443 ØØØ5 ØØ1Ø 5Ø11 546Ø        *,6Z6464#@E@H(I,Ø*
ØØ7Ø   3314 7417 Ø453 7417 ØØ1Ø 5453 33Ø5 7426        *[L<OD+<O@H,+[E<V*
Ø1ØØ   5466 33Ø5 6456 6423 ØØ1Ø 5466 3313 6466        *,6[E464S@H,6[K46*
Ø11Ø   6427 ØØØ5 ØØ1Ø 5Ø11 546Ø 3331 7415 Ø453        *4W@E@H(I,Ø[Y<MD+*
Ø12Ø   7417 ØØ1Ø 5466 333Ø 6466 6427 6453 ØØ1Ø        *<O@H,6[X464W4+@H*
Ø13Ø   5Ø11 ØØØ2 5461 34Ø5 7417 Ø453 7417 ØØ1Ø        *(I@B,1\E<OD+<O@H*
Ø14Ø   5465 3346 6465 6431 ØØØ5 ØØ1Ø 5Ø12 546Ø        *,5[&454Y@E@H(J,Ø*
Ø15Ø   3364 7415 ØØ1Ø 5465 3363 6465 6434 ØØØ5        *[4<M@H,5[3454\@E*
Ø16Ø   6434 645Ø ØØ1Ø 5Ø12 5465 3373 6465 6434        *4\4Ø@H(J,5[;454\*
Ø17Ø   ØØØ5 6434 ØØ!Ø 5453 3377 646Ø ØØ1Ø 5465        *@E4\@H,+[?4Ø@H,5*
```

Figure 1: This shows a dump of block 120 of a tape.

4MAT AND DECTAP

1. 4MAT

This program is a slight modification of DEC's program TOG-8, DEC-Ø8-EUFA, which is used to format new DECtapes. For complete details on how to use it, see the corresponding DEC publication. It is usually used to format a tape using the standard format of 27Ø1 blocks of 2Ø1 words each. To format a tape in this manner, perform the following operations: From an RL system tape containing 4MAT as a system, type RUN 4MAT. After 4MAT is loaded into core, it will rewind and unload the tape currently mounted on tape drive unit 8. You dismount this tape and mount the new tape to be formatted, winding it up just far enough that it holds on the tape drive when tension is applied. Move the readmark-writemark-normal switch to the writemark position. The program types DTA? to which you respond 8. The program types DIRECT? to which you respond MARK and then YES when the program asks for confirmation. Make sure that unit 8 is selected and that write is enabled. Soon the program tells you to place the switch back in the normal position; you do so and type a carriage return to continue. When the program is done, it types DIRECT? again. You can repeat the process on another tape (by typing SAME) or you can hit stop, mount a tape with a good system on it and bootstrap. NOTE: 4MAT is included on the RL Monitor System DECtape but is not on the paper tapes available for this program.

2. DECTAP (and FASTAP)

These are one-page subroutines which allow you to easily read and write blocks of information on DECtapes. You would normally add them to your program (changing the origin setting if necessary). You then add into your program at an appropriate spot, the data constants:

```
          READ, DREAD
   and    WRITE, DWRITE
```

The two routines (one for reading the tape and one for writing on the tape) are then called by one of the following two manners (as a subroutine with 3 arguments):

```
JMS I READ        JMS I WRITE
coreloc           coreloc
mnoblks           mnoblks
block_no          block_no
```

where coreloc is the absolute core address of the spot where the data is to be read from or transferred to.

mnoblks is the (two's) complement of the number of blocks (pages) of information to be transferred.

block no_is the tape block number of the first block on the tape where the data is to be read or written. If mnoblks specifies more than one block, successive consecutive blocks are used.

When the transfer has been affected, these routines transfer control back to the third location following the call (JMS), i.e., to the spot just after the arguments.

Only 2ØØ of the 2Ø1 words on each block of tape are significant. This is the usual mode of operation with most systems, and it is convenient because in this manner, one block of tape corresponds with one page of core.

These routines cannot be used to read or write only smaller parts of a tape than one block, nor can they be used to read or write beginning in the middle of a tape block. The core location at which data is transferred to or from need not start at a page boundary. In general, data should not be read by this means into page 76ØØ of core since that is where the word count and current address locations for the DECtape are located and DECTAP uses these locations. It can be done, however, if you're very careful and know what you're doing.

These two subroutines are packed together and their source is located in an RL source file called DECTAP. No binary file is included since these subroutines are not stand-alone.

If a tape error occurs during execution of these programs, the tape stops, the computer halts and the contents of the DECtape status register B are prominently displayed in the accumulator. The bits that are on indicate the error condition (or you can look on the indicator lights on the tape drive controller). These bit assignments are explained in the Small Computer Handbook (197Ø) page 1Ø6. You can try again and have the tape operation continued or retried by hitting continue.

Caution: Because of an error in the routines, erroneous results occur if you try to read from a tape which is not mounted or for which unit 8 is not selected. If you try to retry, the program tries to write on the tape instead of read. You can not recover in such circumstances. However, the routines do fortunately work in all other cases. In particular, if you try to write on the tape, and write is not enabled, the computer will halt; you can then enable the write and press continue to recover.

Example: To read tape blocks 5 through 12 (inclusive) into core locations 2317 through 3716 (inclusive) you use the following code:

```
JMS I READ
2317
-6
5
```

(All numbers in this document are in octal.)