

PDP-4 PROGRAM LIBRARY

(7-11-I/O)

NUMBER: Digital - 4 - 34 - U

NAME: Tic toc

AUTHOR: J. M. Graetz - DEC

DATE: Writeup revised April 29, 1964

SPECS: AS

Complete program occupies 507₈ locations.

NEEDED: Teletypewriter

ABSTRACT: Teletype input-output conversion package for the PDP-4. Converts input to concise code and packs it three to a word. Converts packed concise code text to teletype and types it out. Routines may be used separately.

DESCRIPTION:

TICTOC is a complete I-O conversion package. No external routines are required. There are four main subroutines in TICTOC which perform the following services:

- | | |
|--------|---|
| tic | converts a string of typed-in characters to concise code and packs them three to a word in the part of memory designated by the user. |
| onetic | will convert one typed character to concise code and present it to the user's program. |
| toc | will type out a string of text packed in the same format used by <u>tic</u> . |
| cotoe | Will convert and type out one character presented to it in concise code. |

The routines may be used separately, except for toc and cotoe which are merely two entries to the same program.

METHOD:

TICTOC uses a table lookup for conversion in both directions. Letters and figures shifts are kept track of internally and do not generate codes for the user. When text is typed in or out, case shifting is done internally and on input is automatically provided where necessary. Tabulations are also performed correctly.

RESTRICTIONS:

The FIO-DEC and Teletype character sets are far from compatible when special characters are required. However, all those characters needed for the Assembler or for Fortran have been provided equivalents on the Teletype set, as shown in the table.

CHARACTER SET EQUIVALENCES

<u>F10-DEC</u>	<u>Teletype</u>
letters	letters
numbers	numbers
.	.
,	,
-	- (minus)
/	/
((
))
+	&
=	:
_ (underbar)	"
\	3
^	\$
— (overbar)	:
× (multiply)	#
↑	:

USAGE:

1. Input

The single character input subroutine, onetic, may be assembled and used independently of tic but not vice versa. In both, letters and figures shifts are taken care of internally; tic also inserts case shifts when needed.

To input a single character, the call is

onetic

When a key has been struck, onetic will convert the character to concise code ~~and~~ exit with this code in the AC. If the resulting character is upper case in the FIO-DEC set, then AC_0 will be set to 1.

By its nature, onetic does not initialize the shift flag. Thus it is recommended that one use the subroutine reset before using onetic for the first time. This is easy enough; merely say

```
reset
.
.
onetic
```

and the job will be done. Reset returns the carriage, gives a line feed, and sets all flags and the tab count to zero.

Onetic keeps track of the tabulator count, and if the tab key is struck, the program will space out to the next tab stop.

To input a string of characters and pack them into memory, the subroutine tic is required. The calling sequence is

```
law stop char
tic
dac buffer
```

where stop char is the concise code of a terminating character. When this character is typed, tic packs it away with the rest of the text and stops listening. Buffer is the address of the first location of ~~E~~ block of storage into which the incoming text is to be packed.

Characters are packed three to a word. The first, second and third characters in order of arrival are packed into the left, middle and the right sections' of the word, respectively. When the terminating character is encountered, it is packed away and the word is filled out with the special code 13. If the last word is complete in itself, an extra word containing nothing but 13's is added to the text.

The normal return from tic is to the instruction following the "dac buffer", with the quantity "dac buffer+n+1" in the AC where n is the number of words used in packing the text into storage.

Two keys have special meaning for tic.

Line feed: causes tic to ignore what has been typed and start over again. LF effectively erases the information in the buffer. A carriage return is provided by the subroutine.

Blank Key: causes tic to return to the main program immediately without filling out the last word. This return is to the same location as normally but with the special code "60" in the AC.

Tic initializes the keyboard so separate use of reset is not necessary.

Onetic occupies 240₈ locations

With tic, the program occupies 317₈ locations.

2. Output

The two output subroutines, toc and cotoc, may not be assembled separately.

To type out a single character, the calling sequence is

```
    law char  
    cotoc
```

where char is the concise code of the character to be typed. It is a good idea here also to call reset before the first use of cotoc, since the latter program makes no assumptions about the state of the keyboard or the case of the incoming letters.

The case shift codes are treated by setting the case flag in cotoc accordingly. No character is typed out. A tab count is kept, and the code for tab (36) will cause the teleprinter to space to the next tab stop.

The carriage can be set to letters mode by the special code "37".

A carriage return will cause both CR and LF to be typed.

Stop code (13) is ignored by cotoc.

To type out a string of text packed accordingly to the format described for tic, the calling sequence is

```
law buffer -1  
toc
```

where "buffer" is the address of the first word of the block of storage containing the text. The last character in the string must be a stop code.

Here again reset is not needed separately as toc does its own initializing. Using cotoc as a subroutine, toc will print the contents of the buffer until it encounters the stop code, at which point it will return to the main program with 13 in the AC.

If either toc or cotoc encounters a code for which no teletype equivalent has been assigned (see table), a "?" will be typed in its place.

The output subroutine occupies 312₈ locations.

3. Useful subroutines.

TICTOC contains some small subroutines which may be of use. The name of the subroutine and its call are identical.

carr This types a CR and a LF and sets the tab counter to zero.

typit This will type out the character given by the code in the low-order five bits of the AC.

tabit This spaces the carriage to the next tab stop. Use of this subroutine requires the user to keep a count of spaces in the storage location "tabc". This count is kept automatically in TICTOC.

ASSEMBLY PROCEDURE:

TICTOC consists of three AS tapes. They may be assembled in any order.

"Subroutines 4c." contains the subroutines, constants and variables essential to all of the TICTOC programs. It must be included when using any one of them.

"Input" contains the two subroutines one tic and tic, in that order. When using the former, only the first part of the tape need be assembled. The two programs are separated by a "start"; to assemble tic, treat the two parts of the tape as two separate programs.

"Output" contains toc and cotoc as one program.

When used as a complete package, TICTOC occupies 507₈ locations.