MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence

Memo Number 164A (Revised 164) June 1970

The Text-Justifier TJ6

R. Greenblatt

B. K. P. Horn

L. J. Krakauer

Introduction

This memo describes the TJ6 type justifying program, which can be used in the production of memos, such as this one. In addition, Appendices *1*, *2* and *3* of this memo contain related information about TECO, the 'Selectric' and the type 37 Teletype, thus gathering most of the information needed for producing write-ups into one location. A sample of input to TJ6 is given in Section IV, and is in fact the very input used to produce this page of output.

The output from TJ6 may be either justified text, with the right margin exactly aligned or it may be 'filled' text, as in this introduction, with the right margin only approximately aligned. The remainder of this memo will be justified.

The sections of this memo are:

I. Introduction
II. Using TJ6
III. Console operation of TJ6
IV. Sample TJ6 input

Appendix *1* Inserting lower case letters into the TECO buffer
Appendix *2* The 'Selectric' output device
Appendix *3* How to use a type 37 Teletype

Section II:

Using TJ6

TJ6 was written by Richard Greenblatt to facilitate the writing of his paper on the chess program and has since been 'improved' by other users. It has recently been used in the production of a few memos and theses, and most of the bugs have been removed.

TJ6 generally takes its input from a file device (UTn: or DSK:) and is line oriented. A line of input is either a command line or a text line. Blank lines are completely ignored (except in nofill-mode, explained later). A line is a command line if the first character in the line is a '.' (except after a LITERA command, explained later). Some commands are followed by a numeric (decimal integer) argument, which will be denoted below by <arg> (The default argument is *1*). Some other commands have characters as arguments. Some commands have a long and a short form. In use, the command must be preceeded by a '.' , which must be the first character on the command line. The command may consist of either upper or lower case characters. Most of the commands which have no explicit inverse have the inverse effect if given an <arg> of *0*.

The novice is fortunate in that the default options for the various parameters are set to give 'reasonable' output. An input file could conceivably have only a END command at the end. Beyond this, the only essential information is in Section III on the console operation of the program.

Details of the operation of TJ6:

Typing _G will close all files and restart TJ6. Output to TTY: or Tnm: will not start until you type a space. Thereafter one more space is required per page, even after the last page.

Hyphen, '-', will be allowed to break lines. '.' , '!' and '?' in adjust mode (or at the end of the line in fill mode) will have two spaces inserted after them (unless they appear inoccuous) and will be preferred in the insertion of justifying spaces. ':' and ';' also have two spaces placed after them under the same conditions, but are not preferred in justifying decisions. Inoccous '.' , '!' , '?' , ':' and ';' 's are defined to be those followed by a letter or digit and those preceded by a letter which in turn is preceded by a space or '.'. (The exact algorithm is more hairy).

'#' quotes the next character on input - a way of putting in spurious CR or LF for example. To get a '#', use '##'. A different quoting character can be defined by the QUOTE command. The quoted character looses its special properties (s.a. '.' being treated specially in justifying decisions and '-' being used to break lines)

Spaces and tabs at the beginning of a lines cause a break and are put in (not slurped up by justification). Sequences of tabs and spaces not at the beginning of the line in adjusted text are compressed into one space before justification.

Error comments:

The place where the error happened is indicated by 4 numbers - (page,line) in input file and (page,line) in output file - one hopes at least one of these will be useful.

The commands are:

<long form>
<short form>|nil
        <Explanation>

END
        Marks the end of the input text.

INDENT
IN

Causes the left margin to be indented <arg> spaces from the physical left margin. The indentation will continue until altered by another command. The command '.INDENT 0' causes indenting to be stopped.

RINDEN
RIN

Causes the left margin to be indented <arg> spaces from where the left margin was before the command. <arg> may be negative.

UNDENT
UN

This command has effect for only the next line of output text, causing it to be started <arg> spaces to the left of the current left margin. UNDENT commands are given within the scope of INDENT commands, to temporarily suspend the indenting for one line.

SPACE

In the output text insert <arg> blank lines.

SP

In the output text, insert <arg> blank lines. SP tries to be clever about page boundaries: if only one line remains on the current page after a SP command, a PAGE is simulated. Similarly if a SP (for one line only) is executed at the top of the page it is ignored. If you don't like these heuristics, use SPACE instead of SP.

PAGE
PA

In the output text, skip to the top of a new page.

CENTER
CE

Center the next line in the input text.

RIGHT

Similar to CENTER except right justifies text in next line instead of centering it.

LITERA
LIT

Take the next line literally, disregarding a possible '.' at the start. This is the only way to have a text line in the input text with a '.' at the beginning, and is the exception mentioned earlier.

COMMENT
C

Treat the remainder of the command line as a comment.

FILL
FI

This command causes the right margin of the output text to be roughly aligned, as it would be by a typist, without actually right justifying the text.

ADJUST
AD

This command enters a mode in which extra spaces are inserted into the output lines as needed in order to line up the right margin of the output text. ADJUST implies FILL.

NOJUST
NJ

Turn off the adjust mode.

NOFILL
NF

Turn off the fill mode. NOFILL implies NOJUST.

BREAK
BR

Lines of the input text will not be filled across this command.

SINGLE
SS

Cause the output text to be single spaced.

DOUBLE
DS

Cause the output text to be double spaced.

HEADER
HE

The remainder of the command line after the space which follows the command itself is taken as a heading which is to be typed in the upper right hand corner of each page of the output text, followed immediately by the page number (with no intervening spaces).

LINE
L

Set line length to <arg> characters. TJ6 will insert spaces at the beginning of each line if needed to cause lines of the specified length to be centered on the paper.

PAPERW

Set the full line width of the paper to <arg> spaces. This command is used to inform the program of the actual width of the paper being used.

PAPER
PL

Set number of lines to be used on each page to <arg>.

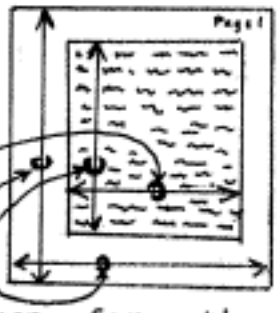
PSIZE

Sets number of lines in full page to <arg>.

Note that PAPERW and LINE set the full paperwidth and the length of the line actually typed. Similarly PSIZE and PAPER set the full length of the page and that number of lines actually typed on (aside from the page-number).

Reasonable parameters are (listed by device):

	KSR 37 (GE TERMINET)	[LPT]	'GE DATANET'
PAPERW	78 (72)	[120]	'44'
LINE	72 (64)	[118]	'44'
PSIZE	66 (66)	[60]	'25'
PAPER	50 (50)	[53]	'18'



For the 'Selectric' use the parameters given for the TERMINET.

DUMMY

Takes single character argument which is replaced by space on output wherever it appears in the output stream. This is useful for leaving multiple spaces for mathematical symbols while in adjust mode. The dummy character is considered as a letter in justifying decisions.

QUOTE

Takes single character argument which is used instead of # to quote characters on input.

TRANS ab,cd,ef ...

Translate on output first character into second character (as many pairs as you like, separated by commas).

EPAGE

Increments page number by <arg> (leaves <arg> blank pages at next page boundary). This command does not cause a break and can therefore appear in the middle of a sentence (on a separate line).

PGN

Switches off output of header and page-number.

DOTS

Causes a line to appear at bottom of page which starts and ends with a '.' - useful for slicing up unperforated paper. The first page of output to DSK: or UTn: will contain only a line starting and ending with '.'.

BSPDIR

Causes back-spaces actually to be output as backspaces.

Some output devices (printer for example) cannot back-space and TJ6 handles back-spaces for them by doing a CR without LF etc. Other devices cannot do a CR without a LF (Execuports and the Selectric for example) - for these one wants to use actual backspaces. Note that some devices (GE datanet) lose in both ways. Also note that TJ6 will not backspace over spaces (to avoid certain justification errors) or the beginning of the line. <Backspace> is _H on GE-Datanet, and <Underscore> is <Backarrow> on some terminals, <Shift O> on GE-Datanet.

FIGURE

Similar to SPACE, but does not break and will delay inserting the <arg> blank lines until the next page if there is not enough space on the current one.

BLOCK

Checks to see if <arg> lines are left on the page, if not simulates a PAGE. Useful for mathematical formulae occupying several lines. This differs from FIGURE in that it breaks and does not continue generating output for the current page.

BEGIN

Start output only at page-number <arg>.

COPY

Produce TJ6-able output which has had some of the processing done to it (such as filling and adjusting). All commands are copied into the output stream. Not all commands are allowed to take effect in this mode - but the output will have about the same paging as the final product. Hopefully TJ6(COPY(FILE)) = TJ6(FILE) . One problem is that words starting with a '.' may find their way to the beginning of a line and be interpreted as commands the next time round.

SECT

Mark next line as section heading - will be output with page number when generating a volume table of

contents - also output as usual otherwise. If a blank line follows this command, a blank line will be inserted in the table of contents.

CONTEN

Output only lines preceeded by a SECT command. The lines output will have no margin and each will have the appropriate page-number.

GENNUM

Add <arg> to the GENNUM counter (initially 0) and output this number in decimal (useful for generating numbers of succesive figures).

GENSET

Reset the GENNUM counter to <arg>.

HYPHEN

Switch off use of hyphens, '-', in breaking lines.

PERIOD

Switch off special treatment of '.' , '!' , '?' , ';' and ':'.

PGNRGH

Causes the page number and heading to be lined up with the outside of the page (set by PAPERW) rather than the typed on region (set by LINE).

THESIS

Causes the page number and heading to appear inside the normally typed area rather than two lines from the top of the page.

TOPM

Causes the top of page margin to be n % of the total available (<arg> of PSIZE - <arg> of PAPER).

SIDM

Causes the left-hand-margin to be n % of the total available (<arg> of PAPERW - <arg> of LINE).

OUTDIR

Direct output of the input file (waits at top of page if to TTY: or Tnm:). Useful if not enough core for TECO or outputting file to 'Selectric' terminal.

SBALL selects table for standard scientific ball.
CBALL selects table for correspondence ball.
TBALL selects table for script ball.
KBALL selects table forkloodgy MAC ball.

HANGUP

Hang up T*10*: line. (This may never return - if so, do a _G)

For some terminals (Selectric and Terminet) a great gain in speed on some type of output can be obtained when using the tabstops. These should be set at intervals of *8* characters (*0,8,16,24* ... for Selectric; *1,9,17,25* ... for Terminet). (Tabs are used for disk and tape output automatically).

TABS

Make use of tabs (implies BSPDIR).

On the Terminets the tabs can be set from the machine:

TABSET

Set tabs on Terminet at intervals of *8*. Wait for it to finish before typing next command. TABSET implies TABS.

On the Selectric a further increase in speed on some types of output is obtained by using the backspaces:

QUICK

Use TAB's, LF's and BS's if possible (instead of CR's and spaces).

TSTOUT

Output all printing ASCII (plus space and rubout) characters as a test.

NOWAIT

Do not wait at top of page for user to type space.

Section III:

Console operation of TJ6

To use TJ6, load it and type as if to MIDAS. If you are not familiar with the MIDAS console command format, read on.

Assume you have prepared a file for processing by TJ6, and it is on device IDV:, with file names FN1 and FN2. Assume further that you want your output on device ODV:, filed under the names FN3 and FN4. Load TJ6 by typing 'TJ6_H', and type:

ODV: FN3 FN4 _ IDV: FN1 FN2 CR

FN1 is required. Omitting FN2 is equivalent to using '>'. Omitting FN4 is equivalent to using (MEMO). Omitting both FN3 and FN4 is equivalent to using FN1 (MEMO). FN3 and FN4 are omitted if ODV is not a file device of course.

Output to DSK: (or UTn:) differs from output to TTY: and LPT: in that pages are separated by form-feeds (FF) and that tabs are used if possible.

The character '_' on the type 37 is equivalent to <BACKARROW> on a type 35 (SHIFT O on a G.E. Console).

The output device need not be a file device, in which case the file names FN3 and FN4 may be omitted. For example, the output device may be LPT:, the line printer; or it may be TTY:, the console you are using. The line printer gives you a quick look at what your output will be like, but has only upper case characters. If you are using a TERMINET, 'Selectric' or type 37 Teletype you may obtain output containing both upper and lower case characters.

Typing '! CR' instead of file-specifications will cause a stand-alone mode to be entered, where HACTRN is flushed and one only occupies 4 blocks of core. 'Q CR' instead of file specifications will cause TJ6 to commit suicide (useful for flushing a stand-alone TJ6).

If instead of the CR ending the file specifications you type a LF, TJ6 will accept input from your TTY: as if it came from the input file until you type a CR followed by space. This is useful for entering commands without modifying the whole input file. If you make an error, type rub-out and re-type the syllable (command without the '.').

Section IV:

Sample TJ6 input:

```
.double
.fill
.center
Introduction
.SP
        This memo describes the TJ6 type justifying program,
which can be used
in the production of memos, such as this one.
In addition, Appendices 1, 2 and 3 of this memo
contain related information
about TECO, the 'Selectric' and the type 37 Teletype,
thus gathering most of the information needed for
producing write-ups into one location.
A sample of input to TJ6 is given in Section IV,
and is in fact the very input
used to produce this page of output.
.sp
        The output from TJ6 may be either justified text,
with the right margin
exactly aligned or it may be 'filled' text,
as in this introduction,
with the right margin only approximately aligned.
The remainder of this memo will be justified.
.sp
        The sections of this memo are:
.nofill
.single

I.      Introduction
II.     Using TJ6
III.    Console operation of TJ6
IV.     Sample TJ6 input

Appendix 1      Inserting lower case letters into the TECO buffer
Appendix 2      The 'Selectric' output device
Appendix 3      How to use a type 37 teletype
.adjust
.pa
```

Appendix 1

Inserting lower case letters into the TECO buffer

Case 1: You are logged in at a TERMINET or type 37 Teletype. In this case, there is no problem. Simply use TECO as usual. When you type a lower case letter, it will be inserted into the buffer as a lower case letter.

Case 2: You are at some other sort of console (type 35 Teletype, G.E. Console). You now are in trouble, since these consoles will neither accept input nor produce output in lower case. In order to get around this problem, a special mode has been added to TECO. To enter this mode, type the command '-1<dollar sign>'. Note that if you type this command by itself, it will appear on your console or paper as '-1$$$', but the first '$' is a real dollar sign, whereas the second two are really alt. modes. TECO responds to this command by retyping its title, and entering -1$ mode. In this mode, letters which are typed are converted to lower case unless they are preceded by a '/', in which case they are left in upper case. Thus the command string 'I/MAC/HAC$$' causes the characters 'MacHac' to be inserted into the buffer. To insert a '/', type '//'.

Since your console cannot output lower case characters any more than it can accept them as input, if you cause the contents of the buffer to be displayed, upper and lower case characters cannot be distinguished. Use of the T command will however output "/"s before capital letters.

To return to the regular TECO mode, type the command "0<dollar sign>".

Appendix 2

The 'Selectric' output device:

The 'selectric' may be any of a number of terminals containing the IBM golf-ball mechanism. It types at 14.8 characters per second (slower in practice because of case changing). To use the Selectric type terminal dial 818-1425 (this may change), put phone on phone coupler - ready and proceed should come on. Output to T10: (this may change) from TJ6 and select one of four internal maps, depending on which golf-ball you mounted:

SBALL - selects table for standard scientific ball.
CBALL - selects table for correspondence ball.
TBALL - selects table for script ball.
KBALL - selects table for kloodgy MAC ball.

Any of these implies BSPDIR. Just before the output starts the proceed light should go out - it should come on again at the end of your file (or after typing a _G). At times the line is not released by the system (i.e. it is busy when you dial it) - after fussing with the dataphone (alternately press second button from left and button on right) and not getting a dial-tone try to get the system to hang up the line by doing a HANGUP (This may never return - if so, do a _G)

THE CHARACTERS ON THE GOLF-BALLS FOR OUR 'SELECTRIC':

ASCII has 94 printing characters:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

The golf-balls have 88 positions - so guess what ...

All maps from the 94 to the 88 (or less) are onto, they certainly are not one-one. If you don't like the mapping use the .TRANS command of TJ6.

Scientific ball: (.SBALL)

misses: [\ ] ^ ` { | } ~ (9)
mapped to: ¦ ¢ ¦ ! ' (|) ¬

extra: ¢ ¬ ¦ (3)
mapped from: \ ~ ¦

Correspondence ball: (.CBALL) (most of the secretaries)

misses: 1 < > \ ^ ` { | } ~ (10)
mapped to: (( ) ¢ ! ' [!] °

extra: ¢ ° (2) + 2 commas + 2 periods
mapped from: \ ~

Script ball: (.SBALL)

misses: ! < > [\ ] ^ ` { | } ~ (12)
mapped to: . ( ) (¢) ± ' (½) ¼

extra: ¢ ½ ¼ ± (4) + 2 commas + 2 periods
mapped from: \ | ~ ^

MAC kludge ball: (.KBALL)

misses: ! " # % & @ ^ { | } ~ (11)
mapped to: . ' ↓ / + → ↑ [¦] ←

extra: ↑ ↓ → ← | (5)
mapped from: ^ # @ ~ ¦

BKPH

Appendix 3

How to use a type 37 Teletype

1. Press the switch labeled 'DATA'.

2. When you get a dial tone, dial the computer. If you are at a 10 character per second Teletype, dial 1474 (this may change). If you are at a 15 character per secnd Teletype, dial 474 (this may change). One can distinguish between 10 and 15 cps type 37 teletypes by the location of the telephone dial. On a 10 cps type 37, the telephone unit is on the table alongside the teletype unit, while on a 15 cps type 37, the telephone unit is vertically mounted below the table, at about knee level.

3. When the computer answers with a 'beep', press the 'RESTORE' button on a 10 cps teletype (on the telephone unit). At a 15 cps teletype, press the 'PROCEDE' button (on the left side, just above the keyboard).

4. Type a _Z ('control Z'). Note that the character set of a type 37 tty is different from that of an ordinary (type 35) tty, and that certain characters must be typed differently as well. The two most important transformations to note: on a

type 37, type <PREFIX> instead of <ALT. MODE>, and type <BACKSPACE> instead of _H ('control H').

5. Typing slowly, log in as usual with <user-name><PREFIX>U. Note that characters that you type will be echoed twice, as the type 37 is still in half-duplex mode. It must be put into full-duplex mode before proceeding. To do this:

6. Type 'lock<BACKSPACE>'

7. When LOCK responds (with a '_'), type '1', then 'q'. This will cause LOCK to initialize the type 37 into full-duplex mode, then to kill itself.

CONGRATULATIONS. You are now ready to use your type 37 as a normal (P) ITS time sharing console.