PHOTON/532/SET
A Text Formatter

by

M.S. Brader

Computer Science Department
University of Waterloo
Waterloo, Ontario

Research Report CS-79-33

PHOTON/532/SET

A Text Formatter

PURPOSE.

Photon/532/Set is a program to allow text to be typeset on MFCF's or Arts's *Photon 532 Fontmaster* phototypesetter. The output is on photographic paper; this is a sample of its appearance. This typesetter has no memory, so before any text is sent to it, the text must be fully formatted and the width of each character computed.

Photon/532/Set accepts as input a sequential ASCII file consisting of text interspersed with "command lines" controlling its output format. The text is formatted as requested, and the result is then outputted in Photon 532 codes. At present the program does not paginate its output; for this manual, that has been done with scissors!

COMMAND LANGUAGE.

Command lines begin with the *control character,* which is initially a dot. (Photon/532/Set resembles a subset of Roff, extended to handle the typesetter's character set.)

The commands are listed below by groups; a summary of all commands follows.

Breaking and Justification.

By default, all text is *filled*—each line contains as many words, delimited by blanks, as can fit into it—and *justified*—the blanks in each line are expanded in width so that both left and right margins are straight.

The **.br** (*break*) command forces a new output line to start. The previous line is not justified: its blanks are printed at the standard minimum width.

If the effect of a .br is desired on every input line, as in setting poetry, the **.nf** (*nofill*) command may be used. When nofill is in effect, a new output line is started whenever a new input line is started, and when the maximum output width is exceeded, but at no other time. There is no justification. The **.fi** (*fill*) command cancels the .nf, resuming filled and justified text.

Output that is filled but not justified, with a ragged right margin and fixed-width blanks, can be obtained when .fi is in effect by the **.nj** (*nojustify*) command, which can be cancelled by .fi or by **.ju** (*justify*). Each of these commands also causes a break when it is issued.

While .nf reproduces the input as closely as possible, one may also want output retaining the original line spacing but with the text aligned by the center or right margin. These are obtained respectively by the **.ce** (*center*) and **.rj** (*right justify*) commands. Each of these acts on the next *input* line of text, also causing a break before and after it. More than one line can also be affected: **.ce n** centers the next n input lines, with breaks before them, between them, and after them; and **.rj n** right justifies n lines.

For example...

```
Opening notrump bids, with a balanced hand:
With 16-18 high-card points, open 1 NT.
With 22-24 high-card points, open 2 NT.
With 25-27 high-card points, open 3 NT.
With 28-29 high-card points, open 4 NT.
.nj
Opening notrump bids, with a balanced hand:
With 16-18 high-card points, open 1 NT.
With 22-24 high-card points, open 2 NT.
With 25-27 high-card points, open 3 NT.
With 28-29 high-card points, open 4 NT.
.nf
(Note how the lines in the
above two parts end on the same words.)
Opening notrump bids, with a balanced hand:
With 16-18 high-card points, open 1 NT.
With 22-24 high-card points, open 2 NT.
With 25-27 high-card points, open 3 NT.
With 28-29 high-card points, open 4 NT.
.fi
Since returning to fill mode automatically turns justification
back on, we don't need a .ju command to get that effect.
Now, every output line is filled with text, except...
.br
where a .br command is used, giving a break.
Similarly, a .nf command always "implies" a .nj to be executed.
```

```
.rj 6
THE CENTRAL POINT
.ce 4
I am the Universe's center.
No subtle skeptics can confound me;
for how can other viewpoints enter,
when the rest is all around me?
by Piet Hein
Poems are normally better set with .nf, of course, but
this example demonstrates how .ce and .nj
break every line until they
run out, and
how the .rj comes back into effect when the .ce runs out,
because these commands are stacked.
However, a .fi, .ju, .nf, or .nj command cancels any stacked commands.
```

...produces this output...

```
Opening notrump bids, with a balanced hand: With 16-18 high-card points, open 1 NT. With 22-24 high-card points, open
2 NT. With 25-27 high-card points, open 3 NT. With 28-29 high-card points, open 4 NT.
Opening notrump bids, with a balanced hand: With 16-18 high-card points, open 1 NT. With 22-24 high-card points, open
2 NT. With 25-27 high-card points, open 3 NT. With 28-29 high-card points, open 4 NT.
(Note how the lines in the
above two parts end on the same words.)
Opening notrump bids, with a balanced hand:
With 16-18 high-card points, open 1 NT.
With 22-24 high-card points, open 2 NT.
With 25-27 high-card points, open 3 NT.
With 28-29 high-card points, open 4 NT.
Since returning to fill mode automatically turns justification back on, we don't need a .ju command to get that effect.
Now, every output line is filled with text, except...
where a .br command is used, giving a break. Similarly, a .nf command always "implies" a .nj to be executed.
                                                    THE CENTRAL POINT
                                                I am the Universe's center.
                                              No subtle skeptics can confound me;
                                              for how can other viewpoints enter,
                                                when the rest is all around me?
                                                                    by Piet Hein
Poems are normally better set with .nf, of course, but this example demonstrates how .ce and .nj break every line until
they run out, and how the .rj comes back into effect when the .ce runs out, because these commands are stacked.
However, a .fi, .ju, .nf, or .nj command cancels any stacked commands.
```

Vertical Spacing.

By default, output lines are placed with their baselines separated by 11½ *points* (1 inch = 72 points). To reset this, use the **.ld n** (*leading, pronounced "ledding"*) command. This will reset the interval between baselines to n points. This text uses 10 points leading.

Also, **.ld +n** will add n to the interval; **.ld -n** will subtract n from the interval; **.ld *n** and **.ld /n** will multiply and divide the interval by n. .ld is not the only command that allows +, -, *, and / in this fashion; there are several others that work the same way. All of them allow all 4 operations; division by 0 produces a 0 result. Decimals are allowed in numeric arguments everywhere, but the value actually used is ruthlessly truncated as appropriate to the command—with .ld, to the ½-point.

Vertical spacing can also be set by the **.ls n** (*line spacing*) command. This has the same syntax as .ld, but instead of taking its operand in points, it takes it in multiples of the last value set by a .ld command. Thus .ls 3 gives triple spacing and .ls 1 returns to single spacing. *Any .ld command causes a .ls 1.*

Also, **.ds** (*double spacing*) and **.ss** (*single spacing*) are respectively equivalent to .ls 2 and .ls 1.

To get blank space inserted once, as before a paragraph, use the **.sp** (*space*) command. Used with no argument, it inserts one blank line—the baselines of the output lines between which it occurs are separated by an extra amount equal to the .ld (*not* .ls) setting. If an argument is given, as in **.sp n**, then n times as much space is inserted.

Each of these commands causes a break. In the case of .ds, .ld, .ls, and .ss, the change goes into effect at the *end* of the first output line following.

For example...

```
(.ld 7.5 now in effect.)
.sp
All very well.
But after all, had he the right to take it?
Granted that Pittsburgh has an atmosphere
rather than an idea,
the attempt to carry away the atmosphere surely borders
on rapacity.
.ld 12
"New Orleans," writes another visitor, "opened
her arms to me and bestowed upon me the soft and
languorous kiss of the Caribbean."
This statement may or may not be true: but in any case
it hardly seems the fair thing to mention it.
```

```
.ds
"Chicago," according to another book of discovery,
"struck me as a large city.
Situated as it is and where it is, it seems destined
to be a place of importance."
.sp
Or here, again, is a form of "impression" that recurs
again and again—"At Cleveland I felt a distinct note
of optimism in the air."
.ss
This same note of optimism is found also
at Toledo, at Toronto—in short, I believe
it indicates nothing more than that someone
gave the visitor a cigar.
.sp .7
Indeed it generally occurs
during the familiar scene in which
the visitor describes his cordial reception in an
unsuspecting American town: thus:
.ld +3.5
"I was met at the station (called in America the depot)
by a member of the Municipal Council driving his own motor car.
After giving me an excellent cigar, he proceeded to drive me
about the town, to various points of interest, including the
municipal abattoir, where he gave me another excellent cigar,
.ld -2
.sp 2
the Carnegie Public Library, the First National Bank
(the manager of which gave me an excellent cigar) and the Second
Congregational Church where I had the pleasure of meeting the pastor.
The pastor, who appeared a man of breadth and culture, gave me
another cigar.
.sp 2
.ls 3
In the evening a dinner, admirably cooked and excellently served,
was tendered to me at a leading hotel."
.ls /2
And of course he took it.
After which his statement that he carried away from the town a feeling
of optimism explains itself:
he had four cigars, the dinner, and half a page of impressions at twenty
cents a word.
```

...produces this output...

(.ld 7.5 now in effect.)

All very well. But after all, had he the right to take it? Granted that Pittsburgh has an atmosphere rather than an idea, the attempt to carry away the atmosphere surely borders on rapacity.
"New Orleans," writes another visitor, "opened her arms to me and bestowed upon me the soft and languorous kiss of the Caribbean." This statement may or may not be true: but in any case it hardly seems the fair thing to mention it.
"Chicago," according to another book of discovery, "struck me as a large city. Situated as it is and where it is, it seems destined to be a place of importance."

Or here, again, is a form of "impression" that recurs again and again—"At Cleveland I felt a distinct note of optimism in the air."

This same note of optimism is found also at Toledo, at Toronto—in short, I believe it indicates nothing more than that someone gave the visitor a cigar.

Indeed it generally occurs during the familiar scene in which the visitor describes his cordial reception in an unsuspecting American town: thus:
"I was met at the station (called in America the depot) by a member of the Municipal Council driving his own motor car. After giving me an excellent cigar, he proceeded to drive me about the town, to various points of interest, including the municipal abattoir, where he gave me another excellent cigar,

the Carnegie Public Library, the First National Bank (the manager of which gave me an excellent cigar) and the Second Congregational Church where I had the pleasure of meeting the pastor. The pastor, who appeared a man of

breadth and culture, gave me another cigar.

In the evening a dinner, admirably cooked and excellently served, was tendered to me at a leading hotel."

And of course he took it. After which his statement that he carried away from the town a feeling of optimism explains itself: he had four cigars, the dinner, and half a page of impressions at twenty cents a word.

Margins.

All margin positions are measured in units of *tenths of an inch.* By default the left margin is at 0 and the right margin at 70, so the line length is seven inches. Owing to hardware restrictions this is the maximum right margin position and hence the maximum line length.

The command **.ll n** (*line length*) will set the line length to n tenths of an inch. As with the .ld command, **.ll +n** and so on are also allowed.

The command **.in n** (*indent*) (or **.in +n,** etc.) will indent the left margin without affecting the right, thus effectively reducing the line length below that set with .ll.

A third command, **.po n** (*page offset*) (or **.po +n,** etc.) moves both margins to the right, thus leaving the line length unaltered. The page offset, indent, and line length values are quite independent—*except:* when a .po or .in forces the right margin beyond 70, the line length is reduced to fit the maximum, and this reduction remains in effect until explicitly cancelled.

These three commands do not cause a break. The margin change takes effect with the beginning of the next output line, and thus can be placed in mid-paragraph.

There is another command that can affect the indent value: **.ti n** (*temporary indent*) will change the indent to n for just the next *output* line. Since this command is normally used at the start of a paragraph, *it does cause a break.* It can also be used in the forms **.ti +n,** etc., with the operator being applied to the indent that *would be* in effect at that point if the command was not there.

There is another form of the temporary indent: **.ti n m** (or **.ti +n m,** etc.) causes a temporary indent for the next m output lines. A .ti command can also be used while another is still in effect, and they will be stacked, with various useful effects possible, such as a "stepped" margin and a "cutout" a specified distance down. *A new .in command cancels any stacked .ti's.*

For example...

```
(.in 5, .ll 65 now in effect.)
.ti +10
Without even having seen the Perry font,
I hardened my heart against Compuscan and turned
my face toward Corporation S.
.ti -3 3
I accepted Tom's offer to run a test batch free
of charge and he sent me typing instructions and
an IBM Perry font bouncing ball in the next mail.
The degree to which the Perry font differed
from the MT/ST as set up at the Institute
of Paper Chemistry required our first major revision
of the Specs.
.in +2.5
Asterisks, dollar signs, plus signs, and percent signs
took the place of the various MT/ST red shifts
(see Figure 2), making the typist's life a bit
.po 5
harder.
But the confusing red brackets at least were gone.
Moreover, she no longer had to worry about forgetting to
insert the page number every time it changed.
.in -10
.po 0
.ll -15
For, entirely without prompting, Vernon
and Dean Stone had just now come around to our way of
thinking and had spontaneously written us to index
by date instead of page.
.ll +1.1
The grapevine had picked up the word that I was looking
for a typist and now a message came back along it to the
effect that Dotty Church, wife of John Church, now
associate high priest
.ti +12.75 5
.ti -12.75 2
to John Brubacher in the Institute of Paper Chemistry's
subterranean computer crypt, was interested
in typing at home.
Just then the whole world changed utterly.
On the 8th of February, 1972, word came that a friend of
Lawrence had matched the National Endowment's offer.
```

```
.ti +3 4
.ti +3 3
.ti +3 2
.ti +3
Not only that, but
when I called Dean Stone with the good news,
he announced that Mrs. Rebecca Logan,
the widow of William van Lennep, editor of Part I,
had given us $5000, and that John Merril
and Elizabeth Knapp had given us $100.
```

...produces this output...

(.in 5, .ll 65 now in effect.)

Without even having seen the Perry font, I hardened my heart against Compuscan and turned my face toward Corporation S.

I accepted Tom's offer to run a test batch free of charge and he sent me typing instructions and an IBM Perry font bouncing ball in the next mail. The degree to which the Perry font differed from the MT/ST as set up at the Institute of Paper Chemistry required our first major revision of the Specs. Asterisks, dollar signs, plus signs, and percent signs took the place of the various MT/ST red shifts (see Figure 2), making the typist's life a bit harder. But the confusing red brackets at least were gone. Moreover, she no longer had to worry about forgetting to insert the page number every time it changed. For, entirely without prompting, Vernon and Dean Stone had just now come around to our way of thinking and had spontaneously written us to index by date instead of page. The grapevine had picked up the word that I was looking for a typist and now a message came back along it to the effect that Dotty Church, wife of John Church, now associate high priest to John Brubacher in the Institute of Paper Chemistry's subterranean computer crypt, was interested in typing at home. Just then the whole world changed utterly. On the 8th of February, 1972, word came that a friend of Lawrence had matched the National Endowment's offer.

Not only that, but when I called Dean Stone with the good news, he announced that Mrs. Rebecca Logan, the widow of William van Lennep, editor of Part I, had given us $5000, and that John Merril and Elizabeth Knapp had given us $100.

Character Sets.

The Photon 532 Fontmaster is capable of setting a range of 32 *fonts* (typefaces) in each of 23 sizes.

The available sizes are 4½, 5, 5½, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 21, 24, 27, 30, 36, 42, 48, 54, 60, and 72 points. The size is measured from the top of the *ascenders* (dfhkl) or the capitals to just below the bottom of the *descenders* (gjpqy). The default size is 10 points; this is 9 points.

To set type in point size n, the **.pt n** (*points*) command is used. The forms **.pt +n,** etc., are also allowed. But note that the result may be truncated—thus if the exact size requested is not available, the next *smaller* size will be selected (except below 4½ points of course). Therefore in 10-point type, the command .pt *5 will give 48-point type and a .pt /5 will then give 9-point, not 10-point.

However, in such cases, what is usually wanted is a *temporary* size change. This is available directly by the **.tp n** (*temporary points*) command, which sets the contents of the next input line at size n points and then returns to the previous size. The forms **.tp +n,** etc., are acceptable, as are **.tp n m** and **.tp +n m,** etc., where the change is applied to the next m *input* lines, and the changes may be stacked as with the .ti command.

The fonts available are numbered from 1 to 32. This default font is Crown Roman, number 20. Fonts can be changed by the commands **.ft n** (*font*) or **.ft +n,** etc. (provided for completeness), and temporarily by **.tf n** (temporary font), **.tf +n,** etc., **.tf n m,** or **.tf +n m,** etc.; the .ft and .tf commands behave exactly analogously to the .pt and .tp.

These commands never cause a break.

When changing the point size, however, the leading is *not* changed automatically to match, so a .ld command is generally necessary. (Setting 72-point type on 12-point leading does not result in legible output!) And the .ld *will,* of course, cause a break.

Since an input newline counts as a blank, it is *impossible*—except by a fudge using the .dc command (see below)—to change font or size without leaving a blank. In the case of a size change, there is absolutely no way to do this at present.

The following fonts are currently set up: (1) **Techno Metabold** (2) Bodoni Roman (3) **Bodoni Bold** (4) Techno Light (5) **Techno Medium Condensed** (6) **Techno Extrabold** (7) **Techno Bold Condensed** (8) **Techno Bold** (9) Pi row (a collection of special characters) (10) *Bodoni Italic* (11) **Bodoni Bold Condensed** (12) Techno Medium (13) **Techno Extrabold Condensed** (14) ***Techno Extrabold Italic*** (15) ***Techno Bold Condensed Italic*** (16) ***Techno Bold Italic*** (17) **Univers Extrabold Wide** (18) Bookman Roman (19) **Poster Bodoni Roman** (20) Crown Roman (21) **Bodoni Campanile** (22) **Univers Medium Extracondensed** (23) **Univers Bold** (24) Univers Light (25) Another pi row (26) *Bookman Italic* (27) ***Poster Bodoni Italic*** (28) **Crown Bold** (29) *Crown Italic* (30) **Univers Extrabold** (31) **Univers Bold Condensed** (32) Univers Light Condensed.

Only the following ASCII characters are acceptable in addition to the alphanumerics:

! # $ % & ' ( ) * + , - . / : ; = ? [ ] _ ‘ |

The single quotes " ‘ " and " ' " correspond to the ASCII grave accent and ASCII single quote (apostrophe) respectively. Double quotes must be built from these characters. Not all characters are available in all fonts, but Photon/532/Set will select an appropriate alternate font when necessary.

If a character is desired that is on the typesetter but does not correspond to any ASCII character, it may be obtained by the **.dc c n** (*define character*) command. This causes the ASCII character c to produce as output the character in disk position n in the current font. Any ASCII character from Control-A (octal 001) to Rubout (octal 177) is acceptable, but if it is already associated with an output character—i.e., is any

alphanumeric, any character in the above list, or any character specified in a previous .dc—then it loses its former association, irrevocably in the cases of "*", "/", and "_" (which are unusual in that they are available in more than one font, but not all). The disk positions run from 1 to 90 and their meanings are tabulated in the appendix.

The .dc command also takes the form **.dc c n m,** which associates the character c strictly with disk position n and font m. (This incidentally provides the fudge mentioned above under .tf, and used again right *here*).

Any characters other than alphanumerics and those listed above (including, of course, all control characters [except Newline] and all characters above Rubout), which have not been defined in a .dc command, will be ignored on input.

For example...

```
.tf 29 2
.tp +4 2
.ld +4
The Underground extends the Central London
and renovates the City & South London
.ld -4
.dc = 62
The first post=war extension was by
.tf 1
Central London
trains, operating over
.tf 1
.tp /1.5
Great Western
tracks, from Wood Lane to Ealing Broadway.
.ft 21
This line, the
.tf 19 2
.tf 27
Ealing & Shepherd's Bush
Railway,
was a Great Western promotion, authorized
.tp *1.6
in
1905
to run from the West London north of Uxbridge Road
to the Acton & Wycombe line at North Acton
.sp
.ld *2
.pt 18
.ft 4
.dc = 46 9
and then south=west to the main line at Ealing;
a short spur was to run to a terminus at
.tp 16
.tf 16
Shepherd's Bush,
with subway connection to the Central London.
.ld /2
.ft 20
.pt 9
This last feature was abandoned when the C.L.
extended to Wood Lane
.tf 28
(for the White City exhibition of 1908),
and the C.L. obtained
.tf 4 9
powers
.tf 12 7
to
.tf 8 5
link
.tf 6 3
with
.tf 1
the
E. & S.B.
north
of
Wood Lane
with running powers to Ealing Broadway.
```

...produces this output...

The Underground extends the Central London and renovates the City & South London

The first post-war extension was by **Central London** trains, operating over Great Western tracks from Wood Lane to Ealing Broadway. This line, the ***Ealing & Shepherd's Bush* Railway**, was a Great Western promotion, authorized in 1905 to run from the West London north of Uxbridge Road to the Acton & Wycombe line at North Acton

and then south=west to the main line at Ealing; a short spur was to run to a terminus at *Shepherd's Bush*, with subway connection to the Central London.

This last feature was abandoned when the C.L. extended to Wood Lane **(for the White City exhibition of 1908)**, and the C.L. obtained powers to **link with the E. & S.B. north** of Wood Lane with running powers to Ealing Broadway.

Control Character Control.

There are two ways to enter input lines that begin with a dot. The **.li** (*literal*) command causes the next input line to be taken literally: if it begins with a dot, there is nevertheless no attempt to treat it as a command. Likewise, **.li n** causes n input lines to be taken literally.

Or if many lines beginning with dots are to be entered, the control character can be changed to something else with the **.cc c** (*control character*) command, which changes the control character to c. If the argument is omitted, it changes it back to the dot.

For example...

```
"Oh, Bunter, don't look so offended.
I mean no harm.
I believe in you, I trust you
.li
... what money have I got?
That'll do.
I knew a man once, Parker, who let a world-famous
poisoner slip through his fingers
.li 2
... because
... the machine on the Underground took nothing but pennies.
.cc ?
?ti +5
"There was a queue at the booking office and the man at
the barrier stopped him,
and while they were arguing about accepting a five-pound note
?tf 29
... which was all he had
... for a twopenny ride to Baker Street, the criminal
had sprung into a Circle train, and was next heard of
?tf 29
in Constantinople
... disguised as an elderly Church of England clergyman
touring with his niece."
?ti +5
?sp
(This example uses all those ...'s because
.cc
.sp
.rj
silly things
?li 2
?sp
?nj
like
?cc *
*li 2
*ce
*ce
*ce
*li
*ce
would never be said by Lord Peter Wimsey...)
*cc
```

...produces this output...

"Oh, Bunter, don't look so offended. I mean no harm. I believe in you, I trust you ... what money have I got? That'll do. I knew a man once, Parker, who let a world-famous poisoner slip through his fingers ... because ... the machine on the Underground took nothing but pennies.

"There was a queue at the booking office and the man at the barrier stopped him, and while they were arguing about accepting a five-pound note ... *which was all he had* ... for a twopenny ride to Baker Street, the criminal had sprung into a Circle train, and was next heard of *in Constantinople* ... disguised as an elderly Church of England clergyman touring with his niece."

(This example uses all those ...'s because .cc .sp .rj silly things ?sp ?nj like *ce *ce *ce would never be said by Lord Peter Wimsey...)

Text Registers.

Sections of text can be stored for later multiple use in *text registers*. Each text register has a name, which can be any sequence of characters, but characters with special meanings in Photon/532/Set are best avoided.

The **.at r** (*assign text*) command creates text register r. All lines following the .at will be saved in the register until the command **.en r** (*end*) is found. The text is saved exactly as given, except for the expansion of *insertion characters* (see below). Commands are *not* executed and the text is saved as ASCII including its original carriage returns—except the final one before the .en, which is dropped.

Note that the register name must be given on both the .at and the .en commands—a register *might* contain a .en for some other register. The register name may be enclosed in parentheses, for compatibility with Roff.

There are two ways to retrieve the contents of a text register. It can be used exactly like a command with no arguments, say **.r,** in which case the .r *expands to* (is replaced by) the contents of the register—and any commands in it will be executed then. Since a carriage return follows the .r, one will be assumed at the end of the register. This method cannot be used with text registers whose names are the same as Photon/532/Set commands, though: .br will still be taken as a break command even if there is a text register "br". It should also be noted that something like .in+5, with *no* space before the +5, will be taken as text register "in+5" if it exists but as .in +5 otherwise. Case is *not* significant in text register names (nor, for that matter, in commands).

The other way to retrieve a text register is by an *insertion character*. This defaults to an ASCII "less-than" sign, alias "opening angle bracket", since that character is not in the Photon 532's character set. But it may be reset by the **.ic c** (*insertion character*) command. This sets the insertion character to c. (If the argument is omitted, it sets it back to the "less-than" sign.) Now if the insertion character is, say, "?", then **?(r)** or (*if* the register name has only 1 character) **?r** expands in exactly the same way as .r—*except*: an insertion character can be used anywhere in a line, not just at the beginning (and not necessarily at the end either, so more text may follow the insertion on the same line).

The parentheses are compulsory when the register name exceeds 1 character: **?(regname)** expands to the contents of register "regname", but **?regname** expands to the contents of register "r" with the string "egname" appended to the last line.

A double insertion character, **??**, expands to a single one, ?, so that insertion character references can be put inside text registers and so that the insertion character can still be used in ordinary text.

The commands .at, .en, and .ic never cause a break; but of course any commands in a text register may do so.

For example...

```
.ic ?
.at h
, with a balanced hand
.en h
.at (bids)
What are the opening notrump bids?h????
With 16 to 18 ??h 1 NT.
With 22 to 24 ??h 2 NT.
With 25 to 27 ??h 3 NT.
With 28 to 29 ??h 4 NT.
.en (bids)
.at h
high-card points, you open with
.en h
.at comment!
Notice the difference in effect between the single
and double insertion characters—one is expanded when the
register "bids" is defined, while the other shrinks
then to a single insertion character and that is expanded
when "bids" is used. Likewise, a quadruple one
will shrink to a double and then to a single...
Incidentally, this example illustrates how all but one of the examples
in this manual were actually produced.
.en comment!
.at demo
??(bids)
.nj
??(bids)
.nf
??(bids)
.en (demo)
.cc +
+nf
+pt -2
+ld -2
+demo
+sp
+tp +2
+tf 16
...yields the following output...
+sp
+fi
+cc .
.demo
.pt +2
.ld +2
```

...produces this output...

```
What are the opening notrump bids, with a balanced hand?
With 16 to 18 high-card points, you open with 1 NT.
With 22 to 24 high-card points, you open with 2 NT.
With 25 to 27 high-card points, you open with 3 NT.
With 28 to 29 high-card points, you open with 4 NT.
.nj
What are the opening notrump bids, with a balanced hand?
With 16 to 18 high-card points, you open with 1 NT.
With 22 to 24 high-card points, you open with 2 NT.
With 25 to 27 high-card points, you open with 3 NT.
With 28 to 29 high-card points, you open with 4 NT.
.nf
What are the opening notrump bids, with a balanced hand?
With 16 to 18 high-card points, you open with 1 NT.
With 22 to 24 high-card points, you open with 2 NT.
With 25 to 27 high-card points, you open with 3 NT.
With 28 to 29 high-card points, you open with 4 NT.
```

...yields the following output...

```
What are the opening notrump bids, with a balanced hand? With 16 to 18 high-card points, you open with 1 NT. With 22 to 24 high-card points, you open with 2 NT. With
25 to 27 high-card points, you open with 3 NT. With 28 to 29 high-card points, you open with 4 NT.
What are the opening notrump bids, with a balanced hand? With 16 to 18 high-card points, you open with 1 NT. With 22 to 24 high-card points, you open with 2 NT. With
25 to 27 high-card points, you open with 3 NT. With 28 to 29 high-card points, you open with 4 NT.
What are the opening notrump bids, with a balanced hand?
With 16 to 18 high-card points, you open with 1 NT.
With 22 to 24 high-card points, you open with 2 NT.
With 25 to 27 high-card points, you open with 3 NT.
With 28 to 29 high-card points, you open with 4 NT.
```

File Insertion.

The command **.so f** (*source*) allows the insertion of another ASCII input file in the midst of the main one. The argument f may be an aft name or a catalog-file description. The inserted file may itself contain .so commands, of course.

There is one problem with this technique, and that is that Photon/532/Set sees one continuous input stream. Therefore, if there should be an error in the input, the line number in the error message will relate to the total number of lines read, not to a line in any specific input file when there is more than one.

It is also possible to intermix with the input actual Photon 532 object codes generated by some other program, as long as they are in the format used by Photon/532/Set itself (6-bit code format, at least one higher bit turned on, delimited at intervals by ASCII linefeeds, octal 012, without any higher bits turned on). This is done by the **.ob f** (*object*) command. The argument f may be an aft name or a catalog-file description. Data will be copied from this file to the output up to and excluding the first stop code, or to end-of-file. The command causes a break, so the insertion will start a new output line, and the next regular line of output will start on a new line and will be unaffected by the insertion.

For example, if file "brian" contains...

```
.at worry
People who worry about computing efficiency might also notice that
the ?(fn) is evaluated twice as often as it need be.
Since the answers are wrong, however, this seems unimportant.
.en (worry)
.at Why
since the function can be integrated by hand.
.en why
```

...and file "elements" contains...

```
.at nxf
is not an exact fraction
.en nxf
.at Fn
function
.en fN
.ic ?
```

...and file "peter" contains...

```
The test that terminates the loop is sensitive to the difference,
and gives us an extra trip around.
.ob moral
.so brian
.ti 5
```

...and file "plauger" contains...

```
Thus 10 times ?1 is not 1.0000..., but 0.9999..., and by extension, when
.1
is added to 1.0 ninety times, the result is not 10.000..., but 9.999....
.so peter
```

...and file "pro/gram/ming" contains...

```
The discrepancy might then have led to further analysis of the program.
```

...and file "style" contains...

```
.at (1)
"0.1"
.en 1
The reason is simple:
.1
.nxf
in a binary machine
```

...while file "moral" contains the object code from...

```
.ft 1
.pt 72
.ld 30
.ce 9999
.dc # 62
#
.tf 21
.tp 21
0.1 times 10.0 is hardly ever 1.0.
#
```

...then this input...

```
.so elements
This should evaluate the
.fn
for X=1.0, 1.1, ..., until X is 10.0, should it not??
But try it, and you will discover that on many machines
it in fact does an extra evaluation, the last one at X=10.09999....
.so style
(in much the same way as 1/3 ?(nxf) in a decimal world);
its nearest representation in most machines happens to
be slightly less than 0.1.
.so plauger
The value of the integral is too high by over 2% for this
.fN
and range.
This error could have been readily caught, ?(why)
.so pro/gram/ming
(?(worry))
```

...produces this output...

This should evaluate the function for X=1.0, 1.1, ..., until X is 10.0, should it not? But try it, and you will discover that on many machines it in fact does an extra evaluation, the last one at X=10.09999.... The reason is simple: "0.1" is not an exact fraction in a binary machine (in much the same way as 1/3 is not an exact fraction in a decimal world): its nearest representation in most machines happens to be slightly less than 0.1. Thus 10 times "0.1" is not 1.0000..., but 0.9999..., and by extension, when "0.1" is added to 1.0 ninety times, the result is not 10.000.... but 9.999.... The test that terminates the loop is sensitive to the difference and gives us an extra trip around.

[redacted]

0.1 times 10.0 is hardly ever 1.0.

[redacted]

The value of the integral is too high by over 2% for this function and range. This error could have been readily caught, since the function can be integrated by hand. The discrepancy might then have led to further analysis of the program. (People who worry about computing efficiency might also notice that the function is evaluated twice as often as it need be. Since the answers are wrong, however, this seems unimportant.)

USAGE.

The MFCF Photon 532 is not usable at this writing. The Arts Photon 532 can be used through a special interface as follows.

Invocation of Photon/532/Set.

On Honeywell TSS, the command **"Photon/532/Set infile obfile"** will take input from file "infile" and place Photon 532 object codes as output on file "obfile". Again, the filenames may be aft names or catalog-file descriptors. "Obfile" will be created if necessary, permanent if a "/" or "$" is in the descriptor, temporary otherwise. **"Infile"** and "obfile" may optionally be preceded respectively by the conventional redirection symbols, the ASCII "less-than" and "greater-than" signs, alias "opening" and "closing angle brackets"; if these are used, the arguments may be in either order.

If "obfile" is omitted, output goes to the terminal, which is practically useless since it is not ASCII; but if "infile" is omitted (and a "greater-than" sign identifies "obfile") input is taken from the terminal, which may be useful. (If this is done inadvertently, the .so command can be used, of course.)

Photon/532/L.

Since the output from Photon/532/Set is not ASCII or anything else readable by mortal man, a special program is provided to interpret it. On TSS, the command **"Photon/532/l obfile dispfile"** will take the presumed Photon/532/Set output from "obfile" and translate it into a readable form on "dispfile". The filenames, again, may be preceded by "less-than" and "greater-than" signs respectively, and may be interchanged if this is done. Again, if "obfile" is omitted, input is from the terminal, which is practically useless; but if "dispfile" is omitted, output goes to the terminal, which may be desirable.

In the output, the widths of all characters are ignored. Most blanks become invisible, since they are created by increasing the width of characters, except at the beginning of lines and in the larger sizes. Photon 532 control codes are translated thus:

" " for a blank not combined with another character
"--" for a "—"
"supX" for a superscript X (where X can be any of 0, 1, ..., 9, $, or c; in some fonts these print differently)
"CR" for a carriage return
"SL" for set standard leading
"AL" for add lead
"RL" for reverse leading
"ZL" for zero leading
"LS" for lens select (this gives the point size in combination with one of the next 2 codes)
"MI" for minifier in
"MO" for minifier out
"DS" for disk level select (this gives the font in combination with one of the next 2 codes)
"DS1" for select disk 1
"DS2" for select disk 2

Photon/532/L is provided purely to facilitate editing and debugging; there is no compulsion to use it before typesetting.

Paper Tape Punching.

Since the Photon 532's originally ran only from paper tape, it was necessary to have a method of punching it; it is described here for those who want it. There is no necessity to punch paper tape now. The files mentioned here are trivial adaptations of existing routines, as altered to handle the object file format described above. The output from Photon/532/Set should be in a permanent file on TSS, with general read permission (or specific read permission for "daemon").

Sign on to Math UNIX (The UNIX userid must have write permission on "/dev/md"), and move the file to UNIX with "hget".

On the Microdata in the UNIX machine room, set switches 1 and 4 down, 2 and 3 up, and the paper tape unit on; then press *Reset* and *Run*. On UNIX, type **"sh /u/msbrader/b"**, which will boot the Microdata in a few seconds.

When the boot is finished and the file has been hgotten, you are ready to punch. If the file is in UNIX file "obfile", type **"/u/msbrader/t ◼obfile | /u/msbrader/pp"** (where "◼" represents a "less-than" sign), and the tape will pour forth, with a date and time header. The command may be repeated with different files as many times as desired. Finally, turn off the paper tape punch and remove the file from UNIX with "rm".

Typesetter and Developer.

The film canister should be mounted in the Photon 532. The correct operating position of the levers is: knife lever (the largest one) and output velvet slot lever *crossed* (knife down, window open), and both levers on the canister to *Open*. **Disks #1 and #2 should be mounted.**

When the machine is powered up, the *No Film* light may come on. This usually only indicates that the door is not properly closed.

If paper tape is being used, it is inserted into the reader (the header appears right side up and to your left as you face the machine) and the reader turned on (toggle switch).

When using the Arts machine and its interface, the toggle switch on the left side of the front panel must be turned on. Then set the Gandalf unit to 66, and make sure the Diablo terminal is at 1200 baud (dial position 6), and has micro-switches 2, 3, 4, 5, 6, and 8 OPEN and 1 and 7 CLOSED. Next, sign on to TSS, and create the "obfile" as above if this hasn't been done yet. Now **"Photon/532/Pass obfile"** will start the transmission. ("Obfile" may be preceded by a "less-than" sign.)

The *TXD* and *RXD* lights on the Gandalf will flash 6 times and then pause.

Now (in either method) press *Clear* and *Rst* to start typesetting.

When finished, hold down *Man Lead* for 10-15 seconds, then open the cabinet and cut the film with the knife lever. Next close the velvet slots with their three levers, and unscrew the canister thumbscrew.

The film is developed the same way as for the Photon Econosetter. The red button turns on the developer. It is advisable to let it run for a while if it has not seen use recently. When ready, pull out a little film and feed it into the developer while placing the cassette under the hood. As soon as the film catches, the hood must be closed. The developer may be turned off momentarily while doing this. Catch the film as it comes out and hang or lay it out. Then turn off the developer with the red button.

Before remounting the canister, return all the levers to operating position and produce about 5 cm of film by holding *Man Lead* down. Thread this into the canister and then press it carefully into place and tighten the screw. *Then* the machine may be turned off.

Appendix A
Summary of Commands

The tabular format of this chart was produced by .ls 0, .ti, and .sp.

In the chart, (r) stands for a register name, optionally enclosed in parentheses; n and m stand for non-negative numbers; +n stands for a non-negative number optionally preceded by any of +, -, *, and /; c stands for any character; f stands for any aft name or catalog-file descriptor, and ■ represents a "less-than" sign.

Command	Break	Default	Meaning
.at (r)	no		Assign text to register R.
.br	Yes		Cause a break.
.cc c	no	c=.	Set control character to c.
.ce n	Yes	n=1	Center next n input lines, break at each.
.dc c n m	no	m=All	Define character c as disk position n in font m.
.ds	Yes		Equivalent to .ls 2.
.en (r)	no		Terminate .at (r) command.
.fi	Yes		Cancel .nf and .nj.
.ft +n	no		Change output to font n.
.ic c	no	c=■	Set insertion character to c.
.in +n	no		Indent left margin n/10 inches.
.ju	Yes		Cancel .nj.
.ld +n	Yes		Set leading to n points.
.li n	no	n=1	Take next n input lines literally.
.ll +n	no		Set line length to n/10 inches.
.ls +n	Yes		Set line spacing to n (times .ld setting).
.nf	Yes		Cause a break on every line.
.nj	Yes		Turn off justification (right margin ragged).
.ob f	Yes		Insert object codes from file f.
.po +n	no		Set page offset to n/10 inches.
.pt +n	no		Change output to n points.
.rj n	Yes	n=1	Right justify next n input lines, break at each.
.so f	no		Insert text from file f.
.sp n	Yes	n=1	Leave n blank lines (times .ld setting).
.ss	Yes		Equivalent to .ls 1.
.tf +n m	no	m=1	Change output to font n for next m input lines.
.ti +n m	Yes	m=1	Indent next m output lines by n/10 inches.
.tp +n m	no	m=1	Change output to size n points for next m input lines.

Appendix B
Character Sets

The following fonts are available:

1 **Techno Metabold**
2 Bodoni Roman
3 **Bodoni Bold**
4 Techno Light
5 **Techno Medium Condensed**
6 **Techno Extrabold**
7 **Techno Bold Condensed**
8 **Techno Bold**
9 *pi row*
10 *Bodoni Italic*
11 **Bodoni Bold Condensed**
12 Techno Medium
13 **Techno Extrabold Condensed**
14 ***Techno Extrabold Italic***
15 ***Techno Bold Condensed Italic***
16 ***Techno Bold Italic***
17 **Univers Extrabold Wide**
18 Bookman Roman
19 **Poster Bodoni Roman**
20 Crown Roman
21 **Bodoni Campanile**
22 **Univers Medium Extracondensed**
23 **Univers Bold**
24 Univers Light
25 *pi row*
26 *Bookman Italic*
27 ***Poster Bodoni Italic***
28 **Crown Bold**
29 *Crown Italic*
30 **Univers Extrabold**
31 **Univers Bold Condensed**
32 Univers Light Condensed

in the following point sizes:

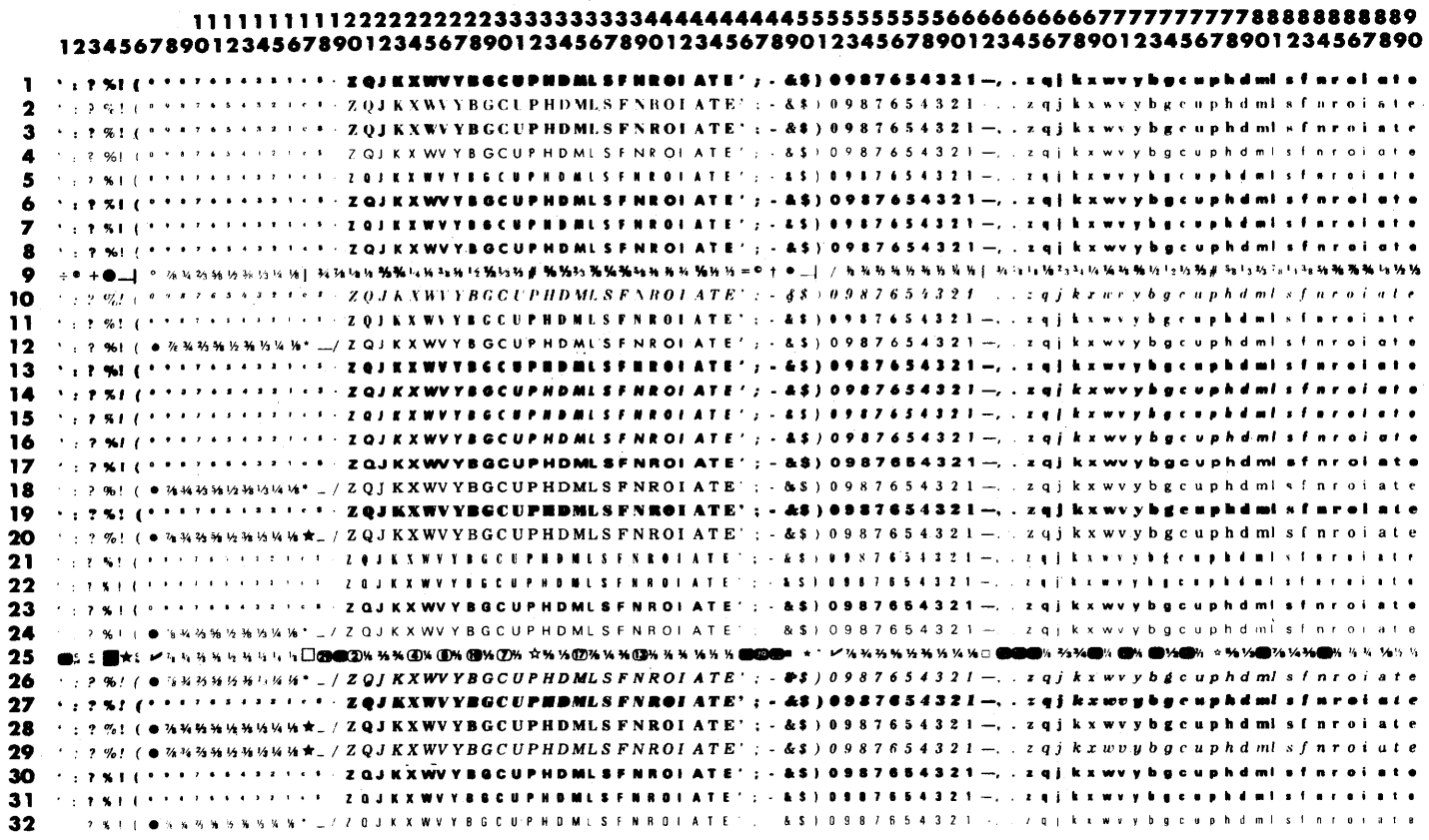
4½, 5, 5½, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 21, 24, 27, 30, 36,

42, 48, 54, 60, 72.

As this shows, certain sizes do not align properly at this writing.

Appendix C
Disk Positions

This was not done by Photon/532/Set.



Appendix D
Errors

All of these messages are prefixed with the input line number where they occurred (but see under "File Insertion", above), and many of them contain phrases in addition to the condensations given here.

".at command not followed by register name" *You tried to use a .at with no argument.*

".at unclosed at end of file" *You used a .at but didn't follow it with a .en with a matching argument. Everything in the input from the .at on went into the text register!*

".ob treated as .br" *The file specified in a .ob couldn't be opened.*

".so ignored" *The file specified in a .so couldn't be opened.*

"Centered line too long, broken" *A line of text to be centered exceeded the line length.*

"Closing) missing after [char]([char] *is the insertion character. An insertion character must be followed by a 1-character register name or by a register name in surrounded by "(" and ")".*

"File or register ends with insertion character" *See previous message.*

"Illegal character argument" *The command expects a 1-character argument, not a string.*

"Illegal disk position" *You tried to .dc a character with a disk position other than 1 through 90.*

"Illegal font" *You tried a .dc, .ft, or .tf with a font other than 1 through 32.*

"Illegal numeric argument" *The command expects a number, not a string.*

"Indent absurd" *An indent must be non-negative and less than the line length.*

"Invalid SETFNT" *Internal error.*

"Invalid SETINC" *Internal error.*

"Invalid SETSIZ" *Internal error.*

"Invalid argument character" *Only characters from Control-A to Rubout (octal 1 through 177) may be defined in a .dc command.*

"Leading absurd" *The leading must be non-negative, but no more than 144 points (2 inches).*

"Length absurd" *A line length must be non-negative and no more than the maximum physical line length (70, i.e. 7 inches) minus the page offset.*

"Length must be reduced" *See previous message. The page offset has been increased, and the line length is now too large.*

"Line ends with insertion character" *See "Closing) missing..." above.*

"Missing argument" *The .dc command requires 2 or 3 arguments.*

"Offset absurd" *The page offset must be non-negative, but less than the maximum physical line length (70, i.e. 7 inches).*

"Temp indent absurd" *See "Indent absurd" above.*

"Text found after register name" *The argument of a .at command began with a "(", but contained more characters after the first ")".*

"Text found following argument" *This command takes only 1 argument, but more text was found on the same line.*

"Undefined register" *A line of input began with the control character and a string of characters that was neither a defined text register nor a valid command.*

"[char] untranslatable" *There is no predefined output equivalent for that character, nor has it appeared in a valid .dc command.*

"[number] points illegal" *The legal point sizes are listed in Appendix B.*

"[operator] ignored" *This command does not allow numbers to be preceded by +, -, *, or /.*