

This is a list of all corrections made to *Computers & Typesetting*, Volumes A–E, between the date of publication (May, 1986) and 15 June 1987. It also includes corrections made to the softcover version of *The T_EXbook*, beginning with the sixth printing (January 1986); these are the same as corrections to Volume A. Corrections to the softcover version of *The METAFONTbook* are the same as corrections to Volume C.

Page A7, fourth line from the bottom (6/28/86)

since control sequences of the second kind always have exactly one symbol after

Page A35, second-last line (1/31/87)

He may run who reads.

— HABAKKUK 2:2 (c. 600 B.C.)

He that runs may read.

Page A43, lines 8–9 (8/23/86)

of Appendix B, which defines % to be a special kind of symbol so that you can use it for comments, defines the control sequence \% to mean a percent sign.

Page A45, lines 10–13 (8/23/86)

T_EX adds 64. Hence code 127 can be typed ^^?, and the dangerous bend sign can be obtained by saying {\manual^^?}. However, you must change the category code of character 127 before using it, since this character ordinarily has category 15 (invalid); say, e.g., \catcode'\^^?=12. The ^^ notation is different from \char, because ^^

Page A76, line 7 (8/23/86)

and extra space; for example, these quantities are 3.33333 pt, 1.66666 pt, 1.11111 pt,

Page A83, bottom line (5/19/87)

[This line should be flush right.]

Page A111, 7th-last line, right-hand column (2/15/87)

if $b = 10000$ and $-10000 < p < 10000$ and $q < 10000$;

Page A117, second-last line (6/10/87)

marks; sometimes also $\$|\$$ (||). You can say, e.g., '\footnote\dag{...}'.

2 Bugs in Computers & Typesetting

Page A124, lines 6–11 (2/26/87)

of insertion; an additional ‘`\penalty-10000`’ item is assumed to be present at the end of the vertical list, to ensure that a legal breakpoint exists.) Let u be the natural height plus depth of that least-cost box, and let r be the penalty associated with the optimum breakpoint. Decrease g by uf , and increase q by r . (If `\tracingpages=1`, the log file should now get a cryptic message that says ‘`% split n to v, u p= r` ’. For example,

```
% split254 to 180.2,175.3 p=100
```

Page A158, lines 6–8 (2/20/87)

the second atom, which has subscript i ; the superscripts are empty except for the last atom, whose superscript is $\bar{n} + \bar{1}$. This superscript is itself a math list consisting of one atom, whose nucleus is $n + 1$; and that nucleus is a math list consisting of three atoms.

Page A171, line 20 (1/26/86)

will be surrounded by more space than there would be if that subformula were enclosed

Page A176, line 1 (8/23/86)

You can insert ‘`\noalign{<vertical mode material>}`’ just after any `\cr` within

Page A248, line 17 (6/17/86)

‘`&`’ or ‘`\span`’ or ‘`\cr`’, it needs some way to decide which alignment is involved.

Page A249, line 20 (6/17/86)

line (see Chapter 8). If you don’t want a `\cr` at the end of a certain line, just type

Page A276, line 19 (1/27/86)

```
| \font<control sequence>(equals)<file name>(at clause)
| <global assignment>
```

[The bottom line of p. 276 will now move to the top of p. 277.]

Page A277, lines 31–32 (1/27/86)

```
<font assignment> → \fontdimen<number><font>(equals)<dimen>
```

Page A286, sixth-last line (4/28/87)

`\sfcode` table as described in Chapter 12; characters numbered 128 to 255 set the

Page A287, line 19 (2/15/87)

- `\-`. This “discretionary hyphen” command is defined in Appendix H.

Page A292, lines 9–10 (2/15/87)

■ \-. This command is usually equivalent to ‘\discretionary{-}{-}{}’; the ‘-’ is therefore interpreted as a hyphen, not as a minus sign. (See Appendix H.)

Page A308, lines 25–26 (6/1/87)

```
\def\appendroman#1#2#3{\edef#1{\csname
\expandafter\gobble\string#2\romannumeral#3\endcsname}}
```

Page A312, lines 10–14 (8/23/86)

12.11. The interline glue will be zero, and the natural height is $1 + 1 - 3 + 2 = 1$ pt (because the depth of `\box2` isn’t included in the natural height); so the glue will ultimately become `\vskip-1pt` when it’s set. Thus, `\box3` is 3 pt high, 2 pt deep, 4 pt wide. Its reference point coincides with that of `\box2`; to get to the reference point of `\box1` you go up 2 pt and right 3 pt.

Page A312, line 21 (8/23/86)

up 4 pt to get to the upper left corner of `\box4`; then down -1.6 pt, i.e., up 1.6 pt, to

Page A319, line 20 (31/3/87)

make ordinary periods act like `\cdot` symbols: Just define `\mathcode‘.` to be “0201,

Page A328, lines 18–19 (5/14/87)

not performed while the expansion is taking place, and the control sequences following `\def` are expanded; so the result is an infinite string

```
A\def A\def A\def A\def A\def A\def A\def A\def A...
```

Page A329, lines 14–15 (8/23/86)

20.5. The `##` feature is indispensable when the replacement text of a definition contains other definitions. For example, consider

Page A356, lines 6–7 (1/30/87)

```
\spaceskip=.3333em \xspaceskip=.5em\relax}
\def\tttraggedright{\tt\rightskip=0pt plus2em\relax}
```

Page A356, line 33 (6/1/87)

```
\vbox to.2ex{\hbox{\char'26}\vss}\hidewidth}}
```

Page A357, tenth-last line (10/13/86)

```
\let\sp=^ \let\sb=_      {\catcode‘\_=\active \global\let\_=_}
```

Page A357, third-last and second-last lines (2/17/87)

```
\def\pr@m@s{\ifx'\next\let\nxt\pr@@@s \else\ifx'\next\let\nxt\pr@@@t
\else\let\nxt\egroup\fi\fi \next}
```

Page A364, fifth-last line (1/30/87)

```
\def\fmtname{plain}\def\fmtversion{2.3} % identifies the current format
```

Page A368, bottom line (2/26/86)

that includes the symbols \leftarrow , \downarrow , \neq , \leq , and \geq , and he finds that this makes it much more

Page A396, line 13 (8/23/86)

```
\hyphenpenalty=10000 \exhyphenpenalty=10000
```

Page A414, line 10 (3/4/86)

```
\font\titelfont=cmssdc10 at 40pt % titles in chapter openings
```

Page A427, line 7 (2/23/86)

the author's book *Computer Modern Typefaces*.)

Page A428, lines 18–20 (6/15/87)

The first eight of these all have essentially the same layout; but `cmr5` needs no ligatures, and many of the symbols of `cmti10` have different shapes. For example, the ampersand becomes an ‘E.T.’, and the dollar changes to pound sterling:

Page A434, lines 25–28 (8/17/86)

from `\nu` (ν). Similarly, `\varsigma` (ς) should not be confused with `\zeta` (ζ). It turns out that `\varsigma` and `\upsilon` are almost never used in math formulas; they are included in plain \TeX primarily because they are sometimes needed in short Greek citations (cf. Appendix J).

Page A447, line 32 (6/1/87)

ters also affect mathematical typesetting: dimension parameters `\delimitershortfall`

Page A455, new paragraph to follow line 9 (2/15/87)



The control sequence `\-` is equivalent to `\discretionary{\char h}{-}{-}`, where h is the `\hyphenchar` of the current font, provided that h lies between 0 and 255. Otherwise `\-` is equivalent to `\discretionary{}{-}{-}`.

Page A457, left column, fifth-last line (2/17/87)

155, 201, 305, 324, 357, 394–395;

Page A458, left column, line 6	(2/15/87)
*\- (discretionary hyphen), 95, 283, 287, 292, 455 .	
Page A458, left column, near the bottom	(5/19/87)
! (exclamation point), 51, 72 , 73, 75, 169 . [This saves a line that otherwise would make the index too long on page 481!]	
Page A458, right column, line 10	(11/27/86)
~ (tilde), 38, 51, 343, 353 ; <i>see also</i> ties.	
Page A458, right column	(6/14/87)
*\accent (general accent), 9, 54, 86, 283, 286 .	
Page A461, entry for boxes	(3/16/87)
boxes, 63–67, 77–83, 221–229.	
Page A461, entry for \centering	(1/28/86)
\centering, 347 , 348, 362.	
Page A462, entry for ⟨code assignment⟩	(1/27/86)
⟨code assignment⟩, 277 .	
Page A464, left column, line 3	(2/15/87)
discretionary hyphens, 28, 95–96, 453, 455 .	
Page A465, right column, line 8	(5/3/87)
expansion of expandable tokens, 212–216, 238,	
Page A466, entry for \font, second line	(1/27/86)
271, 276 .	
Page A466, new entry	(2/3/87)
⟨fontdef token⟩, 271 .	
Page A467, entry for \hideskip	(1/28/86)
\hideskip, 347 , 348, 354.	
Page A468, left column line 2	(2/15/87)
351, 395, 414 , 454, 455.	

6 *Bugs in Computers & Typesetting*

Page A470, entry for <code>manfnt</code>	(1/15/86)
<code>manfnt</code> , 44, 408, 414.	
Page A471, entry for <code>\medbreak</code>	(10/13/86)
<code>\medbreak</code> , 111, 113, 353 , 355 , 419 , 422 .	
Page A471, entry for <code>\moveright</code>	(2/27/87)
<code>*\moveright</code> , 80–81, 221 , 282 .	
Page A471, entry for Mozart, second line	(3/19/86)
Gottlieb (= Theophilus = Amadeus), 409.	
Page A472, the entry for <code>\not</code>	(2/12/87)
[The overprinting here is intentional, since <code>\not</code> is a character of width zero. More than a dozen people have reported this as an error, but it is not!]	
Page A477, entry for <code>\span</code>	(5/3/87)
<code>*\span</code> , 215, 238, 243 , 244 , 245 , 248, 249, 282, 330 , 385.	
Page A479, entry for ties, second line	(11/27/86)
173 , 353 , 404 .	
Page A480, changes to various entries	(6/14/87)
<code>*\underline</code> , 130–131 , 141, 291, 443 . <code>*\unhbox</code> , 120, 283, 285 , 354 , 356 , 361 , 399 . <code>*\unhcopy</code> , 120, 283, 285 , 353 . <code>*\unkern</code> , 280. <code>*\unpenalty</code> , 280 . <code>*\unskip</code> , 222–223 , 280 , 286, 313 , 392 , 418–419 . <code>*\unvbox</code> , 120, 254, 282 , 286, 354 , 361 , 363 , 364 , 392 , 399 , 417 . <code>*\unvcopy</code> , 120, 282 , 286, 361 . <code>*\vadjust</code> , 95, 105, 109, 110, 117, 259, 281 , 393, 454. <code>*\valign</code> , 249, 283, 285–286 , 302, 335 , 397 . <code>*\vcenter</code> , 150–151, 159, 170, 193, 222, 242. <code>*\vfil</code> , 71, 72 , 111, 256, 281, 286, 417. <code>*\vfill</code> , 24, 25, 71, 72 , 256–257, 281, 286. <code>*\vfilneg</code> , 72 , 111, 281, 286. <code>\voidb@x</code> , 347 , 348.	
Page A481, left column	(6/14/87)
<code>*\vss</code> , 71, 72 , 255 , 281, 286.	

 Volume B, in general

(7/28/86)

[A number of entries were mistakenly omitted from the mini-indexes on the right-hand pages. Here is a combined list of all the missing items; you can mount it inside the back cover, say, as a secondary mini-index when the first one fails...]

<i>active_base</i> = 1, §222.	<i>hyf_num</i> : array , §921.	<i>prefixed_command</i> : procedure ,
<i>aux</i> = macro, §213.	<i>index</i> = macro, §302.	§1211.
<i>begin_name</i> : procedure , §515.	<i>inf</i> : boolean , §448.	<i>prev_depth</i> = macro, §213.
<i>big_switch</i> = 60, §1030.	<i>init_col</i> : procedure , §788.	<i>prev_graf</i> = macro, §213.
<i>choice_node</i> = 15, §689.	<i>init_span</i> : procedure , §787.	<i>prev_prev_r</i> : <i>pointer</i> , §830.
<i>cur_boundary</i> : 0 .. <i>save_size</i> , §271.	<i>input_ln</i> : function , §31.	<i>print_err</i> = macro, §73.
<i>cur_c</i> : <i>quarterword</i> , §724.	<i>interaction</i> : 0 .. 3, §73.	<i>r</i> : <i>trie_pointer</i> , §960.
<i>cur_group</i> : <i>group_code</i> , §271.	<i>limit</i> = macro, §302.	<i>reconstitute</i> : function , §906.
<i>cur_i</i> : <i>four_quarters</i> , §724.	<i>line_width</i> : <i>scaled</i> , §830.	<i>resume_after_display</i> : procedure ,
<i>cur_level</i> : <i>quarterword</i> , §271.	<i>macro_call</i> : procedure , §389.	§1200.
<i>do_extension</i> : procedure , §1348.	<i>main_control</i> : procedure , §1030.	<i>save_ptr</i> : 0 .. <i>save_size</i> , §271.
<i>dvi_buf</i> : array , §595.	<i>mem</i> : array , §116.	<i>save_stack</i> : array , §271.
<i>dvi_gone</i> : <i>integer</i> , §595.	<i>mem_bot</i> = 0, §12.	<i>scan_dimen</i> : procedure , §448.
<i>dvi_limit</i> : <i>dvi_index</i> , §595.	<i>mem_end</i> : <i>pointer</i> , §118.	<i>scan_math</i> : procedure , §1151.
<i>dvi_offset</i> : <i>integer</i> , §595.	<i>mem_top</i> = macro, §12.	<i>short_display</i> : procedure , §174.
<i>dvi_ptr</i> : <i>dvi_index</i> , §595.	<i>mlist_to_hlist</i> : procedure , §726.	<i>show_node_list</i> : procedure , §182.
<i>end_graf</i> : procedure , §1096.	<i>mode</i> = macro, §213.	<i>start</i> = macro, §302.
<i>error</i> : procedure , §82.	<i>mode_line</i> = macro, §213.	<i>state</i> = macro, §302.
<i>error_stop_mode</i> = 3, §73.	<i>more_name</i> : function , §516.	<i>str_pool</i> : packed array , §39.
<i>font_base</i> = 0, §12.	<i>mu</i> : boolean , §448.	<i>str_ptr</i> : <i>str_number</i> , §39.
<i>font_info</i> : array , §549.	<i>name</i> = macro, §302.	<i>str_start</i> : array , §39.
<i>get_token</i> : procedure , §365.	<i>nest</i> : array , §213.	<i>tail</i> = macro, §213.
<i>glue_base</i> = 2626, §222.	<i>off_save</i> : procedure , §1064.	<i>trap_zero_glue</i> : procedure , §1229.
<i>half_buf</i> : <i>dvi_index</i> , §595.	<i>open_log_file</i> : procedure , §534.	<i>trie</i> : array , §921.
<i>handle_right_brace</i> : procedure ,	<i>output_active</i> : boolean , §989.	<i>trie_char</i> = macro, §921.
§1068.	<i>p</i> : <i>pointer</i> , §498.	<i>trie_link</i> = macro, §921.
<i>hash_base</i> = 258, §222.	<i>param_stack</i> : array , §308.	<i>trie_op</i> = macro, §921.
<i>head</i> = macro, §213.	<i>pool_file</i> : <i>alpha_file</i> , §50.	<i>vlist_out</i> : procedure , §629.
<i>hyf_distance</i> : array , §921.	<i>pool_ptr</i> : <i>pool_pointer</i> , §39.	<i>write_loc</i> : <i>pointer</i> , §1345.
<i>hyf_next</i> : array , §921.		

 Volume B, in general

(4/6/87)

[The percent signs in all the comments (for example, on pages 7 and 50) are in the wrong font! Change '%' to '%'.]

 Page Bvi, bottom line, and top line of next page

(10/12/86)

puter Science Report 1097 (Stanford, California, April 1986), 146 pp. *The WEB programs for four utility programs that are often used with T_EX: POOLtype, T_FtoPL, PLtoT_F, and DVItype.*

 Page B2, line 32

(4/22/87)

`define banner ≡ `This_is_TeX,Version_2.2` { printed when TEX starts }`

Page B7, new line after line 25 (1/28/87)

```
if max_in_open ≥ 128 then bad ← 6;
```

Page B13, first three lines (4/7/87)

The ‘*name*’ parameter, which is of type ‘**packed array** [*any*] of *char*’, stands for the name of the external file that is being opened for input or output. Blank spaces that might appear in *name* are ignored.

Page B14, line 30 (4/7/87)

31. The *input_ln* function brings the next line of input from the specified file into available

Page B18, line 30 (5/22/86)

```
str_ptr: str_number; { number of the current string being created }
```

Page B21, first line of mini-index, right column (6/14/87)

```
pool_name = "string", §11.
```

Page B34, lines 5–6 (6/14/87)

to delete a token, and/or if some fatal error occurs while T_EX is trying to fix a non-fatal one. But such recursion is never more than two levels deep.

Page B55, lines 12–13 (4/21/87)

```
if r = p then if rlink(p) ≠ p then ⟨ Allocate entire node p and goto found 129 ⟩;
```

Page B57, lines 25–28 (6/14/87)

The first of these has *font* = *font_base*, and its *link* points to the second; the second identifies the font and the character dimensions. The saving feature about oriental characters is that most of them have the same box dimensions. The *character* field of the first *char_node* is a “*charext*” that distinguishes between graphic symbols whose dimensions are identical for typesetting purposes. (See the METAFONT manual.) Such an extension of T_EX would not be difficult; further details are left to the reader.

Page B58, second line of section 136 (7/23/86)

the values corresponding to ‘\hbox{’’. The *subtype* field is set to *min_quarterword*, since that’s

Page B66, lines 2–8 (4/21/87)

location is more efficient than dynamic allocation when we can get away with it. For example, locations *mem_bot* to *mem_bot* + 3 are always used to store the specification for glue that is ‘Opt plus Opt minus Opt’. The following macro definitions accomplish the static allocation by giving symbolic names to the fixed positions. Static variable-size nodes appear in locations *mem_bot* through *lo_mem_stat_max*, and static single-word nodes appear in locations *hi_mem_stat_min* through *mem_top*, inclusive. It is harmless to let *lig_trick* and *garbage* share the same location of *mem*.

Page B67, line 23 (4/13/87)

{ previous *mem_end*, *lo_mem_max*, and *hi_mem_min* }

Page B71, line 17 (4/15/87)

begin while *p* > *mem_min* **do**

[Now *null* can be removed from the mini-index.]

Page B74, line 24 (4/15/87)

procedure *show_node_list*(*p* : integer); { prints a node list symbolically }

Page B74, line 33 (4/15/87)

while *p* > *mem_min* **do**

Page B84, line 12 (2/15/87)

define *relax* = 0 { do nothing (\relax) }

Page B86, third line of section 210 (8/23/86)

that their special nature is easily discernible. The “expandable” commands come first.

Page B88, line 23 (5/22/86)

procedure *print_mode*(*m* : integer); { prints the mode represented by *m* }

Page B93, lines 3–4 (8/17/86)

In the first region we have 128 equivalents for “active characters” that act as control sequences, followed by 128 equivalents for single-character control sequences.

Page B130, ninth-last line (5/7/87)

This variable has six possible values:

Page B151, line 9 (4/22/87)

```
begin if (end_line_char < 0)  $\vee$  (end_line_char > 127) then incr(limit);  
if limit = start then { previous line was empty }
```

Page B160, lines 17–20 (7/28/86)

389. After parameter scanning is complete, the parameters are moved to the *param_stack*. Then the macro body is fed to the scanner; in other words, *macro_call* places the defined text of the control sequence at the top of T_EX's input stack, so that *get_next* will proceed to read it next.

Page B200, top line (5/5/87)

495. When we begin to process a new `\if`, we set $if_limit \leftarrow if_code$; then if `\or` or `\else` or `\fi`

Page B217, lines 15–16 (6/14/87)

DVI format.

Page B224, lines 4–7 of section 560 (10/22/86)

name and area strings *nom* and *aire*, and the “at” size *s*. If *s* is negative, it's the negative of a scale factor to be applied to the design size; $s = -1000$ is the normal case. Otherwise *s* will be substituted for the design size; in this case, *s* must be positive and less than 2048 pt (i.e., it must be less than 2^{27} when considered as an integer).

Page B224, second-last line (4/28/87)

```
done: if file_opened then b_close(tfm_file);  
read_font_info  $\leftarrow g$ ;
```

Page B255, mini-index at the bottom (4/15/87)

mag = macro, §236.

Page B257, lines 11–13 (6/14/87)

```
if  $c \geq qi(128)$  then dvi_out(set1);  
dvi_out(qo(c));
```

Page B260, lines 7–8 (4/15/87)

In the case of *c_leaders* (centered leaders), we want to increase *cur_h* by half of the excess space not occupied by the leaders; and in the case of *x_leaders* (expanded leaders) we increase

Page B267, mini-index at the bottom (4/15/87)

cur_s
: *integer*, §616. *mag* = macro, §236. *pop* = 142, §586.

Page B271, line 10 (8/23/86)

which will be ignored in the calculations because it is a highly negative number.

Page B285, lines 23 and 24 (5/4/87)

the current string would be ‘ $\sqrt{a^{\mathinner{b_{c\over x+y}}}}$ ’.

Page B296, lines 3–5 (5/8/87)

box b and changes it so that the new box is centered in a box of width w . The centering is done by putting `\hss` glue at the left and right of the list inside b , then packaging the new box; thus, the actual box might not really be centered, if it already contains infinite glue.

Page B346, line 19 (5/19/87)

`pass_number`: *halfword*; { the number of passive nodes allocated on this pass }

Page B350, lines 36 and 37 (1/28/87)

`v`: *pointer*; { points to a glue specification or a node ahead of `cur_p` }

`t`: *integer*; { node count, if `cur_p` is a discretionary node }

Page B353, lines 8–22 (1/28/87)

```

s ← cur_p;
if break_type > unhyphenated then if cur_p ≠ null then
  ⟨ Compute the discretionary break_width values 840 ⟩;
while s ≠ null do
  : [as before, but indented one less notch]
end;
```

Page B354, line 6 (1/28/87)

will be the background plus l_1 , so the length from `cur_p` to `cur_p` should be $\gamma + l_0 + l_1 - l$, minus the length of nodes that will be discarded after the discretionary break.

Page B354, lines 12–18 (1/28/87)

```

begin t ← replace_count(cur_p); v ← cur_p; s ← post_break(cur_p);
while t > 0 do
  begin decr(t); v ← link(v); ⟨ Subtract the width of node v from break_width 841 ⟩;
  end;
while s ≠ null do
  begin ⟨ Add the width of node s to break_width and increase t, unless it's discardable 842 ⟩;
```

Page B354, new line after line 21 (1/28/87)

```

if t = 0 then s ← link(v); { more nodes may also be discardable after the break }
```

Page B354, lines 26–34 (1/28/87)

[Change ‘s’ to ‘v’ throughout this section (8 times).]

Page B354, line 9 from the bottom (1/28/87)

842. ⟨ Add the width of node *s* to *break_width* and increase *t*, unless it’s discardable 842 ⟩ ≡

Page B355, lines 1–3 (1/28/87)

```

hlist_node, vlist_node, rule_node: break_width[1] ← break_width[1] + width(s);
kern_node: if (t = 0) ∧ (subtype(s) ≠ acc.kern) then t ← -1 { discardable }
           else break_width[1] ← break_width[1] + width(s);
othercases confusion("disc2")
endcases;
incr(t)

```

Page B355, patches to mini-index at bottom (1/28/87)

```

acc.kern = 2, §155.
incr = macro, §16.
t: integer, §830.
v: pointer, §830.

```

Page B372, lines 12–14 (1/28/87)

```

⟨ Change discretionary to compulsory and set disc.break ← true 882 ⟩
else if (type(q) = math_node) ∨ (type(q) = kern_node) then width(q) ← 0;

```

Page B380, fifth-last line (5/7/87)

b and c, the two patterns with and without hyphenation are a b - cd ef and a bc de f. Thus the

Page B386, lines 2–4 (5/21/87)

hyphenation, T_EX first looks to see if it is in the user’s exception dictionary. If not, hyphens are inserted based on patterns that appear within the given word, using an algorithm due to Frank M. Liang.

Page B397, line 28 (5/21/87)

$h = z - c$. It follows that location *trie_max* will never be occupied in *trie*, and we will have

Page B415, the mini-index (4/6/87)

[Delete the spurious entry for ‘c’.]

Page B419, mini-index entry for *c* (4/6/87)

```

c: integer, §994.

```

Page B422, line 24	(8/23/86)
<i>prev-p</i> : <i>pointer</i> ; { predecessor of <i>p</i> }	
Page B435, line 16	(10/12/86)
$width(p) \leftarrow font_info[k].sc;$ { that's <i>space(f)</i> } $stretch(p) \leftarrow font_info[k+1].sc;$ { and <i>space_stretch(f)</i> } $shrink(p) \leftarrow font_info[k+2].sc;$ { and <i>space_shrink(f)</i> }	
[And the mini-index gets three new entries: <i>space</i> = <i>macro</i> , §558. <i>space_shrink</i> = <i>macro</i> , §558. <i>space_stretch</i> = <i>macro</i> , §558.]	
Page B495, lines 18 and 19	(2/15/87)
[delete these lines, since the cases cannot occur]	
Page B510, line 8	(12/15/86)
("Pretend that you're Hercule Poirot: Examine all clues,")	
Page B527, new line to follow line 13	(6/17/86)
This program doesn't bother to close the input files that may still be open.	
Page B534, fourth-last line	(5/4/87)
define <i>write_stream</i> (#) \equiv <i>info</i> (# + 1) { stream number (0 to 17) }	
Page B544, left column	(1/28/87)
<i>acc.kern</i> : 155, 191, 837, 842, 879, 1125.	
Page B546, entry for <i>c</i>	(4/6/87)
[Add a reference to section 994.]	
Page B547, left column	(4/7/87)
<i>char</i> : 19, 26–27, 520, 534.	
Page B547, left column	(6/14/87)
Chinese characters: 134, 585.	
Page B553, entry for <i>font_base</i>	(6/14/87)
[Insert a reference to section 134.]	
Page B555, right column, new entry	(10/25/86)
Huge page..., 641.	

Page B556, entry for <i>incr</i>	(1/28/87)
[Add a reference to section 842.]	
Page B557, entry for <i>is_char_node</i>	(1/28/87)
[Delete the reference to section 881.]	
Page B557, right column	(6/14/87)
Japanese characters: 134, 585.	
Page B560, right column	(1/28/87)
<i>max_in_open</i> : 11, 14, 304, 328.	
Page B561, left column, line 10	(4/15/87)
169–172, 174, 178, 182, 1249, 1312, 1334.	
Page B561, left column	(5/1/87)
Missing font identifier: 577.	
Page B563, left column, line 2	(4/15/87)
136, 145, 149–154, 164, 168–169, 175–176, 182,	
Page B563, right column	(6/14/87)
oriental characters: 134, 585.	
Page B569, right column, in appropriate places	(10/12/86)
<i>space</i> : 547, 558, 752, 755, 1042.	
<i>space_shrink</i> : 547, 558, 1042.	
<i>space_stretch</i> : 547, 558, 1042.	
Page B570, third-last line	(1/28/87)
786, 795, 809, 819–820, 822, 837, 842–844, 866,	
Page B571, right column	(10/25/86)
The following...deleted, 641, 992, 1121.	
Page B571, right column	(4/7/87)
<i>text_char</i> : 19, 20, 25, 47.	
Page B573, right column	(5/1/87)
[Delete the entry for 'Undefined font code'.]	

Page B576, line 2 (1/28/87)

⟨ Add the width of node s to *break_width* and increase t , unless it's discardable 842 ⟩
 Used in section 840.

Page B591, line 6 from the bottom (1/28/87)

⟨ Subtract the width of node v from *break_width* 841 ⟩ Used in section 840.

Page C14, top two lines (3/16/87)



The recursive midpoint rule for curve-drawing was discovered in 1959 by Paul de Casteljau, who showed that the curve could be described algebraically by the remarkably simple formula

Page C54, sixth-last to fourth-last lines (10/13/86)

Jonathan H. Quick (a student) used 'a.plus1' as the name of a variable at the beginning of his program; later he said 'let plus+'. How could he refer to the variable 'a.plus1' after that?

Page C76, line 14 (10/13/86)

$x_4 = w - .01in$ Point 4 should be one-hundredth of an inch inside

Page C103, line 12 (10/12/86)

$ht\# = body_height\#; .5[ht\#, -dp\#] = axis\#;$

Page C105, line 13 (10/13/86)

The vertical line just to the right of the italic left parenthesis shows the italic

Page C113, lines 20–27 (8/23/86)



The command 'erase fill c ' is an abbreviation for 'cullit; unfill c ; cullit'; this zeros out the pixel values inside the cyclic path c , and sets other pixel values to 1 if they were positive before erasing took place. (It works because the initial **cullit** makes all the values 0 or 1, then the **unfill** changes the values inside c to 0 or negative. The final **cullit** gets rid of the negative values, so that they won't detract from future filling and drawing.) You can also use 'draw', 'filldraw', or 'drawdot' with 'erase'; for example, 'erase draw p ' is an abbreviation for 'cullit; undraw p ; cullit', which uses the currently-picked-up pen as if it were an eraser applied to path p .

Page C124, line 9 (6/17/86)

$branch_2 = flex((30, 570), (10, 590), (-1, 616))$

Page C130, 3rd-last line (9/25/86)

Geometry 1 (1986), 123–140]: Given a sequence

Page C144, sixth line of the program (8/23/86)

```
6  y2 = .1h; top y3 = .4h;
```

Page C148, the line before the illustration (11/27/86)

are polygons with 32 and 40 sides, respectively:

[New illustrations are needed here, since METAFONT version 1.3 improves the accuracy of pen polygons.]

Page C149, 7th line after the illustration (10/24/86)

($200, y + 100 \pm \alpha$), where $\alpha = \sqrt{5}/4 \approx 0.559$. If we digitize these outlines and fill the

Page C178, second-last line (8/23/86)

(If $t_3 = t_1$ transum t_2 , then z transformed $t_3 = z$ transformed $t_1 + z$ transformed t_2 ,

Page C198, fifth-last and fourth-last lines (10/13/86)

```
top y2 = round(top beta).
```

Such operations occur frequently in practice, so plain METAFONT provides convenient

Page C212, lines 9–11 from the bottom (8/23/86)

```
| point <numeric expression> of <path primary>
| precontrol <numeric expression> of <path primary>
| postcontrol <numeric expression> of <path primary>
```

Page C233, lines 13–14 (2/15/87)

one column of white pixels, if the character is $2a$ pixels wide, because the right edge of black pixels is specified here to have the x coordinate $2a - 1$.

Page C247, lines 23–25 (11/27/86)

16.2. ‘**pencircle** scaled 1.06060’ is the diamond but ‘**pencircle** scaled 1.06061’ is the square. (This assumes that $fillin = 0$. If, for example, $fillin = .1$, the change doesn’t occur until the diameter is 1.20204.) The next change is at diameter 1.5, which

Page C262, lines 1–4 (7/28/86)

When we come to macros whose use has not yet been explained—for example, somehow **softjoin** and **stop** never made it into Chapters 1 through 27—we shall consider them from a user’s viewpoint. But most of the comments that follow are addressed to a potential base-file designer.

Page C266, line 16 (8/17/86)

variables; they have the side effect of changing the variable’s value.

Page C276, line 26 (6/23/86)

```
if charic<>0: r((w+charic*hppp,h.o_),(w+charic*hppp,.5h.o_)); fi
```

Page C286, lines 24–26 (10/13/86)

but METAFONT won't let you. And even if this had worked, it wouldn't have solved the problem; it would simply have put **ENDFOR** into the replacement text of **ast**, because expansion is inhibited when the replacement text is being read.

Page C290, line 1 (8/23/86)

2. *Fortuitous loops.* The 'max' and 'min' macros in Appendix B make use of the fact

Page C298, third-last line (8/23/86)

$$t[u_1, \dots, u_n] = t[t[u_1, \dots, u_{n-1}], t[u_2, \dots, u_n]]$$

Page C304, 14th-last line (2/15/87)

[replace this '\smallskip' by a \smallskip between lines!]

Page C307, fifth-last line (12/7/86)

```
adjust_fit((left sidebearing adjustment), (right sidebearing adjustment));
```

Page C312, line 34 (10/12/86)

```
params[2] = "sans_params";      fontname[2] = "cmsb10";
```

Page C316, lines 19–21 (8/17/86)

example, '(some charht values had to be adjusted by as much as 0.12pt)' means that you had too many different nonzero heights, but METAFONT found a way to reduce the number to at most 15 by changing some of them; none of them had to be

Page C319, line 3 (8/23/86)

specified by saying, e.g.,

Page C321, line 6 (7/28/86)

```
special "identifier " & font_identifier_;
```

Page C334, line 2 (6/23/86)

```
currentpicture := currentpicture shifted-(1,1); pix := currentpicture;
```

Page C339, tenth-last line (2/4/87)

```
Jackie K=\aren {\L}au\ra Mar{\'\i}a N\H{a}ta{\l}{\u\i}e {\0}ctave
```

Page C343, second-last line	(8/23/86)
<i>the precise needs of a precise but limited intellectual goal.</i>	
Page C346, 2nd line of entry for ‘;’	(1/12/87)
217, 223–224, 263, 312.	
Page C348, line 6	(6/17/86)
concatenation, of paths, 70–71, 123, 127,	
Page C348, just before ‘debugging’	(3/16/87)
de Casteljau, Paul de Faget, 14.	
Page C348, right column	(3/16/87)
[The entry for ‘define_whole_vertical_blacker_pixels’ should be moved up before the entry for ‘define_whole_vertical_pixels’.]	
Page C352, left column	(6/1/87)
*kern, 97, 316, 317.	
Page C352, right column	(3/8/87)
[The entry for ‘lowres’ belongs before the entry for ‘lowres_fix’.]	
Page C353, left column	(3/8/87)
[The entries for ‘mode’ and ‘(mode command)’ belong before the entry for ‘mode_def’.]	
Page C353, entry for mode_def	(8/17/86)
mode_def, 94, 189, 270, 278–279.	
Page C355, right column	(4/15/86)
[The entry for ‘rulepen’ belongs before the entry for ‘rules’.]	
Page C355, right column	(8/5/86)
screenstrokes, 191, 277.	
Page C355, 2nd line of entry for ‘semicolons’	(1/12/87)
217, 223–224, 263, 312.	
Page C356, full names for the Stanfords	(4/10/86)
Stanford, Amasa Leland, 340. Stanford, Jane Elizabeth Lathrop, 340.	

[A number of entries were mistakenly omitted from the mini-indexes on the right-hand pages. Here is a combined list of all the missing items; you can mount it inside the back cover, say, as a secondary mini-index when the first one fails...]

- add_or_subtract*: **procedure**, §930.
after: **array**, §427.
arg_list: *pointer*, §720.
b: *pixel_color*, §580.
bad_exp: **procedure**, §824.
before: **array**, §427.
begin_name: **procedure**, §770.
bin1: **procedure**, §968.
binary_mac: **procedure**, §863.
blank_rectangle: **procedure**, §567.
boc_c: *integer*, §1162.
boc_p: *integer*, §1162.
cf: *fraction*, §298.
clockwise: *boolean*, §453.
ct: *fraction*, §1198.
cubic_intersection: **procedure**, §556.
cur_pen: *pointer*, §403.
cur_rounding_ptr: 0 .. *max_wiggle*, §427.
cur_spec: *pointer*, §403.
cur_x: *scaled*, §389.
cur_y: *scaled*, §389.
dely: *integer*, §557.
dep_finish: **procedure**, §935.
dep_list = macro, §587.
dimen_head: **array**, §1125.
dx: *integer*, §495.
dy: *integer*, §495.
d1: 0 .. 1, §464.
end_name: **procedure**, §772.
eqtb: **array**, §201.
error_stop_mode = 3, §68.
firm_up_the_line: **procedure**, §682.
get_next: **procedure**, §667.
gf_buf: **array**, §1152.
gf_offset: *integer*, §1152.
gf_ptr: *gf_index*, §1152.
halfword = *min_halfword* .. *max_halfword*, §156.
hash: **array**, §201.
index = macro, §629.
input_ln: **function**, §30.
interaction: 0 .. 3, §68.
j: 0 .. *move_size*, §357.
known_pair: **procedure**, §872.
limit = macro, §629.
m_spread: *integer*, §357.
materialize_pen: **procedure**, §865.
max_allowed: *scaled*, §403.
max_c: **array**, §813.
max_link: **array**, §813.
max_tfm_dimen: *scaled*, §1130.
mem_top = macro, §12.
mem: **array**, §159.
memory_word = **record**, §156.
more_name: **function**, §771.
m1: *integer*, §464.
n: *screen_col*, §580.
n_sin_cos: **procedure**, §145.
name = macro, §629.
negate_dep_list: **procedure**, §904.
new_knot: **function**, §871.
node_to_round: **array**, §427.
n1: *integer*, §464.
octant_dir: **array**, §395.
o1: *small_number*, §453.
o2: *small_number*, §453.
paint_row: **procedure**, §568.
param: **array**, §1096.
param_stack: **array**, §633.
path_length: **function**, §916.
perturbation: *scaled*, §1119.
phi: *angle*, §542.
pool_ptr: *pool_pointer*, §38.
post_head: *pointer*, §843.
pre_head: *pointer*, §843.
print_err = macro, §68.
print_macro_name: **procedure**, §722.
quarterword = 0 .. 255, §156.
recycle_value: **procedure**, §809.
row_transition: *trans_spec*, §579.
scan_text_arg: **procedure**, §730.
scroll_mode = 2, §68.
set_controls: **procedure**, §299.
sf: *fraction*, §298.
show_context: **procedure**, §635.
sorted = macro, §325.
st: *fraction*, §298.
start = macro, §629.
start_sym: *halfword*, §1077.
str_pool: **packed array**, §38.
str_ptr: *str_number*, §38.
str_start: **array**, §38.
take_part: **procedure**, §910.
tfm_changed: *integer*, §1130.
tol: *integer*, §557.
tt: *small_number*, §843.
tx: *scaled*, §954.
txx: *scaled*, §954.
txy: *scaled*, §954.
ty: *scaled*, §954.
tyx: *scaled*, §954.
tyy: *scaled*, §954.
unsorted = macro, §325.
uw: 0 .. *bistack_size*, §557.
xy: 0 .. *bistack_size*, §557.
x1: *scaled*, §542.
x2: *scaled*, §542.
x3: *scaled*, §542.
y1: *scaled*, §542.
y2: *scaled*, §542.
y3: *scaled*, §542.

Volume D, in general	(4/6/87)
[The percent signs in all the comments (for example, on pages 7 and 42) are in the wrong font! Change ‘%’ to ‘%’.]	
Page Dvii, line 9	(9/25/86)
<i>Discrete and Computational Geometry</i> 1 (1986), 123–140. <i>Develops the theory</i>	
Page D2, line 27	(6/17/86)
define <i>banner</i> ≡ ‘This _□ is _□ METAFONT _□ , _□ Version _□ 1.3’ { printed when METAFONT starts }	
Page D18, line 30	(5/22/86)
<i>str_ptr</i> : <i>str_number</i> ; { number of the current string being created }	
Page D23, second line of mini-index, right column	(6/14/87)
<i>pool_name</i> = “string”, §11.	
Page D30, lines 33–34	(6/14/87)
to delete a token, and/or if some fatal error occurs while METAFONT is trying to fix a non-fatal one. But such recursion is never more than two levels deep.	
Page D63, lines 13–14	(5/5/87)
[These two lines can be eliminated, since the variable <i>temp_ptr</i> is no longer used! If you delete them, also remove §158 from the list of sections where global variables are declared (pages D7 and D552), and remove <i>temp_ptr</i> from the index on page D540.]	
Page D66, line 6	(5/22/86)
function <i>get_node</i> (<i>s</i> : <i>integer</i>): <i>pointer</i> ; { variable-size node allocation }	
Page D66, lines 31–32	(3/16/86)
controlled growth helps to keep the <i>mem</i> usage consecutive when METAFONT is implemented on “virtual memory” systems.	
Page D67, lines 7–8	(4/21/87)
if <i>r</i> = <i>p</i> then if <i>rlink</i> (<i>p</i>) ≠ <i>p</i> then ⟨ Allocate entire node <i>p</i> and goto <i>found</i> 171);	
Page D86, second line of section 198	(2/27/87)
Individual class numbers have no semantic or syntactic significance, except in a few instances	
Page D101, line 2	(3/16/86)
like ‘x’, or they can combine the structural properties of arrays and records, like ‘x20a.b’. A	

Page D102, line 24 (3/16/86)

In other words, variables have a hierarchical structure that includes enough threads running

Page D127, line 10 (5/5/87)

[Variable r can be eliminated, since it is not used in this procedure! If you delete it, also remove 280 from the corresponding index entry on page D536.]

Page D129, line 15 (5/5/87)

[This line can be eliminated, since *sine* and *cosine* are not used in this procedure! If you delete them, also remove 284 from the corresponding index entries on pages D538 and D521.]

Page D142, line 23 (4/24/87)

$(7 - \sqrt{28})/12$; the worst case occurs for polynomials like $B(0, 28 - 4\sqrt{28}, 14 - 5\sqrt{28}, 42; t)$.

Page D178, third-last line (7/30/86)

The following code maintains the invariant relations $0 \leq x_0 < \max(x_1, x_1 + x_2)$, $|x_1| < 2^{30}$,

Page D228, line 13 (7/30/86)

while *max_coef* < *fraction_half* **do**

The mini-index at the bottom of the next page should also receive the following new entry:

fraction_half = macro, §105.

Page D228, 10th-last line (5/5/87)

begin *right_type*(p) $\leftarrow k$;

[Also eliminate ‘ q ,’ seven lines above this, and delete 497 from the index entry for q on page D536.]

Page D248, lines 16–21 (11/27/86)

alpha \leftarrow *abs*(u); *beta* \leftarrow *abs*(v);

if *alpha* < *beta* **then**

begin *alpha* \leftarrow *abs*(v); *beta* \leftarrow *abs*(u); **end**; { now $\alpha = \max(|u|, |v|)$, $\beta = \min(|u|, |v|)$ }

if *internal*[*fillin*] $\neq 0$ **then**

$d \leftarrow d - \textit{take_fraction}(\textit{internal}[\textit{fillin}], \textit{make_fraction}(\textit{beta} + \textit{beta}, \textit{delta}))$;

$d \leftarrow \textit{take_fraction}((d + 4) \textit{div} 8, \textit{delta})$; *alpha* \leftarrow *alpha* **div** *half_unit*;

Page D263, line 20 (3/16/86)

instead of *false*, the other routines will simply log the fact that they have been called; they won’t

Page D268, line 2 (4/28/87)

Given the number k of an open window, the pixels of positive weight in *cur_edges* will be shown

Page D301, line 6 of section 652 (5/5/87)

[This line can be eliminated, since variable *s* is not used in this procedure! If you delete it, also remove 652 from the corresponding index entry on page D537; remove 652 from the index entries for *param_size* and *param_start* on page D534; and remove *param_size* from the mini-index on page D301.]

Page D376, lines 17 and 18 (11/14/86)

[these two mysterious lines should be deleted]

Page D380, line 11 (5/5/87)

[Variables *q* and *r* can be eliminated, since they are not used in this procedure! If you delete them, also remove 862 from the corresponding index entries on page D536.]

Page D429, line 14 (5/5/87)

begin *p* ← *cur_exp*;

[Also eliminate line 12, and delete 985 from the index entry for *vv* on page D543.]

Page D455, line 5 (5/5/87)

[This line can be eliminated, since variable *t* is not used in this procedure! If you delete it, also remove 1059 from the corresponding index entry on page D540; remove 1059 from the index entries for *small_number* and *with_option* on pages D539 and D544; and remove *with_option* from the mini-index on page D455.]

Page D463, line 10 (12/15/86)

("Pretend that you're Miss Marple: Examine all clues,")

Page D465, lines 17–18 (6/14/87)

[Delete these two lines.]

Page D474, 5th-last line (3/16/86)

depths, or italic corrections) are sorted; then the list of sorted values is perturbed, if necessary.

Page D481, line 12 (6/17/86)

print_nl("Font metrics written on"); *print*(*metric_file_name*); *print_char*(".");
b_close(*tfm_file*)

The mini-index at the bottom of this page should also receive the following new entry:

print_char: **procedure**, §58.

Page D510, new line to follow line 5 (6/17/86)

This program doesn't bother to close the input files that may still be open.

Page D510, just before the fifth-last line	(8/5/86)
<i>internal[fontmaking] ← 0; { avoid loop in case of fatal error }</i>	
Page D520, right column	(6/14/87)
Chinese characters: 1147.	
Page D526, left column, lines 1–2	(7/30/86)
<i>fraction_half</i> : 105, 111, 152, 288, 408, 496, 543, 1098, 1128, 1141.	
Page D526, left column, lines 6–7	(7/30/86)
478, 497, 499, 503, 530, 540, 547, 549, 599, 603, 612, 615, 815–816, 917, 1169–1170.	
Page D528, right column	(6/14/87)
Japanese characters: 1147.	
Page D530, right column, line 45	(7/30/86)
<i>max</i> : 539, 543.	
Page D533, right column	(6/14/87)
oriental characters: 1147.	
Page D535, right column, line 27	(6/17/86)
1134, 1163–1165, 1182, 1194, 1200, 1205, 1213.	
Page D547, bottom two lines	(11/27/86)
[These lines, and the top two on the next page, should move down so that they appear in alphabetical order just before ‘Compute test coefficients’.]	
Page Exiii, lines 1–2	(7/28/86)
February 11–13, 1984), 49. <i>An example meta-character of the Devanagari alphabet, worked out “online” with the help of Matthew Carter.</i>	
Page Exiii, line 6	(7/28/86)
<i>and western alphabets work also for Devanagari and Tamil.</i>	
Page E12, lines 15 and 19	(7/23/86)
[change ‘17.32’ to ‘17.28’ in both places]	

Page E12, third-last line (12/18/86)

[change ‘41’ to ‘40’]

Page E13, lines 3, 4, and 20 (12/18/86)

[change ‘40’ to ‘41’, ‘48’ to ‘47’, ‘17’ to ‘7’]

Page E18, line 20 (7/23/86)

[change ‘17.32’ to ‘17.28’]

Page E18, line 29 (12/9/86)

[change ‘236’ to ‘212’ in the `cmss9` column]

Page E170, top illustration (11/2/86)

[There should be no “dish” or depression in the vicinity of point `3x`; the top edge of the character should be straight. This error appears also in the other uses of ‘`no_dish_serif`’ throughout the book, since the illustrations were made before ‘`no_dish_serif`’ was added to the program. See page E180 (twice at the top), E370 (twice), E374 (twice), E376 (twice), E378 (top), E390 (bottom), E398 (top), E402 (top), E406 (top), E453 (twice).]

Page E179, new line to be inserted after line 6 (10/13/86)

if `shaved_stem < crisp.breadth`: `shaved_stem := crisp.breadth`; **fi**

Page E219, line 29 (6/2/87)

`top y1 = h; x1 = x2; filldraw stroke z1e -- z2'e; % stem`

Page E279, seventh line from the bottom (7/20/86)

that delicious but restrained humor which her readers found so irresistible.

Page E301, new line to be inserted after line 28 (5/15/87)

if `lower_side > 1.2upper_side`: `upper_side := lower_side`; **fi**

Page E554, bottom half of page (12/18/86)

[The letters will change slightly because of the corrections to `cmr17` noted on pages 12 and 13.]

Page E561, line 3 (12/9/86)

[The numerals should be ‘0123456789’ (i.e., 2/3 point less tall) because of the correction made to page 18.]

Page E562, line 9 (12/9/86)

[The numerals should be ‘*0123456789*’ (i.e., 2/3 point less tall) because of the correction made to page 18.]

Page E572, entry for *breadth* (10/13/86)

breadth, 59, 75, 79, 91, 93, 179, 225, 233,

Page E573, entry for `cmcsc10` (8/17/86)

`cmcsc10`, 30–31, 567.

Page E576, tenth-last line (5/15/87)

`lowres_fix`, 550.