

# Learning about the histories of computerizing publishing and desktop publishing, 2017–19

David Walden

Draft of June 30

This is my third TUG annual conference presentation about the history of publishing, printing, and typesetting. For TUG 2012 in Boston (where I live), I thought an appropriate topic was *Printing & Publishing in Boston: An Historical Sketch* ([walden-family.com/bbf/bbf-printing.pdf](http://walden-family.com/bbf/bbf-printing.pdf)). This kindled an interest in digital typography history; thus, my topic in 2016 in Toronto was *An Informal Look into the History of Digital Typography* ([tug.org/tug2016/walden-digital.pdf](http://tug.org/tug2016/walden-digital.pdf)). I then hoped to spend some time expanding my 2016 paper into a small monograph, but first an opportunity came for me to learn a lot more about the history of digital typography.

## 1 Desktop publishing meeting

I was invited to be an observer at a two-day May 2017 meeting at the Computer History Museum of pioneers of desktop publishing (DTP) ([history.computer.org/annals/dtp](http://history.computer.org/annals/dtp)). Pioneers in attendance at the meeting were Chuck Bigelow (Bigelow & Holmes type design studio), Paul Brainerd (Aldus), Liz Crews (was Liz Bond, Xerox PARC and Adobe), Charles Geschke (PARC, Adobe), Steve Kirsch (Frame Technology), Don Knuth (Stanford,  $\TeX$ ), Butler Lampson (PARC), Lee Lorenzen (Ventura Software), John Scull (Apple), Jonathan Seybold (ROCAPPI, Seybold Publications and Seminars), John Shoch (PARC), Charles Simonyi (PARC, Microsoft) Bob Sproull (PARC), Larry Tesler (PARC and Apple), John Warnock (PARC, Adobe), and Richard Ying (Atex).

The meeting was organized by Burt Grad, co-founder of the museum's Software History Special Interest Group, and David Brock, director of the museum's Center for Software History. They sought advice from Jonathan Seybold (more about him later) about which pioneers to invite to the meeting. This was the fourteenth pioneer meeting Burt has organized or co-organized since 2003, and these meetings have resulted in eight special issues of the *IEEE Annals of the History of Computing* plus additional stand alone *Annals* articles, and 130 oral histories for the museum's collection.<sup>1</sup>

This desktop publishing pioneer meeting is resulting in two (closer to three) desktop publishing special issues of the *Annals*.

Issue 1 (*Annals* vol. 40, no. 3, July-September 2018)

Desktop Publishing: Laying the Foundation by Burton Grad and David Hemmendinger

Rocappi: Computerizing the Publishing Industry by Jonathan W. Seybold

How Atex Helped an Industry Change the World by Douglas Drane

More about Atex by Jonathan Seybold

The Xerox Alto Publishing Platform by Robert F. Sproul

How Modeless Editing Came To Be by Lawrence G. Tesler

The Origins of PostScript by John E. Warnock

$\TeX$ : A Branch of Desktop Publishing, Part 1 by Barbara Beeton, Karl Berry, and David Walden

Interview with Charles Bigelow by David Walden

---

<sup>1</sup>Burt invited me to the meeting because he knew of my work for many years as a member of the editorial board of the *IEEE Annals of the History of Computers*.

Issue 2 (*Annals* vol. 41, no. 3, July-September 2019)

Desktop Publishing: Building the Industry by Burton Grad and David Hemmendinger

Seybold Publications and Seminars by Jonathan Seybold

Founding and Growing Adobe Systems Inc. by John Warnock and Charles Geschke

Paul Brainerd, Aldus Corporation and the Desktop Publishing Revolution by Suzanne Crocker

Desktop Publishing: The Killer App That Saved the Macintosh by John Scull and Hansen Hsu

Interview with Tim Gill (Quark) by Jay Nelson

Frame Technology and FrameMaker by David J. Murray

The Ventura Story by Lee Lorenzen

TeX: A Branch of Desktop Publishing, Part 2 by Barbara Beeton, Karl Berry, and David Walden (published in *Annals* vol. 41, no. 2, April-June 2019)

Oral History of Liz Bond Crews by Paul McJones (to be published in the *Annals* in early 2020)

Font Wars parts 1 and 2 by Charles Bigelow (to be published in the *Annals* in early 2020)

Burt Grad and David Hemmendinger are the special issue guest editors. The Computer History Museum has posted on its website the transcripts of the nine sessions of the two-day-meeting. The Museum also has or soon will have interviews or oral histories of Charles Bigelow, Paul Brainerd, Charles Greschke and John Warnock, Steve Kirsch, Donald Knuth, Butler Lampson, Lee Lorenzen, John Schull Jonathan Seybold, Robert Sproull, Gary Starkweather, Larry Tesler, and Charles Thacker.

From the meeting and from reading the above materials and helping prepare them for publication, I learned about the following topics that were missing from my 2016 paper.

- Computerizing newspaper, periodical, and book publishing
  - John Seybold and ROCAPPI
  - Michael Barnettts PAGE-1
  - Bringing “all digital” to newspapers, e.g., Atex
- Jonathan Seybold and the Seybold Reports and Seminars
- Development of the desktop publishing technology and market: Xerox PARC, Adobe, Aldus, Apple, Frame, Interleaf, Quark, Ventura
- The “Font wars” of 1989 to 1995 and prior technology

I will sketch a bit about all but the last of these in this paper. My short monograph will cover more.

As I see things, use of digital computers in typesetting and publishing followed two more or less parallel paths in the 1960s and 1970s. (1) various commercial vendors were working to computerize the publishing industry, initially via computer control of phototypesetting machines; I discuss this in the next section. (2) various individuals in universities and research laboratories were bringing out a succession of computer programs to format text for their typewriter printers and line printers; I discussed this in my 2016 presentation. Then in the 1980s with laser printers, page description languages such as PostScript, and “desktop publishing”, the two worlds came together; I discuss this in section 3.

## 2 Making publishing digital

Digital technology started coming to the typesetting world in the 1930s when linotype machines began to be able to input paper tapes, either coming from wire services (TeleType-Setting) or punch on local TTS keyboards. Fax-like systems were also in use to transmit images. In the 1950s, phototypesetting systems began to be available and by the 1960s their use was spreading widely replacing linotype machines. The early phototypesetters were driven by paper tapes punched from a keyboard; next the typesetters were connected directly to keyboard using dedicated electronics, and then general purpose computers drove the typesetters; whole pages could then be specified — text plus layout.<sup>2,3</sup> Eventually

<sup>2</sup>John W. Seybold, *The World of Digital Typesetting*, Seybold Publications, Inc., Media, PA, 1984, [computerhistory.org/collections/catalog/102740425](http://computerhistory.org/collections/catalog/102740425).

<sup>3</sup>Frank Romano, *History of the Phototypesetting Era*, Graphic Communications Institute at Cal Poly State University, 2014.

the computer systems capabilities expanded to encompass all the functions involved in producing a newspaper or periodical. Throughout this process there was lots of backward compatibility. For instance, a computer system capable of complex typesetting and page layout would still need to be able handle the electronic equivalent of paper tape input from wire services.

## 2.1 Going from phototypesetting to digital

There were a number of key people, newspapers, and vendors that pioneered and spread the increasingly digital technology, originally with phototypesetters but heading toward all digital. Marcus and Trimble<sup>4</sup> noted some of these companies, such as the *Minneapolis Star Tribune*, the *New York Daily News*, and the Mergenthaler Linotype Company.

Another useful summary of the early activities in computer typesetting comes from a note by Jonathan Seybold.<sup>5</sup> Table 1 (derived from Seybold’s note) shows some of the steps in computerizing the printing and publishing industry (newspapers, periodicals, books) in the phototypesetter era. An exception in the table is Saltzer’s 1964–5 development of RUNOFF. RUNOFF’s purpose was to nicely print people’s documents on their personal terminals or office line printer; it was not aimed at the challenges of the publishing industry. In the rest of this section, I will say something about the efforts marked with \*\*\* in the table.

---

Table 1: Phototypesetting-to-digital for industry\*

- 1961–1964, Michael Barnetts experiments at MIT \*\*\*
- 1962, John Duncan began research on computer typesetting at the University of Newcastle-upon-Tyne
- RCA 301 and IBM 1620 based hyphenation and justification at newspapers
- 1963–1970, John Seybolds Rocappi company \*\*\*
- 1964–5, IBM 1401 and 1130 and DEC PDP-8 based typesetting systems
- 1964, Saltzers RUNOFF at MIT—interactive text formatting
- 1966–1967, PAGE1 computer composition system, produced in the Graphics Systems Division of RCA \*\*\*
- 1967 on, other similar systems
- 1971, Seybold Reports started by John and Jonathan Seybold \*\*\*
- 1973–1981, Atex offers full office newspaper/periodical/etc. system \*\*\*

\* This table is derived from a September 2018 note by Jonathan Seybold titled “Early steps in computer typesetting in the 1960s” and posted at [history.computer.org/annals/dtp/rocappi-typesetting.pdf](http://history.computer.org/annals/dtp/rocappi-typesetting.pdf).

---

## 2.2 Michael Barnett

Jonathan Seybold believes<sup>5</sup> that Michael Barnett’s document shown in Figure 1 was one of the first two documents phototypeset from output generated by a computer; the other was a press release also produced by Barnett. The Wikipedia reports that Barnett also typeset a number of books with his computer composition system.<sup>6</sup>

Barnett was at MIT and was working with a Photon 560 “film setting” machine. Text and instructions (as shown in Figure 1) were typed on a Friden Flexowriter that output the typed characters on paper tape. This paper tape was converted by Barnett’s TYPRINT program running on MIT’s IBM 709 computer into another paper tape in a format understandable

---

<sup>4</sup>Mike Marcus and George Trimble, Taking Newspapers from Hot Lead into the Electronic Age, *IEEE Annals of the History of Computing* vol. 28 no. 4, 2006, pp. 96-100.

<sup>5</sup>Jonathan Seybold, Early steps in computer typesetting in the 1960s, September 2018, [history.computer.org/annals/dtp/rocappi-typesetting.pdf](http://history.computer.org/annals/dtp/rocappi-typesetting.pdf).

<sup>6</sup>[en.wikipedia.org/wiki/Michael\\_P.\\_Barnett](http://en.wikipedia.org/wiki/Michael_P._Barnett)

```

[indn77d12ls24st1,,36cnxs1]
EXCERPT FROM ALICE IN WONDERLAND
[nl1s18]
December 6, 1961
[sp4st2,10,36st3,11,36st4,12,36st5,13,36st6,14,36s
st8,16,36st9,17,36st10,18,36ls14d11xs2r1]
[sc19sc19]Fury said to
[xs3]a mouse, That
[nlxs6]he met
[xs7]in the
[xs8]house,
[xs9][sc19]Let us
[xs7]both go
[xs6ls12]to law:
[xs5d19]I[d11] will
[xs3]prosecute
[xs2d19]you.
[nlxs3d11] Come, I'll
[nlxs4]take no
[nlxs5]denial:
[nlxs7]We must
[nlxs8ls11]have a
[nlxs9]trial[sc47]
[nlxs10] For
[xs7]really
[xs6]this
[nl] morning
[nlxs8ls10] I've
[xs6]nothing
[xs5]to do.'
[xs4]Said the
[xs2]mouse to
[xs1] the cur,
[nlxs3][sc19]Such a
[nlxs4ls9]trial,
[xs3]dear sir,
[xs2] With no
[xs2]jury or
[xs1ls8]judge,
[nlxs2]would be

```

```

"Fury said to
a mouse, That
he met
in the
house,
'Let us
both go
to law:
I will
prosecute
you.
Come, I'll
take no
denial:
We must
have a
trial;
For
really
this
morning
I've
nothing
to do.'
Said the
mouse to
the cur,
'Such a
trial,
dear sir.
With no
jury or
judge,
would be
wasting
our breath.
'I'll be
judge,
I'll be
jury.'
Said
cunning
old Fury:
'I'll try
the whole
case.

```

Figure 1: Barnett's reproduction of a page from chapter 3 of *Alice in Wonderland* with phototypesetter commands

by the Photon 560. Another program in the 709, TABPRINT, could input papers tapes from non-Flexowriter sources.

Barnett wrote a book on his work computer composition work at MIT which is widely cited.<sup>7</sup> His work during this 1961–64 period also apparently was useful in terms of helping other people see that they could do themselves. Barnett's book also is a useful reference for what happened before his work and suggests state-of-the-art when he was working.

Having slipped into the domain of computer composition from physics because of a need to format some physics formulas, Barnett became further involved with the worlds of computer composition (see section 2.5), publishing, and libraries.<sup>6</sup>

## 2.3 Rocappi

John Seybold and his son Jonathan had impact on the printing and publishing industry from 1963 to 1990. They were involved with the Rocappi company (1963–1970), as I discuss below. Post Rocappi they (first both and then Jonathan alone) produced the Seybold reports and seminars (section 3.2). What I sketch below is primarily based on

<sup>7</sup>Michael T. Barnett, *Computer Typesetting: Experiments and Prospects*, MIT Press, 1965.

Jonathan's paper in the first *Annals* special issue on desktop publishing.<sup>8</sup>

John Seybold joined fledgling world of computerized phototypesetting in 1963 with the formation of his company Rocappi—Research on Computer Applications in the Printing and Publishing Industries. In 1962 John knew the publishing industry but not computers when he saw an early computer typesetting system at a newspaper. He immediately envisioned many ways a computer could speed along the move away from hot metal type that was already underway with phototypesetting. He started his company to participate in and help advance what he saw as a coming revolution. Rocappi had inordinate impact for its size.

Rocappi didn't actually do research. Rather it took on a variety of typesetting jobs, primarily from publishers, and used a computer to carry out the jobs under the principle that new programming for any job should be done in a general enough way that it could lead to a general class of jobs. Their software ran on an RCA 301 computer, and their software could generate instructions for various different phototypesetters. In Jonathan Seybold's paper on Rocappi,<sup>8</sup> he describes several aspects of the software created at Rocappi.<sup>9</sup>

**Device independent markup.** For instance, since the different phototypesetters required different commands to drive them, a person keyboarding the text to be typeset could give a command that said which phototypesetter the output was for (e.g.,  $\beta a$  for the first kind of typesetter,  $\beta b$  for the first kind of typesetter) and then ignore phototypesetter differences in specifying text formatting commands. Also, the formatting commands were generally abstract rather than actual device codes (e.g.,  $\$hb$  for second level heading,  $\$hc$  for third level heading) and had different implementing code depending on the document style and which typesetter was specified with the prior  $\beta$  command. The  $\$$ -codes could also indicate an actual device code.

**Pattern based hyphenation.** Jonathan Seybold has sketched the hyphenation method used by Rocappi.<sup>8,10</sup>

The Rocappi routine. . . looked at successive blocks of five consecutive letters. Each five-letter combination pointed to a position in a table of bits. If it was permissible to place a hyphen between the second and third letters, the bit would be one. If it was not permissible, the bit would be zero.

The bit table was generated by running a heuristic program against a large dictionary. The program was left running overnight, night after night, until it stopped improving itself.

According to Seybold, this approach to hyphenation was used by Rocappi from its earliest days (ca. 1963) and was developed by Colin Barbar ("an exceptional programmer").<sup>11</sup> I couldn't learn enough about Rocappi's hyphenation method to compare it in detail with the Frank Liang's approach which is used in  $\text{\TeX}$ .

**Hyphenation correction.** Rocappi's computer did not have the capability of holding the entire dictionary. Thus they used the pattern based hyphenation method described in the prior paragraph. However, they felt that the market required perfect hyphenation. After running their H&J program, the hyphenated words in a document were sorted into alphabetical order and then compared against the dictionary to catch any mis-hyphenated words.

To deal with the book publishing market—more demanding than the newspaper market—Rocappi's typesetting system also supported kerning, tracking, and ligatures.

<sup>8</sup>Jonathan W. Seybold, Rocappi: Computerizing the Publishing Industry, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 8–24.

<sup>9</sup>Jonathan, having had his first experience with a computer as a student of economics in 1964, joined his father's company in 1965 where within a year he was effectively running Rocappi's production operations.

<sup>10</sup>Jonathan Seybold, email of 2019-05-08.

<sup>11</sup>Rowley Atterbury, Colin Barber: Computer pioneer who launched the greatest printing revolution since Gutenberg, *The Guardian* obituary, October 3, 2006, [theguardian.com/technology/2006/oct/04/news.guardianobituaries](https://www.theguardian.com/technology/2006/oct/04/news.guardianobituaries)

**Character width changing.** Jonathan Seybold also has described the following technique Rocappi used (nearly 30 years before Zapf and Karow’s similar ideas were published):<sup>10</sup>

I found at Rocappi that allowing the composition program to vary the set width of the characters on a line of type in very fine increments gave the composition program a great deal of added flexibility in producing beautifully justified type. Changing the type set width by a tenth of a point or so results in changes to character and word shapes that are imperceptible to the human eye, but which make a considerable difference over the length of a line.

This was for a Harris Intertype Fototronic CRT typesetter which had been modified to allow type to be sized to 1/10 of a point (1/720 of an inch).<sup>12</sup> Rocappi used this technique, for instance, to typeset the King James Bible.

**Pagination and vertical justification.** Again to deal with having a small computer, Rocappi’s system scanned over a hyphenated and justified text file and extracted just enough information about the text to calculate where page breaks would go—the text itself was not needed. Using the resulting “text facts” file, the page makeup program could calculate the best places for page breaks including calculating “vertical justification” adjustments within pages and being able to make adjustments to prior pages that improved later pages. The program “produced a compact page descriptor file that specified what was to go on every page and what spacing adjustments were required to make the page come out right.”<sup>8</sup>

For the Bible project, Jonathan discovered that slight changes in interline spacing within a column to make columns the same length were not noticeable to readers even if one column had one more line than the other column.

Over the life of Rocappi, the Seybolds had many connections throughout the publishing industry. Visitors came to see what they were doing. Jonathan wrote a book describing and comparing all of the then extant CRT typesetters. They did some consulting. Jonathan has written,<sup>8</sup> “[We] viewed Rocappi as an opening chapter in what we expected would be a revolutionary change in publishing technology. Sharing what we were trying to do was a way to help kindle that revolution.”

In 1967, Rocappi was sold to Lehigh Press, with the Seybolds continued working for company. Jonathan left Rocappi in 1970 “having played a role in the embryonic states,” and wanting find a way to play a role in the bigger field that was going to move very fast. His father left the same year. (The Seybolds’ story is continued in section 3.2 on page 13.)

## 2.4 Saltzer’s RUNOFF

As stated above, RUNOFF was aimed at interactive use by authors drafting and formatting their own documents rather than for use by professional typesetters in the publishing industry, which is what Barnett and Rocappi were working on. Gerald Saltzer released RUNOFF in 1964 for use under the CTSS operating system on MITs IBM 709 and 7090 computers; it was written in the MAD programming language.

RUNOFF originally had only 16 formatting commands, as shown on the left in Figure 2. From that command list and the RUNOFF output shown on the right in the figure, you can understand the limited but still useful nature of RUNOFF.

RUNOFF was influential leading to RUNOFFs for other computers, Script for CP/CMS, and roff, which led in turn to nroff, troff, ditroff, and groff. Over time the text formatting systems for individual interactive use that followed RUNOFF gained in capability and became able to produce quality that would be acceptable to the publishing industry. With desktop publishing’s rise in the 1980s, the two worlds came together.

---

<sup>12</sup>Email 2019-05-13.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC

abbreviation	control word	automatic break
.ap	.append A	no
.ll	.line length n	no
.in	.indent n	
.ss	.single space	yes
.ds	.double space	yes
.bp	.begin page	yes
.ad	.adjust	yes
.fi	.fill	yes
.nf	.nofill	yes
.nj	.nojust	yes
.pa	.page (n)	yes, if n
.sp	.space (n)	yes
.he	.header xxxx	no
.br	.break	yes
.ce	.center	yes
.li	.literal	no

TO: CTSS users.

FROM: J. H. Saltzer

SUBJECT: TYPSET and RUNOFF, Memorandum editor and type-out commands.

DATE: November 6, 1964

The command TYPSET is used to create and edit 12-bit BCD line-marked files. This command permits editing and revising by context, rather than by line number. The command RUNOFF will print out a 12-bit BCD line-marked file in manuscript format. RUNOFF contains several special features not available with the DITTO command, including type-justification.

These two commands provide an alternative to the MEMO, MODIFY, and DITTO commands, and are intended to provide experience with a different approach to editing symbolic files.

This memorandum was prepared with the aid of these two commands.

Figure 2: Left: Summary of RUNOFF control words. Right: Saltzer used RUNOFF to produce the RUNOFF manual.

## 2.5 PAGE—1

During 1965–1966 Michael Barnett was employed by the Graphic Systems Division of RCA to develop of the PAGE—1 computer composition system.<sup>13</sup> The PAGE—1 system, written in assembly language for the RCA Spectra 70 computer, was released in 1967 for use with an RCA VideoComp® 70|820 Electronic Photocomposer. PAGE—1 appears to have been primarily aimed at typesetting books or book-like documents.

PAGE—1 was programmable in a rudimentary way (unlike RUNOFF or, I believe, Barnett’s experimental work at MIT). PAGE—1’s capabilities included:

- Thirteen typographic variables such as top boundary (**tb**), maximum interword space (**mx**), and the typeface in use (**tf**).
- Three read-only variables for horizontal position (**cx**), vertical position (**cy**), and current characters (**cc**, decimal code for most recently set character).
- Several types of global variables
  - page number (**pn**)
  - footnote counter (**fn**)
  - standard paragraph indentation (**pi**)
  - up to 201 general variables (**gvn** where  $0 < n \leq 200$ ) and where almost 150 of them conventionally held particular information, e.g., point size for footnote text (**gv14**) and space between primary text and footnotes in points (**gv172**).
  - up to 9 indirect variable (**ivi** where  $1 \leq i \leq 9$ ) [I don’t know what these were for.]

- Six arithmetic operators; the example for sum is as follows

[**ad**,variable,parameter,parameter]

- Six conditional procedures where two parameters are compared and if the condition is true, the following action is taken, for example

[**lt**,parameter,parameter[text to be set]]

[**eq**,parameter,parameter[[code-to-be-executed]]]

[**gr**,parameter,parameter[[code-to-be-executed]text-to-be-set]]

- Eight instructions for editing the text from within the system; these were dropped from PAGE—2 as it was simpler to use a separate text editor to edit the source text and insert the PAGE—1 markup.

- Names for synonyms and formats:

Synonyms had two character names (e.g., **x1**) where the first character is a letter from t through z and the second character is a number from 1 through 9; these names are given to strings of text and/or code for use within a job.

Formats had two character names (e.g., **a3**) where the first character is a letter from a through s and the second character is a number from 1 through 9; these names are given to strings that are in a central library for use from job to job.

- An instruction for assigning a sequence of instructions and/or text strings to a name, for example (from first line of Figure 3):

[**sy**,**x1**[[**gt**,**cx**,**gv1**[[**gv1**,**cx**]];**nl**]]]

The above sequence makes variable **x1** a **synonym** for the rest of the characters between the open and final close square brackets. Assuming **gv1** (the saved horizontal position) has previously been initialized to 0, the rest of the characters do the following. If **cx** is greater than (**gt**) **gv1**, then **gv1** takes the new value of **cx**. Either way, a newline (**nl**) finishes off the line.

Figure 3 shows a bit of PAGE—1 programming (paraphrased from the Pierson book).<sup>13</sup> I think the example works as follows.

<sup>13</sup>John Pierson, *Computer Composition Using PAGE—1*, John Wiley & Sons, 1972.

```

[sy,x1[[gt,cx,gv1[[gv1,cx]];nl]]
[sy,x2[[x1;df,gv1,rb,gv1;qo,gv1,gv1,2;us]]]
...
...
...
[gv1,0;su;lb,gv1]
Some text[x1]
Some more text[x1]
And this[x2]
...
...

```

Figure 3: A small example of PAGE—1.

The first line of this example was already explained in the description above of how the synonym instruction worked.

The second line defines another synonym, x2. When the x2 code is executed, first the x1 code is executed. Then the difference (df) between rb (right boundary) and gv1 is taken and becomes the new value of gv1. Next the quotient of gv1 over 2 is taken and becomes the new value of gv1. Then, typesetting is unsuspended (us).

The next instruction shown in the example first sets gv1 to zero. Then typesetting is suspended (su). Then the left boundary is set to gv1, i.e., also set to zero.

After the first line of text is scanned, x1 is executed. This checks if the current character position (cx) is bigger than the current value of gv1. Since it is, i.e., cx is 9 and gv1 is zero, gv1 is set to the value of cx (9), and processing goes on to the next line.

This works the same as the prior line. At the end of the line, cx is 14 which is greater than 9, the previous value of gv1, so gv1 is set to 14, and processing goes on to the next line.

The next line is processed, and x1 within the definition of x2 is executed which finds that 8 is not greater than 14, so gv1 remains 14.

Executing the rest of x2, let's suppose that the right boundary (rb) is 60. Subtracting 14 (gv1) from rb gives 46 which becomes the new value of gv1. The quotient (qo) of gv1 and 2 is 23 which becomes the new value of gv1.

Typesetting is then unsuspended (us) which apparently causes typesetting to jump back to where it was suspended (su), i.e., in the middle of an instruction to where the left boundary (lb) is set—now to 23, the value of gv1. This second time through the three lines of text, that same calculation happens but none of it matters as the three lines all begin at position 23, calculated such that the longest bit of text is centered on a line. (I didn't try to figure out from the book how PAGE—1 deals with proportional fonts.)

Programming in T<sub>E</sub>X macros doesn't seem too hard compared to PAGE—1. A more real PAGE—1 example is shown in Figure 4.

In time there was also an expanded version of PAGE—1 known as page PAGE—2. Still later Information International Inc. delivered PAGE—1 and PAGE—2 on one or two computers other than the Spectra 70. Barbara Beeton has told me that the AMS used PAGE—2 before the Science Typographers, Inc. system which came before T<sub>E</sub>X.

## 2.6 Atex

In Table 1, the Seybold reports, started in 1971, come before Atex which was started in 1973. However, even through the Seybolds are mentioned in this section, I will discuss the Seybold Reports together with the Seybold Seminars in section 3.

Jonathan Seybold has summarized nicely desire for automation by newspapers in the early 1970s.<sup>14</sup>

<sup>14</sup>Douglas Drane, How Atex Helped an Industry Change the World; *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 25–29; Jonathan W. Seybold and David Walden, More about Atex, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 30–36.

B3 ≡ [gv143,gv123;gv149,gv123;gv168,gv88;gv169,gv88;  
 gv195,0;gv196,0;gv197,0;gv198,0;gv200,0;iv1,98;  
 gv1,0;gv3,3;ad,pn,pn,1;fm,1;  
 sb,6;lb,0;tb,0;rb,gv58;bb,gv88;pd,gv88;  
 ne,gv123,0[[bb,gv87;pd,gv87]];ib;  
 sb,2;lb,0;rb,gv58;bb,gv88;ps,gv15;bl,ps;tf,gv45;y4,nl;  
 sb,3;lb,0;rb,gv59;bl,gv29;ju;wb[[y3]];cb;  
 ab[[iv;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];gv182,0;  
 iv1,99;gv1,1;gv3,5;ab[[iv;pg;sb,1;bb,gv168;ad,pn,pn,1;  
 pd,gv88;ne,gv123,0[[pd,gv186;by,2,0]];ib;sb,2;lb,0;rb,gv58;  
 y2,nl;sb,3;lb,0;rb,gv59;ju;cb;by,1,gv143;by,4,gv149;  
 gv98,gv168;gv99,gv169;gv144,gv143;gv150,gv149;  
 gv195,gv197;gv196,gv198;gv168,gv88;gv169,gv88;  
 gv143,gv123;gv149,gv123;gv197,0;gv198,0;  
 gv200,0;iv1,98;gv1,0;gv3,3]]];wb[[y1]]]

B4 ≡ [ad,gv144,cy,gv110;gv150,gv144;nl;by,1,gv144;by,4,gv150;  
 df,gv98,gv88,gv144;ne,gv123,0[[df,gv98,gv87,gv144]];  
 gv99,gv98;sb,1;lb,0;rb,gv59;bb,gv98;ps,gv11;bl,gv26;tf,gv41;ju]

B6 ≡ [gv192,cy;bb,gv88;gv182,1;eq,gv1,0[[sb,5]]]

B7 ≡ [eq,gv1,0[[jd;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]]];  
 gv182,0;iv1,99;gv1,2;gv3,5]];  
 eq,gv1,1[[jd;pg;sb,1;bb,gv168;ad,pn,pn,1;ib;  
 sb,2;lb,0;rb,gv58;y2,nl;sb,3;lb,0;rb,gv59;  
 ju;cb;by,1,gv143;by,4,gv149;gv98,gv168;  
 gv99,gv169;gv144,gv143;gv150,gv149;gv195,gv197;  
 gv196,gv198;gv168,gv88;gv169,gv88;gv143,gv123;  
 gv149,gv123;gv197,0;gv198,0;gv200,0;iv1,98;gv3,3]];  
 df,gv1,gv1,1]

[sy,y1[[B7]];sy,y2[[\*\*]];sy,y3[[B6]];sy,y4[[††]]

\*\* Desired running head. †† Desired drop folio.

B8 ≡ [eq,gv200,1[[eq,gv198,0[[eq,gv197,1[[gc.gv169,gv6[[gc.gv6.gv92[[gv198,1]];  
 lt.gv6.gv92[[ne,gv149,gv123[[gv198,1]];eq,gv149,gv123[[ad,gv149,gv149,gv6]]];  
 df,gv169,gv169,gv6]]];  
 eq,gv197,0[[lt,gv168,gv6[[gc.gv169,gv6[[gc.gv6.gv92[[gv198,1]];lt.gv6.gv92[[ne,gv149,gv123[[gv198,1]];  
 eq,gv149,gv123[[ad,gv149,gv149,gv6]]];  
 df,gv169,gv169,gv6]]];  
 gc.gv168,gv6[[gc.gv6.gv92[[gv197,1]];  
 lt.gv6.gv92[[ne,gv143,gv123[[gv197,1]];eq,gv143,gv123[[ad,gv143,gv143,gv6]]];  
 df,gv168,gv168,gv6]]]]]]];  
 eq,gv200,0[[eq,gv196,1[[gc.gv6.gv92[[gv197,1]];lt,gv6.gv92[[ad,gv143,gv143,gv6]];df,gv168,gv168,gv6;gv200,1]];  
 eq,gv196,0[[eq,cy,0[[gv90,0;ne,cx,0[[gv90,ps]]];  
 ne,cy,0[[eq,cx,0[[df,gv90,cy,bl;ad,gv90,gv90,ps]];ne,cx,0[[ad,gv90,cy,ps]]];df,gv90,bb,gv90;  
 eq,gv1,0[[eq,gv195,1[[gc.gv6.gv92[[gv196,1]];  
 lt.gv6.gv92[[ne,gv150,gv123[[gv196,1]];eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]];  
 df,gv99,gv99,gv6]];  
 eq,gv195,0[[gc.gv90,gv6[[gc.gv6.gv92[[gv195,1]];  
 lt.gv6.gv92[[ne,gv144,gv123[[gv195,1]];  
 eq,gv144,gv123[[ad,gv144,gv144,gv6;by,1,gv144]]];  
 df,bb,bb,gv6;gv98,bb]];  
 lt.gv90,gv6[[gc.gv6.gv92[[gv196,1]];lt,gv6.gv92[[ad,gv150,gv150,gv6;by,4,gv150]];  
 df,gv99,gv99,gv6]]]]];  
 eq,gv1,1[[gc.gv90,gv6[[gc.gv6.gv92[[gv196,1]];lt,gv6.gv92[[ne,gv150,gv123[[gv196,1]];  
 eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]];  
 df,bb,bb,gv6;gv99,bb]]];

**T**HIS SET OF FORMAT STATEMENTS will generate two-column pages with footnotes and allocate white space for subsequent “strip in” of graphics. The format statements B3 and B4 are used together to provide for a full, two-column chapter opening followed by two columns of text to complete the chapter opening page. The format statements B7 and B6 are the text block and footnote block bottom boundary procedures respectively and they must be equated with the synonym names Y1 and Y3 in the job specification.

White space is cited by assigning to gv6 a value equal to the depth of white space desired (in points) and then calling B8; the amount of white space requested should equal an integral number of text lines.

The footnote format statements appear and their use is described on page B11.

The two-column style generator (for use with these format statements) provides for user selection of two basic typographic formats. See discussion on page B11.

Figure 4: Some PAGE—1 definitions used in marking up text with two columns, footnotes, and allocated white space for subsequent “strip in” of graphics.

By the early 1970s, many reasonably sized businesses were using computers for support functions, especially for accounting, billing, inventory, and so forth. In addition to these functions, newspapers were also using small computers (IBM 1130s and DEC PDP-8s) programmed to perform hyphenation and justification (H&J) to increase productivity in the composing (typesetting) room.

Next, newspapers set out to do something far more ambitious: computerize the entire process of creating and producing their product. The news copy for the newspaper would be written, edited, formatted, and composed on interactive terminals. All of the “copy flow” between writers and editors would take place within the computer system. All classified ads would be taken, priced, and composed on the same system. Ultimately, all display advertising and all page makeup would be done using interactive graphic display terminals.

Figure 5 shows a computerized newsroom.



Figure 5: Atex terminals in *Newsday* newsroom on Long Island, 1977 [photo by John Seybold, courtesy of Jonathan Seybold].

In a 1991 report of the National Academy of Engineering, Wilson Locke gives a detailed description of the 1970s effort of the *Los Angeles Times* to computerize and the reasoning behind the effort.<sup>15</sup>

Living in the Boston area since 1964, I was aware of the existence of Atex, but I knew nothing about it until the May 2017 Computer History Museum desktop publishing pioneers meeting and the writings about Atex<sup>14</sup> that appeared in the first desktop publishing special issue of the *IEEE Annals of the History of Computing*.

Atex was started early in 1973 by Douglas Drane and brothers Charles and Richard Ying. They met Jonathan Seybold at a conference and Jonathan learned what they intended to do. Jonathan’s father John was consulting to *U.S. News and World Report* magazine and knew *U.S. News* was seeking a new all digital system like the Atex partners were planning. *U.S. News* gambled on Atex, providing upfront funding and the specification of the system they wanted. Atex got the initial system working on *U.S. News*’ tight deadline and over the

<sup>15</sup>Wilson R. Locke, Telecommunication in the News Industry: The Newsroom Before and After Computers, a chapter in *People and Technology in the Work Place*, National Academy of Engineering, Washington, D.C., 1991.

next few years supplied systems to many other companies and institutions, becoming the most popular supplier of computer systems for newspapers and periodicals.

Each customer installation was a custom system based on Atex's highly efficient and relatively inexpensive hardware and software configuration, including considerable hardware they developed themselves. The Atex systems could drive whatever phototypesetter the customer had. In time Atex was delivering a full editorial system including digital images and had many different systems in its product line (Figure 6).

Atex	
Atex 8000 (PDP-11/05 or 11/35, or 04 and 34). Computer also serves as controller for Atex-built terminals.	Release 4 software (1981). Spellcheck (1982).
9000 newspaper system.	5000 newspaper system (1982).
7000 newspaper system (April 1981).	Atex 1000 remote terminal (1981).
Models 7032, 7048 and 7064.	Atex 1500 "full-function" remote (1981).
4000 (see also AKI).	Atex 9080 remote cluster (1981).
4000S (with Release 4 software) 1984.	Atex 500 remote terminal (1982).
GT68 Graphics and pagination terminal. (Motorola 68000) (1982).	Atex library or morgue system.
Classified ad pagination.	Electronic library system jointly with Infotex (1981).
Release 3 introduced 1979.	Acquired by Eastman Kodak 1981.
Atex PC interface software supports virtually any PC running virtually any word processing and communication software.	TPE (Total Publishing Environment) products.

Figure 6: Atex product line circa 1985

Naturally competition for the Atex systems developed and managing the business became harder as the company became bigger. Thus, in 1981 the founders sold the company to Kodak. Charles Ying remained with the company until Kodak closed the part of the company for which he was working. After Atex, he stayed close to the publishing industry, for instance serving at different times as president of Information International and BitStream.<sup>16</sup>

### 3 Desktop publishing

Today desktop publishing is everywhere. However, in 1980 commercial typesetting for newspapers, periodicals, and books was still a separate domain, and commercial word processing products were a relatively new product and mainly not thought of as a tool for publishing.

#### 3.1 Xerox PARC

The commercial desktop publishing market developed over the 1980s. However, the much of the technology enabling what we now think of as desktop publishing was developed in the 1970s. Of course, computing and electronics technology had been evolving for a long time with the work of many companies and people. But a surprising amount of the relevant technology was demonstrated by Xerox Corporation, particularly at Xerox Palo Alto Research Center (PARC).

Being a research organization in Xerox, PARC did development that (nominally) related to computerization of the office. Below is a list of some of what they developed—at least prototypes and sometimes distributed fairly widely within Xerox or outside the

<sup>16</sup>Andrew Tribute, Charlie Ying 1946–2010, *What They Think*, October 20, 2010, [whattheythink.com/articles/47263-charlie-ying-1946-2010/](http://whattheythink.com/articles/47263-charlie-ying-1946-2010/)

company.<sup>17,18,19,20,21</sup>

- The Alto networked (via Ethernet) workstation (1973) with a raster display which therefore could provide a graphical user interface.
- Laser xerographic printers that could print high resolution bit maps for output pages.
- Printer servers on the local area network (the Electronic Array Raster Scanner, EARS).
- “Press files” that could intermingle text and graphics.
- The Fred program to create (on the Alto) outline fonts for printing and display using cubic splines.
- The Draw program to create figures made up of text, lines, and curves, again using cubic splines.
- The Press program to print Press files.
- The Bravo and Gypsy WYSIWYG editors.
- Interpress page description language (the predecessor of PostScript).

The above technology was known outside of PARC and thus aspects of it were highly influential as the desktop publishing world developed. It also fed into the word processing world.

### 3.2 Seybold Reports and Seminars

I began the story of the Seybolds’ activities in the publishing world (starting in 1963) in Rocappi section (section 2.3 on page 4). Another way the world got ready for desktop publishing was through the activities of the Seybolds throughout the 1970s during which they found ways to keep current about and to push forward publishing technology.

The sketch in this section is taken from Jonathan Seybold’s paper about the Seybold reports and seminars.<sup>22</sup>

#### Seybold Reports

After leaving Rocappi, John Seybold remained on the east coast and established a company, John W. Seybold & Associates, to consult to publishing companies interested in applying computing technology. Jonathan Seybold moved to southern California where he initially helped Autologic company by specifying the typographic capabilities of their new APS-4 phototypesetting system. By the spring of 1971, John and Jonathan had begun to discuss writing another book but decided the publishing world was moving too fast for a book. A better idea would be a bi-monthly “newsletter,” except that it would only contain in-depth analysis. The first year, the Graphic Communications Computer Association (GCCA) operated the newsletter, which they insisted would be called *The Seybold Report*, while the Seybolds provided all the content, typically a single long article on one product or product line or occasionally a tutorial on important technology or market trends. The Seybolds took great care to be accurate and conflict free while producing issues of 12–16 pages of typewriter copy. About six months in, they added some pages of news at the back of each issue, so there was something of interest of readers not interested in the feature report in the issue. Industry trade shows provided good sources of news.

For the second year the Seybolds took over the business of the Report themselves with John handling the business and Jonathan handling the “intellectual side.” This was the

---

<sup>17</sup>Gardner Hendrie interview of Gary Starkweather, Computer History Museum, 2010, [computerhistory.org/collections/catalog/102702465](http://computerhistory.org/collections/catalog/102702465)

<sup>18</sup>Robert F. Sproull, The Xerox Alto Publishing Platform, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 38–54.

<sup>19</sup>John E. Warnock, The Origins of PostScript, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 68–76.

<sup>20</sup>Lawrence G. Tesler, How Modeless Editing Came To Be, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 55–67.

<sup>21</sup>Robert F. Sproull, Publishing a Computer Graphics Book With Prototype Desktop Publishing Tools, *IEEE Annals of the History of Computing*, vol. 40 no. 4, 2018, pp. 69–76.

<sup>22</sup>Jonathan W. Seybold, Seybold Publications and Seminars, *IEEE Annals of the History of Computing*, vol. 41 no. 3, 2019.

start of Seybold Publications. The Report then expanded to 20-24 phototypeset pages. By the end of the second year, 25 percent of subscribers were from outside the U.S.

Seybold Publications, i.e., John and Jonathan, also gave two-day tutorials several times a year on latest developments in the industry. These were arranged in the U.S. by GCCA and by the Printing Industry Research Association in the U.K. As with the trade shows, the tutorials were an opportunity to meet and get to know more people thus building their network and knowledge. They also continued their consulting work they had begun before starting the Report, between them for little newspaper groups, big technology companies, and big publishing companies.

While the Report's subscription base and other aspects of the Report grew, it never made much money. Without advertising, the report was priced in the hundreds of dollars which limited potential subscribers and encouraged reading of a subscription by multiple people and even piracy. However, Jonathan says,

The Report was highly successful in achieving its primary objective: The technological base of an entire industry was being re-made in a single decade. . . . We were right in the middle of all of this. I like to think that we played an important and constructive role in helping to shape how it all came out.

By the early 1980s, Jonathan's sister Patty had started the *Seybold Report on Word Processing* and a little later Jonathan and Patty started a report on personal computers, and to sort this all out, they renamed the reports, i.e.,

- *The Seybold Report on Publishing Systems*—the original Seybold report
- *The Seybold Report on Office Systems*—the prior report on word processing
- *The Seybold Report on Professional Computing*—the PC report

When John Seybold eventually retired, Jonathan kept the publishing report, Patty kept the office systems report, and another son Andy took the computing report.

John Seybold had been “a true pioneer in automated typesetting.”<sup>23</sup> Frank Romano in his dedication to his book on the phototypesetting era continues, saying of Seybold: Rocappi served “as the world's first commercial computer typesetting service bureau.” Seybold published books on “the new typesetting machines, companies, concepts, and applications.” That Seybold was first to apply “what you see is what you get” to “display screen applications.” “He played a key role in the decision by *U.S. News and World Report* to become the first customer for the Atex Publishing System.” And with Jonathan, John created the *Seybold Report*, which (page 296 of Romano's book) John called, “a book that had to be constantly updated” and promised to cover every photocomposition and text editing device on the market”, and “they tested every system and reported their results and critiques.”

## **Seybold Seminars**

As Jonathan saw the coming world of what became desktop computing, he started the Seybold Seminars as a way . . .

. . . to get the people involved together for a conference designed to encourage interchange.

Four 1½ hour sessions per day with generous time for a group lunch and generous morning and afternoon breaks to encourage lots of informal interaction. Two presentations per session. No sales pitches.

The seminar was an annual event and, realizing Jonathan's goal, became a place where developers of desktop publishing systems and other relevant parties got together.

From near the beginning some attendees of the Seybold Seminars wanted to bring equipment to show. In 1986 Jonathan put together a desktop publishing conference that included a trade show. He also launched a new *Seybold Report on Desktop Publishing* at

---

<sup>23</sup>Reference 3, Dedication page.

the same time. In time, the original conference also grew into a trade show, with one held annually in San Francisco and the other (a little smaller) held in Boston. Both combined the conference with the trade show with the former having a few thousand attendees and the latter having tens of thousands of attendees. Also, as the technology evolved, the distinction between professional publishing and desktop publishing disappeared. The Seybold Seminars continued to expand.

Seybold and his people had always helped the press when asked for answers or pointers to other people. This was consistent with their mission to help the industry change happen. (Being quoted in the press was also good PR.) In time they established an explicit press liaison office to help the press. The same staff members also helped the PR people in the companies they dealt with who might be inexperienced and need pointers within the industry.

In 1989 Jonathan established a *Digital World* conference, independent of the publishing conferences, for people interested in the ever increasingly digital world. Naturally, a monthly publication came next — *Digital Media*.

In 1990 Jonathan sold the Seybold Seminars and Seybold Publications to Ziff, while he continued working in the activities for the next few years. He then left. Of this he says,

For me, it had been a great ride for a quarter century. I was able to play a role in three successive revolutions: the computerization of the print publishing industry, the democratization of publishing (desktop publishing), and then helping a little to lay the foundations for our current Digital World.

On pages 296 and 297 of his book,<sup>3</sup> Frank Romano vouches for Jonathan's role in helping create the revolution discussed in the next section.

### 3.3 Commercial desktop publishing

The systems to be discussed in this subsection are what we now call desktop publishing systems — DTPs. For many people, the definition of DTP involves a WYSIWYG interface running on a desktop computer. To my mind, the work stations from SUN, Apollo, etc., on which some of the systems in this section initially ran were the early versions of today's personal computers.

The companies mentioned in the following sketch of creation of the desktop publishing business are Adobe,<sup>24,25,26</sup> Aldus,<sup>27,28</sup> Apple,<sup>29</sup> Frame Technology,<sup>30,31,32</sup> Interleaf,<sup>33,34</sup> Quark,<sup>35</sup> and Ventura.<sup>36</sup> There were also other companies that I won't mention.

In 1981 the Seybold Seminars (as described in the prior section) had been started as a way for people to get together who might advance the use of digital technology for publishing; many of the involved people also subscribed to the *Seybold Report* or were reading issues from other people's subscriptions. Seybold was constantly scouting out

<sup>24</sup>John E. Warnock, The Origins of PostScript, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2018, pp. 68-76.

<sup>25</sup>Pamela Pfiffner, *Inside the Publishing Revolution: The Adobe Story*, Peachpit Press, Berkeley, CA, 2003.

<sup>26</sup>John Warnock and Charles Geschke, Founding and Growing Adobe Systems Inc., *IEEE Annals of the History of Computing*, vol. 41 no. 3, 2019.

<sup>27</sup>Paul Brainerd oral history: [tinyurl.com/brainerd-oralhistory](http://tinyurl.com/brainerd-oralhistory)

<sup>28</sup>Suzanne Crocker, Paul Brainerd, Aldus Corporation and the Desktop Publishing Revolution, *IEEE Annals of the History of Computing*, vol. 1 no. 3, 2019.

<sup>29</sup>John Scull and Hansen Hsu, Desktop Publishing: The Killer App That Saved the Macintosh, *IEEE Annals of the History of Computing*, vol. 41 no. 3, 2019.

<sup>30</sup>Charles Cornfield interview: <http://tinyurl.com/cornfield-interview>

<sup>31</sup>David J. Murray, Frame Technology and FrameMaker, *IEEE Annals of the History of Computing*, vol. 1 no. 3, 2019.

<sup>32</sup>David Murray, The FrameMaker Document Model, <https://history.computer.org/annals/dtp/>

<sup>33</sup>Four Interleaf documents and three books may be found at [bitsavers.org/pdf/interleaf/](http://bitsavers.org/pdf/interleaf/)

<sup>34</sup>Paul M. English, Interleaf active documents, *Electronic Publishing*, vol. 7 no. 2, June 1994, pp. 75-87, [researchgate.net/publication/228057523\\_Interleaf\\_Active\\_Documents](https://www.researchgate.net/publication/228057523_Interleaf_Active_Documents).

<sup>35</sup>Jay Nelson, Interview with Tim Gill, *IEEE Annals of the History of Computing*, vol. 41 no. 3, 2019.

<sup>36</sup>Lee Lorenzen, The Ventura Story, *IEEE Annals of the History of Computing*, vol. 41 no. 3, 2019.

what was happening in the publishing and publishing technology industry for his reports and seminars; he tended to know what everyone was doing. Also, by 1980 outline fonts were available to publishers from Linotype and Compugraphic, but bitmapped fonts were still typically used for laser printing, screen display, and in the newspaper industry.

Adobe was founded in 1983 to push Geschke and Warnock's vision for a page description language that Xerox PARC had not been interested in pursuing, and in 1983 they were able to demonstrate a prototype PostScript laser printer. In 1984 Adobe released Level 1 PostScript. Along with this came Adobe's Type 1 and Type 3 fonts. Adobe also did a deal to use ITC fonts in PostScript. With PostScript, outline fonts began to spread for laser printing and screen display.

Also in 1984 an Apple Mac computer with a graphical user interface was available. Steve Jobs had excitedly shown Jonathan Seybold a Mac the year before, and in 1983 Jobs called Seybold back to Apple to show him a Mac connected to a LaserWriter with built-in Postscript. Apple and Adobe had done a deal about PostScript and raster output devices. Seybold says that seeing the Mac-Laserwriter-PostScript combination told him that a revolution in the publishing world was immanent; he also knew what Aldus Corporation was doing.

Aldus Corporation was founded in 1983 by Paul Brainerd. Out of college he worked in operations for the *Minneapolis Star and Tribune* while they converted from hot type to computer-based typesetting. Atex was a key supplier. Next Brainerd went to Atex and stayed there until it was sold. Then he started Aldus which created PageMaker, initially for the Mac. Brainerd is credited with coining the term "desktop publishing." PageMaker was aimed at small businesses and also used by professional and amateur book designers and others. Jonathan Seybold encouraged Brainerd to get together with the right group at Apple to see the Mac with its Postscript laser printer and also encouraged people at Apple to talk to Brainerd.

The deal between Apple and Adobe resulted in the 1985 product release of the Laserwriter with built-in PostScript with Adobe's Type 1 and 3 font technology. In 1985 Aldus PageMaker for "desktop publishing" also was released, and a group at Apple, people at Adobe, and Aldus people began an informal collaboration marketing desktop publishing to small businesses. Apple needed sell Macs, Adobe got paid for every PostScript Laserwrite that Apple sold, and Aldus sold its PageMaker software package. It was in everyone's interest to help each other selling this "desktop publishing solution". They found a significant untapped market, and pushing desktop publishing was a major benefit for all three companies. (Adobe was also licensing PostScript to other printer and computer manufacturers.)

Interleaf also released a desktop publishing system in 1985—their Interleave Technical Publishing Software (TPS) (the company had been founded in 1981). Their product was aimed at technical publishing and distribution with integrated text and graphics. It originally ran on Sun and Apollo workstations. The system was programmed in Lisp (Interleaf Lisp), and users could modify the system.<sup>37</sup> By 1987 Interleaf was also running on more workstations, on the Mac, and under Windows.

Also in 1986, Ventura Publisher for the IBM PC was released by Ventura Software which had been founded the previous year. The founders felt they had an innovative way (better than PageMaker) to layout multi-story documents; their system also became the first popular desktop publishing system for the IBM PC class of computers.

Yet another desktop publishing company was founded in 1986—Frame Technology. They released FrameMaker for the SUN and other Unix workstations. Charles Corfield had developed FrameMaker which was a competitor of Interleaf and was aimed at publishing large and very large and complex documents. (David Fuchs was the fifth employee of the company after the four founders.)

In 1987 QuarkXPress 1.0 for the Mac was released aimed a high-end publishing—the

---

<sup>37</sup>Karl Berry was with Interleaf for a while. Some readers may remember author Tracy Kidder attending the 2014 TUG conference in Portland, OR. Kidder wrote a book called *A Truck Full of Money*, published in September 2016, that talks a good bit about Interleaf as a development organization and business.

professional typesetting and page layout market — and it competed with PageMaker in that Market. (Quark had been founded in 1981 and did other things before going after desktop publishing.) Within only two or three years, QuarkXPress was putting serious pressure on PageMaker and Aldus. Aldus brought out successive versions of PageMaker but also broadened its product line through acquisitions of other products.

By 1989 the tiny staff of Ventura was getting tired (they never had more than five employees), and in 1990 Ventura was acquired by Xerox which had been the distributor of Ventura Publisher from the beginning. Three years later, Xerox sold the Ventura business to Coral, which continues to sell the system today as Coral Ventura. Also in 1990, Interleaf was sold to Broadvision which renamed the product Quicksilver. In parallel Quark brought out QuarkXPress 3.1 for Windows, and Quark became the dominate player and QuarkXPress was the industry standard in the market.

By 1993 PageMaker had lost lots of market share to Quark and other desktop publishing systems. With such increasing competition for PageMaker and Aldus' other products not generating enough sales for the company to continue its early extraordinary growth and profit, in 1993 Paul Brainerd initiated talks with Adobe and in 1994 Aldus was acquired by Adobe.

Meanwhile, Frame Marker had been made to run on Unix, Macs, and Windows PCs, and the company tried to also compete in the home desktop publishing market which was a loss of business focus and led to near insolvency. Adobe bought the product in 1995, refocused on the business market, and the product still has a significant following today.

QuarkXPress 4.0 continued Quark's market dominance. Adobe countered by rewriting PageMaker and brought the resulting package out as InDesign 1.0. Over time InDesign cut deeply into Quark's market, although I think there is still competition between Quark and InDesign today. InDesign is used by professional book designers and typesetters (and by amateurs who want good typesetting and would never think of using L<sup>A</sup>T<sub>E</sub>X).

Desktop publishing tapped a large market that effectively included consumer products as well as products for professionals. A common method of document interchange, both layout and fonts, was a natural outgrowth. Adobe and PostScript won the battle for dominance over other companies and technologies. The competition for digital font technology dominance resulted in a compromise.

By 1986, Adobe had pushed PostScript into graphics applications, and a couple of years later brought out Encapsulated PostScript (EPS) graphics format.

In 1989 Apple and Microsoft began what is known as the "font wars" by claiming at a Seybold conference that their (unfinished) TrueType font technology was superior to PostScript font technology. Naturally, Adobe forcefully disagreed, and Adobe brought out Adobe Type Manager for Mac, Windows, and other operating systems to counteract TrueType. Apple's first release of TrueType was in 1991, and Microsoft released TrueType for Windows 3.1 in 1992. In parallel with the competition about font technology, Adobe kept pushing PostScript, bringing out PostScript Level 2 in 1991, and Acrobat and Portable Document Format (PDF) in 1993. With OpenType in 1996, Adobe, Apple, and Microsoft combined their competing font technology approaches.

I have heard PostScript being described as a page description language or as a language for creating vector graphics. It was originally aimed at driving printers and first became well known by its use in Apple's computers. PostScript (and EPS and PDF) have clearly changed the way the typesetting and printing worlds work.

In about a dozen years, the desktop publishing market had developed and then consolidated.

## **Next step and acknowledgments**

I do intend to finish the monograph I mentioned at the beginning of this paper. The monograph will include relevant parts of my TUG2012 and TUG2016 papers, the content

of this paper, and a bit more.<sup>38,39</sup> It will be posted at [walden-family.com/texland/digitype-monograph.pdf](http://walden-family.com/texland/digitype-monograph.pdf)

Over the past half a dozen years, dozens of people have answered questions about the topic of this paper (and its 2016 predecessor) or otherwise helped me with the paper(s). I greatly appreciate the help of each of them.

I must specifically acknowledge Burt Grad who invited me to participate in the May 2017 desktop publishing pioneers meeting at the Computer History Museum.

In the past several years, I have also had more or less frequent contact on a variety of topics with Barbara Beeton, Karl Berry, Chuck Bigelow, David Hemmendinger, and Jonathan Seybold. Each contributed to my education on the history of digital typography.

I welcome feedback on this draft: [dave.walden.family@gmail.com](mailto:dave.walden.family@gmail.com)

---

<sup>38</sup>Charles Bigelow, Font Wars parts 1 and 2, *IEEE Annals of the History of Computing*, to be published in early 2020.

<sup>39</sup>David Walden, Interview of Charles Bigelow, *IEEE Annals of the History of Computing*, vol. 40 no. 3, 2019, pp. ???.