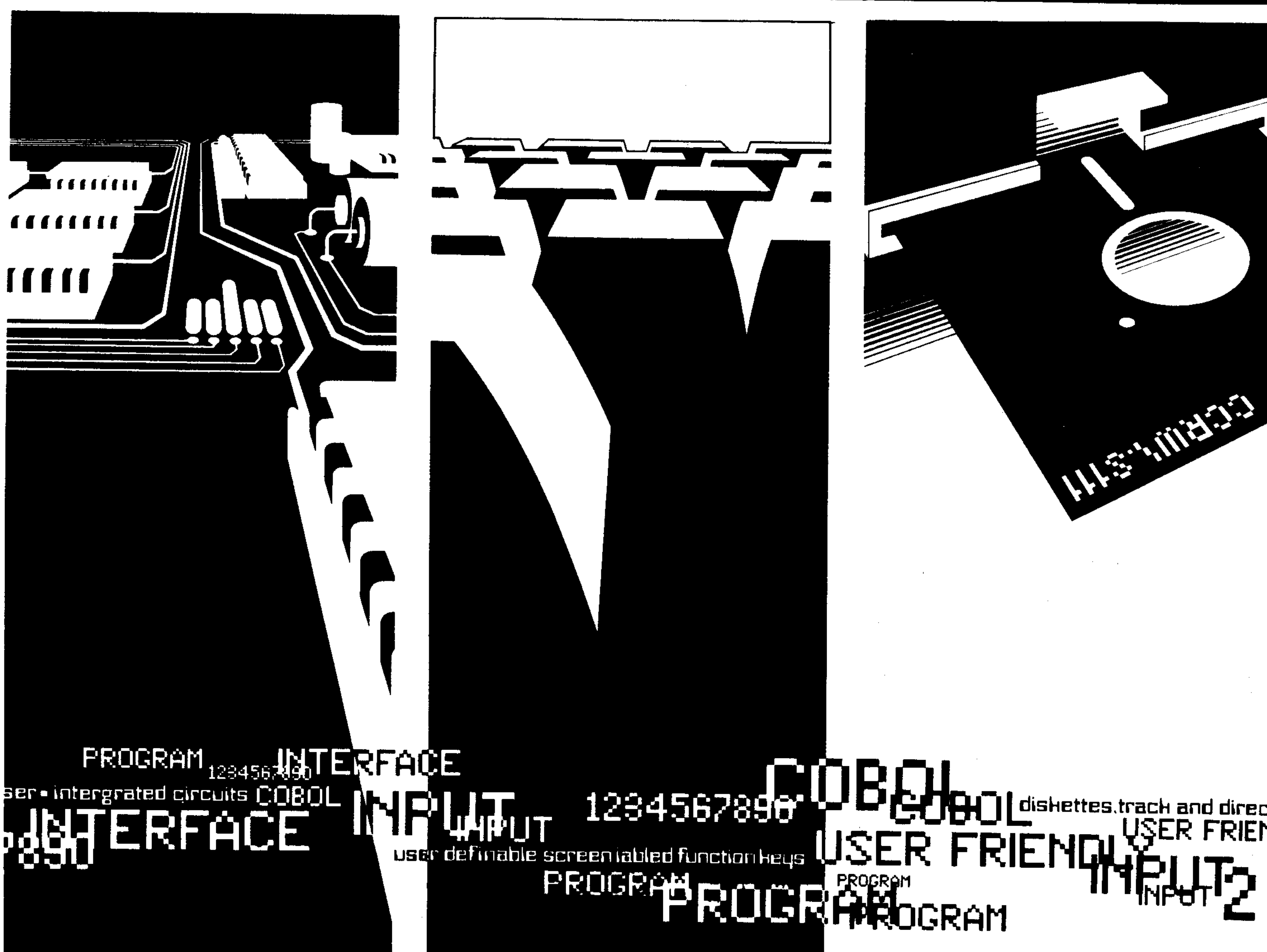
RESEARCH IN

WORD PROCESSING

NEWSLETTER

South Dakota School of Mines and Technology Rapid City, SD 57701 (605) 394-2481
Volume 5 Number 6 September 1987

UPCOMING COMPUTER CONFERENCES
see page 8



RESEARCH IN
WORD PROCESSING
NEWSLETTER

Research in Word Processing Newsletter. Volume 5, Number 6. Copyright © 1987 by the South Dakota School of Mines and Technology. All rights reserved. ISSN: 0748-5484. Also indexed by **ERIC, INSPEC, COMPUTER LITERATURE INDEX,** and **SOFTWARE REVIEWS ON FILE.** The *Research in Word Processing Newsletter* is published 9 times a year (September through May) by the South Dakota School of Mines and Technology, Rapid City, South Dakota 57701-3995; (605)394-2481. Postage paid at Rapid City, South Dakota. **SUBSCRIPTION:** $15 per year (U.S.); $21 per year (Canada); $27 per year (foreign). Address all subscription inquiries and manuscripts to the Editors, *Research in Word Processing Newsletter,* SDSM&T, 501 E. St. Joseph, Rapid City, SD 57701-3995. Please allow 4 to 6 weeks for subscription processing. **POSTMASTER:** Send address changes to RWPN, South Dakota School of Mines and Technology, 501 East St. Joseph, Rapid City, South Dakota 57701-3995.

Building Text Filters in *Turbo Pascal*

Joe Weixlmann

Today's widespread use of word processing means that the production of quality typescripts is a simpler matter than it was in a less technologically sophisticated era. The ability to encode and manipulate text electronically has made writing and revision easier; computerized spelling checkers and grammatical programs help authors to make their essays more mechanically ept; and those submitting typescripts to would-be publishers are able to produce clean copy at the push of a button. Until recently, though, authors and publishers have made little use of the computer's ability to massage electronically encoded data in order to produce printed magazine or book copy directly.

The way in which we handle accepted essays at the editorial offices of *Black American Literature Forum* typifies a burgeoning movement in the publishing industry. For over a year, we have been encouraging writers to submit electronic copy, and we have recently purchased an inexpensive optical character reader in order to further limit the amount of text that we must enter manually. Some publishers now go so far as to *require* that accepted works be presented in electronic form, and such demands are likely to escalate, especially within academic publishing, where typesetting can account for twenty-five percent or more of a book's or journal's production costs. Indeed, academic publishers who fail to use the new technology may be committing fiscal suicide.

Publishers who "set" type from the electronic texts they receive face three major problems: (1) converting the data sent by their authors into a format which their word processor can read; (2) converting the codes which their word processor uses to produce italics, superscripts, boldface characters, and the like into symbols a typesetting machine can interpret; and (3) changing their word-processed files into "pure" ASCII files which a typesetting machine can read.[1] Many word processors come with utility programs that do a limted amount of the first kind of conversion; *WordPerfect*, for example, lets one read a variety of file types in and out, including those in IBM mainframe DISOSS format as well as those created in *WordStar* and *MultiMate*. And even more versatile standalone programs exist.

The programs and techniques discussed in this article assume that—directly, or following conversion—one is working with *WordStar* files, although the programs' logic may be readily applied to other file types.[2] An extremely popular program, *WordStar* makes extensive use of "high-bit" characters, those with ASCII values above decimal 127 (hexadecimal 7F). As a consequence, *WordStar* files require substantial massaging before a computerized typesetting machine can make direct use of them. WS_CNVRT and WS_STRIP address the second and third tasks enumerated above: the former converts print codes; the latter produces pure ASCII files. Written in Version 3.0 of *Turbo Pascal*, they are extremely fast—approximately twenty-five times quicker than related utilities which come packaged with the Compugraphic 8400 compositor our University printshop uses.

The WS_CNVRT Filter

WordStar users are accustomed to seeing symbols like ^S (which denotes underscoring), ^B (boldface), and ^T (superscripts) on their screens. The main function of WS_CNVRT is to convert these and other embedded *WordStar* print commands into codes that computerized typesetting equipment understands. Translations of boldface, italic, boldface italic, superscript, subscript, and non-break space symbols are all provided for in WS__CONVRT. Moreover, since computerized typesetting machines ordinarily do not permit the use of more than one space between words and sentences (blocks of space require special codes), WS__CNVRT eliminates multiple spaces within paragraphs and converts spaces at the beginning of a paragraph to an indentation code. Although the program appears rather lengthy and would seem to analyze a file a byte at a time, it is extremely fast. My IBM PC, running a Seagate ST-4038 hard disk drive, processes a 30K file in less than half a minute.

Commented source code for the utility follows:

```
{$G16384,P16384,D-}                                  {I/O redirection statement}

program WS_Code_Converter;                                       {program name}

var  ch,                                {character and integer variables declared}
     ch1,
     ch2,
     ch3,
     ch4   : Char;
     boldflag,
     spaceflag,
     subflag,
     superflag,
     italflag  : Integer;

procedure  Boldface;                                       {boldface subroutine}

  begin
    boldflag := boldflag + 1;                           {increment boldface flag}
    ch3 := ^@;                                                       {remove ^B}
    if ch4 = chr(19) then                               {if preceding ^S located}
      begin
        ch4  := ^@;                                {remove ^S and set italic flag}
        if italflag = 0 then
          italflag := 1
        else
          italflag := 0;
      end; {if}
    if ch2 = chr(19) then                                {if trailing ^S located}
      begin
        ch2 := ^@;                                 {remove ^S and set italic flag}
        if italflag = 0 then
          italflag := 1
        else
          italflag := 0;
      end; {if}
    if ch4 >= chr(32) then
      write (ch4);   {the 3rd character has been trapped, so write the 4th now}
    ch4 := ^@;                                  {prevent its being rewritten later}
    if boldflag = 1 then    {based on flag settings, mark bold or bold italics}
      if italflag = 1 then
        write ('<bold italics on>')
      else
        write ('<bold on>')
    else                             {or turn off bold and mark italics or roman}
      begin
        boldflag := 0;
        if italflag = 1 then
          write ('<italics on>')
```

```
      else
        write ('<roman on>');
    end; {if}
end; {proc}

procedure Non_Break;                                    {non-break space subroutine}

  begin
    ch4 := ^@;                                                          {remove ^O}
    write ('<non-break space code>');              {insert <non-break space code>}
end; {proc}

procedure Underscore;                                           {italics subroutine}

  begin
    italflag := italflag + 1;                              {increment italics flag}
    ch4 := ^@;                                                          {remove ^S}
    if italflag = 1 then {based on flag settings, mark italics or bold italics}
      if boldflag = 1 then
        write ('<bold italics on>')
      else
        write ('<italics on>')
    else                             {or turn off italics and mark bold or roman}
      begin
        if boldflag = 1 then
          write ('<bold on>')
        else
          write ('<roman on>');
        italflag := 0;
      end; {if}
  end; {proc}

procedure  Superscript;                                     {superscript subroutine}

  begin
    superflag := superflag + 1;                        {increment superscript flag}
    ch4 := ^@;                                                          {remove ^T}
    if superflag = 1 then  {based on flag setting, enter <superscript on code>}
      write ('<superscript on code>')
    else                        {or enter <superscript off code> and reset flag}
      begin
        write ('<superscript off code>');
        superflag := 0;
      end; {if}
  end; {proc}

procedure  Subscript;                                         {subscript subroutine}

  begin
    subflag := subflag + 1;                              {increment subscript flag}
    ch4 := ^@;                                                          {remove ^V}
    if subflag = 1 then      {based on flag setting, enter <subscript on code>}
      write ('<subscript on code>')
    else                          {or enter <subscript off code> and reset flag}
      begin
        write ('<subscript off code>');
        subflag := 0;
      end; {if}
  end; {proc}
```

```
procedure  Spacing;                                        {multiple-space subroutine}

  begin
    if ch4 = chr(13) then           {if spaces follow carriage return_line feed}
      begin
        write (#13#10,'<indent code>');       {send cr_lf and mark <indent code>}
        ch4 := ^@;                                    {cancel written characters}
        ch3 := ^@;
        ch2 := ^@;
        chl := ^@;
        spaceflag := 1;                                          {set space flag}
      end {if}
    else                                        {if cr_lf does not precede spaces}
      ch2 := ^@;                                         {delete one of the spaces}
  end; {proc}

begin {body of program}
  ch3 := ^A;                                                {initialize variables}
  ch2 := ^A;
  chl := ^A;
  boldflag := 0;
  spaceflag := 0;
  subflag := 0;
  superflag := 0;
  italflag := 0;
  while not eof do begin                                             {main loop}
    Read (ch);                                                    {read in data}
    ch4 := ch3;
    ch3 := ch2;
    ch2 := chl;
    chl := ch;
    if ord(chl) <> $8D then              {except for a mid-paragraph line feed}
      chl := char(ord(chl) and 127);      {express character in "low-bit" form}
    if ch3 = chr(2) then                                         {if ^B located}
      Boldface;
    if ch4 = chr(15) then                                        {if ^O located}
      Non_Break;
    if ch4 = chr(19) then                                        {if ^S located}
      Underscore;
    if ch4 = chr(20) then                                        {if ^T located}
      Superscript;
    if ch4 = chr(22) then                                        {if ^V located}
      Subscript;
    if spaceflag = 1 then                {eliminate spaces after <indent code>}
      if chl <> chr(32) then
        spaceflag := 0
      else
        chl := ^@;
    if (chl = chr(32)) and (ch2 = chr(32)) then       {if double space located}
      Spacing;
    if (ch4 >= chr(32)) or (ch4 = chr(10)) or (ch4 = chr(13)) then
      write (ch4);                          {most characters written to file here}
  end; {while}
  write (ch3);      {write final three characters to file after loop completed}
  write (ch2);
  write (chl);
  write (^Z);                                                       {close file}
end. {program}
```

Those who understand Pascal programming should be able to read and understand the program. Those who do not need most to know that, after performing minor modifications, they can compile this program and use it. I have used the expressions <bold italics on>, <bold on>, <italics on>, <roman on>, <non-break space code>, <superscript on code>, <superscript off code>, <subscript on code>, <subscript off code>, and <indent code> *in lieu of* the specific codes demanded by specific typesetting machines. For example, the printshop I use employs the code <FT45> to turn on the roman font of the type in which the journal I edit is printed. <FT46> turns on italics; <FT44> turns on boldface italics; <P, followed by a digit, prints a superscript number, and a trailing > terminates the superscript mode; etc. These codes will vary from printshop to printshop. Writers need to check with their publishers before embedding specific codes; publishers need to check with their printshops. The search-and-replace feature of *Turbo Pascal* (invoked with ^QA) can be used to make the program changes once one knows the requisite codes. The program should then be compiled as a .COM file with the *Turbo* compiler.

Some remarks about the program are in order for those who would like to tinker with it in order to incorporate other features (e.g., the handling of quotation marks, dashes, and the like). The speed of the program resides in its first line. *Turbo Pascal 3.0* permits the redirection of input and output using G (get) and P (put) compiler directives. The first line of WS_CNVRT permits 16K (or 16,384 bytes) of data to be read through the filter at a time. Second, the way in which I delete a character is by changing its value to ^@ (decimal 0), the null character. Third, in the body of the program, after the characters are read into the file and incremented, I reduce all high-bit characters except decimal 8D, which keeps the text in paragraph form, to their low-bit equivalents to simplify data handling.[3] Finally, the eighth line from the end of the program allows only characters equal to or greater than decimal 32 (a space) to pass through the filter, along with decimal 10 (line feed) and decimal 13 (carriage return).

The syntax for using the WS_CNVRT program is: WS_CNVRT <oldfilename >newfilename. One wishing to convert the print codes of a file named THISESSY would, at a DOS prompt, type WS_CNVRT <THISESSY >THATESSY and hit the return key. A short time later, the converted file, THATESSY, would be ready for copy editing. The copy editing completed, an editor would want to send the file to the printer's, and a program like WS_STRIP would be necessary in order to ready the file for transmission.

The WS_STRIP Filter

WS_STRIP allows the user to reduce an essay to a one-line, pure-ASCII text file. The program's main function is to strip away all of the carriage returns and line feeds within and between paragraphs and to substitute codes that a typesetting machine requires. (A stripped file would be a nightmare to read in *WordStar*: It would begin at the left-hand side of the first line and run well off the right-hand side of the screen.) WS_STRIP also strips away "soft hyphens" and multiple spaces, and permits *only* alphanumeric data and common symbols—characters equal to or greater than decimal 32 (a space)—to pass through the filter. WS_STRIP takes slightly less time to run than WS_CNVRT.

Commented source code for the utility follows:

```
{$G16384,P16384,D-}                                {I/O redirection statement}

program  WS_Text_Stripper;                                        {program name}

var   ch,                                {character and integer variables declared}
      ch1,
      ch2,
      ch3,
      ch4    : Char;
      flag   : Integer;

procedure  Para_LFs;                {subroutine to handle mid-paragraph line feeds}

  begin
    ch2 := ^C;                  {change the line feed to ^C, which becomes a flag}
    ch1 := ^C;
    if ch3 <> ^C then       {recognizes only the first in a series of line feeds}
      if (ch3 <> chr(32)) and (ch3 <> chr(45)) then       {if not space or hyphen}
```

```
      begin
        if ch4 >= chr(32) then              {write any preceding alphanumerics}
          write (ch4);
        if ch3 >= chr(32) then
          write (ch3);
        ch4 := ^@;                              {cancel the written characters}
        ch3 := ^@;
        write (#32);                                             {add a space}
      end; {if}
  end; {proc}

begin {body of program}
  ch3 := ^A;                                          {initialize variables}
  ch2 := ^A;
  chl := ^A;
  while not eof do begin                                       {main loop}
    Read (ch);                                              {read in data}
    ch4 := ch3;
    ch3 := ch2;
    ch2 := chl;
    chl := ch;
    if ord(chl) <> $8D then              {except for a mid-paragraph line feed}
      chl := char(ord(chl) and 127);     {express character in "low-bit" form}
    if (chl = chr(30)) or (chl = chr(31)) then     {if a "soft hyphen" located}
      chl := ^@;                                                {remove it}
    if ord(ch2) = $8D then               {if a mid-paragraph line feed located}
      begin
        Para_LFs;
        flag := 1;                    {set flag to monitor for trailing spaces}
      end; {if}
    if flag = 1 then            {eliminate spaces after mid-paragraph line feed}
      if chl = chr(32) then
        chl := ^@
    else
      if chl <> ^C then
        flag := 0;
    if (chl = chr(32)) and (ch2 = chr(32)) then      {if double space located}
      ch2 := ^@;                                   {delete one of the spaces}
    if ch4 = chr(13) then                        {if a carriage return located}
      write ('<cr_lf code>');                               {mark <cr_lf code>}
    if ch4 >= chr(32) then
      write (ch4);                    {alphanumeric characters written to file here}
  end; {while}
  if ch3 >= chr(32) then           {if the last three characters are alphanumeric}
    write (ch3);                                       {write each to the file}
  if ch2 >= chr(32) then
    write (ch2);
  if chl >= chr(32) then
    write (chl);
  write (^Z);                                                   {close file}
end. {program}
```

Only one modification would need to be made to make the source code listed above executable: Replace <cr_lf code> with the code(s) that your printshop requires to mark a carriage return/line feed, and the program is ready to be compiled as a .COM file. The execution syntax of WS_STRIP is the same as that for WS_CNVRT: WS_STRIP <THATESSY >DONEESSY.

Conclusion

Individuals as well as publishers involved in "electronic typesetting" can benefit tremendously from text filters that permit them to automate text massaging, and *Turbo Pascal*'s speed and versatility make it an excellent language in which to write such utilities. Indeed, the two utilities discussed in this article demonstrate but a few of the ways in which the power of *Turbo Pascal*'s I/O redirection capabilities might be tapped.

Notes

[1]In an effort to avoid these problems, and to accommodate authors who lack access to a word processor, some publishers prefer to automate their typesetting by eschewing electronic media and, instead, reading typescripts into the computer with a sophisticated optical character reader. Ordinarily, though, small publishers do not have access to a Kurzweil scanner, which costs about $40,000, or a kindred piece of hardware. Moreover, even if they did, a new set of problems would surface: The publisher might still need to "teach" the machine to handle the type font to be read and would need to add italic, boldface, and related codes to the computer file once the text had been transmitted.

[2]In order to build text filters for a variety of word processing programs, one needs to learn which codes a particular word processor uses both to express alphanumerics and to mark line feeds, carriage returns, print commands, and the like. The tool for uncovering this information is the DEBUG program in DOS, which permits one to "dump" text in hexadecimal code, with an accompanying display of alphanumeric characters. Instructions for using DEBUG appear in the DOS manual. Ordinarily, one begins by dumping a file at address 100.

[3]When the *WordStar* user enters the letter *f*, it appears on the screen. However, if, using the DEBUG program in DOS, one examines a word like *fief* in a text file, the first *f* will almost certainly be coded as hexadecimal 66 (decimal 102), whereas the second may be coded as hexadecimal E6 (decimal 230, the high-bit equivalent of *f*). The line *chl := char(ord(chl) and 127* assures that both f's will be expressed at the code level as hexadecimal 66. (Although one doesn't need to understand this line to use it, I might note that its success rests on the concept of Pascal's "bitwise' and" operator. Anyone interested in pursuing the point can find a lucid explanation in Jeff Duntemann's *Complete Turbo Pascal* [Glenview: Scott, Foresman, 1986], pp. 124-25.) Forcing characters into their low-bit form at the beginning of the program precludes the computer's having to look repeatedly for the low- *and* high-bit equivalents of each character one is trying to isolate and analyze.

Joe Weixlmann is Professor of English and Associate Dean of the College of Arts and Sciences at Indiana State University. He edits the quarterly journal *Black American Literature Forum*.

IEEE Professional Communication Conference in Winnipeg

Nearly 60 specialists in technical communication will speak at IPCC 87—the IEEE's International Professional Communication Conference—which will be held this year from October 14 to 16 at the Winnipeg Sheraton. Presentations, workshops and seminars will feature various approaches to using computers in communication, including

■ Electronic literary Tools ■ Learning to Write with the New Electronic Tools ■ English Etudes for an Engineer/Editor ■ Writing User-Friendly Manuals for the Information Systems Environment ■ Teaching a Course in Writing for the Computer Industry ■ How To Conduct a Usability Test ■ Desktop Publishing Systems: Where's the Communication? ■ The Future of Desktop Publishing in Technical Communications ■ Desktop and Electronic Publishing Implicit Language Codes—A Morphologic Connection. There will also be a seminar entitled ■ Resume Writing and Portfolio Building: Exploiting Developments in Word Processing and Page Make-up Programs, in addition to a range of papers and workshops covering a wide-range of topics in technical communications.

Contact Conference Ron S. Blicq, IPPC 87, 569 Oxford Street, Winnipeg, MB, Canada R3M 3J2, or call (204) 452-6480 or 284-4978.

Call for Papers in Technical Communications

The Society for Technical Communication has announced September 25, 1987, as the deadline for paper, panel, and workshop proposals for its 35th International Technical Communication Conference, to be held May 10-13, 1988, in Philadelphia. Papers tend to fit into one of the following categories: advanced technology applications; management & professional development; research, education, & training; visual communication; and writing & editing. Papers will be published in conference proceedings. Contact JoAnn T. Hackos, Comtech, 133 S. Van Gordon St., Suite 206, Denver, CO 80228, or phone (303) 986-9534.

Call for Papers: Computers and Humanities

The Association for Computers and the Humanities (ACH) will sponsor a conference on "Teaching Computers and the Humanities Courses" at Oberlin College in Ohio on June 9-11, 1988. Avoiding computer-assisted instruction, the conference will teach humanities faculty how computers are used, and may be used in the future, as a tool and thematic framework within their disciplines. Papers will tend to address practical curricular approaches, including ■ What should be included in such courses? ■ How should they be taught? ■ What level and mix of students should take the course? ■ Should a higher-level programming language be taught? ■ If so, which language is most suitable? ■ Should computing skills be taught before or after the student is familiar with applications of computers within his or her field? ■ Which applications software should be taught? ■ In how much detail?

Five copies of 1000-word paper-abstracts or panel-proposals (including list of participants) are due by November 30, 1987. Contact Prof. Robert S. Tannenbaum, ACH Conference Chair, Department of Computer Science, Hunter College CUNY, 695 Park Avenue, New York, NY 10021.

1987 CAUSE National Conference in Florida

The Professional Association for Computing and Information Technology in Higher Education, an organization of institutional members concerned with the administration of computing services, will hold its annual conference in Tarpon Springs, Florida, on December 1-4, 1987. The conference theme will be "Leveraging Information Technology." Contact CAUSE87, 737 29th Street, Boulder, CO 80303, or call (303) 449-4430.

California's Computer Using Educators Conference

Computer Using Educators, serving California teachers, has set its 8th annnual fall conference, "Reach for Tomorrow," for October 16-17, 1987, in San Jose, California. Contact Computer-Using Educators, P.O. Box 2087, Menlo Park, CA 94026 or call (415) 328-2248.

Conference on Natural Language Processing

The Association for Computational Linguistics will have its "Second Conference on Applied Natural Language Processing" in Austin, Texas, on February 9-12, 1988. The meeting will focus on the application of natural language processing techniques to real world problems, including invited and contributed papers, panel discussions, tutorials, exhibits, and demonstrations.

Papers will cover areas such as human-machine interfaces (including databases, expert systems, report writers, etc.), speech input and output, information retrieval, text generation, machine translation, office automation, writing aids, computer-aided instruction, tools for natural-language processing, and applications to medical, legal, or other professional areas. Papers will tend to present applications, evaluations, limitations, and general tools and techniques. Contact Donald Walker, ACL, Bell Communications Research, 445 South Street, MRE 2A379, Morristown, NJ 07960, or call (201) 829-4312.

Call for Papers: Microcomputers in Education

The Eight Annual Microcmputers in Education conference—"The Emerging Frontier: Interactive Video, Artificial Intelligence, and Classroom Technology"—will take place March 6-8, 1987, at Arizona State University in Tempe. November 15, 1987, is the deadline for submitting research studies, articles, and papers on all facets of microcomputers in education. Conference proceedings will be published. For guidelines, contact Maureen Miller or Gary Bitter, Microcomputers in Education Conference, FMC-Educational Media and Computers, Arizona State University, Tempe, AZ 85287-0111, or call (602) 965-7363.

Bibliography Update

Bradford A. Morgan

Allen, James. *Natural Language Understanding*. Menlo Park, CA: Benjamin Cummings, 1987.

Antonoff, Michael. "Setting Up for Desktop Publishing: In-house Publishing Pays Off Once a System's Set Up, but What's Involved and Who Should Oversee It?" *Personal Computing*. 11:7 (July 1987), pp. 74-82.

————. "Writing in a Spreadsheet." *Personal Computing*. 11:7 (July 1987), pp. 51-54.

Batchelder, Ned and Trevor Darrell. "Bringing *troff* Up to Speed: Until Now, *troff* Has Been Unable To Fully Exploit the Capabilities of Modern, Low-Cost Laser Printers. But the Development of a Versatile PostScript Preprocessor Now Means the Formatter Can Keep Current with Technology." *Unix Review*. 5:7 (July 1987), pp. 45-51.

Beil, Don. "What Key?: Several Simple Techniques for Macintosh Users Can Help To Locate the Correct Keys for the Characters You Need." *Personal Publishing*. 3:7 (August 1987), pp. 68-70.

Berry, John D. "The Technology of Type Has Changed in the Last Few Hundred Years. The Type Has Changed, Too." *Personal Publishing*. 3:7 (August 1987), p. 106.

Berst, Jesse and James Cavuoto. "Tips for Tagging: The Real Power of *Ventura Publisher* Lies in Its Style Sheets. Here Are Some Tips and Tricks To Use Those Style Sheets More Efficiently." *Personal Publishing*. 3:7 (August 1987), pp. 58, 62-64, 67.

Betz, David. "An Adventure Authoring System: A Tour of AdvSys, a Tool for Writing Text Adventure Games." *Byte*. 12:5 (May 1987), pp. 125-135.

Bierman, Howard. "What Accountants Think about Desktop Publishing: *Computers in Accounting*'s Survey of Readers on Desktop Publishing Reveals Users Views on Costs, Training Time and Benefits Derived from this New Technology." *Computers in Accounting*. 3:8 (June 1987), pp. 58-60, 62.

Bobker, Steven. "So What's New: Real Professional Quality Typesetting Is Now a Good Deal More Available." *MacUser: Special Supplement* 5. 3:6 (June 1987), p. 4.

————. "A Peek Between the Pages: Looking Forward to *PageMaker 2.0*? Here's What We've Seen So Far." *MacUser: Special Supplement* 5. 3:6 (June 1987), pp. 34-36.

Bohn, Lawrence and David Weinberger. *Unix Review*. 5:7 (July 1987), pp. 28-34.

Bortman, Henry. "In Your Write Mind: *MindWrite* Is a Serious, Heavy-duty Word Processor. Powerful Integrated Outlining and True Writers' Features Add Up to a Superior Program." *MacUser*. 3:8 (August 1987), pp. 90-95.

————. "A Library of PostScript Routines That Smooths the Learning Curve for Beginning PostScript Pilots." *Personal Publishing*. 3:6 (July 1987), pp. 96-98.

————. "The Fine Points of Kerning and Letterspacing Inside the Adobe Font Metrics, and Kerning with *XPress*." *Personal Publishing. 3:7 (August 1987), pp. 96-98.*

Bove, Tony and Cheryl Rhodes. "Searching for Page Makeup Perfection: The Ideal Page Layout Program Has Yet To Appear." Publish. 2:6 (July 1987), p. 97.

Brown, Eric. "Building Manuscripts Block by Block: Creating a Complex Document Can Seem As Intimidating a Task As Building a Pyramid. So Pharaohs Take Note: Lotus's First Word Processor Is Designed Primarily To Handle Problems Unique to Technical Writing, but Its Tightly Integrated Outliner and Advanced Print Formatting Make It Ideal for Any Monumental Work." *PC World*. 5:5 (May 1987), pp. 214-223.

Bryan, Marvin. "*Lotus Manuscript*: This New Word Processor Offers Tools for Engineers and Scientists, Though with Only Limited Support for Multiple Typefaces and Sizes." *Personal Publishing*. 3:6 (July 1987), pp. 78-85.

————. "Higher *Glyphix*: Instead of Buying Hundreds of Bitmapped Fonts for Your LaserJet, Why Not Just Buy Typeface Outlines and Produce the Fonts and Variations That You Want." *Personal Publishing*. 3:7 (August 1987), pp. 44-50.

Buckles, Mary Ann. "Interactive Fiction As Literature: Adventure Games Have a Literary Lineage." *Byte*. 12:5 (May 1987), pp. 135-142.

Buckley, Bradford D. "A Multiuser Solution to Electronic Publishing: Talaris' System Lets VMS Users Merge *TeX* Documents with Graphics and See the Results on a Shared Printer." *Digital Review*. 4:14 (July 13, 1987), pp. 65-68.

Caernarven-Smith, Patricia. "Computers and Composition: It May Be Computerized Art, but...." *Technical Communication*. 34:2 (May 1987), pp. 117-118.

"Center for Computer Analysis of Texts (CCAT)." *ACH Newsletter*. 9:2 (Summer 1987), pp. 4-6. (Association for Computers and the Humanities)

Crider, Bill. "Pages According to Plan: With *Harvard Professional Publisher*, Careful Preparation Pays Off in Quality Typography." *Publish*. 2:6 (July 1987), pp. 72-75.

Custer, Linda. "Stat's Incredible: *Parameter Manager* Takes the Tedium Out of the Laboratory Notebook." *MacUser*. 3:7 (July 1987), pp. 76-79, 194.

D'Acquisto, Domonic R. "XPressive Pages: Quark's New Page Makeup Program Is the Favorite Around Apple These Days. It Handles Type and Graphics Like a Champ, and Produces High-quality Output." *Personal Publishing*. 3:6 (July 1987), pp. 24-26, 29-31, 34-36.

DeMaria, Russel. "Hypertext-based *Guide* Helps Users with Interactive Program Development." *PC Week*. 4:30 (July 28, 1987), pp. 61,70.

Diltz, Carl. "*Laser Envelopes*: Simple Tasks like Addressing Envelopes Are Not Always Simple with Laser Printers, but This Easy-to-use H-P Utility Can Make Trips to the Typewriter a Thing of the Past." *Personal Publishing*. 3:7 (August 1987), p. 76.

Donovan, Ronald John. "The Electronic Writer: Compacts and Convertibles. We Take Some Exciting New Technology for a Spin." *Writer's Digest*. (July 1987), pp. 42-43, 46.

Duin, Ann. "Computer Exercises to Encourage Rethinking and Revision." *Computers and Composition*. 4:2 (April 1987), pp. 66-105.

Flammang, James M. et al. "Review Portfolio: *Mindwrite, Spellbinder, Laser Author, Cybertype, Mac Calc, PC Search and Replace, Blueberry, Stepping Out*." *Publish*. 2:5 (June 1987), pp. 86-92.

Forest, Robert B. "On Writing: Whether It's Your User Newsletter or a Report, You Don't Have To Fear Writing. Approach the Task As a Programmer Approaches an Application." *Information Center*. 3:7 (July 1987), pp. 71, 74-75.

Gillis, Philip D. "Using Computer Technology to Teach and Evaluate Prewriting." *Computers and the Humanities*. 21:1 (January-March 1987), pp. 3-19.

Graham, Gord. "Edit Trails: In This Age of Word Processors and Electronic Manuscripts, It Can Be Hard To Keep Track of Changes. Here Are Some Ways To Figure Out Who Did What and When." *Personal Publishing*. 3:5 (June 1987), pp. 70-71, 74-75.

Gralla, Preston. "Examining the Value of Electronic Writing Aids." *PC Week*. 4:21 (May 26, 1987), pp. 121-122.

Green, Michael. *Zen and the Art of the Macintosh Discoveries on the Path to Computer Enlightenment*. Philadelphia, PA: Running Press, 1987.

Harris, Kevin. "Systemisation and Computerization in Literature Museums: Bronte Parsonage, Dickens House, Keats House." *Humanities Communication Newsletter*. 8 (1987), pp. 2-4.

Hart, Roger. "The Big Bundle: Tips on Buying an All-in-one PC Publishing System." *Publish*. 2:6 (July 1987), pp. 52-59.

Herold, Steve. "*Fontographer* at Work: Most People Won't Want To Design Whole Typefaces, but for Logos, Special Characters, and Some Illustration Tasks, *Fontographer* Is Hard To Beat." *Personal Publishing*. 3:5 (June 1987), pp. 60-68.

Hewes, Jeremy Joan. "The Latest *Word*: Microsoft's Popular Macintosh Word Processing Program Blossoms with New Features." *Publish*. 2:6 (July 1987), pp. 64-67.

Hixson, Amanda C. et al. "Dueling Lasers II: The Desktop Publishing Face-Off. Armed with Two of the Newest Turnkey Systems, We Venture Further into the World of MS-DOS Desktop Publishing." *Personal Computing*. 11:6 (June 1987), pp. 121-130.

Hocking, Joan. "What Can English Instructors Expect from CAI?" *Collegiate Microcomputer*. 5:2 (May 1987), pp. 159-162.

Hoenig, Alan. "*TeX* for the Mac: *TeX* Runs on Many Different Machines and Offers Some Amazing Powers. Here Are Two Versions for the Macintosh." *Personal Publishing*. 3:7 (August 1987), pp. 82-87.

Holmes, Thom. "Make My Page! Here's How to Choose the Best Desktop-Publishing Program for Your Needs." *Byte*. 12:5 (May 1987), pp. 159-166.

House, William J. and Elisabeth Bell. "THOUGHT-LOG: A Thought Processing Program in the Form of a Journal." *Collegiate Microcomputer*. 5:3 (Autumn 1987), pp. 291-293.

Jantz, Richard. "Publishing Meets Its *PageMaker*: Cast Off Your Codes and Quads! If You Love Snappy-looking Documents, Hate Typographic Arcana, and Lack a Mac, You're in Luck. *PageMaker* for the PC Matches Mac-based Publishing with Professional PC Tools for Typographers and Graphic Designers Alike." *PC World*. 5:5 (May 1987), pp. 198-207.

Kahle, Jacqueline R. "Departmental Publishing—Developing a Sensible Approach to Producing and Distributing Information: The Departmental Publishing Concept Offers Organizations a Cost-effective Way to Bring Publishing In-house and to Integrate and Improve on Existing Capabilities." *Hardcopy*. 7:6 (June 1987), pp. 73-74, 76.

Kershenbaum, Peg. "Latin: Requiescat in PC." *Collegiate Microcomputer*. 5:2 (May 1987), pp. 139-147.

Kleper, Michael L. *The Illustrated Handbook of Desktop Publishing and Typesetting*. Blue Ridge Summit, PA: Tab Books, 1987.

————. "All about Type: From Metal to Laser Type, from Gutenberg to Bitstream, the Art of Putting Letters on Paper." *Personal Publishing*. 3:5 (June 1987), pp. 24-26, 29-30. (Reprint from above)

Kochan, Stephen G. "PostScript Tutorial, Part One: The Coordinate System." *The PostScript Language Journal*. 1:1 (March 1987), pp. 4-18.

Krasnoff, Barbara. "Users See the Darker Side of Desktop Publishing: In Some Companies, It's Shaking Up the Workplace and Casting Workers into Roles They Weren't Prepared For." *PC Week*. 4:24 (June 16, 1987), pp. 41, 48.

Lehmann, P. F. "Putting Together a Poster Fast: Fancy Font and Sign Designer." *Binary: The Newsletter of the Society for General Microbiology Computer Club*. 1:10 (1987), pp. 18-19.

Lewis, Barbara and Robert Lewis. "Do Style Checkers Work Work? Plagued by Purple Prose? Style Analysis Software Can Help You Identify Problem Passages—but It Won't Turn Gibberish into English." *PC World*. 5:5 (May 1987), pp. 246-252.

Little, Sherry Burgus. "The Computer As Audience: Using HOMER, a Text Analysis Program." *Computers and Composition*. 4:2 (April 1987), pp. 106-120.

Lombardi, John. "Office Word Processing: The Current Crop." *InfoWorld*. 9:28 (July 13, 1987), pp. 41, 44, 48-49, 52.

Makrias, Stephanie. "What's New in Low-priced Word Processors: Power Features like an On-line Thesaurus, an Integrated Spelling Checker, and Multiple Windows Represent an Emerging Trend in Inexpensive Software." *Personal Computing*. 11:8 (August 1987), pp. 97-103.

McGilton, Henry and V. Bill Tuthill. "Progress through Accretion: People Have Wanted To Do Away with *troff* for Years, but the Darned Thing Keeps on Absorbing New Capabilities. If WYSIWYG Vendors Are To Come Up with a Viable Alternative in the UNIX Space, They'll Need To Offer More Than Just an Improved User Interface." *Unix Review*. 5:7 (July 1987), pp. 36-41.

McLean, Alice Musick; Robert Morrissey, and Donald A. Ziff. "ARTFL: A New Tool for French Studies. 1700 Texts on Line, from Apollinaire to Voltaire." *Scholarly Communication*. 1:8 (Spring 1987), pp. 1, 6-9.

Meeker, Michael. "A Simple Database Manager for Writers." *Collegiate Microcomputer*. 5:3 (Autumn 1987), pp. 255-259.

Minicucci, Rick. "Text Retrieval Systems Find the Right Words(s): Combing the Far Reaches of Text Databases for Words That Will Help Track Down Essential Documents Can Be Time Consuming, Complicated and Even Fruitless. Text Retrieval Systems Can End the Frustration, Performing Searches That Are Fast, Thorough and On Target." *Today's Office*. 21:12 (May 1987), pp. 50-54.

Mullins, Carolyn J. "The Layout of Desktop Publishing: Everyone Is Talking about Desktop Publishing, but the Lay of the Land Is More Than a Little Confusing. Guidelines for Novice Publishers." *Information Center*. 3:6 (June 1987), pp. 19-23, 27.

————. "Desktop Design: Good Document Design Is a Lot Like a Good User Interface—They Should Both Be Transparent. Here's How To Become Invisible." *Information Center*. 3:7 (July 1987), pp. 81-84.

————. "Computer Training and Technical Communications." *Technical Communication*. 34:2 (May 1987), pp. 117-118.

————. "Vendors, Corporations Increasingly See Benefits of Multilingual Software." *PC Week*. 4:26 (June 30, 1987), pp. 91, 94-95.

Nace, Ted. "Better Business Reports with *Ventura Publisher*: Before You Sit Down and Typeset a Book with *Ventura Publisher*, Learn the Basics of Creating a Sharp Business Report with All the Fixin's." *PC World*. 5:5 (May 1987), pp. 264-273.

Nelms, John David. "Search and Replace! A Program To Perform Multiple String Substitutions in Any Number of Files." *DEC Professional*. 6:7 (July 1987), p. 82.

Nelson, Robin. "Word Processing: Let's Hear It for CAW." *Personal Computing*. 11:8 (August 1987), pp. 49-52. (computer-aided writing)

————. "Word Processing: Trading Up Is Hard To Do." *Personal Computing*. 11:6 (June 1987), p. 65.

Norton, Peter. "Style Sheets Offer a Finely Tuned Text Format." *PC Week*. 4:30 (July 28, 1987), p. 31.

Olsen, Mark. "Beyond SNOBOL: The Icon Programming Language." *Computers and the Humanities*. 21:1 (January-March 1987), pp. 61-66.

O'malley, Christopher. "The Output Side of Data Bases: 1) Placing Specific Information in the Order That You Need It, 2) Getting the Data in a Presentable Report You Can Distribute, 3) Doing It Quickly with Macros That Make Reporting Automatic." *Personal Computing*. 11:7 (July 1987), pp. 89-97.

Pallato, John. " 'Creeping Elegance:' When Appearance Slights Functionality. A Little Embellishment Can Be Dangerous, Some Managers Are Finding Out from Overly Enthusiastic PC Users." *PC Week*. 4:27 (July 7, 1987), pp. 43, 50.

Parker, Roger. *Aldus Guide to Basic Design*. Seattle, WA: Aldus Corporation, 1987, 68 pp.

Pelli, Denis G. "Programming in PostScript: Techniques and Applications in Vision Research, Plus a Survey of PostScript Hardware and Software." *Byte*. 12:5 (May 1987), pp. 185-202.

Perdue, Lewis. "Pepping Up the Old PC: Turn Your 8088-based PC into a Desktop Publishing Machine." *Publish*. 2:5 (June 1987), pp. 80-85.

Petkus, Jennifer. "Think BIGGER: How To Put Together a Tabloid from Letter-Size Pages." *Publish*. 2:6 (July 1987), pp. 68-71.

Pountain, Dick. "Sorting Out the Sorts: A Primitive Method Indexes a Book Faster Than More Sophisticated Algorithms." *Byte*. 12:8 (July 1987), pp. 275-276, 278, 280.

Ristroph, John H. "Use of Microcomputers To Prepare Scientific Documents." *Collegiate Microcomputer*. 5:3 (Autumn 1987), pp. 260-268.

Robinson, Phillip. "Word Processors, The Top Seven—Three for the Macintosh and Four for the IBM PC: *WriteNow, MindWrite, Word for the Macintosh, Manuscript, WordPerfect, Word for the PC, Nota Bene*." *Byte: Application Software Today*. 12:7 (Summer 1987), pp. 55-67.

Rochester, Jack B. "Beyond Desktop Publishing: Desktop Publishing Isn't the Only Game In Town: An Interview with Daniel Kennedy of Texet Corporation." *Information Center*. 3:6 (June 1987), pp. 28-30, 32.

Rodarmor, William. "Teaching *WordStar* New Tricks: After Years of Procrastination, MicroPro Has Finally Released a *WordStar* Upgrade. Loyalists Are Largely Rewarded, but Will Version 4.0 Attract New Converts?" *PC World*. 5:5 (May 1987), pp. 224-229.

Rosenbaum, Daniel J. "For Technical Pros/e: Illuminating *Manuscript*. Lotus' New Technical Word Processor is a Whiz at Organization." *Publish*. 2:5 (June 1987), pp. 64, 66-8.

Rowell, Dave. "Translatable Text: Port Text Files from Computer to Computer by Making Them Fluent in the Universal Language of ASCII." *PC Resource*. 1:3 (June 1987), pp. 46-48.

Saenz, Mike. "PostScript in Color: Personal Publishing Is Mostly a Black and White Affair. With the Advent of New Tools and Color Separation Techniques, However, PostScript Color Is Arriving." *Personal Publishing*. 3:4 (May 1987), pp. 30-34.

Saffo, Paul. "Desktop Publishing: The Paper Revolution." *Personal Computing*. 11:7 (July 1987), pp. 43-46.

Schargel, David. "Do Not Pass Go: *Ready, Set, Go!* 3 Makes a Valiant Effort To Be the Premier Page Layout Program. And It Comes Close—Maybe Close Enough for You." *MacUser: Special Supplement* 5. 3:6 (June 1987), pp. 26-30.

Selfe, Cynthia L. "Creating a Computer Lab That Composition Teachers Can Live With." *Collegiate Microcomputer*. 5:2 (May 1987), pp. 149-158.

Sell, William. "In So Many Words: Mastering the Art of Glossary Writing." *Access 87*. 4:10 (June 1987), pp. 18, 59.

Seybold, John W. "The Desktop-Publishing Phenomenon: Personal Printing Has Come a Long Way in a Short Time." *Byte*. 12:5 (May 1987), pp. 149-154.

Seybold, John and Fritz Dressler. "Digital Type Foundry: A New Breed of Type Designers Use Technology To Build on Historic Principles Creating Digital Type Fonts of the Future—Here Today!" *Personal Publishing.* 3:5 (June 1987), pp. 40-42.

Shapiro, Ezra. "Personally Designed: Here's How To Select the Right Desktop Publishing Software for Each Project." *MacUser.* 3:6 (June 1987), pp. 90-96.

Simpson, David. "Database Development Tools: An Evaluation. An Independent Company Evaluates over 50 Database Development Tools Available for Applications Developers, Software Houses and End Users." *Mini-Micro Systems.* 20:5 (May 1987), pp. 77-86.

Small, Jocelyn Penny. "Computer Index of Classical Iconography." *Humanities Communication Newsletter.* 8 (1987), pp. 4-8.

Smith, M. W. A. "The Revenger's Tragedy: The Derivation and Interpretation of Statistical Results for Resolving Disputed Authorship." *Computers and the Humanities.* 21:1 (January-March 1987), pp. 21-55.

Sorensen, Karen. "Corporate Desktop Publishing: Who's In Charge. As Large Corporations Adopt Desktop Publishing, Their Workers Must Adapt to New Job Roles." *Publish.* 2:5 (June 1987), pp. 56-61.

Sperberg, Luther. "For Technical Pros/e: Deep in the Heart of Texes. Multiply Your PC's Typesetting Skills with *Micro Tex* and *PC Tex*." *Publish.* 2:5 (June 1987), pp. 65, 69-71.

"Spiral Text." *The PostScript Language Journal.* 1:1 (March 1987), pp. 19-20.

Strange, William C. "Hamlet on the Macintosh: An Experimental Seminar That Worked." *Collegiate Microcomputer.* 5:2 (May 1987), pp. 113-122.

Strauss, Ken and Averil. "Connecting the IBM PC and the Apple LaserWriter." *The PostScript Language Journal.* 1:1 (March 1987), pp. 40-42.

Strickland, James. "Computers, Invention, and the Power to Change Student Writing." *Computers and Composition.* 4:2 (April 1987), pp. 7-26.

Strom, David. "LAN Helps Deaf Sharpen Writing." *PC Week Connectivity.* 4:27 (July 7, 1987), p. C/4.

Susser, Bernard. "Computers and Composition: A Bibliographic Introduction." *The Language Teacher.* 11:5 (May 1987), pp. 10-19. (The Japan Association of Language Teachers)

Sutcliffe, Judy. "The Exquisite Square: Starting with a Typewriter and Nickels for the Corner Copier, *Emigre* Magazine Has Blossomed into a Tour de Force of Avant Garde Personal Publishing." *Personal Publishing.* 3:5 (June 1987), pp. 44, 46, 50.

Thomas, Jean. "Introducing Students to Free Verse on the Word Processor." *The NCTE-ACE Newsletter.* 2:4 (April-June 1987), pp. 1, 4.

Tortora, Lou. "How Commercial Typesetters Can Survive—and Thrive—in the World of Desktop Publishing." *TypeWorld.* 11:10-2. (May 1987), pp. 12, 14.

Ulick, Terry. *Personal Publishing with PC PageMaker.* Indianapolis, IN: Howard W. Sams, 1987.

————. "Finding Fonts: If You Need To Match an Existing Type Style, or Wish To Explore beyond Times and Helvetica, There Is Good News: Desktop Publishing's Type Library Is Growing Daily." *Personal Publishing.* 3:5 (June 1987), pp. 52-53, 58-59.

"University of Pittsburgh's Computerized Writing Program: The Fine Art of Editing. As a Writing Tutor, MINA's Objective Is Not To Correct Errors for the Students, but To Teach Them To Correct Their Own." *Edu: The Education Magazine of Digital Equipment Corporation.* 1:44 (Spring 1987), pp. 23-24.

Urso, Bob. "Of Size and Space: Type Size and Spacing Are Two of the Most Elusive Concepts for Novice Personal Publishers. Here Are Some Tips and Examples To Show You the Way." *Personal Publishing*. 3:5 (June 1987), pp. 34-38.

Wasson, Gregory. "Typecasting: Have You Ever Thought of Going Professional with Desktop Publishing? Better Read This First." *MacUser: Special Supplement* 5. 3:6 (June 1987), pp. 18-22.

Webster, Bruce. "Publish and/or Perish: Serious Business at the MacWorld Exposition, and a Look at Desktop Publishing." *Byte*. 12:5 (May 1987), pp. 279-290.

Weigand, C. J. "Decorative and Display Type Can Mean the Difference Between Being Noticed or Being Ignored." *Personal Publishing*. 3:5 (June 1987), pp. 92-93, 96-98.

————. "The Word on *Word*: The Macintosh Version of *Microsoft Word* Spent Years As the Poor Stepchild to the IBM Version. With *Word 3.0* on the Macintosh, All That Has Changed." *Personal Publishing*. 3:7 (August 1987), pp. 24-28, 30, 34-35.

Wheatley, William F. "Quality Typography." *TypeWorld*. 11:7-1. (April 1987), pp. 18, 24.

Wolff, Bob. "An Accountant's Guide to Desktop Publishing: In the Past, Only Large Accounting Firms Could Afford To Do In-house Publishing, but Today's Desktop Publishing Systems Can Put a Print Shop on Your Desk. This Report Will tell You What You Need To Know To Get Started." *Computers in Accounting*. 3:9 (July/August 1987), pp. 70-81, 84-96.

Wood, Patrick. "The Font Dictionary." *The PostScript Language Journal*. 1:1 (March 1987), pp. 36-40.

Yudkin, Marcia. "Proofing Problems: Spell Checkers Are All Very Well and Good, but They Don't Replace Proofreading. Here Are Some Techniques for Using the Two Together." *Personal Publishing*. 3:7 (August 1987), pp. 78, 80.

Manuscript Submissions Welcome

The *Newsletter* welcomes article submissions that pertain to word-processing, text-analysis, and research applications in professional writing situations. Also, hardware and software reviews are encouraged, but please contact Dr. Jim Schwartz, Hardware/Software Review Editor, **before** submitting them (call Jim at 605-394-1246). Manuscripts may be submitted either as hard copy or on 5¼" diskettes using *WordStar*, *WordPerfect*, DCA, or standard ASCII code. If submitting disks, please make sure they are formatted either in MS-DOS, PC-DOS, or a popular CP/M format (Kaypro, Zenith, etc.) The Editors reserve the right to edit manuscripts, if necessary. If you want your manuscript or diskette returned, please send enough postage to cover the return along with a self-addressed envelope. Address all correspondence to the Editors, *Research in Word Processing Newsletter*, South Dakota School of Mines and Technology, 501 E. St. Joseph, Rapid City, SD 57701-3995. Jim Schwartz may also be reached on *CompuServe* (70177,1154).

On The Meaning of the Term "Desktop Publishing"

Tom Carney

First, a word of thanks to George Moberg for setting out so clearly, in the April 1987 number of the *Research in Word Processing Newsletter*, the variety of meanings, and the media hype, associated with this term. What follows is an attempt to explore the core meanings of this term in its current usage. Presumably, the term wouldn't get so much play if it weren't giving its users *some* advantages that the other terms so well discussed by George don't give.

Basically, the term "desktop publishing" is currently used to mean that you can create text and graphics, compose the results into pages with the text flowing round the graphics, and produce camera-ready copy for either a laster printer or a phototypesetter. You CAN'T reproduce this copy en masse by this technology; you have to get a traditional printer to do that. Distribution may well be done by someone else, too. So the "publishing" part of the desktop publishing term means something rather different from the word "publishing" when it's used on its own (but let's not get into the range of meanings associated with "publishing" here).

There seems to be at least three basic concepts implicit in the term *desktop* publishing, as it's currently used. *The first is that you can do page makeup with a microcomputer*—that you can publish a document (a pamphlet, a newsletter or a book) by using your microcomputer (which is usually positioned on your desk). You don't need a mainframe computer or a minicomputer or "workstation", the computers associated heretofore with the terms "computer-assisted publishing" and "corporate electronic publishing". Your personal microcomputer will suffice.

With the micro that you've got sitting on your desk you can set type, produce graphics and make up pages, flowing the text round the graphics. You don't need all the paraphernalia—drafting board, special lighting, George's "razor blades, waxes and paste pots," etc.—traditionally associated with making up pages for a publication. You simply produce a text file, on a computer disk, containing the made-up pages.

Next, still at your desk, you send this text file, via the LAN (local area network) or, via a modem over the telephone, to a printer. If it goes via the LAN, it usually goes to a laser printer that you and others share in-house. If it goes by phone, you're usually sending it to an outside lasersetting house or—more likely—a computerized typesetting shop. The latter prints it out, generally on a phototypesetter.

A laser printer such as the LaserWriter (which is the printer most associated with desktop publishing) produces a resolution of 300 (X 300) dots per inch (dpi). This is camera-ready copy: at 90,000 dpi, it looks like print to most people. A phototypesetter produces densities of over 1200 dpi (the Linotronic 100) or of over 2400 dpi (the Linotronic 300), turning out photographic plates, which a printer prints from (the Linotronic gives better results than many phototypesetters used to print expensive books).

Running off the made-up pages accounts for about 10% of publishing costs (IF distribution costs aren't included therein); so you've just produced a 90% saving in getting your document printed. As desktop publishing is so quick and easy, you've usually cut at least 30-40% off the time taken by traditional printing. Maybe your graphics took longer than your graphic artist would have taken to do them. But, with desktop publishing, you've now got a set of graphics (which can be easily and quickly adapted for other purposes), for future use or sale.

The second basic concept is that of interactive page makeup. Desktop publishing is strongly associated with WYSIWYG (what you see [on the screen] is what you get [printed out]) on your page. WYSIWYG means that you don't have to become expert in the programming—the coding or tagging—associated with batch processing, which is done "blind" (you can't see the results of your programming till it's printed out) and which requires considerable expertise and training. Moreover, desktop publishing programs can now easily do things (such as flowing text round irregularly shaped graphics) that traditional page makeup programs had great difficulty with. With very little training, nearly anyone can make up a page using an interactive WYSIWYG page makeup program. It's this ease of learning and use, combined with the cheapness of the technology, which has made desktop publishing so popular.

It's true that much of the product of desktop publishers is amateurish. But this is another issue. Anyway, much of the product of publishers using batch processors, which have their problems with graphics and fancy page layouts, was reader-unfriendly: page after page of (easy-to-set) monolithic text blocks. Desktop publishing goes with visually informative text, which involves a distinctively different use of language, entailing lots of graphics and diagrams and use of layout as punctuation at the level of the page. Books are changing, becoming more readable as a result of this technology. Another reason for its popularity—though not in the eyes of some traditionalists.

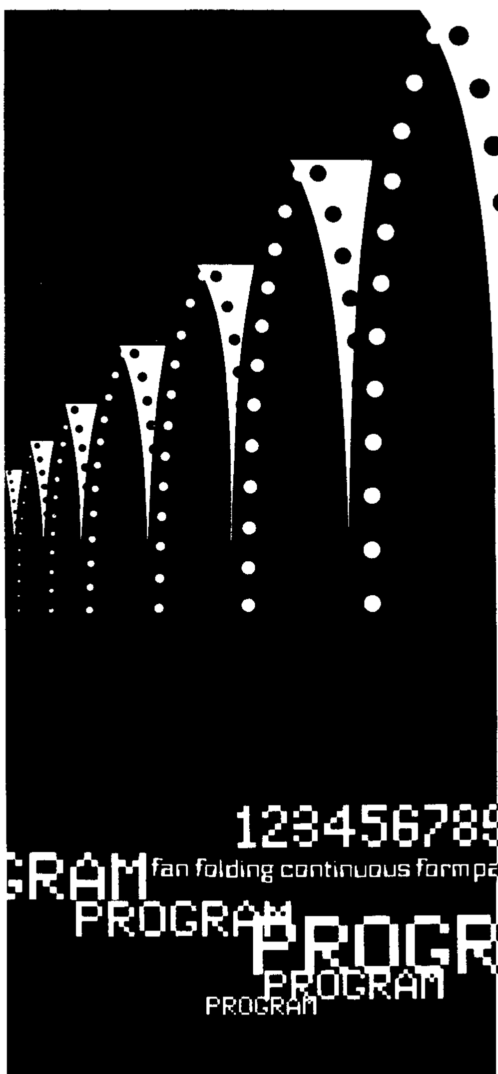
The third basic concept has to do with the "desktop" part of the term. The desktop metaphor is basic to *the Macintosh's user interface* (what the Mac uses in place of the PC's command-driven operating system) and so to desktop publishing (the PC has had to appropriate most of the Mac's desktop features to do desktop publishing). This "desktop" (an image of a desktop is what you actually see when the Mac's screen comes on) consists of a graphics-based micro, with much use of icons to represent visually the operations to be performed; a point-and-click "mouse" to activate them; "windows" that open to enable you to have several "pages" open on your desktop at once; and "desktop accessories". The latter are the electronic equivalents of the paraphernalia found on a desk. They're there in the background as you work, for you to pull out and use while working on your "pages". Desktop publishing is publishing using *this* "desktop".

The facts are that 70% of all desktop publishing is done on the Macintosh and the LaserWriter, generally using the page makeup program *PageMaker*. This was the combination that made desktop publishing feasible. People don't talk about "PC publishing" because the term "PC" is associated with the IBM PC and its clones. You can't do desktop publishing on a PC/PC clone unless it's been considerably up-graded. You really need something like an AT—and even then it's easier to do interactive page makeup involving lots of graphics on a Macintosh.

So desktop publishing isn't primarily associated with the PC in most computer-user's minds. Rather, it's associated with the Macintosh *and the latter's easy-to-use "desktop"*—the main metaphor of the Macintosh, and what most distinguishes its operating environment from the command-driven, MS-DOS environment of the PC.

So the term desktop publishing is a very convenient piece of shorthand. It carries a considerable freight of implications and associations. None of the other terms considered above carry this penumbra of associations. It will indeed be interesting to see if it stands the test of time.

Contributing editor **Tom Carney** teaches in the Department of Communication Studies at the University of Windsor, 401 Sunset Avenue, Windsor, Ontario N9B 3P4.



In Future Issues. . .

Textpack

Micro-OCP

Nevada Pilot

Minnesota SNOBOL 4.2

Hercules Graphics Plus Card

Approaches to Multi-Lingual Text