RESEARCH IN
WORD PROCESSING
NEWSLETTER

IN THIS ISSUE . . .

BIBLIOGRAPHY UPDATE

(MONTHLY)

THE LATEST IN WORD-PROCESSING RESEARCH

A Typology of Word-Processing Programs[1]

Bryan Pfaffenberger

Finding one's way through the word-processing jungle unaided requires the bravery of Indiana Jones: dozens of word-processing programs seem alluring, but each conceals some dark secret—and you might not find out about it until you've spent $350. It's by no means certain, moreover, that you'll get help from computer-using colleagues. Those who have adopted a word-processing program tend to behave like new converts to an intolerant religion. Or, so immersed are they in computer jargon that, as Carolyn Heilbrun found, you feel as though you had asked a classicist about Antigone and received a reply—in classical Greek.[2]

Yet, there's a rational way through this jungle. The apparent diversity disappears when it's realized that all word-processing programs are based on just four basic design options, producing just four basic program types: **text editor/batch formatters, online formatters, advanced online formatters, and integrated word-processing programs.**[3] More than any other single factor, a word-processing program's basic advantages (and disadvantages) are shaped by its underlying design.

You'll want to know, therefore, which design underlies a program you're thinking about buying. The trouble is, software publishers seldom give you this information. They emphasize, rather, the program features that cast it in a good light (``Proportional spacing! Context-sensitive, onscreen help!''), leaving you to discover the ones that don't (``Oh no! It can't double-space footnotes!''). Much of the detective work you'll need to do before choosing a program rightly focuses on finding the program that has the features you need. Yet, this emphasis on features—one shared by both publishers and buyers—is only part of the story. The pages to follow tell the rest of it.

TEXT EDITORS

Text editors, the earliest type of word-processing programs, were devised not for writers but for programmers. Programs are written and edited as text files: that is, as documents containing nothing but the standard characters available on the computer's keyboard. Once written, the program is translated and saved to disk in a form that the computer can read and execute. The program is printed only to provide backup security; there is little concern for its appearance on the printed page.

Text editors, therefore, contain no provisions for the formatting and printing of documents; their job, rather, is to provide the user with the most efficient possible tools for entering and revising text on the display screen and saving it to disk. As the text is entered, it is displayed in a standard (and usually unalterable) format of single-spaced paragraphs aligned along the left margin. Few provisions (if any) are made for showing the user how the text will appear when it is printed. For writing purposes, text editors are obviously incomplete; that is why **batch formatters** (see below) have been devised to work with them.

Word-processing programs that fit into the text editor/batch formatter category are easily recognized because **two separate programs**—the text editor and the batch formatter—are necessary to complete the writing process. Sometimes the fact that two programs are needed is obvious from the program's name (for instance, *Edix/Wordix*), but in other cases it's not (for instance, *Perfect Writer* or *Proofwriter* [*RWPN*, Sep. '85]). To find out whether a prospective program fits into this category, ask: ``Is a separate pass by a formatting program required to prepare the text for printing?''

ADVANTAGES OF TEXT EDITORS

A text editor encourages the writer to concentrate on the text itself—that is, on writing and revising—to the exclusion of formatting considerations. It is possible, to be sure, to concern oneself with formatting while writing with text editors, but—for reasons that will become clear below when batch formatting programs are discussed—few writers find this strategy satisfactory.

Some prefer text editors precisely because there is little temptation to be waylaid by formatting considerations while writing. Author David Stang, for instance, started out with the online formatter *WordStar* [*RWPN*, Feb. '85], which simulates on the screen the text's appearance when printed—it's a "what you see is what you get" word-processing program. But he found himself distracted by the "visual thrill on the screen." Stang found himself getting "caught up in the [onscreen] formatting and in fooling with margin width, page breaks, headers, and end-of-paragraph widows," wasting precious writing time. Finally, Stang switched to *Edix/Wordix*, a text editor/batch formatter, and the problem ceased. "Since with these programs what you see is not what you get, the thrill was gone."[4] You can put the formatting off until later and concentrate on writing.

A second advantage of text-editing programs is their uncluttered screen display. All that is seen (with the exception of a status line containing, perhaps, the name of the document and the location of the cursor) is the text, which is almost invariably placed into single-spaced paragraphs with left flush alignment.

A third advantage of text editors is the comparatively short length of the program code. Because text editors lack formatting and printing functions, they take up little room in the computer's memory. Few (if any) of the program's functions have to be paged from disk, a feature that slows memory-hungry programs by 50% or more. For all save lengthy documents, the whole text can be kept in memory with the program, allowing almost instantaneous scrolling from one page to the next.

DISADVANTAGES OF TEXT EDITORS

The chief disadvantages of text editors for scholarly purposes are entirely those of the batch formatting programs with which they must operate. See **Batch Formatting Programs** below for the details.

EXAMPLE: *EDIX*

Edix (Emerging Technology Consultants, Inc., 2031 Broadway, Boulder, CO 80302, 303-447-9495, available for the IBM PC and PC-compatible computers) is a text editing program of exceptional quality. It well exemplifies the strengths of text editors for writing purposes.

Edix comes equipped with a full panoply of features appropriate for writing and revising the text, including a full set of block commands, search and replace, variable tabs and margins, and multiple windows. It does not include any features or commands for formatting or printing. (A companion batch-formatting program, *Wordix*, provides the formatting and printing features.) *Edix* displays text on the screen in single-spaced, left flush format, no matter how you decide to format it for printing.

Text is entered indirectly into the computer's main memory up to the limits imposed by the computer's memory capacity. An IBM Personal Computer with 320 kilobytes of main memory, for example, can accommodate an *Edix* document of approximately 50,000 words. Because the entire text is present in the main memory, scrolling and other commands execute at high speeds. *Edix* saves text to disk in standard ASCII format, meaning that it can easily be used by many other programs, communicated via modems to other writers, or submitted on disk for computer-aided typesetting.

EXAMPLE: *PROOFWRITER GRAPHICS*

Proofwriter Graphics [*RWPN*, Sep. '85] (Image Processing Systems, 6409 Appalachian Way, P.O. Box 5016, Madison, Wisconsin 53705, available for the IBM PC and PC-compatibles) comes with a text editor (SEDIT) and a batch formatter (FMAT). Both are designed to exploit fully the bit-mapped video display made possible by the PC's optional color-graphics adapter.

The program comes with a massive, disk-based database of extended-character sets, providing all the characters necessary for writing in Russian, classical Greek, and modern European languages. Also included are a wide variety of symbols for mathematicians, engineers, and statisticians, and the program included a special equation mode that facilitates the entry and printing of equations (which it can automatically number). The special characters and equations are displayed on the screen, and if you have one of several popular dot-matrix printers, it prints them without any special intervention on your part. If the bewildering variety of special characters does not suffice, a character-generation program, supplied with the program disks, permits you to construct your own characters and save them to disk.

BATCH-FORMATTING PROGRAMS

Batch-formatting programs, or separate programs which are intended exclusively for formatting the text of a document, are designed to equip text editors with the formatting and printing features necessary for writing (not just programming). Programmers created batch formatters after it was realized that text editors were suitable for tasks other than writing programs. Even so, the text editor/batch formatter still shows signs of its birth in the data-processing department: as will be seen below, the combination is at its best when producing complex manuals or technical documentation—precisely the writing task its creators had in mind for it.

The term **batch** provides an important clue to the way these programs function. Working with these programs provides an opportunity to experience the computer's past, in which all computers employed a user-computer interaction technique called **batch processing**. When you use a computer in the batch-processing mode, you feed the program and data into the computer, wait while it does the processing, and look at the results when the printer spits them back at you. That's exactly how batch-formatting programs work—and that's exactly what's wrong with them.

Batch-formatting programs require the user to insert **embedded formatting commands** in the document's text using a special **marking character**, such as @ or %. One formatting program, for instance, requires the insertion of the following inserted or embedded command to justify the right margin of text that follows the command:

.bw 65

The period is the marking character, which alerts the formatting program that a command is to follow. **BW** stands for "body width," or the length of lines in the body of the document, and **65** is an **argument**, or specification (in this case, 65 characters).

The embedded commands are placed in the text using the text editor. After the text has been marked in this way, the formatting program reads the text and produces an **output file**, which is subsequently routed to the printer. Formatting with a batch formatter is like programming: you put in the code, compile it, and run the program to see if everything works right. For the programmers who devised these programs to assist them in writing technical documentation, the strategy made perfect sense. If you've no experience with programming, however, you'll almost certainly need weeks or even months of experience with the program before you work up to high proficiency.

ADVANTAGES OF BATCH-FORMATTING PROGRAMS

Batch-formatting programs are so inconvenient that they can be recommended only to those who wish to take advantage of their merits (which, though few, do indeed exist.)

Even though formatting a document with a batch formatter isn't much fun, these programs do offer a wide variety of formatting and printing options even when used on computers with limited memories. Generally included are all or almost all of the following formatting features: automatic footnote numbering, automatic pagination, chain printing, floating footnotes, headers and footers, gutters, microjustification, multiple columns, print merging, printer font selection, proportional spacing, and table of contents generation.

Moreover, because batch-formatting programs can concentrate on formatting and printing tasks, they are capable of an exceptional degree of control over the printing process. Already mentioned was *ProofWriter Graphics*, and another example is provided by *Fancy Font* (SoftCraft, Inc., 222 State Street, Madison, Wisconsin 53703, for most CP/M-80 and MS-DOS computers). *Fancy Font* is a batch-formatting program designed to make full use of the graphics mode of inexpensive dot-matrix printers. It forces the printer to make several passes over the printed copy, producing a typescript in a variety of fonts and sizes that is close to the quality produced by professional typesetting firms.

DISADVANTAGES OF BATCH-FORMATTING PROGRAMS

Entering the embedded commands into the text is a slow, tedious, and difficult business. Worse, the outcome of the commands is not known for certain until the document is printed or displayed in its printed form on the computer's screen. Figure 1, for instance, shows a document prepared for formatting by *Perfect Formatter* (the companion program of the text editor *Perfect Writer* [Thorn-EMI Computer Software, Inc., 3187-C Airway Avenue, Costa Mesa, CA 92626, available for CPM-80, IBM PC, and PC-compatible computers]). These embedded commands produce the result shown in Figure 2—that is, they do if the commands are properly placed and punctuated.

@style (spacing 2 lines)
@Center (@Bold[Introduction])

  A pioneering work in the study of schizophrenia is Gregory Bateson's 1969 paper entitled "Double Bind." @footnote {Reprinted in his @underline [Steps to an Ecology of Mind] (New York: Ballantine Books, 1972), pp. 271-278.}

Figure 1. Document with Embedded Commands

Introduction

  A pioneering work in the study of schizophrenia is Gregory Bateson's 1969 paper entitled "Double Bind."[1]

_____

[1]Steps to an Ecology of Mind (New York: Ballantine Books, 1972), pp. 271-278).

Figure 2. Printed Document

As these two figures illustrate, using batch-formatting programs can be tedious and difficult due to the high potential for erroneous use of the embedded commands. Commands for *Perfect Formatter*, the formatting program included with the text editor *Perfect Writer*, require (with certain exceptions) that the text to be formatted be enclosed with parentheses or brackets. Leaving one of the parentheses or brackets out will produce spurious results, but the user may not learn of the error until the document has been printed.

Using batch-formatting programs almost always results in a frustrating cycle of making changes to the document's embedded commands, saving the document to disk, running the formatting program, viewing the output file, finding new errors, reloading the document, making more changes, etc. The better batch-formatting programs reduce the tedium somewhat in two ways: (1) by warning the user of inappropriate uses of formatting commands before the document is printed and (2) by permitting the user to preview the output file on screen before it is printed. Even so, the productivity gains made possible by text editors may be lost as you spend hours reformatting the document to print properly.

Another serious drawback of batch-formatting programs is that the embedded commands vitiate the document's onscreen readability, making further editing difficult. As it is displayed on the screen, a document fully prepared for formatting and printing resembles a wilderness of special symbols, parentheses, and commands.

EXAMPLE: *WORDIX*

Wordix (Emerging Technology Consultants, Inc., 1877 Broadway, Boulder, CO 80302, available for CP/M-80, IBM PC, and PC-compatible computers), is designed to work with ETC's *Edix* and similar text editors. *Wordix* well illustrates the strengths and limitations of batch formatting programs generally. It is a comprehensive program, offering virtually every text-formatting feature imaginable (with the unfortunate exception of automatic footnote numbering). Included are automatic foonote placement, automatic hyphenation, automatic numbering of sections in the text, automatic table-of-contents generation, multi-line page headers and footers, mailing list processing, proportional spacing, and (with the optional program *Indix*) automatic index generation.

TEXT EDITOR/BATCH FORMATTER APPLICATIONS

Reading over this list of *Wordix* features suggests precisely the kind of writing job that text editor/batch formatters were originally designed to do: automating the production of technical documents or manuals to be directly reproduced from the program's printouts. You write up the document, insert the appropriate commands, and (after much gnashing of disks) you've got a complete technical manual, including an automatically-generated table of contents, automatically-numbered sections (and even paragraphs), and an automatically-generated index.

Even though the authors of text editors/batch formatters clearly had technical documents in mind when they created them, some of these programs are suitable for scholarly writing. When scholars write for publication, they must adhere to rigid formatting specifications that, for instance, specify precise line lengths, header locations, and other formatting minutiae. Many text editors/batch formatters can do a good job of printing such documents correctly, but watch out for one major problem: footnotes. Technical documents such as software manuals are written for direct reproduction from program printouts and, as such, the footnotes in them are single-spaced. But scholars submit manuscripts that will, if accepted, receive further editing, and all material in them must be double-spaced. Rare is the text editor/batch formatter that can double-space footnotes; the ones mentioned in this essay, with the exception of *Perfect Writer*, can do the job.

ONLINE FORMATTERS

Online formatters offer **interactive** or **online formatting**—that is, a formatting process in which formatting commands, entered via menus or keyboard commands, are directly reflected in the appearance of the text on the video display. The writer works, in other words, with an electronic representation of the page. Ideally, margins, tabs, line

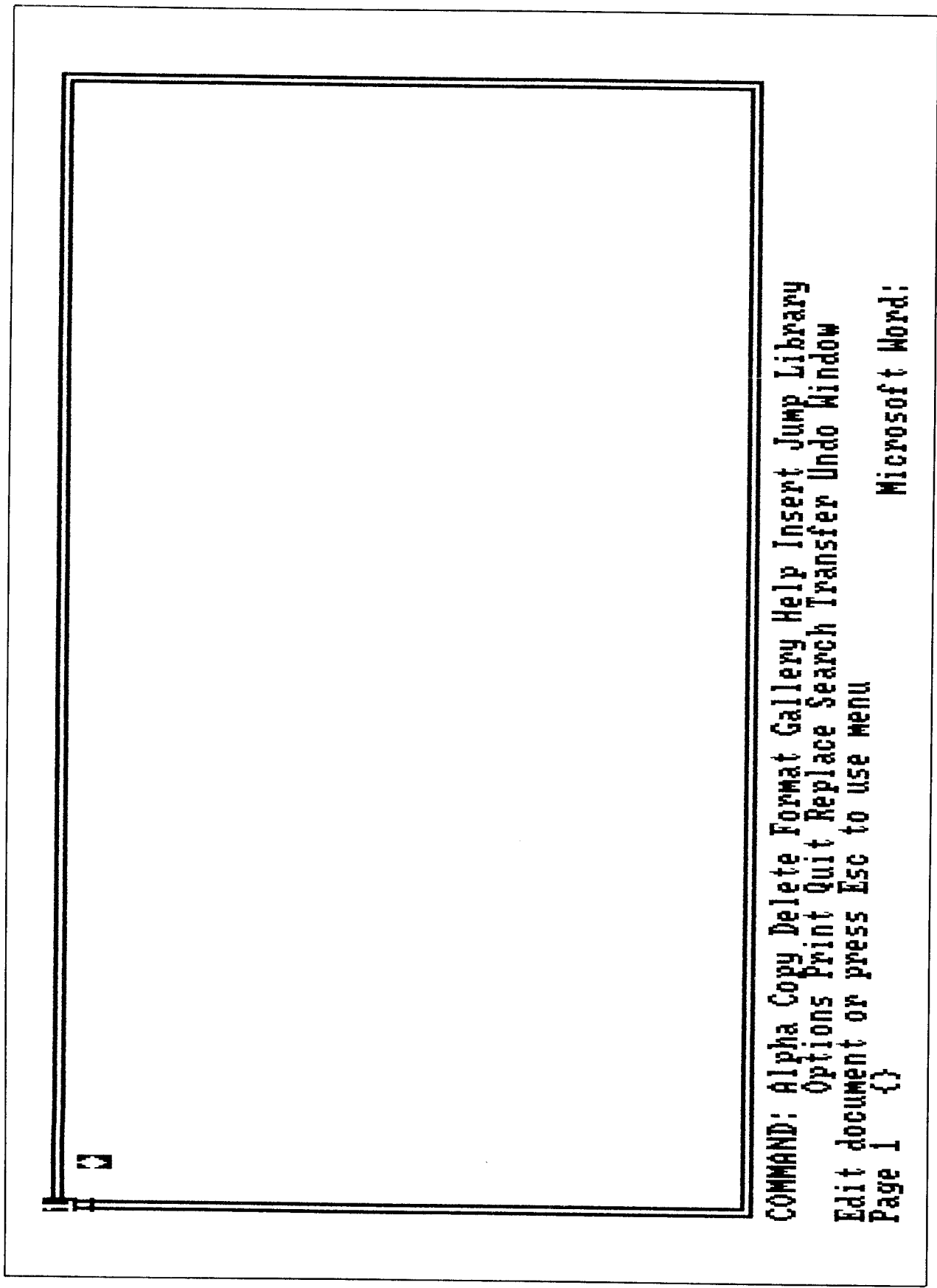COMMAND: Alpha Copy Delete Format Gallery Help Insert Jump Library
Options Print Quit Replace Search Transfer Undo Window
Edit document or press Esc to use menu   Microsoft Word:
Page 1 {}

Fig. 3: *Microsoft Word editing screen.*

spacing, page breaks, and other aspects of page formatting are shown on the screen just the way they will print on paper. What you see, as one firm's advertising stresses, is what you get.

ADVANTAGES OF ONLINE FORMATTERS

True online formatters do away with embedded commands entirely, and for this reason they are much easier to use than batch formatting programs. The formatting process is more intuitively sensible and much more easily managed: in the place of embedded commands are keyboard commands (commands entered by holding down a control key and pressing another key) or menu commands (commands entered by selecting one of several options in a list) which directly and visibly affect the formatting of the text. In one program, for example, holding down the "CONTROL" key and pressing "C" centers the selected text on the screen, and that is precisely the way it will be printed.

You'll probably finish your writing tasks more rapidly with an online formatter than you would with a text editor/batch formatting team. There's no mystery about the formatting command's effect; it's immediately visible. And if the effect if unsatisfactory, you can fix it right away without having to repeat the tedious and time-consuming operation of saving the document to disk, running the formatting program, and viewing the results.

Because there are no extraneous embedded commands to interfere with the document's readability, the documents produced by online formatters are easier to proofread on the screen. Moreover, line and page breaks are shown just the way they will appear when printed, making it possible to pay attention to design considerations such as balance and proportion.

DISADVANTAGES OF ONLINE FORMATTERS

A major drawback of online formatters, however, is the limited number of formatting features they offer. Unlike the text editor/batch formatter design, which gives programmers the luxury of putting editing functions in one program and formatting ones in another, online formatters include both functions in a single program. And because most of these programs were designed for computers with only 64K of memory or, at the most, 128K, memory was lacking to include advanced formatting features as automatic cross-referencing, footnotes, index generation, multiple columns, and table of contents generation. That may rule these programs out for the two writing jobs text editors/batch formatters do so well, technical documents and scholarly articles.

An additional drawback of online formatters is that they invariably use non-standard (that is, non-ASCII) file formats. Many desirable accessory programs, such as spelling and readability checkers, will not work with non-standard text files unless they're specifically designed to do so. If you plan to choose any word-processing program that does not use the standard file format, make sure that the accessory programs you want, such as spelling- and word-choice checkers, are available for it.

EXAMPLE: *MACWRITE*

MacWrite [*RWPN*, Sep. '85] (Apple Computer, 20525 Mariani Ave., Cupertino, CA 95014, available only for the Apple Macintosh) is a true online formatter in the most genuine sense of the term; the video display shows precisely what will appear on the printed page, right down to the size and style of the several character fonts which the program makes available. Active page breaks, which are adjusted when insertions or deletions are made, are displayed by a dotted line across the screen.

Like most online formatters, *MacWrite* (which was designed for the 128K Macintosh) is limited in its features. No provisions are made for such features as footnotes, automatic index generation, printing with multiple columns, automatic section numbering, and the like. Nevertheless, the program offers some attractive features, including onscreen display of the Mac's standard set of foreign language and mathematical characters, the ability to blend

graphic images with text, printer font selection (and onscreen display of the selected font), and a printer control system that (with the Apple Imagewriter printer) produces a high-quality printout that corresponds precisely to the video display.

EXAMPLE: *WORDSTAR*

WordStar [*RWPN*. Feb. '85] (MicroPro International Corporation, 33 San Pablo Avenue, San Rafael, California 94903, available for CP/M-80, IBM PC, and PC-compatible computers) is not a true online formatter; character formats (such as boldface, italics, and underlining) and printing characteristics (such as line height and running heads) are handled by the insertion of embedded commands. Nevertheless, WordStar displays most paragraph formats (such as line spacing, line width, centering, and right margin justification) precisely the way they will appear when printed. The program displays active page breaks, which are adjusted as insertions and deletions are made.

Like *MacWrite*, *WordStar* lacks many features found in more comprehensive word processing programs, such as automatic reformatting, footnotes, index generation, multiple columns, onscreen display of character formats, print merging, printer font selection, proportional spacing, section numbering, table of contents generation, and windows.[3]

ONLINE FORMATTER APPLICATIONS

Online formatters such as *MacWrite* and *WordStar* may not be the best choice for writing technical documentation or scholarly articles, but they're hard to beat for writing short documents in which the aesthetic quality of text arrangement is of importance. *MacWrite*, in particular, is great to have around for writing nicely-designed course syllabi, examinations, letters, and memos.

ADVANCED ONLINE FORMATTERS

Like early online formatters such as *WordStar*, advanced online formatters try to present an interactive representation of the printed page on the display screen. But they go one step further: they try to give you all or almost all the formatting features of a batch formatter such as *Wordix*.

That this goal was set is hardly surprising, given the situation prevailing in the early 1980s. On the one hand, text editors/batch formatters were powerful, but far from interactive. On the other hand, early online formatters were interactive, but far from powerful. The challenge was obvious. Like Mount Everest, it was there. What is more, meeting it became feasible as memories of 256K and more became common.

Whether the results are praiseworthy is another matter. Inevitably, advanced online formatters are big, memory-hungry programs that run sluggishly on systems lacking hard disks. One may question, moreover, whether these programmers put the right goal before themselves. The formatting power of batch formatters is best employed for the production of technical documents to be directly reproduced from program printouts. To be sure, many writers will benefit from having formatting tools appropriate to this task, but there is more to creating good software than providing the wilderness of formatting commands it requires.

The question was not whether the job could be done; with the new, bigger memories, it surely could. The question, rather, was whether the new programs would represent significant advances in making word processing software more versatile and easier to use. And, as will be seen, the results are a mixed bag.

ADVANTAGES OF ADVANCED ONLINE FORMATTERS

Advanced online formatters combine the advantages of batch formatting programs (a wide variety of formatting features) with the online formatter's interactive, visual presentation of the page. The result: sophisticated formatting features such as footnoting or complex patterns of running heads are available online, without exacting the price of embedded commands or batch processing. As such, advanced online formatters are superior to their predecessors—or would be, rather, if they didn't run so sluggishly.

DISADVANTAGES OF ADVANCED ONLINE FORMATTERS

Advanced online formatters tend to be memory-hungry, a fact that is not surprising when it is remembered that they combine the characteristics of batch formatting programs and online formatters. For that reason, they are not available for 8-bit computers; as a rough guideline, these programs require a minimum of 128K or 192K or main memory. Even with a 16-bit computer with its larger memory capacity, however, the whole program cannot fit into the memory all at once: portions of it have to be read from disk when needed.

That means, in short, slow operation and tedious waits with systems lacking a hard disk. Maximum performance from these programs is obtained with systems which are, at this writing, state-of-the-art (such as the IBM Personal Computer AT, with its super-fast 80286 microprocessor, 640K main memory, and 20-megabyte hard disk). Performance on IBM Personal Computers equipped with the standard 8088 chip, 256K of main memory, and dual-floppy-disk drives is intolerably sluggish in comparison to fast programs such as *Edix*.

A second disadvantage of advanced online formatters is that, like early online formatters, they do not use the ASCII file format, and that can cause problems when you want to use the program with accessory software. Before buying an advanced online formatter, make sure the accessory programs you want can work with its file format. A third disadvantage of advanced online formatters is their complexity. Complexity is a double-edged sword: it makes a wide variety of formatting features possible, but it forces the user to accept a long, frustrating learning period as dozens or even hundreds of commands are mastered. Worse, so much effort went into equaling the formatting prowess of batch formatters that comparatively little went into improving the program's ease of use. Programs in this category tend, therefore, to present major challenges to beginners in computing.

EXAMPLE: *WORDSTAR 2000*

WordStar 2000 (MicroPro International Corporation, 33 San Pablo Avenue, San Rafael, California 94903, for IBM PC and PC-compatible computers), exemplifies the conventional thinking that fueled efforts to achieve an advanced online formatter. In its optional *WordStar 2000* "Plus" configuration, the program offers virtually every word-processing feature found in the best batch formatters, including index generation, section numbering, and table of contents generation. As an online formatter, the program resembles the earlier *WordStar* (which is, in fact, a completely different program and is still available in its 3.3 version).

WordStar 2000's programmers succeeded well enough in their quest to unite the batch formatter's formatting power with the online formatter's interactive page display, but the victory—truly Pyrrhic—does little to advance the design of word-processing software. The new program makes some improvement over *WordStar*'s user interface, which beginning users find forbidding. And to top it off, the program runs so sluggishly that MicroPro recommends it only to those whose computers are equipped with hard disks.

EXAMPLE: *MICROSOFT WORD*

Microsoft Word [RWPN, Mar. '85 (MS-DOS); Dec. '85 (Macintosh)] (Microsoft Corporation, 10700 Northrup Way, Box 97200, Bellevue, Washington 98009, available for IBM PC, PC-compatible, and Apple Macintosh computers) is an advanced online formatter that, like *WordStar 2000*, appears to have had its origins in the quest to unite batch

formatting power with an interactive page display. Unlike *WordStar 2000*, however, *Word* achieved some noteworthy innovations.

That the original goal was the same as the one set for *WordStar 2000's* programmers can be seen in the Version 1 manual, in which a note promises that future versions of the program would include automatic section numbering (and, it follows, automatic table of contents generation). Yet the goal was given up; the Version 2 manual makes no mention of it. For the Version 2 revision, Microsoft decided to devote attention to ease of use and other matters of concern to users rather than trying to out-perform batch formatters.

Word departs from the conventional thinking underlying *WordStar 2000* in another respect, too. The program's little-understood **style sheet** feature represents a major innovation in microcomputer word processing. In brief, *Word's* style sheets let you define your own set of keyboard commands to suit precisely the kind of writing you do. If you write for a journal that requires headings to be boldfaced and centered, with three blank lines above and two below, you can create a keyboard command (entered with the PC's "ALT" key, such as "ALT-H1") that **simultaneously** attaches all those formatting characteristics to the cursor. After entering the command, everything you write will appear on the screen (and will print) in precisely that format until you enter another command that begins a new one.

When you write with *Microsoft Word*, in short, you are not saddled with the formatting commands programmers thought appropriate for you; on the contrary, you are given the tools you need to create precisely the formats you want.

Microsoft Word could well be described as an **high-level programming language** for the formatting of text. A high-level programming language is one that lets you instruct the computer using English terms or other intuitively sensible and familiar operations. *Word* does precisely that by letting you define keyboard commands using a series of menus.

Sadly, this innovation appears to have preceded the market's sophistication. Microsoft found in a survey of *Word* users that most did not grasp the style sheet's potential for their writing and made no use of it. Yet writers who have discovered it and understood its appeal tend to love the program. Screenwriters saddled with studio's bizarre formatting strictures provide a case in point, and a major screenwriter's association has recommended the program to its members.
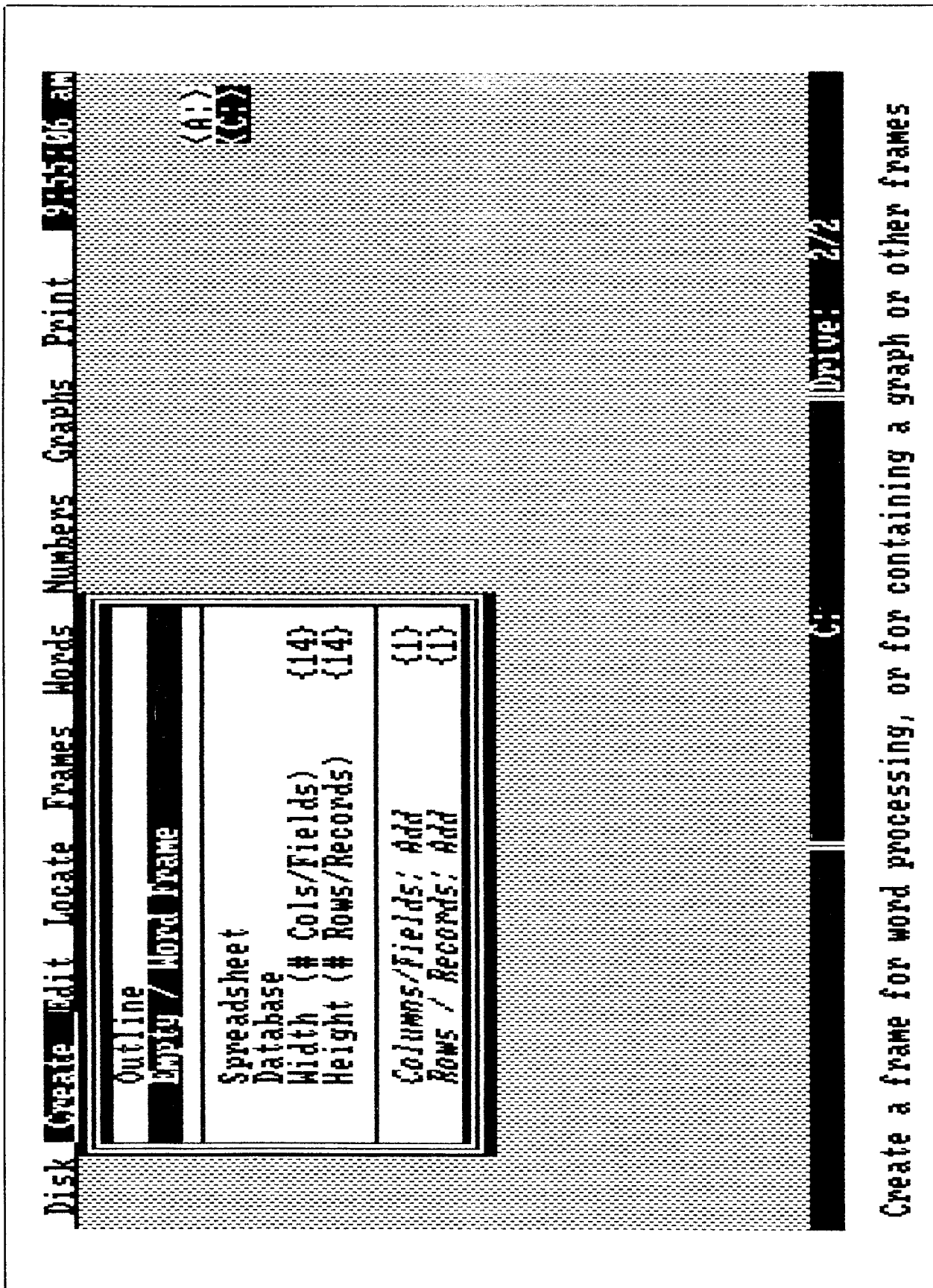
INTEGRATED WORD-PROCESSING PROGRAMS

With the 1984 arrival of the 512K Macintosh, a standard IBM-PC configuration of 256K, and a dramatic fall in the cost of 64K RAM chips, the stage was set for some fresh thinking about word processing software. All that additional memory opened the possibility of adding new kinds of program functions to the old standard ones.

Happily, a few programmers thought beyond the traditional confines of word processing programs, and the result is a new genre of **integrated word-processing programs**; word-processing programs that include one or more word-processing accessories not usually included with word-processing software, such as word-choice checkers (thesaurus) or idea processors.[6]

A word-processing program, after all, is only one of several programs useful for writing. Other programs, such as idea-processing programs and word-choice checkers, can work with word processing software to help you do a better job organizing and proofreading your text. But using these accessory programs forces you back into the old batch-processing mold. You write your document, save it to disk, leave the word-processing program, and run the accessory program on it. Now here's the bright idea: why not have these tools available online—active in the computer's memory and within the reach of your word-processing program's commands, so that you can use them while you're writing?

WordStar 2000 exemplifies, in a limited way, this idea's potential. Its spelling-correction program, *CorrectStar*, is

```
Disk  Create  Edit  Locate  Frames  Words  Numbers  Graphs  Print    9:55:46 am

                                                              <A:>
                                                              <C:>

    Outline
    Empty / Word Frame

    Spreadsheet
    Database
    Width  (# Cols/Fields)          {14}
    Height (# Rows/Records)         {14}

    Columns/Fields: Add             {1}
    Rows / Records: Add             {1}




                                                    C:    Drive:   2/2

Create a frame for word processing, or for containing a graph or other frames
```

Fig. 4: *Framework "Create" pull-down menu.*

available online while you're writing; you can check, for instance, the spelling of a particular word or paragraph and then go on writing. But this is hardly a radical innovation. A program conventionally sold with word-processing software was simply moved into the program so that it was available online. A truly integrated word processing program would make a **variety** of word processing accessories available online.

The idea of integrated word-processing software is so new that one can speak only provisionally of its inherent advantages and disadvantages. The advantage of such software, clearly, is that you would have several word-processing accessories available online while writing, but just how useful this feature would be depends almost entirely on which accessories were provided. A poet, for instance, would have little use for an automatic index-generation program, but would surely find helpful a program that scanned an 80,000 word dictionary for all the words that rhyme with the one on which the cursor is positioned.

As for disadvantages, one may look for instruction at the efforts to blend all five software functions (word processing, database management, spreadsheet, graphics, and communication) in such programs as Lotus Development Corporation's *Symphony*. Often, one function is highly developed—*Symphony*'s spreadsheet, for instance, is the state of the art—but the others may have been pared down or deprived of features so that they would fit.

EXAMPLE: *FRAMEWORK*

Framework [*RWPN*, Apr. '85] (Ashton-Tate, 10150 West Jefferson Blvd., Culver City, CA 90230, for the IBM-PC and PC-Compatibles) isn't an integrated word-processing program in the sense used here, but it points the way towards one direction such software may take. It includes (among other things) a sophisticated **idea processor** or outlining program linked to a word processor (an online formatter with limited features).

Idea processing programs such as Living Videotext's *ThinkTank* let you create and restructure an outline. By linking an idea processor with its word processor, *Framework* goes a step further: each heading in the outline contains a hidden block of text. You can look at your document then, either in terms of its overall structure (the outline view, in which the text is hidden) or the text itself (the text view, in which the outline is hidden). What is more, making changes in the idea processor's outline actually restructures the text of the document.

The combination of an idea processor and word processor makes for a powerful and elegant solution to a major drawback of word processing: the limited view of your document provided by that 24-line display and slow scrolling pace. Unfortunately, the current version (1.01) of *Framework*'s word processor does not include features necessary for scholarly writing, such as unlimited document length and superscripts.

EXAMPLE: *NOTA BENE*

The promise of things to come is evident in *Nota Bene* [*RWPN*, Jan. '86] (Dragonfly Software, 409 Fulton Street, Suite 202, Brooklyn, New York 11201, for the IBM PC and PC-Compatibles), a remarkably complex program that combines an advanced, full-featured online formatter with a text-oriented, free-format data base management program. Among other things, this combination permits a writer to pause while writing a document, search a large file of notes (using logical or Boolean operators) for needed material, and paste the material directly into the document.

CONCLUSIONS

This overview of the four basic categories of word processing software has been roughly chronological; text editors/batch formatters date back to pre-microcomputer days, and were followed by online formatters, advanced online formatters, and integrated programs. Just because text editors/batch formatters have been around for years, however, doesn't necessarily mean you should rule them out. If you can live with embedded commands and batch processing, the features of a text editor/batch formatter may offer ample compensation indeed.

The wave of the future, however, is clearly represented by integrated programs (such as *Framework* or *Nota Bene*), which attempt to place idea processing, text-oriented database managers, or other accessory functions at a writer's disposal **within** the word-processing program. No integrated word-processing program has yet appeared that can be called fully satisfactory, but the efforts to date are promising indeed. New microcomputers with larger memories and faster processors will make such programs all the more feasible.

What is especially appealing about this new direction in word processing software is its potential affinity with the way people actually write. As Linda Flower and John Hayes of Carnegie-Mellon University have so convincingly demonstrated,[7] writing cannot be broken down into rigidly sequenced steps of (1) planning, (2) writing, and (3) revision/proofreading. That is the way you have to write with today's software: you begin with an outlining program, and then write with a word processor; finally, you use spelling and word choice checkers to assist the proofreading and revision. But Flower and Hayes found that writers like to cycle back and forth among the three steps, experimenting (for instance) with writing something and then going back for more planning before proceeding.

If Flower and Hayes are correct, then we have plenty of **word-processing** programs but have yet to see one that is truly for **writing**.[8] But at last the path toward such a program is becoming clear. Imagine a program, for instance, that combines all the following features and lets you use **any** of them at **any** time:

- **Planning**—An idea processor with a *Framework*-style link to the word processor, coupled with online access to a text-oriented database management program that stores research notes, outlines, and bibliographic citations;

- **Writing**—A full-featured, advanced online formatter like *Microsoft Word*;

- **Revising/Documenting**—A variety of functions for text revision (such as readability analysis, word choice checking, spelling checking, vocabulary richness analysis, and word frequency analysis) and others for documentation (such as automatic generation of bibliographies from key word citations placed in the text, automatic generation of footnote text from key word citations placed in the note, and automatic generation of indices).

This imaginary program may sound like something from the distant future. But all the pieces needed to create it already exist, and Flower and Hayes have provided the theoretical foundations. We may speak, in short, of a chronology and typology of word-processing programs, but the more exciting story—the history of **writing** programs—has only just begun.

NOTES

[1]This essay is excerpted from *The Scholar's Personal Computing Handbook: A Practical Guide* (Boston: Little, Brown, 1986).

[2]"Some Words for the Trepid," *A Grin on the Interface: Word Processing for the Academic Humanist* (New York: Modern Language Association, 1984), p. 2.

[3]This terminology, it should be noted, is by no means universally accepted; some prefer the term "screen-oriented" to online formatter or "stand-alone formatter" to batch formatter.

[4]Cited in Curt Supplee, "The PC Tapping at the Chamber Door," *PC: The Independent Guide to IBM Personal Computers* (May 29, 1984), p. 254.

[5]Accessory programs such as *Footnote* (Pro/Tem Software, 814 Tolman Drive, Stanford, CA 94305) are available that equip *WordStar* with some of these features, but at the cost of turning the program into a batch formatter.

[6]Note that integrated word-processing programs should be distinguished from the kind of integrated programs which, like *Symphony*, attempt to put all five basic software functions (word processing, database management, spreadsheet, graphics, and communications) together into a single package. We are speaking here of programs that integrate several formerly separate **word-processing** programs and accessories.

[7]"A Cognitive Process Theory of Writing," *College Composition and Communication*, 32 (December, 1981), pp. 365-387.

[8]Mainframe programs such as *Writer's Workbench*, to be sure, represent a major step in this direction. But *Writer's Workbench*'s tools are available as a collection of separate programs; they're not available online as you write.

Contributing Editor **Bryan Pfaffenberger** is a writer and anthropologist who teaches in the Division of Humanities, School of Engineering & Applied Science, University of Virginia. He's the author of numerous articles and books, including *The College Student's Personal Computer Handbook* and *Macintosh for College Students* (both published by Sybex Computer Books). His more recent *The Scholar's Personal Computing Handbook: A Practical Guide*, will be available this year from Little, Brown and Company. Bryan is currently working on another text, *Dynamics of Microsoft Word*, in both IBM and Apple Macintosh editions for Dow Jones/Irwin. Comments and dialogue are welcome; contact Bryan at 218 Sunset Ave., Charlottesville, VA 22903.

Academic's Guide to Microcomputer Systems

The University of Toronto Computing Services (Dr. Willard McCarty and others) has recently published a 217-page *Academic's Guide to Microcomputer Systems*, a well-rounded survey of those aspects of microcomputing which the typical professor is likely to need to know. Designed as a resource for University of Toronto faculty, the guide teaches about necessary computer concepts in easy-to-follow explanations.

Perhaps even more important, the guide offers numerous hardware and software reviews which keep the microcomputer-using academic in mind. It is available in different forms for under $18.00, including postage and handling. Contact Ms. Dale Wright, Information Office, University of Toronto Computing Services, 255 Huron Street, Room 350, Toronto, Canada M5S 1A1.

IEEE Professional Communication Society Conference

The IEEE Professional Communication Society plans to take its annual conference to Canada for the first time in 1987. The chosen site is Winnipeg, Manitoba in the mid-West, which has direct airline access from Chicago (1½ hours), Minneapolis (1 hour), and Denver (1½ hours), as well as from Toronto and Vancouver in Canada. The conference is scheduled for October 14-16, 1987, at the Sheraton Winnipeg.

The theme for the three-day event will be "Engineering Communication: A Byte into the Future," although conference chairman Ron Blicq says that topics will range broadly from oral communication to teaching report writing by computer. Those who would like to be placed on the conference mailing list should contact The Steering Committee, PCC87, Box 181, Postal Station C, Winnipeg, Manitoba, Canada R3M 3S7.

Teaching Computers and the Humanities Courses

A workshop on *Teaching Computers and the Humanities Courses* will be held at Vassar College from the evening of July 31 through noon on August 2, 1986. The intent of the workshop is to provide a forum for discussion of appropriate orientation and content for computing courses designed for humanities students as well as how best to present the material to these students.

Panel discussions on these topics will be interleaved with small group discussions in order to encourage open dialogue between workshop attendees and panelists. The workshop is sponsored by the Association for Computers and the Humanities and the Alfred P. Sloan Foundation. Contact Elle Gohl, Workshop Coordinator, Department of Computer Science, Box 252, Vassar College, Poughkeepsie, NY 12601.

San Francisco Conference Includes Computer Emphasis

"A Literacy Debate: Reasoning, Style, and Writing" is the theme of a conference to be held at the University of San Francisco next June 12-14, 1986. Speakers will include Walter Ong, S.J., on the topic of "Plato, Writing, Print, and Computers." Workshops will include one led by Michael Cohen on "The Role of Computers in the Literacy Debate." Registration has been set at $155.00 per person. Contact Dr. Brenda Schildgen, University of San Francisco, San Francisco, CA 94117.

NCTE Spring Conference in Arizona

K-12 and teacher-education faculty attending the March 20-22, 1986, NCTE Spring Conference in Phoenix, Arizona, will find a few sessions dealing with computers in the classroom:

Thursday: ■ Integrating the Microcomputer into the Elementary Classroom ■ Computers and Teaching Writing

Friday: ■ A Computer Lab for Composition: Those Magnificent Kids and Their Writing Machines ■ Teaching Teachers to Use Computers to Teach Writing

Saturday: ■ Computers as a Supplement to Classroom Instruction ■ Computers in Teacher Education

---

Contact the National Council of Teachers of English, 1111 Kenyon Road, Urbana, IL 61801.

Conference on College Composition and Communication

New Orleans will host the 37th Annual CCCC Convention between March 12-15, 1986. Computer applications appear in the following workshops, sessions, and Special Interest Group meetings:

Wednesday, March 12

■ *Introduction to the Computer: Opportunities for Practice and Learning.* (Workshop Leaders: James Gifford, Donald Pattow, Bruce Staal, William Wresch)

■ *Evaluating Computer Software.* (Workshop Leaders: Michael G. Southwell, Elray Pedersen, David N. Dobrin)

Thursday, March 13

The Possibilities of Computer-Assisted Instruction ■ Research on Computers and Composition ■ Networking Computer-Based Rhetoric ■ Word Processing, Composing, and the Learning-Disabled Writer ■ Interacting with Student Writing Present and Future: Impossible Commentary Tasks Made Possible by Computers ■ The Fifth C: Computers

---

Friday, March 14

Writing, Rhetoric, and Computers in Business ■ Computer Conferencing as an Educational Tool ■ Sentence Combining, Revising, and Computers: Toward New Applications ■ Expanding Views of Computer Conferencing ■ Using Computers to Write about Literature ■ The Role of the Computer in Revising ■ Innovative Uses of the Computer

---

Saturday, March 15

■ Designing a Comprehensive Curriculum for Writing with Computers

---

Contact the CCCC, National Council of Teachers of English, 1111 Kenyon Road, Urbana, IL 61801.

Bibliography Update

By Bradford A. Morgan

Barnes, Karen and Dan Epstein. "Changing Character: Technical Problems in the Multilingual Support of UNIX." *UNIX Review.* 3:12 (December 1985), pp. 36-42.

Bigelow, Charles and Donald Day. "Digital Typology: Most Type Is Now Produced Not by Casting Metal or by Photography but by Computer. The Digital Typesetter Can Create New Letterforms with the Flexibility of a Scribe at up to 15,000 Characters per Second." *Scientific American.* 249:2 (August 1983), pp. 106-119.

Burns, Diane and S. Venit. "Word into Type: Meeting the Corporate Challenge. Laser Printers, Typesetting Front-ends, and Page Composition Software Are Taking Corporate Publishing by Storm. A $10,000 Invest-ment Can Turn Your Company's Printed Pages into Professional-Quality Documents." *PC: The Indepen-dent Guide to IBM Personal Computers.* 4:25 (December 10, 1985), pp. 180-182.

----------------. "PCs and Typesetters: A Mixed Marriage. Thanks to Front-end Software, PCs Can Now Drive Typesetters and Laser Printers Directly, Generating Both Reproduction-Quality Copy and Real Cost Savings." *PC: The Independent Guide to IBM Personal Computers.* 4:25 (December 10, 1985), pp. 194-195, 202.

----------------. "Page Design on the PC Screen: Companies with One or More PCs Can Now Put Away Their Knives and Glue. These Seven Page-Composition Programs Bring Design, Coding, and Editing In-House." *PC: The Independent Guide to IBM Personal Computers.* 4:25 (December 10, 1985), pp. 180-210.

Cavuoto, Jim. "Word Processing Software: Demand for Quality Output Will Force Packages To Improve Their Cast of Characters." *PC Week.* 2:46 (November 19, 1985), pp. 133, 135, 142.

Chiang, Katherine S. et al. "Creating Bibliographies for Business Use: Literature-Retrieval Programs Can Help You Organize Unwieldy Collections of Brochures, Legal Documents, and Corporate Reports, As Well As Academic Books and Journals." *PC: The Independent Guide to IBM Personal Computers.* 4:23 (November 12, 1985), pp. 249-260.

Cowart, Robert. "Database Utilities: Enhancements Ease the Chores of Sorting, Indexing and Packing with dBase." *PC Week.* 2:49 (December 10, 1985), pp. 57-78, 90.

Garrick, Michael D. "Word Processing Software Simulates Cloning Procedures." *BioTechniques.* 3 (1985), pp. 453-454.

This article describes the use of several common word processing programs to simulate restriction cut-ting and DNA ligation in handling DNA and RNA sequence data. The handling of sequence data is one area of word processing which has not been well researched or documented, although almost everyone involved in genetics and biotechnology has had to improvise some way of dealing with it. It is an area which really needs some creative product development.

Everett Larsen, Department of Agricultural Chemistry, Oregon State University.

Gaunt, Marianne I. "Rutgers Inventory of Machine-Readable Texts in the Humanities." *Data Bases in the Humanities and Social Sciences.* ed. Robert F. Allen. Osprey, FL: Paradigm Press, 1985, pp. 283-290.

Greitzer, John. "Setting Type with PCs: Micros Help Corporate America Move Toward In-House Publishing." *PC Week.* 2:49 (December 10, 1985), pp. 37-39.

Haegele, Gerald E. "An Automated School Newspaper." *English Journal.* 75:1 (January 1986), pp. 35-37.

Hayes, Brian. "A Progress Report on the Fine Art of Turning Literature into Drivel." *Scientific American.* 249:5 (November 1983), pp. 18-28.

Heller, Ron and Mark Hizer. "Flexing *WordStar*'s Character Widths: Give Your Pages 'Character' with the Character Width and Page Offset Commands." *IBM PC Update.* 2:12 (December 12, 1985),. pp. 6-8.

Jeffries, Ron. "Goodbye, Gutenberg! Within the Next 15 Years the Printed Book May Be Replaced by Notebook-Style Computers with Optical Disk Readers. The Reasons, Notes Gary Kildall in this Interview, Are Mainly Economic." *PC: The Independent Guide to IBM Personal Computers.* 4:23 (November 12, 1985), pp. 95-98.

Johnson, Bonnie M. and Ronald E. Rice. "Reinvention in the Innovation Process: the Case of Word Processing." *The New Media: Communication, Research, and Technology.* ed. Ronald E. Rice. Beverly Hills, CA: Sage Publications, 1984, pp. 157-183.

Lemley, Brad. "PCs Paginate USA Today: Front-Page Editor Ray Gniewed Writes an Efficient and Cheap Compiled BASIC Layout Program for USA Today." *PC: The Independent Guide to IBM Personal Computers.* 4:25 (December 10, 1985), p. 202.

Microcomputer Support Group. "Word Processing Packages." *Academic's Guide to Microcomputer Systems.* Toronto, ON: University of Toronto Computing Services, 1985, pp. 62-109.

Morrissey, Robert and Claude Del Vigna. "A Large Natural Language Data Base: American and French Research on the Treasury of the French Language." *Data Bases in the Humanities and Social Science.* ed. Robert F. Allen. Osprey, FL: Paradigm Press, 1985, pp. 17-21.

Mullins, Carolyn J. "Word Processing and Graphics: Software Vendors Are Bringing Text and Images Together." *Graphics: Supplement to PC Week.* 2:46 (November 19, 1985), pp. S/19-S/21.

Paijmans, J. J. "Natural Language Databases on Microcomputers." *Data Bases in the Humanities and Social Sciences.* ed. Robert F. Allen. Osprey, FL: Paradigm Press, 1985, pp. 341-346.

Pepper, Jon. "Banish Typos with Spelling Checkers: Spelling Errors and Typos Can Undermine Your Carefully Chosen Words Faster Than You Can Say You're Sorry. These Eight Software Programs Mean You'll Never Have To." *PC: The Independent Guide to IBM Personal Computers.* 4:26 (December 24, 1985), pp. 199-213.

Raimondi, Donna. "Humanities Take a Course in Automation, Save a Bundle: Wang Helps NEH Cut the Cost of Patronage." *Computerworld.* 19:51 (December 23, 1985), pp. 25-26.

Satterwhite, Robb. "Subject Indexing and Creation of a Controlled Vocabulary: The MLA Bibliography and the MLA Thesaurus." *Data Bases in the Humanities and Social Sciences.* ed. Robert F. Allen. Osprey, FL: Paradigm Press, 1985, pp. 209-217.

Slatta, Richard W. "Data Base Management Software and Historical Research." *History Microcomputer Review.* 1:2 (Fall 1985), pp. 19-27.

--------------. "Teaching Historical Research Methods with Data Base Software." *History Microcomputer Review.* 1:2 (Fall 1985), pp. 28-33.

Software Digest. *The Ratings Book, 1984/85 Edition: IBM PC Word Processing Programs.* Vol. 1. Wynnewood, PA: Software Digest, 1984.

Standera, O. L. "Electronic Publishing: The Predicament of Occasional Users in the Editorial Process." *Journal of the American Society for Information Science.* 36:4 (July 1985), pp. 230-240.

Wallach, Wendall. "Text-Management Programs Bridge Applications Gap." *Office Systems '85.* 2:12 (December 1985), pp. 38, 40.

Weiner, E. Judith. "The Role of the Knowledge Network in the Computational Processing of Literal and Non-Literal Language." *Data Bases in the Humanities and Social Sciences.* ed. Robert F. Allen. Osprey, FL: Paradigm Press, 1985, pp. 402-409.

Addendum to *Microsoft Word* (Macintosh)

Tom Carney

Readers of the *Newsletter* are in William Kemp's debt for marshaling such a concise but wide-ranging survey [Dec. '85] of what *Word* has to offer. The following notes are submitted to round out what will be for many of us a quick guide to *Word* that we can hand to potential Apple Macintosh users.

- *Word* is now at version 1.05 and supports the Apple LaserWriter. Other new features of 1.05 include the following: the "Word Convert" program moves files back and forth between the Macintosh and the IBM-PC (and compatibles); graphics pictures can now be reduced to 10% of their original size without distortion; bottom-of-page footnotes now work without font resizing hassles.

- Variable line spacing is possible: all you have to do is change "auto" in the "Paragraph Formats" menu to the numerical specifications—in inches or centimeters—that you want.

- Keyboard macros *are* available if you install a keyboard enhancer such as *MacTracks* as a desk accessory.

- *Word* facilitates the integrated use of many other applications along with it:

    1. It accepts graphics from *MacPaint,* and *Paint* accepts (and can enhance) diagrams from *MacDraw,* Microsoft *Chart,* and *Multiplan.*

    2. You can transfer textfiles both ways between *Word* and *ThinkTank 512.* So, effectively, if you're using *Switcher* and a 512K Macintosh, you can have an idea processor running along with your word processor.

    3. *PageMaker,* currently one of the most popular desktop-publishing programs, accepts *Word* (and, in fact, *requires* its text to be input from a word processor). Thus, you can do sophisticated editing of newsletters, manuals, and so on with your Macintosh, using *Word* for the all-important, text-preparation stage.

Professor **Tom Carney** teaches in the Department of Communication Studies at the University of Windsor (401 Sunset Avenue, Windsor, Ontario, Canada N9B 3P4). Look for a feature from Tom on Desktop Publishing in a future *RWPN.*

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 accepted, but please contact Dr. Jim Schwartz, Hardware/Software Review Editor, **before** submitting them (call Jim at 605-394-1246). Manuscripts either may be submitted as hard copy or on 5¼" diskettes using *WordStar, WordStar 2000.* 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. The Editors may also be reached on *CompuServe* (70177, 1154) and the *Source* (AAH500)

NOTES