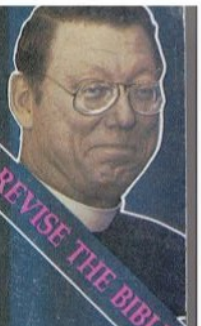
PC
The Independent Guide to IBM Personal Computers
Volume 1 Number 6 $3.00

Getting the Most out of Your IBM Personal Computer

USING THE PC TO REVISE THE BIBLE



REVIEWS
of UCSD p-System, Concurrent CP/M, IBM COBOL, and The Programmer.

PLUS -
Writing Data Files, How to Copyright Your Software, Tips on Memory Expansion, Setting Up an Executive Work Station, and an Introduction to RatBAS, a Language for Structured BASIC Programming.

Making The Elephant Tap Dance

IBM EYE/JIM EDLIN

Making The Elephant Tap Dance

Whatever your Personal Computer sales representative knows about PCs, chances are he or she learned it from Hal Jennings.

As a Marketing Support Representative for IBM's Systems Products Division in Boca Raton (the division that makes the PC), Hal Jennings' official responsibility is teaching the courses that authorize people to sell the PC. But over the last several months Jennings has volunteered himself for an additional task; he has become a principal author of the demonstration programs IBM provides dealers to help the PC sell itself.

Enroute to creating the demonstration programs, Jennings also created some powerful software tools. One steamy Florida afternoon (when even the air conditioning could barely preserve IBM's traditional blue-suited cool), Jennings took time out to show off his tool kit to a visitor from *PC* and to explain some of the techniques behind the tools.

"The dealers were packing 'em in," says Jennings, describing the necessity behind the development of his Graphics Character Generator (GCG) program. "There were layers of people viewing the demonstrations, and the regular-size characters on the screen just weren't readable for people behind the front rank."

Hal's solution was to create a program, the GCG, that uses the medium-resolution graphics mode of IBM's color/graphics adapter to create screens with large characters that have an almost typeset appearance. The GCG, written in BASIC, has three major subparts: one for designing video "type fonts," one to create "screens" using the fonts, and one to display screens after they are created.

Once a video type font is created with GCG, it is stored as a disk file. A font file may contain anything from a single character up to a full alphabet in two variations—about 200 characters. A font is created by designing one character at a time, so if you wanted to create a complete alphabet of 26 capital letters, 26 lowercase letters, ten numerals, and a helpful assortment of punctuation marks and symbols, you would have to go through the design process 80 or 90 times. If you wanted to design a second version of the font (say an italic variation of your original creation), you would have to repeat the process an-

other 80 or 90 times. Hal Jennings has programmed shortcuts to make this somewhat less daunting than it sounds.

A Blank Grid

On the GCG program's initial menu the character design process is called "Define." When Jennings demonstrates this part of the tool, the screen clears and divides vertically into two work areas. The left area is a straightforward presentation of facts concerning the design in progress; the right area displays a blown up representation of the character being designed. Initially the right part is a blank grid, somewhat like graph paper, on which the character shape will be sketched.

The large characters of Jennings' type fonts, like characters displayed on computer screens in general, are made up of dots. It is the video equivalent of a dot matrix printer. Normal size characters on the PC's color/graphics display are arrangements of dots within a grid, or matrix, seven columns wide and nine rows high (on the monochrome display the grid is nine wide by 14 high); however, Jennings' program lets you set the grid size up to a maximum of 24 high by 24 wide. A separate grid height and width can be specified for each character designed, allowing the creation of proportionally spaced character sets while economizing on storage space for descriptions of characters that require less than the full grid. If a grid smaller than the maximum is chosen, matrix positions are eliminated from the bottom or right side while the upper left corner holds a constant position.

JENNINGS estimates he spent about 30 minutes designing each character.

The first step in designing a character is selecting the code number (between 1 and 254) the computer will use internally to signify that character. If you were about to design a capital *A* for your new type font, for example, you might designate it as character number 65. (This choice would be practical because 65 is the number used to identify capital *A* in the PC's internal code for its standard character set, which is an expanded version of the widely used ASCII code. The program doesn't require that your numbering conform to the ASCII conventions, but it's generally more convenient to do so.) GCG will let you enter the code in either the decimal or hexadecimal numbering systems, and will then display the number in both systems, along with the standard PC character corresponding to the number. Type in &H42 and the display will show: "Decimal 66, Hex 42, B". At this stage you can also define both the horizontal and vertical sizes of the design grid within which you want your character to fit.

Designing a Typeface

Once the preliminaries are done, a cursor appears in the right section of the screen. You can move the cursor to any dot in the grid, and then switch it from on to off or vice versa. A portion of the left-side display shows the evolving character's actual size as you create its greatly magnified version on the right. You can build up a character shape in this sample fashion, dot by dot. Or, once you get started, you can employ more powerful editing functions, such as copying the dot pattern in one horizontal row to the row below it.

Jennings has also provided for second thoughts. If you are designing a capital *P* and belatedly decide you want the top section more elongated, you can move the cursor to the point at which you want more elongation, spread apart two adjacent rows, and insert a new row of dot positions between them. Such an insertion pushes the bottom row of dot information off the grid. Where does it go? "To the great bit bucket in the sky," says Hal Jennings.

Deletion of a dot row is also possible, but it brings only a row of blank dot positions into the grid. No dots are retrieved from the celestial bit bucket. Another aid to font creation is the ability to take the shape of a character you have already created and edit it to create others; so the *B* can also become the basis for the *P*, *F*, *E*, and so forth.

"Developing this tool took quite a while," says Jennings. But after he developed it, he realized that he'd won only half the battle. With the design program ready to use, Jennings had to exchange his programmer's hat for a typographer's visor. He chose to make a video version of the type style Garamond, which IBM had settled on as the corporate typeface for the Personal Computer. (Garamond is used in all IBM advertising for the PC, in the user manual, and even on the nameplate of the computer itself.) This became a process of trial and error, attempting to recreate the printed versions of the letter forms using dot patterns on the grid, then refining and refining again. Jennings estimates he spent about 30 minutes designing each character—about a solid week's work.

The process held some surprises for Jennings, particularly when he got around to designing the italic letter forms used to print "Personal Computer" on the screen, the same way it appears on the PC nameplate. "I have a scientific background," Jennings explains, "and scientists like to see things symmetrically. So when I start-

The Hoofing Pachyderm

Jennings' animated reply to an IBM critic.

It seems a computer industry observer commented in print that IBM would have a great deal of trouble being successful in the personal computer business. "It would be like teaching an elephant to tap dance," the analyst said. This comment really stuck in Hal Jennings' craw. So one day he sat down and programmed a picture of a friendly-looking elephant up on his hind legs. A bit more code and there was a nice, rhythmic sound track. Soon the elephant's legs were animated to tap along to the music while the tail wagged to beat time. Finally, Jennings, as both programmer and appreciative audience, programmed in a little crowd reaction to the elephant's performance. "Encore! Encore!" shouts the off-screen crowd. And the elephant dances again. —*J.E.*

ed designing the italic letters, I gave them all the same slant. But they didn't look right. Finally I figured out that some of the italic letters, *l* and *p* for example, have different slants. It blew my mind." Perhaps these problems explain why Jennings never finished a complete italic version of the Garamond typeface; he did a full set of the roman Garamond, but in italic he designed only the letters used to spell personal computer. Then, to go with these, he made vid-

SOME OF the italic letters have different slants. It blew my mind.

eo versions of the three letters *IBM* as they appear in the company logotype, allowing a very close reproduction of the PC name as rendered on its nameplate.

Shapes Stored as "DRAW" Strings

When you tell the GCG program you are satisfied with a character design, it processes the pattern you entered in the dot grid. The result is a string of text that is later used as instructions for PC Advanced BASIC's DRAW statement, which recreates your character shape on the screen. (The text in a DRAW string serves as a series of commands for the action of an imaginary pen moving across the screen and leaving dots of light instead of ink.) GCG's screen creation and display parts are built around the DRAW statement. If the shape of a graphics character is very complex, the DRAW string can be as long as 700 characters. This string is stored as the characters' entry in the font file on disk.

With a font designed, you are ready to use the screen creation part of the GCG program. This section of the program can theoretically be used with any font file stored on disk, though Garamond was the only one Jennings had designed at the time of the demonstration.

Jennings' screen creation section allows keyboard control of all the variables in appearance that are possible using the PC's medium-resolution graphics mode. Controls activated by the ALT key allow both foreground color and background color (analogous to ink and paper) to be set either character by character or for the screen as a whole. A clever bit of programming lets text be displayed in certain colors the PC doesn't normally offer, since Jennings' DRAW strings cause the imaginary pen to draw characters by moving up and down in a set of parallel rows, and he allows you to select different colors for alternating rows. Using this ability, you could order odd-numbered rows in magenta and even rows in white. The eye tends to mix colors when they are in alternating one-dot-wide lines, creating the appearance of a pastel purple.

Normally, the screen creation program places a big, blinking square on the display to serve as a giant cursor, initially in the upper left corner. Characters are typed by pressing keys that generate the appropriate ASCII code; so if you had assigned your graphic capital *B* to the same code used for that letter in standard text, then pressing [SHIFT] and [B] together would bring the graphic *B* to the screen. As with normal text typing, the cursor moves right to the next position after a letter is entered, and it will wrap around to the beginning of the next "line" if there isn't enough room for a character on the present line. Both character width and line height are determined by the grid size used in creating the character font.

Since a typeface file can include two variations on a font, such as roman and italic, an ALT key sequence allows switching back and forth between them; thus either a roman or italic *B* could be produced with the B key, depending on which half of the font file was currently selected. If you press a key for which no character has been defined, the program beeps the PC speaker and waits for another choice. Another ALT key sequence allows intermixing of standard-size text with the graphics characters, so the two sizes can be used together in headline and text fashion. In an enhancement of what is usually possible in PC medium-resolution graphics, GCG allows you to select the color of standard-size text. "You have to know the right place to POKE," says Jennings.

Unlike the standard PC display characters, those designed with the GCG can have varying widths and proportional spacing in the same manner as print typesetting—*m* very wide, *a* average, *i* rather

narrow, and so forth. The screen creation program will take account of this in placing characters along a line, so a line full of the letter *m* might accommodate fewer characters than a line full of the letter *i*. Jennings is especially proud that the backspace key is fully functional on lines of varying-width characters. "How often do you see proportionally spaced characters on a computer screen let alone a destructive backspace that works properly with them?" he asks.

A Keystroke Recorder

To capture screen designs for later use, an ALT sequence turns on a "keystroke recorder" that makes a disk file of all subsequent keys pressed. This facility allows

for a limited range of text animation, since the recorded keystrokes can include control codes for both a timed pause and the clearing of the screen. Another ALT sequence commands playback to the screen of a recorded keystroke set, and the playback commands and display file name can be included when recording, so that one display file can call for the presentation of another. Chains and continuously repeated sequences are possible in this way.

Jennings says characters larger than the 24 by 24 matrix he provides are possible by using the scaling function available with the DRAW statement. Doubling the character sizes this way would be easy, he suggests. Anything much larger wouldn't allow fitting more than a few letters on the screen. He acknowledges that this method of sizing is a compromise because good type design would call for subtle variations as type changes in size. Comparing his work to that of Stanford University's Donald Knuth, a leading figure in computerized design of typefaces, Jennings says, "Knuth is a perfectionist; I'm not in the same class."

Plenty of time went into the programming of the GCG and the creation of the Garamond font. But Jennings says the

work saved time overall, since he is now able to create text screens for demonstration programs very rapidly. "A good tool goes a long way," he adds.

Two Other Tools

Jennings has also developed two other tools he uses in developing demonstration programs. One is for designing and editing informational screens using the text instead of graphics display mode. This works with the monochrome as well as the color/graphics adapter and allows positioning of text anywhere on the screen. It also allows for changing text attributes such as boldface, underline, blinking, and reverse with keystroke control, and creation of borders using the line-drawing characters provided in the text mode. Once a screen is designed with this tool, it generates a file of BASIC code on disk with all the proper LOCATE, COLOR, and PRINT statements that will reproduce the screen when run. Code produced this way can then be merged as modules into a larger program incorporating a whole sequence of screens.

The third software tool is for creating picture-oriented graphics displays including animation and sound. One of these is an animated sequence of butterflies and flowers that has become a staple in IBM showrooms. Jennings says he was able to develop it over one weekend. The animation works, he explains, by having a portion of the program watch the PC's internal real-time clock, and post messages to the screen management part of the program, informing it when the next step in the animated sequence is due for display. **/PC**

Hal Jennings' demonstration-development tools have obvious potential as commercial software products beyond their internal IBM applications, though it appears no attempt has been made to polish them for consumer sale or develop appropriate user manuals. Following a long-standing IBM policy, neither Jennings nor the company will comment on whether versions of the GCG and other tools might be developed for sale, though IBM spokesperson Jeanette Maher did point out that the company does have procedures set up whereby employee-developed programs can be submitted for possible inclusion in its software line.

The Pachyderm's Choreographer

What does an IBM wiz do away from the job?

Hal Jennings comes across as the kind of guy IBM makes a special effort to bring into its fold. He is earnest, dedicated, and friendly, but when he gets to talking about the work he does, the enthusiasm bubbles right through his otherwise reserved manner.

Jennings is 31, married, and has a 3-year-old daughter. He says he has been fascinated by computers since he was a senior in high school, and the fascination carried him through both a bachelors and masters degree in computer science.

One aid to Jennings' productivity is the IBM PC he bought to use at home—a 64K system with two disk drives, the IBM matrix printer, and both monochrome and color/graphics displays. The computer is kept in the family room and has already begun to interest the youngest Jennings.

Jennings says he likes to program, and that's why he got involved in creating demonstration software, even though he asserts, "Nothing in my job description says I had to do this." His next programming project will be family-related instead of job-related, however. His wife is secretary for her bowling league, and he plans to whip up a bowling league scoring system for her. Jennings seems determined to bring his whole family over to the computer corner of the family room. —*J.E.*