The Software Toolworks
Walt Bilofsky, Prop.

14478 GLORIETTA DRIVE
SHERMAN OAKS, CALIFORNIA 91423

TELEPHONE
(213) 986-4885

PIE Full Screen Text Editor
H8/H89/Z89 Version 1.5b
Thomas Crosley
Walt Bilofsky
January 14, 1981

1. INTRODUCTION

PIE is a two dimensional cursor based screen editor designed specifically for use with CRT systems. PIE is human engineered to remove the conceptual barriers that ordinary, line oriented editors place between the user and the file being edited. In PIE, the terminal screen acts as a "window" displaying a portion of the file being edited. Instead of hard-to-remember commands, PIE uses function keys, performing simple operations which are immediately reflected in the file as displayed on the screen. What you see on the screen is what you get in the file.

The screen window may be positioned to display any part of the file. Cursor motion buttons allow positioning the cursor anywhere on the screen, and changes to the file may be typed right on the screen at the cursor position. Other features include:

- Character and line insert and delete anywhere on screen.
- String search forward and backward.
- Macro facility for search/replace and other functions.
- Move and copy single and multiple lines.
- Insert mode for inserting text into a line.
- Scrolling of text in the window.
- Append and clear to end of line
- Continuous typing mode ("hot zone") for rapid text entry.

PIE generates standard text files, with tab characters used wherever possible to conserve disk space. (This feature can be suppressed; see Section 11.)

In order to perform word processing, PIE may be used to prepare files for input to the TEXT formatting program (available from The Software Toolworks). PIE is also handy for entering and revising program and data files.

PIE runs on the H8, H89 and Z89 computers. It is available in two versions, for the HDOS operating system or for CP/M. It is necessary to have the correct version for the operating system you are using. PIE requires at least 32K of memory to run. If more memory is available, PIE will be able to edit larger files. The amount of memory available for editing varies with the operating system and the machine configuration, but typically 20K to 24K characters of memory are occupied by PIE and the operating system, with the remainder available to hold the file being edited.

> NOTE: PIE will not run on the H8 computer under CP/M unless the H19 terminal is connected through an H-8-4 serial interface (or a compatible interface using the type 8250 UART at port address 350). This restriction does not apply when using a computer other than the H8, or when using the H8 with HDOS.

2. RUNNING PIE

The PIE 1.5 disk contains two versions of the software: PIE, which runs only on the H89, and PIE8, which runs on the H8 with the H19 terminal. Be sure to use the correct version for your computer.

> Important: When running PIE under HDOS, the system disk in drive SY0: should not have a write protect tab. If the system disk is write protected, PIE may not run correctly, particularly when used to edit large files.

PIE is run by typing the command PIE, followed by the name of the file to be edited, followed by a RETURN. Thus, to edit a file named DOCUMENT.TXT, type the command

```
PIE DOCUMENT.TXT
```

If no file name is provided, PIE will request one. PIE will read the file in, and scroll the first 24 lines onto the screen. If the file does not exist, PIE will create it.

Editing is done by positioning the cursor anywhere on the screen using the cursor motion keys, and typing text just the way it is to appear in the file. Function keys f1 through f5 are used to move the file so that different portions of it appear on the screen. The following sections explain these operations in more

detail, and describe the editing commands. Section 12 provides a handy chart showing the location of all the command keys.

In the lower right hand corner of the screen, PIE displays the line number of the file on which the cursor is positioned, and the number of free bytes of memory remaining. When this number drops below 2000, PIE will highlight it and give a warning message. If this occurs, EXIT IMMEDIATELY; DO NOT CONTINUE EDITING. See Section 11 for more information about free space.

There are several ways to end a PIE editing session. The simplest is to type control-E (hold down CTRL and type "e"). This will write the file, with the changes made during the editing session, back onto the disk. The other options for ending the session are described in Section 9 below.

3. ENTERING TEXT

Text is typed into the file by positioning the cursor at the point where the text is to be entered, and typing the text. How to position the cursor is explained below.

Typing past column 72 causes the bell to sound. Attempting to type past column 80 causes the bell to sound, and the typed character is ignored.

The cursor motion buttons (the four buttons with the arrows) can be used to position the cursor at any position on the screen. To change the file at the cursor position, simply type the change onto the screen. Whatever is typed on the screen will be inserted in the file, just as it appears on the screen.

Other keys that can be used to position the cursor are:

HOME — Move the cursor to the upper left hand corner of the screen.

HOME (shifted): Move the cursor to the lower left hand corner of the screen.

TAB — Move the cursor right to the next tab stop (multiples of 8 columns). (Note that this positions the cursor, but does not necessarily insert a tab in the file. PIE inserts tabs automatically whenever they will save room in the file.)

ESC — Move the cursor left to the next tab stop.

-> (shifted): Move the cursor, alternately, to the beginning of the current line and to the end of the text on the current line.

To move the cursor a long distance, use the TAB and ESC functions, or the cursor motion buttons with the REPEAT key.

Two keys have a dual function: they move the cursor and also perform another action.

BACKSPACE Moves the cursor left one position, and erases the character at that position.

RETURN Moves the cursor to the left margin, and down one line. If the cursor was on the bottom line of the screen, the file scrolls up one line in the window. (This is especially useful when typing text at the end of a file, since the window keeps repositioning itself as RETURN is typed.)

4. INSERT MODE

Normally, changes are simply typed over any existing text on the screen, and replace the existing text. In Insert Mode, a typed letter is inserted at the cursor position, and a space is opened up for it by shoving all the characters on the line, starting at the cursor, one position to the right. In Insert Mode, the BACK SPACE key not only deletes the character to the left of the cursor, but also pulls the rest of the line, starting at the cursor, one position to the left to close up the space. (It sounds more complicated than it is; the easiest way to visualize it is to try it and see what happens on the screen.)

Insert Mode is turned on and off by the IC function key. When Insert Mode is on, a special message appears at the bottom of the screen.

If an attempt is made to insert a character in a full line (80 characters), the character is ignored, the bell sounds, and the message "Line full" appears at the bottom of the screen. In this situation, the DIVIDE function (ctrl-D; see Section 8) may be used to split up the full line into two lines, so that typing may continue.

5. ARGUMENTS TO COMMANDS

Some of the function key commands may have their range specified by an argument, which is specified by typing the ENTER key, a numeric or string argument, and the function key. For example, the DL key usually deletes one line, but the sequence ENTER 3 DL deletes three lines. Arguments may be used with many of the commands described below.

6. SCREEN POSITIONING

Several function keys are used to position the screen at any desired point in the file.

f1 +PAGE. Moves the screen 24 lines (one screenful) down in the file (but never past the end of the file).
ENTER n f1 moves n screenfuls down in the file.

f2 +LINE. Moves the screen one line down in the file. Other ways to move the file one line down are (shift) up-arrow, and, if the cursor is on the bottom line of the screen, RETURN or down-arrow.
ENTER n f2 moves n lines down in the file.
ENTER f2 moves the screen down in the file so that the current cursor line becomes the first line on the screen.

f3 GOTO. Moves the screen to the top of the file.
ENTER f3 Moves the screen so that the last line of the file appears on the screen.
ENTER n f3 Moves the screen so that line n of the file is at the top of the screen.

f4 -LINE. Moves the screen one line up in the file (but never past the top of the file). Other ways to move the file one line up are (shift) down-arrow, and, if the cursor is on the top line of the screen, up-arrow.
ENTER n f4 moves n lines up in the file.

f5 -PAGE: Moves the screen 24 lines (one screenful) up in the file (but never past the top of the file).
ENTER n f5 moves n screenfuls up in the file.

ENTER s 0 (the 0 on the numeric pad): +SEARCH. Searches for the next occurrence of the string s in the file following the position of the cursor, and moves the line containing s to the top of the screen. If s is not found, beeps and does not change the screen.

0 (the 0 on the numeric pad): A search forward is made for the last string given to a search command.

ENTER s . (the . on the numeric pad): -SEARCH. Searches for the first occurrence of the string s in the file above the first line on the screen, and moves the line containing s to the top of the screen. If s is not found, beeps and does not change the screen.

. (the . on the numeric pad). A search backward is made for the last string given to a search command.

Lines can be added to the end of the text file by moving to the

last line of text and typing on one of the lines below it. When the file is written out, PIE will ignore any blank lines which follow the last line containing text.

If a command attempts to move far past the last text line (such as moving 100 pages down a short file), PIE will move to a point where the last line is near the top of the screen. It is still possible to move further down, but not faster than one page at a time.

7. EDITING FUNCTIONS

The full power of PIE lies in the special editing functions, which allow inserting, deleting and moving lines of text, and other special capabilities. Many of these functions use a text buffer called the temporary lines buffer. Lines can be placed in this buffer using the DL or PICK functions. The contents of the buffer can then be put back into the file at any location with the PUT function.

The editing functions are:

DC — Delete the character at the current cursor position. The characters to the right of the current cursor position are moved left to fill the space.

IL — IL: Insert a blank line at the current cursor line, moving the lines below the cursor down.
ENTER n IL inserts n blank lines at the current cursor line.

DL — Delete the current cursor line, placing it in the temporary lines buffer. The previous contents of the buffer, if any, are lost.
ENTER n DL deletes n lines (24 maximum), beginning with the current cursor line, and place them in the temporary lines buffer. The previous contents of the buffer, if any, are lost.

DL — (shifted): Delete the current cursor line, but do not change the contents of the temporary lines buffer. The contents of the current line are lost.
ENTER n DL (shifted) deletes n lines (255 maximum) beginning with the current cursor line, but do not change the contents of the temporary line buffer. The contents of the deleted lines are lost.

PUT — (white-square key): Insert the line(s) in the temporary line buffer into the file at the current cursor line. Move the current line and those below it down to make room for the inserted lines. The contents of the temporary line buffer are not changed, and may be inserted again anywhere in the

file.

PICK (red-square key): Place the current cursor line into the temporary line buffer. The line is not deleted or altered.
ENTER n PICK places n lines (24 maximum), beginning with the current cursor line, into the temporary line buffer. The lines are not deleted or altered.

ERASE Erase all characters, beginning with the current cursor position, up to the end of the line.

Ctrl-K QUOTE. Used to insert control characters in the file. The next character typed is entered on the screen as a control character. If it is a text character, the corresponding control character is entered instead. Control characters display on the screen in inverse video, but are otherwise treated as printing characters for editing purposes. Certain control characters, such as NUL (ctrl-@), tab, return and line feed may not be entered in this way.

8. MOVING TEXT BETWEEN LINES

PIE is a text editor, and not a word processor. It is not intended to prepare text in a neat format for output to a printer. Thus, there are not many commands for moving around pieces of text lines.

However, PIE is a convenient way to prepare text for input to a text formatter (such as the TEXT program available from The Software Toolworks). In order to make PIE's share of this task easier, two features are included to help in entering and editing text, assuming the formatting is left to another program.

Ctrl-D DIVIDE. Divides the current line at the cursor position. The characters at and to the right of the cursor are moved to the beginning of a new line inserted below the current line

Ctrl-W WRAP. Turns word wrap mode on, and sets a right margin at the current cursor column (or column 76 if the cursor is in column 1). When word wrap mode is on, typing past the right margin will cause a new line to be inserted, and the word(s) extending into the right margin will be moved down to the beginning of that line. The effect is to allow you to type words continuously, without ever worrying about hitting RETURN at the end of the line.

Pressing WRAP (ctrl-W) again turns word wrap mode off. Word wrap mode is normally off when PIE starts up, but PIE can be patched to start up with word wrap mode on; see Section 11.

9. EXITING, SAVING, AND CHANGNG DISKS

Ctrl-R REPLACE. (This function does not operate under CP/M.) Ctrl-R allows dismounting and mounting of disks during an editing session. For example, the sequence "ENTER sy1: ctrl-R" will dismount the disk mounted in SY1: (if any), request a new disk, mount it, and continue with the editing session.

Generally, it is not possible to reset SY0:, since the system overlays reside on this disk. However, if the system is running in SET HDOS STAND-ALONE mode, and a RESET SY0: command has been executed at HDOS command level since the last boot, ctrl-R may be used to reset SY0:.

NOTE: If a REPLACE is attempted on a disk which does not exist on your system (SY2:, for example, on a two drive system), current versions of HDOS will hang up. Be careful not to get into this condition; if you do, see Section 11 for what to do in the event of a crash.

Ctrl-E EXIT. Writes the edited file out, and return to HDOS command level.

ENTER s Ctrl-E does not change the original file, but writes the changed text out on the file named s instead, and then returns to HDOS command level.

ENTER Ctrl-E aborts the editing session. PIE will request confirmation, and, if it is received, will return to HDOS command level. No files will be written out. All files will remain as they were at the start of the editing session, and all editing done during the session will be lost.

Ctrl-V SAVE. Writes the edited file out, and then resumes editing at the top of the file.

ENTER s Ctrl-V does not change the original file, but writes the changed text out on the file named s instead. Then editing resumes at the top of the file.

If there is a file system error in writing out the file when an an EXIT or SAVE function is invoked, PIE gives the error message, does not exit, and returns to the editing state just before the EXIT command, without changing anything. Normally, PIE writes the new version out without deleting the old version, so that some version of the file exists on the disk at all times.

If there is not enough room on the disk for both versions, PIE asks if you want to delete the old version first. This can be dangerous if enough text has been added to the file so that the disk can not hold the new version even with the old version deleted. If you delete the old version and then find yourself in this situation, you can either (1) delete enough lines from the version you are editing to make it fit, (2) save it as another file which already exists on the disk, deleting the old version of that file first, (3) use the REPLACE (ctrl-R) function to insert a

disk with free space, or (4) abort the session, losing both versions of the file. To minimize the chance of losing the file, PIE will not exit until you do one of these four things.

10. MACROS: THE DO KEY

The DO key (the key marked with a blue square) is a user-definable macro key which can be easily programmed to perform many functions, including search and replace.

The DO key is used to record a macro, or sequence of function and text keys, as they are typed to perform an editing operation. Then the DO key is used again to repeat the macro operation once or many times.

The DO key commands are:

ENTER DO — (DO is the blue square key.) Begin recording the typed keystrokes. All typed commands and text appear normally on the screen, but in addition the keystrokes are recorded. This recording mode is terminated by striking the DO key again.

DO — Perform the recorded macro.

ENTER n DO — Perform the recorded macro n times. N must be a number.

Ordinarily, macro execution continues until the macro has been performed the specified number of times. However, if any error occurs the macro execution will terminate, and the error message will be displayed. Macro execution may also be halted by typing any key; the macro will stop and the typed key will be executed.

There is a limit on the number of keystrokes that can be recorded in a macro. The exact limit depends on the version of PIE in use, and the mix of text and function keys in the macro. At least 250 text keys, or 83 function keys, may be used. When the limit is exceeded, an error message appears and the macro recording is aborted.

The operations which the DO key may be used to perform are limited chiefly by the imagination of the user. Several illustrations will be given here.

The most common DO operation is the search and replace function, which replaces all occurrences of one text string by another string. For example, to replace all occurrences of "man" by "person", type the following sequence:

```
ENTER DO                Begin recording.
ENTER man +SEARCH       Search to next "man".
per                     Type "per" over "man".
IC                      Enter insert mode.
son                     Insert "son" after "per".
IC                      Leave insert mode
DO                      Terminate recording
ENTER 999 DO            Execute the macro 999 times.
```

If there are fewer than 999 instances of "man" in the file, the search will fail after the last instance has been replaced, and this error will halt the macro execution.

Sometimes you may want to inspect each instance of a string in a file, and decide for each one whether to replace it with another string. This is easily done by programming the DO key to perform the replacement, but not the search. Then the +SEARCH key may be pressed repeatedly to move to each occurrence of the string, and the DO key can be pressed to perform the replacement whenever that is desired.

Another useful operation is to indent a number of lines of text. Suppose the cursor is positioned at the first line of 50 lines of text, and it is desired to indent all 50 lines by placing ten blank spaces at the beginning of each line. This can be done by:

```
IC                      Enter insert mode.
ENTER DO                Begin recording.
"          "            Type 10 blank spaces.
RETURN                  Move to next line.
DO                      Terminate recording
ENTER 49 DO             Insert spaces in next 49 lines.
```

11. HINTS AND KINKS

Error Conditions.

Error conditions are indicated in one of two ways. If PIE is asked to do something it can't do, like search for something that isn't in the file, or if an illegal function key is pressed, PIE will sound the bell. If something special happens, like running out of memory for editing, a warning message will appear on the bottom line of the screen.

Free Space.

In the lower right hand corner of the screen, PIE displays the line number of the file on which the cursor is currently positioned, and the number of free characters remaining in the computer memory. As more text is entered, the number of free characters will decrease.

When this number drops below 2000, PIE will highlight it and

display a warning message. If this happens, you must exit almost immediately. The characters on the screen, which may number as many as 1944, need to be stored in free space in order to exit, so when the free count goes under 2000 PIE is almost completely out of space.

If this happens, it is a good idea to break the file up into smaller files if possible. If you continue editing, PIE may abort and any editing done will be lost.

(Depending on the operating system, PIE may make use of system overlay areas to utilize additional space when needed. This may result in an increase in the free space count, and an increase in the number of disk accesses at the beginning and end of a session when editing a large file.)

Exiting without Confusing Your Terminal

Sometimes the PIE command is given, and then you decide not to edit a file (or can't remember the name of the file to edit). You may abort PIE in this situation by hitting ctrl-C in CP/M versions, or ctrl-Z in HDOS versions. Once a file is displayed on the screen, ctrl-Z should not be used to exit, as this may leave the terminal, keypad, and console driver in strange modes. Instead, the editing session may be aborted by ENTER Ctrl-E (see Section 9).

Losing Tabs and Other Patchables

To conserve disk space, PIE normally uses tabs to represent two or more blanks ending on a tab stop. This feature can be suppressed by a simple patch, which will cause PIE to replace all tabs with multiple spaces on every line which is displayed on the screen during an editing session. The file PATCHES.DOC on the PIE distribution disk lists the patch and location for each current version of PIE.

PIE uses timing to detect the difference between the ESC key and the function keys. If PIE is operated at terminal speeds below 1200 baud, the delay between characters may interfere with proper reading of the function keys. Although operation of PIE at such low speeds is not recommended, it may be accomplished by patching the timing location shown in file PATCHES.DOC. That location currently contains -10 (366 octal or F6 hex). Try replacing it with a smaller number; 350 (E0) should suffice for 300 baud operation.

Word wrap is normally off when PIE starts up. To patch it to be on, enter the column number (in octal) at which to wrap in the byte indicated in PATCHES.DOC.

In HDOS, patching is done using the PATCH program on the HDOS distribution disk. Ctrl-D is used to return to the previous prompt or to exit; otherwise the program is self-explanatory. To

patch under CP/M, use DDT and the SAVE command; see the appropriate CP/M manuals.

In Case of Emergency.

PIE is quite reliable, as programs go, but occasionally something goes wrong and things lock up. There could be a hardware glitch or a power line spike, or you could try to reset a non-existent disk drive. You might even find a bug in PIE.

Well, for whatever reason, suppose you've been typing for three hours without saving and now the keyboard is dead. You make firm but unhelpful resolutions about saving every fifteen minutes (next time), and you're too much of a gentleperson to call me at four A.M. Is there any way at all to save all that typing?

Absolutely maybe. Desperate situations call for desperate measures, so here's one. Grit your teeth and reboot the system. Mount a disk with plenty of free space. If you're on CP/M, execute the command

```
SAVE 255 LASTHOPE.TXT
```

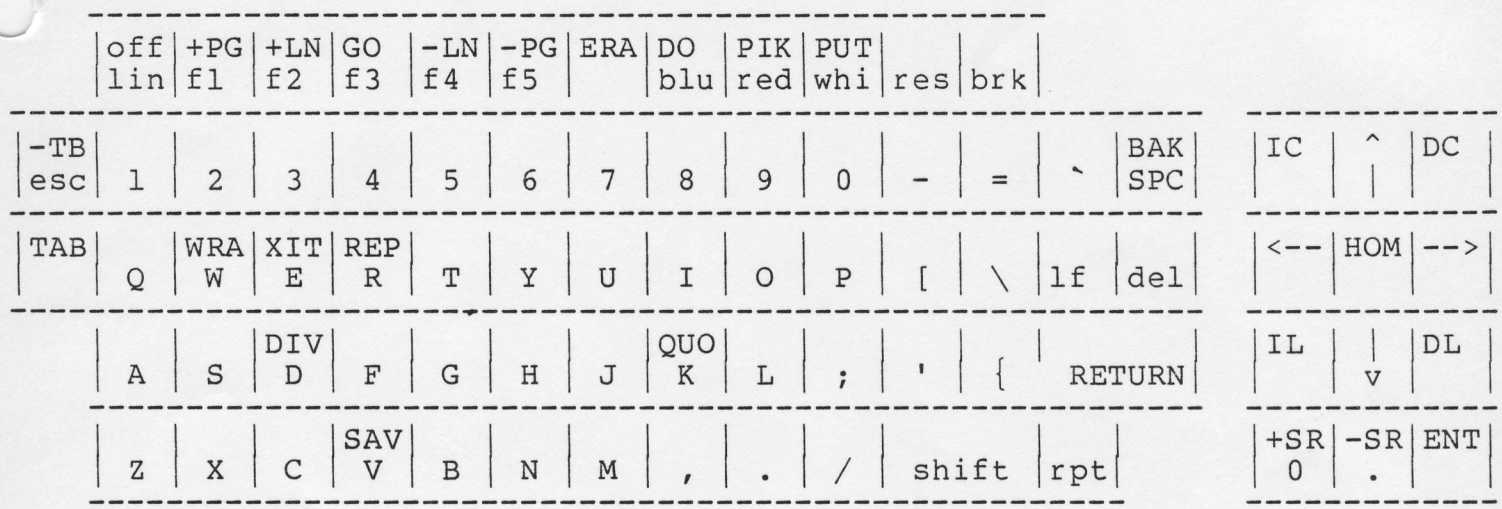
but instead of 255 use four times your memory size, in K, less one. On HDOS, run DBUG, and execute the command

```
DUMP LASTHOPE.TXT,60000-377377
```

but instead of 377377 use the last word address of your memory - 377377 for 56K, 337377 for 48K, 237377 for 32K. Then ctrl-D out.

Now you have a file LASTHOPE.TXT. It's huge, and full of garbage, but somewhere in there is your text. It's probably split up into two chunks, and whatever was on the screen is lost or in a funny format, and it will take some creative editing with ED or EDIT to get the text out (the file is too big for PIE). But most of your typing is in there somewhere. Good luck.

12. PIE FUNCTION KEY LOCATION



FUNCTION KEY LABEL

Because the function key commands are so simple, labeling the keys may provide enough of a reference for normal use of PIE. You may find it helpful to paste labels with the function name on the front of each function key. Alternatively, this label strip may be cut out and placed above the top row of the keyboard to provide a quick reference to most PIE function key commands.

+PAGE	+LINE	GOTO	-LINE	-PAGE		DO	PICK	PUT	Keypad: 0 +SRCH . -SRCH