

1 Welcome to the User Level!

The User Level documentation assumes you have read, understood, and practiced the material in the Learner Level. Along the way, you've gained enough experience in communicating with the Apple that some things have become a habit. Therefore, in the User Level, we won't tell you when to press <CR> in the menu dialogue. We won't say things like "Enter the number for the drive where the data disk is located, then press <CR>"; BEX presents a numbered list of chapters, and prompts you to choose from the list by entering numbers." Instead, we say: "Scan a drive for chapters."

Part 1: More Options and Shorter Prompts

BEX's prompts also get more concise at the User Level. Instead of prompting
Main Menu
Enter Option:

BEX prompts `Main Menu: Press <CR>` at this point, and you see that there are 16 options available, seven more than at the Learner Level. You can now quit at any menu, and there are four more options on the Second Menu and one more on the Page Menu. Another distinctive change from the Learner Level is how BEX prompts for chapters. Instead of `Drive number or chapter name:` BEX prompts `Drive or chapter:` When you type the name of one chapter, BEX *re-prompts* `Drive or chapter:` This means you can individually type the names of several chapters, rather than choosing from a numbered list. There are a number of other shortcuts for chapter selection: details appear in Section 4.

- ***Information Exchange***

Three of the Main Menu's new options help you quickly exchange information between the Apple and the tape-based VersaBraille. Options A - Autoprint from VersaBraille, F - From VersaBraille, and T - To VersaBraille

are discussed in Section 11. The BEX Interface Guide contains step by step explanations for connecting your VersaBraille to the Apple.

On the Second Menu, option I - Input through slot allows downloading data from other computers, including the disk-based VersaBraille, Kurzweil Reading Machine, and many others--details in Section 12. Also new to the Second Menu is option W - Write textfile, which converts BEX chapters into sequential textfiles for use by other Apple programs.

- ***New Editor Features***

In Section 5 you'll learn all about the *clipboard* a fun feature that makes manipulating text very easy. Also discussed is braille keyboard, which lets you use six keys and the spacebar on the Apple's keyboard to enter braille text.

- ***Back Translator From Grade 2 Braille into Print***

Since you can import braille information from other computers and enter braille information in the Editor, you may find use for option B - Back translate from grade 2 on the Main Menu. Back translate allows you to prepare text in a braille environment (including entering format commands) and then process it through BEX for print output.

- ***Greatly Expanded Printing Capability***

Option M - Multi-function print on the Main Menu adds three features to option P - Print, which are explained in Section 6. Section 7 deals with the great variety of format commands that allow you to place information exactly where you want it.

- ***Replace Characters***

Option R - Replace characters was discussed briefly at the Learner Level. In Section 8 we go into the guts of this *very powerful* feature, which lets you automate many, many reformatting tasks, and develop your own keyboard shortcuts.

Part 2: Expanding the Environment

Most computer software outputs the "user dialogue" information only to the computer screen. BEX has four channels for this dialogue. At the Learner Level, we only used two channels: the screen channel and the voice channel. The screen channel mode was limited to what you described in your configuration. The voice channel output was restricted to the SlotBuster and the Echo family of synthesizers.

At the User Level, the voice channel can be connected to either an Echo, SlotBuster or a serial voice device. You can choose from ten different types of screen display in the Editor and when printing to the screen. The User Level also opens the third, braille channel for the information going to the computer screen.

Section 2 provides background information on the choices available in this expanded user environment.

Part 3: How to Advance to the User Level

You tell BEX to unlock all these new features by creating a User Level configuration. Section 3 steps you through the process of defining a User Level configuration, and provides guidelines for answering the many new configuration questions. Nitty-gritty technical details, as always, are available in the BEX Interface Guide.

2 Controlling the User Environment

We use the term *user dialogue* to describe the communication between you and your Apple. Most sighted computer users read the user dialogue on the screen and instruct the Apple through the regular keyboard. When a sighted user wants to examine a particular part of the user dialogue in detail, they may make a hard copy printout for further study.

There are a number of ways to make what appears on the screen accessible to blind and partially sighted people: BEX tries to support as many as possible. This means that you've got a lot of choices, and that, as a blind or partially sighted computer user, you need to learn a little bit about *input* and *output*. The information in this Section provides you with the background you need to answer some User Level configuration questions.

Part 1: BEX's Two Roles

At the Learner Level, we focussed on *BEX as word processor*. At the User Level, we explain more of BEX's word processing features. In addition, we begin to explore BEX's other role: *BEX as I/O system*.

What's *I/O*, and why should I care about it? *I/O* is the abbreviation for *input/output*, and is an important piece of computerese to understand. Most sighted computer users never think about *I/O*: they simply use the regular keyboard for *input* (getting information into the computer), and the screen and printer for *output* (getting information out of the computer). In fact, most software for the Apple only uses the Apple's built-in *I/O* features. But when you use BEX, BEX takes control of all *I/O*, creating a unique user environment. BEX certainly lets you use the regular keyboard and screen, but also lets you use alternative *I/O* channels.

Four output channels

BEX has four *output channels*: screen, voice, braille, and print. The first two channels are the most efficient way to present the user dialogue, because they are fast and transitory. With BEX, you can set the screen channel for five different sizes (from 80-column to 5-column) and for two modes (print letters or braille dots.)

BEX lets you combine voice output with large letters on the screen. Most of the time, using these two channels suffices.

The braille and print channels are useful when you want a permanent record of your dialogue with the computer. They are much less efficient for general communication with the Apple, because of the time required for a print or braille device to make hard copy records. The braille channel sends all the information on the screen to a braille device--you can use a braille embosser or a paperless braille display. The print channel sends all the information on the screen to an inkprint printer. At the User Level, you can configure BEX so that all the user dialogue appears on the braille channel. Output of the user dialogue to the print channel is only available at the Master Level.

Don't confuse the braille and print channels with *printing* to a braille or printer. The braille and print channels let you record the user dialogue: menu prompts and your responses. When you send formatted text to a printer or braille, then you're *printing* to that device.

To summarize, BEX controls all I/O for the Apple. You tell BEX what I/O you want and BEX makes it happen. At the Learner Level, you defined your I/O preferences in your configuration. At the User Level, you can change some of BEX's I/O as you use the program, without having to reconfigure. At the Master Level, you are able to independently control all four channels at any time.

Part 2: What's the Best Access Strategy?

When you can see the 80-column or 40-column screen, you have access to a lot of information at once. You can review several lines of the user dialogue, and you can fit the whole list of menu choices on one screen. When you instruct BEX to make 20, 10, or 5-column display, less information fits on one screen. When you listen to the dialogue through a voice device, you only hear one word at a time. Therefore, BEX has to make accommodations for your need to get the whole picture of the user dialogue.

When you use BEX on the Apple, all input and output go *through* BEX. This means that the speed of the dialogue is controlled by the speed of the slowest

I/O channel that's currently in use. For example, suppose you want to hear the list of options at the Main Menu. You told BEX to send output to the Wide (80-column) screen and the Echo. At the Main Menu prompt, you enter <CR> to get the list of options. BEX sends the information in the list to two channels: the 80-column character generator on your 80-column card, and the TEXTALKER software that controls the Echo circuit card. Speaking the list of options takes longer than displaying them on the screen, so the words show up on the screen at the same time they are spoken by the Echo.

Sometimes you won't want to listen to the whole list. The TEXTALKER software has two ways to let you shut up the Echo. As the Echo is speaking, you can enter control-X; this disconnects the TEXTALKER software, and the Echo shuts up. TEXTALKER is automatically reconnected when the Apple needs more input. (Control-X performs the same function with the SlotBuster.) Control-X doesn't affect BEX's screen channel, so the list is quickly displayed on the screen. When BEX prompts for more input with the Main menu, then the Echo can start speaking. Since the characters are on the screen, you can use screen review at this point to review the information. (More information on Echo screen review appears in Learner Section 10. For details on SlotBuster screen review, see the SlotBuster manual.)

The other way to shut up the Echo is by pressing a valid command. Suppose you want to know which key to press to choose the Grade 2 translator. Press <CR> at the Main Menu and you hear the list of options. As soon as you hear `G - Grade 2 translator` you can press G. The Echo stops talking, and passes the G to BEX, which loads the Grade 2 translator. The next thing you hear is `Grade 2 translator`

When you instruct BEX to use one of the large print screen displays (20, 10 or 5 column) you find yourself in a similar situation. All of the menu options can't fit in one screen, so it takes a while for the list of menu options to scroll by. If you don't care to wait for the entire list, you can enter control-Z to temporarily disconnect the screen channel. Just like control-X, control-Z automatically reconnects the screen channel when the Apple wants more input.

Part 3: The Screen Channel

BEX allows you to choose between five sizes of print on the screen, from tiny to enormous, and also allows you to choose two styles of display: regular letters or braille dots. As we've mentioned, the sizes are defined in *columns* (how many characters fit on one line on the screen). The actual size of the letters depends on the size of your monitor. Most screen options draw letters on the screen--they're called *HI-RES* because they use the Apple's high resolution drawing routines. With any HI-RES screen option, you can control the rate of scrolling with the open-Apple (or command) key and solid-Apple (or option) key.

Two screen options use the Apple's built-in character generator. You can only use screen review software (like TEXTALKER or SCAT) with the 80-column Wide and 40-column Non-HI-RES screen displays, so size is relevant even if you depend only on the voice channel for output of the user dialogue. The Echo still speaks the menus even when you choose a HI-RES screen display. Each size and style is specified with one letter; the complete list of ten letters appears in Section 5, Part 7; Section 6, Part 1, and on the Thick Reference Card.

When you configure with 80- or 40-column screen *without* voice output, BEX switches to HI-RES screen for two options. This switch allows sighted users to see control characters, using the same shapes as in the Editor. This switch to HI-RES happens with option P - Printer control code display at the Starting Menu, and when you type changes directly with Replace characters.

- ***Controlling Screen Channel***

There are four ways to control screen display in BEX; the one you use depends on your level and your whim. For all levels, you specify what size screen display you want in your configuration. (The default values are N for menus and H for the Editor.) At the Learner Level, this is the only method of controlling the screen channel that we document.

At the User Level, you can change screen display in the Editor by entering control-S S followed by one of the ten screen-size letters. For example, you configure with 20-column screen, and then wish to use 40-column braille dots in the Editor, you enter control-S S B. (Details in Section 5.)

At the User Level, you can also specify varying screen sizes when you print to the screen. When BEX prompts `Which printer:` you enter `S <CR>` to print to the screen in the size you defined in your configuration. When you follow the `S` with one of the ten screen-size letters, BEX prints to that size screen. For example, you have configured with 40-column screen. You want to print text to the 20-column screen. Enter `SL <CR>` at the `Which printer:` prompt. (Details in Section 6.) At the Master Level, you learn how to change the screen display at the menus, too.

- ***Deciding Which Screen to Use***

Each screen mode has its plusses and minuses. In the Editor, `W` and `N` mode show all control characters as a checkerboard pattern. For the fully sighted user, you'll get more information from a HI-RES mode, because BEX draws a unique pattern for each control character. (`A <CR>`, for example, shows as the letters `C` and `R` jammed together; a control-`T` shows as an underlined `T`.) When you use 20-column or larger screen and you are a speedy typist, most of the text and commands you type are stored temporarily in BEX's keyboard buffer. This can be a drag if you inadvertently hold down an arrow key for five seconds!

When you can't see the screen at all, `W` and `N` modes combined with the Echo or other output device give just as much information as a HI-RES mode, and are also a very short amount speedier. The HI-RES screen modes require a longer time to display because BEX has to instruct the Apple what every character looks like. `W` or `N` modes are also good for Echo users, because you can use Echo screen review. Of course, for the partially-sighted user, 20-column or larger display at the menus is helpful.

- ***Advancing to the Next Screen of Data***

There are three different contexts for BEX screen display: at menus, in the Editor, and when printing to the screen. During screen display at menus, you have *scrolling*: When there's more information than can appear on one screen, every line on the screen moves up one. The top line disappears, and the new text shows up on the bottom line.

In the Editor, the screen display changes to match your cursor movement. When you advance your cursor two paragraphs, the screen display jumps so that your cursor appears in the center lines of the screen.

Printing to the screen is like printing to a printer that has pause on form feed. Each screen's worth of data is one *output page*. When the screen is full, BEX makes a low boop and pauses. Press <space> for the next page.

Printing to a braille previewer or Review class printer is a special case, discussed in Learner Level Section 5 and again in User Level Section 6, Part 3 and Section 7, Part 9.

• ***Controlling Large Print Screen Display***

For 80 and 40-column screen, BEX uses the Apple's built-in scrolling abilities. As a new line of text appears, every line on the screen moves up one, and the top line *scrolls* off the top into oblivion. For all other screen sizes, BEX provides a different scrolling system, which is available at all menus. You can control the speed of scrolling; you can momentarily freeze the screen display, and you can slow the screen display down to a crawl. Here's how:

- Changing scrolling rate - Solid-Apple-number (or Option-number on the IIgs) controls the speed of scrolling; 1 is slowest and 9 is fastest. When you configure with large print screen, BEX's initial speed is 9. You can only change the scrolling rate *during* scrolling; you *cannot* change the rate when BEX is waiting for input, for example, at the Main Menu : prompt. To initiate scrolling, press <CR> at any Menu prompt. Now, as the characters scroll by, depress the solid-Apple (or Option) key. Press and release a digit between 1 and 9, then release the solid-Apple (or Option) key. The new rate lasts until you change it or turn off the power.
- Freeze large print - Whenever you depress just the open-Apple key (or command key on the IIgs), large print scrolling momentarily freezes.
- Crawl large print - Whenever you depress just the solid-Apple (or Option) key, large print scrolling slows to a crawl.

Temporarily suspend large print output

When you want to temporarily suspend output to the large print screen, enter control-Z. Control-Z parallels the function of control-X for the Echo and

SlotBuster. Control-Z stops output to the screen until the next time the Apple is waiting for input.

Here's when control-Z could be handy. You want to see all the files on a disk, so you press D for a Disk catalog at a BEX menu, then press <space> after the list of chapters. The file you're interested in shows up in the first ten files of the DOS catalog. You don't need to see the rest of the catalog. Press control-Z and BEX suppresses large print display for the rest of the catalog.

You may run into a problem here, however. DOS 3.3 has a built-in function that pauses the catalog display every 24 lines. When your disk has more than 24 files, you must press any key to see the rest of the catalog. Suppose your disk has 50 files on it. After you see the 14th file, you don't want to look at the rest of the catalog. When you press control-Z, BEX suppresses the remaining ten files in the current catalog screen; all you see is the cursor on the bottom screen line. Until you press any key, DOS 3.3 waits.

In this situation, press control-Z twice. The first control-Z functions as the "any key" DOS needs to continue the catalog display. BEX gets the second control-Z and suppresses the catalog display for the 25th through 48th files. Finally, press control-Z twice to skip over the last two files. Because BEX is suppressing large print output, you do not see the `Main Menu:` prompt. However, you do hear the low boop that means BEX is waiting for a menu choice.

Part 4: Three BEX Functions that Suppress Large Print Screen

The 20, 10, and 5-column HI-RES screen displays use a part of the Apple's memory called, logically enough, the *HI-RES memory*. Unfortunately, a number of other parts of BEX also require accommodation in the HI-RES memory. When you have 20, 10 or 5 column screen display at menus, and you choose a menu option that uses HI-RES memory, you temporarily change from HI-RES to N screen. This doesn't mean anything is wrong with your BEX disk. It's an unavoidable result of how the Apple's memory is arranged.

The following items harmlessly conflict with HI-RES screen at the Main menu. When the option is completed, you return to the HI-RES screen.

- The braille translation tables - These are loaded into memory when you choose options B - Back translate from Grade 2; G - Grade 2 translator; A - Auto print from VersaBraille; or H - Heading test.
- Transformation chapters from disk - Option R - Replace characters uses the HI-RES memory to store the text of a transformation chapter that's read from disk. When you specify a chapter name at the Transformation chapter name: prompt, the HI-RES screen display disappears. The last Continue? Y prompt is shown in 40-column screen; you must press <CR> to begin replacing. There's no conflict when you directly key in changes.
- Large print dot-matrix printer - When you configure a large print printer with BEX, the program loads the 14 or 18 point font into HI-RES memory. After you enter the number of a large print printer at the Which printer: prompt, the Apple screen shows random lines. Once the printout is done, the menu prompt returns in large print.

One Disk Drive: BEX needs to load the large print font from the program disk every time it prints. You must copy the font files to your data disk. More details in Appendix 3.

Part 5: The Voice Channel

At the Learner Level, BEX limits output on the voice channel to a member of the Echo family or the SlotBuster. These devices are *integral* voice devices. When you combine the circuit card with the synthesizers' TEXTALKER or SCAT software, speech is integrated into all the Apple's operations.

At the User Level, you can connect a *serial voice device* instead of an integral device to the voice channel. A serial voice device is like a *talking printer*. You send it some text, and it talks. BEX has many ways to send text to a serial voice device. When you press <CR> at any menu prompt, the list of options is sent out the voice channel. When you enter control-T in the Editor, that sentence is sent out the voice channel. When you add +V to a printer destination, the text is sent to both the printer and to the voice channel.

No serial voice device has screen review capabilities. BEX's Review class printer depends on TEXTALKER'S and SCAT'S screen review features-- details in Learner Level Section 10 and Appendix 2.

- ***Controlling the Voice Channel at Menus***

BEX lets you send commands to your serial voice device in a similar manner to sending Echo or SlotBuster. At any BEX menu, control-O begins a command that's sent to the voice channel. After you enter control-O, your keystrokes are all directed to the voice channel, so you can enter any commands for your serial voice device. Finish the command with <CR>; this signals BEX that you have ended the command. For example, to set the volume on the Echo GP, enter control-O control-E 12 V <CR>.

- ***Controlling the Voice Channel in the Editor***

As with Echo and SlotBuster commands, you can send commands to the voice channel from inside the Editor, too. A plain control-O is the Output text Editor command, so you must precede it with control-S, and finish the command sequence with <CR>.

In Section 3, we explain how you configure a standard series of commands, or an *automatic set-up sequence*, for voice, braille, and print devices. Once you give this information to BEX, it automatically sends the sequence every time it addresses the device. When you want a voice device to use a particular set of parameters that's different from the default, you establish an automatic set-up sequence with the commands that create the mode.

Part 6: The Braille Channel

In *addition* to an integral or serial voice device, BEX allows you to have a braille device display all the information in the user dialogue. The braille device can be a braille computer terminal, like either VersaBraille or a

Cranmer Brailer; or simply a braille printer, such as the MBOSS-1 or Thiel. The output on the braille channel is *screen braille*: every inkprint character is represented by one braille cell. Since there are 96 *printa* inkprint characters and only 64 braille cells, some braille cells do double duty, representing more than one inkprint character. Details on screen braille appear in Appendix 1.

While you can send all the user dialogue to the braille channel, it may not be the most efficient way to do word processing. When you are using a braille terminal, then you have to keep track of two cursors: the terminal's own, and the Apple's. When you are using a braille printer, then you have to wait for the braille to be embossed. In either case, voice output is faster.

The tape-based VersaBraille is a special case. Many people find the best way to exploit the power of the VersaBraille and Apple is to prepare material on the VersaBraille, and then send the data to the Apple for back-translation and printing. Similarly, print oriented material is prepared in the Apple environment, then translated and transferred to the VersaBraille for close review. Three Main Menu options enable very fast transfer of information between the Apple and the tape-based VersaBraille--details in Section 11 and the Interface Guide.

- ***Sending Commands to the Braille Channel***

The control-O commands described in Part 5 actually direct keystrokes to both the voice and braille channels. When you don't have a device on the voice channel, then the control-O commands go just to the braille channel. At BEX menus, enter control-O, the braille device commands, and finish with <CR>. In the Editor, enter control-S control-O, the braille device commands, and finish with <CR>.

As mentioned in Part 5, you can establish an *automatic set-up sequence* for your braille device. BEX always sends this sequence to the braille device. For example, the three characters <control-N> ; <control-O> ensures that the VersaBraille II display shows uppercase with vibrating pins. More information on this feature is provided in Section 3.

Part 7: Alternative Input Devices

There are six ways you can put text into a BEX chapter. At the Learner Level, we explored two methods. Most of the attention focussed on typing text on the Apple keyboard. Learner Level Section 12 demonstrated option R - Read textfile to chapter on the Second Menu, which lets you copy the information from a DOS 3.3 or ProDOS textfile into a BEX chapter.

At the User Level, you have access to the four other ways to get text into BEX. Section 11 explores the VersaBrailles in detail; BEX has many features designed to make it easy to input text through a VersaBraille. Section 12 discusses option I - Input through slot on the Second Menu. BEX contains a limited *terminal* function; you tell BEX to accept information that's being sent from another computer. That other computer could be a special-purpose device like PortaBraille, Keynote, or SmallTalk, an optical scanner like the Kurzweil Reading Machine, or another general-market computer, like the IBM-PC or Macintosh.

In Section 5, Part 7, we discuss BEX's *braille keyboard mode*. BEX can change how the keys on the Apple keyboard are interpreted; you can use six keys and the spacebar like a Perkins keyboard to perform braille data entry.

<i>Apple</i>	Braille keyboard is not available with a true Apple IIgs keyboard.
<i>IIgs:</i>	Braille keyboard <i>is</i> always available through the Apple IIe keyboard, even when you obtain a "performance upgrade" to change the IIe to a IIgs.

Finally, at the User Level you can tell BEX to ignore the Apple keyboard and accept input through a different device. Section 3 discusses the configuration questions relating to remote keyboards. For example, you could configure a VersaBraille's keyboard as a remote keyboard. You then enter all BEX commands on the VersaBraille's six-key keyboard instead of the Apple's.

3 Configuring at the User Level

Establishing a configuration at the Learner Level involved answering three basic classes of questions: using a voice device; size for the screen display; and details about your printers. At the User Level, you are asked many more questions. You can connect braille devices to the braille channel. (See Section

2 for a discussion of BEX's four channels.) In Part 3, You learn how to define an *automatic set-up sequence* to send to any printer, to the voice or braille channels, to the VersaBraille, to a paperless brailler, and to a remote serial device to input text through slot, such as an optical scanner, or other serial computers. As at the Learner Level, you can press <CR> at any configuration question to get a help message.

You don't need to describe in one configuration every device you might use. When you work in both print and braille, you might want to establish two different configurations. You can make three sets of parameters for the same inkprint printer, plus a Review class printer. The other configuration can contain an inkprint printer, a brailler, and a braille previewer with voice. The only limit to the number of configurations is the space available on disk.

Part 1: Getting Access to the Configuration Dialogue

To define a User Level configuration, you enter the asterisk character at the `Enter configuration:` prompt. When BEX loads an existing configuration from disk, it's loading a list of your *input/output* preferences. When you are establishing a new configuration, BEX does not yet know about your I/O preferences. When you have an Echo or SlotBuster voice device, BEX can speak the configuration dialogue. But when you don't have an integral voice device and can't see the screen, you must use one of two "bootstrap" codes--that is, codes to get your computer and voice device going--to get output of the configuration questions in an accessible medium:

- 1. To get output to a serial voice or braille device interfaced through an Apple port or slot, enter the slot number of the device after the User Level asterisk at the `Enter configuration:` prompt. For example, to establish a User Level configuration and send output to an Echo GP in slot 2 during the configuration process, enter `*2<CR>` at the `Enter configuration:` prompt.
- 2. To get both input and output through a braille terminal, add the letter *V*, then the slot number. For example, to establish a User Level configuration and send input and output through a Cranmer Brailler in

slot 5 during the configuration process, enter *V5<CR> at the Enter configuration: prompt.

Part 2: Overview of New Questions

Do you have a remote keyboard?

When you want to use a different device for input to the Apple, you answer Y to this question. The keyboard may be a stand-alone peripheral, or it can be part of a computer terminal, like the VersaBraille or Cranmer Brailier. You may need a specific serial interface card for this option; see the Interface Guide for details. When you use a Remote keyboard, it interferes with other capabilities, such as the braille keyboard mode, so answer Y to this question only when you need it.

The voice channel

When BEX recognizes an Echo or SlotBuster in your Apple, it asks if you want Echo (or SlotBuster) speech. When you do not have an Echo or SlotBuster, or when you answer N, BEX asks if you have a voice device for all the material going to the screen. When you answer Y, then you have the opportunity to enter an automatic set-up sequence for the device (see Part 3). BEX can only output to one device on the voice channel. Therefore, you must choose between an integral and a serial voice device. You can use the Echo for voice channel and still define a serial voice device as one of your printers.

The braille channel

In addition to output on the voice channel, you can also have the same information sent to a braille device. Only answer Y when you want *all* the computer dialogue sent to a serial braille device. Don't answer Y when you want to emboss braille documents; configure your embosser as one of your four printers. Again, you can enter an automatic set-up sequence for the interface card or braille device.

Do you have a tape-based VersaBraille?

The answers here only affect your use of options A - Auto print from VersaBraille, T - To VersaBraille, and F - From VersaBraille on the Main Menu. These options transfer files between the VersaBraille and the Apple faster than Printing and Input through slot. (See Section 11, and Section 9 of the Interface Guide.)

Do you have a remote serial device to input text through slot?

The answers here only affect your use of option I - Input through slot on the Second Menu. You give the slot number of a device that sends serial data to BEX--see Section 12 for further details.

The screen display

You have the same options here that you did at the Learner Level. You have more ways to change screen display while you are using the program, so the answers here only supply default values. When you have a BDP or a DP-10, ask for the 40-column, non-HI-RES screen mode, or for the 80-column screen display. As we stated in Learner Level Section 3, Part 4, HI-RES screen display in the Editor is designed to make it easier for users with no vision impairment to read the screen. HI-RES screen display refreshes slightly slower, so when you are depending on voice alone, answer N to this question.

The printer section

In addition to choosing among the eight printer classes, you can specify an automatic set-up sequence for each one.

In Learner Level Section 3, Part 5 we described Generic printers, Brailers, Large print, and Review class printers. Here we describe three more classes of printers: Specific printers, Voice devices, and Paperless brailers.

BEX supports many specific inkprint printers. At the Master Level, you learn how to use specific commands for your printer, to get different types of printing fonts.

You can print to a slot-based voice device as a class V - Voice device printer; simply press V when prompted for the printer class. Class V printers are a

special class of printers: because they are *voice only*, they need no carriage width or form length. When you configure your voice device as a class V printer, you are not asked for carriage width or form length. You may establish an automatic set-up sequence if you wish. However, BEX has a hard time coping when there's no carriage width or form length, so values are automatically provided. These values do not affect your voice device. When you press ? <CR> at the `Which printer:` prompt, your class V printer is listed as a printer 40 by 0. Don't configure your Echo as a class V printer: BEX won't let you. The class V printer configuration is meant for high-quality voice devices hooked up to a slot, such as a DECtalk.

You use a class P - Paperless braille to send unformatted text to any computer device. Printing with a Paperless braille sends exactly what's in your chapter except for several control characters, which are stripped out. These control characters are <control-T>s, <ASCII 30>s, <ASCII 31>s, and <control-S>s. What these control characters do is discussed in Master Level Section 5. Paperless brailles are discussed fully in Section 6, Part 4.

Disk drives

Just like at the Learner Level, you can only answer 1 or 2. At the Master Level, we tell you how to configure up to eight drives.

Naming your configuration

As we said in Learner Level Section 3, configuration names follow two simple rules: the first character must be a letter, and the name cannot exceed ten characters in length. Choose a name you can easily remember. When the name you choose already exists on the Boot side of BEX, then the new information overwrites the old.

After you type the name, finish with <CR>. If you want, you can establish a *default* configuration by entering <CR> alone as your configuration name. You then press <CR> for that configuration whenever you boot.

Whenever you forget the name of your configuration, press ? <CR> at the `Enter configuration:` prompt. The name of your default configuration (the one with <CR> as its name) is listed as DEFAULT in the list of configurations. Enter it as <CR>, as you specified in your configuration.

Part 3: Automatic Set-up Sequences

You can define automatic set-up sequences for voice and braille devices, any of your printer channels, for the VersaBraille, and for the interface card used for a remote serial device to input text through slot. An *automatic set-up sequence* is a series of characters that BEX sends out a particular slot whenever that device is addressed. This series of characters "sets up" the device according to your instructions. For example, you can use an automatic set-up sequence to set a left margin on printer, so that you don't need to enter `$$ml#` in every chapter you print.

Whether the commands are for the interface card or the device attached to the interface card is up to you. For the voice channel, the sequence is sent every time you switch menus, and every time you press control-Reset and type RUN at the BASIC prompt. For printers, BEX sends the sequence every time you specify the printer number at the `Which printer:` prompt.

- ***How To Type Automatic Set-up Sequences***

You enter an automatic set-up sequence when prompted while configuring. An automatic set-up sequence can contain any character you can type on the regular Apple keyboard. Every key you press when defining the set-up sequence is entered into the sequence: you can't use the left and right arrows to correct mistakes. You may wish to set Echo punctuation to pronounce All so you can hear the keys as you press them. Control characters do not appear on the screen. Press the delete key to signal the end of the set-up sequence.

At the beginning of configuring, you depressed your Caps Lock key. When you need to enter lowercase characters in the set-up sequence, release the Caps Lock key. Remember to depress it again after you finish the sequence.

When you are finished configuring, you can see your automatic set-up sequences with option V - View a configuration on the Starting Menu. When you use this option, BEX encloses any control character or space in angle brackets.

BEX uses an automatic set-up sequence to set up a large print printers. However, you can still establish an automatic set-up sequence when you configure a large print printer. When you use option V - View a configuration, you see the automatic set-up sequence that BEX sends as well as the one you entered.

- ***What You Type***

You decide which characters to type for your automatic set-up sequence by reading up on the manuals for the voice device, printer, or interface card you're addressing. Details on addressing the Super Serial Card and Apple IIc ports appear in the BEX Interface Guide. You *cannot* enter BEX format commands as part of an automatic set-up sequence. Format commands are always instructions to BEX's formatter, not directly to the device or interface card.

- ***When To Use Automatic Set-up Sequences***

You can enter some very useful information into an automatic set-up sequence. For example, with the Ohtsuki printer, you can set a specific output mode using an automatic set-up sequence. The Ohtsuki's default is to accept grade 2 text input, and output grade 2 braille and back-translated print on alternate lines. Suppose you want either braille output or print output. You then configure the Ohtsuki printer twice; once with an automatic set-up sequence of <ESC> B for braille only output, and once with an automatic set-up sequence of <ESC> P for print only output.

When your printer always needs a left margin, you can define it with an automatic set-up sequence. For help with establishing an automatic set-up sequence for your printer, see Part 4.

Another example deals with the Super Serial Card. BEX's standard parameters set the Super Serial Card at 9600 baud. Suppose you want to use one Super Serial Card to interface a VersaBraille at 9600 baud, and a Macintosh at 19,200 baud. When you configure, you include an automatic set-up sequence for both interfaces: For the Macintosh, you enter control-A then type 15 B <CR> to change the baud rate to 19,200. For the

VersaBraille, you enter control-A then type R <CR> to reset the Super Serial Card to its default switch settings.

- ***Alternatives to Automatic Set-up Sequences***

Since a BEX chapter can contain any of the 128 ASCII characters, you can also write a BEX chapter with the same information as you might put in a set-up sequence. Then, to effect the instructions, you print this *set-up chapter* before any chapters you print to the device. The big advantage to a set-up sequence is that BEX always remembers to send it. When you are experimenting with various commands, do your trial and error research with BEX chapters. When you know what works, enter it as an automatic set-up sequence in your configuration.

Part 4: Configuring Printers

At the Learner Level, we focused on three printer classes: G - Generic, B - Braille, and L - Large Print. In that discussion, we covered all the basic issues in configuring a printer. In Part 2, we covered two more printer classes: V - Voice devices, and P - Paperless braille. As always, you can press <CR> at any question to get a help message. If you're wondering what *Auto linefeed* or *Pause on form feed* means, go back to Section 3 of the Learner Level.

The BEX Interface Guide has pages and pages about printers, including the wonders of Dipner Dots and details about the LaserWriter. In the next paragraphs, we present a method to help you establish margins for your printer. See Section 7, Part 10, for examples of Review class printers.

When you configure a large print printer, BEX does some calculations for you. BEX uses a combination of point size (the size of the letters) and extra spacing to calculate carriage width; the answer for linespacing affects the calculation of form length. To get these values, press <CR> at the prompt.

The horizontal and vertical test grids

In Learner Level Section 5, Part 3, we discussed how to use the RP GRID chapter, the LP GRID chapter, the LONG GRID chapter, and the V GRID chapter on your BEXtras disk to establish margins and a carriage width for your printer. What we did not cover was how to use these chapters to determine values to use in an automatic set-up sequence for your printer. In this part we will go through the same sequence again, only with the aim of using automatic set-up sequences.

These four chapters print reference grids using BEX's horizontal and vertical numbering system. Enlisting the assistance of a sighted person, if necessary, you can ascertain the appropriate horizontal and vertical numbers you need for your printer. See Learner Level Section 5, Part 3 for instructions on how to configure your printer for testing.

- ***Printing the Vertical Test Grids***

The chapter named V GRID provides you with a guide to set form length, top-of-form, and top margins. Establish a workable landmark on your printer, as discussed in Learner Level Section 5, Part 2. Then print the V GRID chapter to this printer.

Where line 1 appears in this printout is where BEX prints the first line of every page. When your test sheet shows line 1 too high on the paper, then you have several ways to establish more aesthetically pleasing top and bottom margins.

Establishing a top margin

At the Learner Level, we describe two ways to set top margins: by using \$\$mt# and by pressing the line feed button.

The third way to establish a top margin is to use an automatic set-up sequence to send a top margin command to your printer. Your printer manual should have the exact sequence you need. For example, you can use the printer's internal top margin command, if it has one, for an automatic set-up sequence. The ImageWriter doesn't, but the Diablo 630 does. To set a top margin of three lines, you enter five characters:

<CR> <CR> <CR> <ESC> T

To enter this command as an automatic set-up sequence, you go through the configuration process until you come to the printer section. When you are prompted for an automatic set-up sequence, you type in three returns plus <ESC> T as your sequence:

Establish an automatic set-up sequence for PRINTER ONE? Y <CR>

Type it EXACTLY. Press Delete key to end sequence: <CR> <CR> <CR>
<ESC> T

Using these characters as an automatic set-up sequence insures that the Diablo 630 always has an aesthetic top margin.

As we've stressed, to maintain an accurate top-of-form you must never advance the paper by rolling the platen. When you are done printing a document, turn the printer off-line, and press the formfeed button. The paper advances into position for the next document you print. (The design of some printers make it difficult to tear off the sheet after just one formfeed. In that case, press the formfeed button twice, sacrificing one sheet of paper to the cause of beauty.)

Establishing form length

In addition to establishing the appropriate top margin, the V GRID chapter can provide you with the appropriate value for your form length. Once you've decided on the value for your top margin, find the set-up sequence for that command, and insert it into the V GRID chapter. Turn your printer off and then on to reset it so that the new command will work. Establish your correct top-of-form, then print the V GRID chapter again.

To analyze your form length, first fold the paper in half the long way (so that the top and bottom edges meet). Note which grid line number towards the bottom of the sheet meets line number 1. Use this number as your form length when you configure. When you specify that number as your form length, you have equal top and bottom margins.

- ***Printing the Horizontal Test Grids***

The chapters named RP GRID, LP GRID, and LONG GRID provide a ruler for you to determine left margin and carriage width for your printer. Use the grid chapter that is appropriate for you: The RP GRID chapter prints six rulers; this chapter is designed for regular print printers. The LP GRID chapter prints three rulers; it is designed for large print printers. The first ruler in these two chapters prints 39 characters, the second prints 49 characters, and so on up to 59 characters for LP GRID and 89 characters for RP GRID. The LONG GRID chapter prints one long ruler with 159 characters for printers with condensed print. Use the RP GRID, LP GRID, or LONG GRID chapters once for each different character size or pitch you plan to use.

Setting the left margin

Once you have your sample RP GRID, LP GRID or LONG GRID printouts, analyze them to determine left margin and carriage width. Learner Level Section 5, Part 3 tell what the rulers should look like and how to interpret the rulers.

The left edge of the ruler is BEX's *position zero*. On some printers, this position zero appears at the left edge of the paper, which means that BEX's position zero would make a very ugly printed page. When you are unhappy with the left margin on the test printout, get a *real* ruler. Measure a nice margin of one inch or seven-eighths of an inch and make a pencil mark at this point. Compare the pencil mark with the grid ruler.

Suppose the pencil mark occurs at position number 8; you now know that BEX's left margin of eight positions the printhead at a good place. You can use this value of 8 to set a left margin, using `$$ml8` when you print. Include the `$$ml8` command at the start of every chapter you print. Or you can use a *setup* chapter that contains escape sequences that set margins for your printer.

It's fastest to establish an automatic set-up sequence to set your left margin. Check your printer manual for the exact sequence; different printers use different strategies. Some printers let you enter a numerical value for the left margin. On the ImageWriter, for example, you set a left margin of eight by sending it five characters:

```
<ESC> L 008
```


Other printers use a command meaning "set the left margin at this point." For the Diablo 630, you set a margin of eight by sending eight spaces followed by a two character command:

```
<space> <space> <space> <space> <space> <space> <space> <space> <space> <ESC> 9
```

Once you've discovered the appropriate printer control codes, create a BEX chapter that contains just those characters. Copy the GRID chapter you are working with to a MY GRID chapter. Insert the control codes at the start of MY GRID. Turn your printer off and on to reset it. Then print MY GRID to your printer to confirm that you're using the right number of characters for your left margin.

When you're satisfied with the left margin, you then reconfigure this printer, and answer Y to the Establish an automatic set-up sequence? question. When you have established an automatic set-up sequence for your top margin, enter both that sequence and the sequence for your left margin. Type the printer control codes exactly, and BEX automatically sets the left margin (and top margin) each time you print to this printer.

When your printer uses a command like the Diablo's, it's possible for your left margin to start creeping leftward because the effect of the set-up sequence is cumulative. To prevent this from happening, include the "master reset" or "remote reset" command at the beginning of your set-up sequence. For the Diablo, this is <ESC> <CR> but you must find the command for *your* printer in its manual.

Setting the carriage width

When you've set the left margin with an automatic setup sequence, you need to set a carriage width to work with that sequence. After you reconfigure, print your horizontal grid chapter again, this time with your left margin. When you've set the left margin, you need to set a carriage width to work with that left margin. Take your printed copy of RP GRID, LP GRID, or LONG GRID (or MY GRID) and find the first ruler that is too long. This is the ruler you use to measure your carriage width. Take a *real* ruler and measure a distance from the right edge of the paper that's equal to the left margin, and mark this position. Determine the value for your carriage width by counting over from the nearest vertical bar. This value is the number you use for your carriage width.

For example, suppose you have a left margin of one-half an inch. You measure one-half inch over from the right edge of the paper, making a mark along the first ruler that's too long. This mark falls at the third lowercase *o* after the vertical bar numbered 50. Counting over from that vertical bar, you determine that your carriage width is 53.

Now you reconfigure, using the values you obtained from the tests for carriage width and form length.

Part 5: Troubleshooting Configuration Problems

Inevitably, some problems may occur when you are configuring. We cover some of them here. If you are having problems saving the configuration file onto disk, see Learner Level Section 3, Part 8. When you encounter other problems, call our Technical Hotline for help.

- Problem: During configuration, I can't get BEX to recognize a card I have.
- Solution: Finish configuring without using that card. When you get to the Starting Menu, use option R - Recognition of circuit cards to force BEX to recognize your card. See the Interface Guide Section 15 for further help.
- Problem: BEX accepted my configuration fine, but when I got to the Starting Menu, everything froze.
- Possible Cause: You have configured a braille device for everything going to the screen. However, the device is off-line, and BEX hangs. Or the device is turned off.
- Solution: Press the on-line button on your device, or plug it in. You may have to reboot.
- Problem: My first page of output is fine, but then everything stops.
- Possible Cause: You configured this printer with "pause on form feed", which directs the formatter to stop printing after each page.
- Solution: Press <space> to resume output after each pause. If you don't want to pause after each form feed, set up a new configuration and answer N to the Do you want pause on form feed question.
- Problem: My printer prints over each line, so that the text is unreadable. I can't hear the paper move forward for each line; it only advances on form feed between pages.

- Solution: Reconfigure, and answer Y to the auto linefeed question.
- Problem: Everything prints fine but I can't get my text to be single-spaced. I get double-spaced text when I want single-spaced, or triple-spaced text when I want double-spaced.
- Cause: You have answered Y to the auto linefeed question in configuration.
- Solution: Reconfigure, and answer N at the auto linefeed question.

4 Working with Chapters

BEX has many methods for choosing which chapter or chapters to work with. We use the term *default data drive* to mean the drive whose number appears as the default when you press D at any menu. When you have a two-drive system, your default data drive is always drive 2. When you have a one-drive system, your default data drive is always drive 1. At the Master Level, you can have many more data drives, so this concept is more important there.

Part 1: The Story Thus Far

At the Learner Level, BEX prompts `Drive number or chapter name :` when it wants to know which chapter to work with. You either type the chapter name, or enter 1 or 2 to scan the disk in the drive. Precede a chapter name with the number of the drive you want BEX to scan, if the drive is other than the default data drive. When you tell BEX to scan a drive, BEX presents a numbered list of chapters to work with. BEX asks `Use entire list? N` and you enter `Y <CR>` to use every chapter. You can accept the N default by pressing `<CR>`. When you accept the N default, BEX prompts you to choose chapters by number. You enter the chapter numbers; when you are finished, you enter `<CR>` alone to the `Chapter number :` prompt. You then move to the next step in the option.

The next step after you specify a chapter or chapters, is either directing action on the *source chapter* list (editing, page menu options, printing, or killing) or providing BEX with a *target chapter naming method* to use in creating new

chapters on disk. You have the responsibility of checking to make sure there's enough room on the disk for the target chapters). Press # at any menu prompt to get the number of free sectors on your data disk. A full explanation of how much space on disk a chapter uses is provided in Part 4.

Part 2: New Features at User Level

As we mentioned in Section 1, we assume you are acquainted with the Learner Level features. The User Level builds on these features, adding more flexibility in terms of specifying drives and chapters.

- ***Reprompting for Chapters***

All BEX prompts are shorter at the User Level. Instead of `Drive number or chapter name:` BEX prompts for `Drive` or `chapter:` Typing a chapter name individually produces different results at the User Level. After finding this chapter, BEX prompts you with `Drive` or `chapter:` again. At this point, you can type another chapter name or enter a drive number. This reprompting lets you mix and match three methods for specifying chapters: You can specify a list of chapters entirely by name, preceding chapter names with drive numbers where needed; you can combine specifying chapters by name and specifying a drive to scan; or you can just specify a drive to scan by entering the drive number.

- ***Restrictive Scanning with the Slash***

To obtain a subset of all the chapters on your default data drive, enter a slash followed by one character at the `Drive` or `chapter:` prompt. The character following the slash is the last character of the chapter names you wish to specify. BEX presents a numbered list restricted to those chapters whose last character matches the character after the slash.

When you are systematic about naming your chapters, you can use them for your subsets. For example, if you always end your grade 2 braille chapter names with the digit 2, BEX can present a list restricted to braille chapters when you specify /2 at the `Drive or chapter:` prompt.

You precede the slash with a number to specify drive 1. You can enter 1/X or 2/X to scan chapters ending in X on drives 1 or 2, or /X with no number to scan the default data drive. You can use the slash alone at the `Drive or chapter:` prompt to get a list of all the chapters on your default data drive. When you enter the slash alone, then you haven't provided BEX with the distinctive last character. Therefore typing / <CR> at any `Drive or chapter:` prompt builds a numbered list of all chapters on your default data drive, drive 2.

- ***Scanning More than One Drive with the Plus Sign***

You can get BEX to scan more than one drive by preceding the drive number with the plus sign. You must enter a drive number after the plus sign; a plus sign alone just returns you to the menu.

For example, you wish to choose from a numbered list of the chapters on both drives 1 and 2. Enter +1 at the first `Drive or chapter:` prompt. BEX presents the numbered list of chapters on drive 1 and prompts you to choose by chapter number. When you finish this list, BEX prompts `Select more chapters` and then asks `Drive or chapter:` again. At this point, you have exactly the same range of choices as you do at the first `Drive or chapter:` prompt.

You can repeatedly specify the same drive using different restricted scans: At the `Drive or chapter` prompt, enter +2/Q and get a list of all the chapters ending in Q. Make your choices, then BEX prompts: `Select more chapters` again. Enter +2/Z and get a list of all the chapters ending in Z. Make choices from this list, and you get still another `Select more chapters` prompt. Just for variety, enter +1/M and get a list of the chapters that end in M on drive 1.

When you don't want to make another chapter selection, press <CR> alone at the `Drive` or `chapter:` prompt. BEX stops asking for chapters and moves to the next step in the process.

Part 3: Target Chapter Naming Methods

With many options, BEX creates new chapters. Working with the `chapter` or `chapters` you specify, BEX changes the data in some way and writes it to disk. Whenever you are only working with one chapter, BEX prompts `Target chapter name:` and you type it in. You can precede a target chapter name with a drive number to tell BEX where to write the chapter. When you've specified a list of chapters, BEX prompts `Target chapter naming method:` When you use the name of a chapter that's already on the disk, then the text in the new target chapter overwrites the existing chapter.

- ***Methods Available***

Single letter codes tell BEX how to modify the source chapter name to create the target chapter name. Now that you know about restricted scanning with the slash, you understand why the target chapter naming methods affect the final characters of BEX chapter names. A summary of your options appears when you enter ? <CR> at the `Target chapter naming method:` prompt. They are:

- S - Same name for source and target chapter.
- I - individually name target chapters. You may also individually specify which drive to write the target chapters on by preceding the name with the drive number.
- AXYZ - add the characters XYZ to the target chapter names. You may add any number of characters, provided the target chapter name is 25 characters or less.
- LXYZ - changes last character of source name to It, then adds YZ. The character may be any character appropriate to a chapter name.
- D - delete the last character of source names to make target names.

You may precede any of these five letter codes with a digit 1 or 2 for drive 1 or 2. When you don't precede the method with a digit, the chapters are written to the default data drive.

- ***Target Naming Methods with Another Chapter Selection***

Watch out! You can get into big trouble when you combine scanning two drives with a target chapter naming method. When you precede the code letter with the drive number, then *all* the source chapters, no matter which drive they came from, end up on the target drive. Make sure you have room for these target chapters! At the Master Level, you can have up to eight disk drives--you'll learn about more target chapter naming methods to use.

Part 4: Managing Your Data: Notes on Disks, Chapters, and Pages

Whenever you create a new chapter, you must make sure there's room for it on the disk. A standard BEX data disk has 528 sectors available for your text. Each sector can contain 256 characters. Theoretically, then, each disk could contain 528 times 256 or 135,168 characters. However, the way BEX stores data uses approximately 15 per cent of the disk space for overhead. A BEX chapter containing 30 4096-character pages would just fill one disk, for a total of 122,880 characters per disk.

To find out the number of free sectors on your disk, press # at any menu prompt. The number of sectors free also appears on the first line of a DOS catalog.

The rule of thumb

To avoid disk-space problems, limit your pages to 3300 characters. Limit the total number of pages in all chapters on a disk to 30. Always move to a new disk when FS = 100 or less. Use option W - Whole disk catalog on the Page Menu to get the total picture about a disk. Whole disk catalog tells you

how many pages are in each chapter, and the total number of characters on the disk. When the total is near 100,000, switch to a new data disk.

To get a rough estimate of how many pages you can have per sector, you have to do a little math. An average braille page has about 880 cells on it; each cell is one character. With 256 characters per sector, that yields about 3.5 sectors on disk for every braille page. An average single-spaced print page has 56 lines, with 70 characters per line. 56 times 70 is about 3920 characters per page; 3920 divided by 256 characters per sector gives a little more than 15 sectors per print page. An average double-spaced print page, 56 by 70, has about 1960 characters per page. 1960 divided by 256 characters per sector yields about eight sectors per print page.

- ***More Nitty-Gritty Details***

It's not easy to state how many characters fit on one disk. Page size, number of pages, and number of chapters all interact to determine how many sectors are used. Every directory file uses three sectors, and every page file uses two sectors, plus one sector for each 256 characters (these are the 15 per cent overhead). This means that two chapters with the same number of total characters can occupy very different amounts of disk space, depending on the number of pages in each chapter.

Two chapters compared

Chapter SKINNY contains three pages, and each page contains 3584 characters. 3584 divided by 256 characters per sector yields 14 sectors. Add two sectors overhead for each page file, to make 16 times three or 48 sectors used for page files. Add three sectors for the directory file, for a grand total of 51 sectors used to store 10,752 characters.

Chapter BLOATO also contains 10,752 characters, but it requires 63 sectors. BLOATO has nine pages: eight pages contain 1280 characters each, and one page contains just 512 characters. 1280 divided by 256 yields five sectors; add two sectors overhead for each of these page files to get seven times eight or 56 sectors. 512 divided by 256 yields two sectors, adding two sectors overhead for this little page file, a total of four sectors. Finally, three more

sectors for the directory file means three plus four plus 56 for a grand total of 63 sectors.

Because each directory file requires three sectors, each chapter on the disk adds to the overhead burden. Because each page file requires two sectors, each page in a chapter adds to the overhead burden.

- ***Page Size***

While each BEX page can contain 4096 characters, you should never make a page that full. You always want to leave enough room for additions and modifications. Particularly when you back-translate a great deal, you must leave enough room for the expansion of contractions. For these reasons, limit your pages to around 3300 characters. A 3300 character page occupies approximately 15 sectors on disk. Following the rule of thumb above, limit each disk to 30 pages. 30 times 15 equals 450 sectors, leaving 78 sectors free for directory files and later fiddling with your data.

When you use options where text is brought in from other systems, such as options I - Input through Slot, R - Read textfiles to chapters and F - From VersaBraille, BEX determines the size of the pages. These sizes vary depending on the option. With Input through slot, BEX creates pages of about 3840 characters for print text, and pages of about 3072 for braille text. With Read textfiles to chapters, BEX uses the 3300 character page limit. With From VersaBraille, BEX stores a number of complete VersaBraille pages into each BEX page, so the resulting page size varies with the size of your VersaBraille pages.

- ***Page Lettering***

As you create each page in the Editor, BEX creates a page file on disk. BEX names this page file by adding a two-character extension to the chapter name: a period plus a letter of the alphabet. BEX starts out with .A then takes the next available letter of the alphabet.

When you cut pages in the Editor or manipulate pages with options on the Page Menu, the numerical order of the pages no longer matches the alphabetical order of the page files on disk.

For example, you create PAPER that contains two pages; BEX creates the PAPER . A page file for page 1 and the PAPER . B page file for page 2. You edit page 1 and enter control-C control-P. BEX uses the next available letter of the alphabet for the new page 2; its page file is named PAPER . C

Most BEX options create target page files with the same extension as the source page files. However, two BEX options create target chapters whose page files are relred so that . A is page 1, . B is page 2, and so forth. These options are options M - Merge pages, and A - Adjust page sizes on the Second Menu. Option F - Fix chapter directory on the Second Menu does the opposite: it rennumbers pages according to their letter extension.

- ***Ordering Chapters on a Disk***

How DOS 3.3 manages disk files determines the order of chapters on a disk. The position of the directory files on disk determines the chapter order when BEX presents a numbered list of chapters on that disk. When you want to create a specific order of chapters on disk, use any option which creates new chapters to copy the chapters onto disk in that order.

For example, suppose you want three braille chapters in the order

CHAPTER1

CHAPTER2

CHAPTER3

However, your print chapters are in the order

CHAPTER2

CHAPTER1

CHAPTER3

When you translate with the Grade 2 translator, specify the print chapters by number, in the order CHAPTER1, CHAPTER2, CHAPTER3. BEX translates them in the specified order, and the braille chapters will be in the correct order on disk.

To maintain your chapter order, be careful which options you use on your chapters. Any option which rewrites the directory file of a chapter takes that chapter out of order, since the new directory file is created before the old file is deleted.

- ***The Lure of the Ready Chapter***

Once you get to the Master Level, you'll find that editing with the Ready chapter is slightly addictive, because it's so fast. The Ready chapter has a maximum of six pages on the Apple IIc and IIe, and a maximum of 20 pages on the Apple IIgs. You can safely use the Ready chapter to edit longer chapters but you lose some speed. This means that writing chapters of six pages or less, or 20 pages or less on the Apple IIgs, allows maximum use of the Ready features.

An *ideal* chapter of six 3300-character pages uses approximately 93 sectors on disk. (Around 15 sectors for each page file, and three sectors for the directory file.) This means that you can have five of these chapters on a 528-sector disk. Since you can readily chain together options like printing, translating, and replacing, and since, at the Master Level, you can have up to eight disk drives, these guidelines won't limit you too much.

5 The Editor

At the Learner Level we discussed only a few Editor commands. In this Section we introduce all the Editor commands. In addition to a broad variety of ways to move the cursor, we explore the clipboard and view mode. The clipboard is a flexible tool that makes editing text a delight. The clipboard allows you to move sections of text around very quickly. View mode lets you preview your text without leaving the Editor.

Part 1: Control Characters

In the Learner Level Section 4, Part 4, we described the smallest working unit of data as a *character*. A character in BEX can be any member of the group of the 128 *ASCII* characters. "ASCII" rhymes with "passkey" and is a well-established computer standard for internally representing characters by numbers. At the Learner Level, we mentioned the characters on the keyboard: all the lowercase and uppercase letters; the space character; the digits 0 through 9; punctuation like comma, period, percent sign, parenthesis, etc. This group of characters numbers 96, and is sometimes known as the *printable characters*.

In addition, there are 32 non-printing characters, called *control characters*. You use these characters to control the behavior of a printer or a computer. At the Learner Level, we mentioned one of these control characters: the carriage return, also known as control-M. BEX allows you to type any control character directly into your text. This means that a BEX chapter can contain instructions for the Apple, for an interface card, or for any printer.

- ***Typing Control Characters***

When you want to type a control character in your text, first enter the Editor command control-C. The next key you press is interpreted as a control character and is placed as in your text. In the 80-column Wide mode and the 40-column Non-HI-RES screen mode, all control characters appear as the delete checkerboard. In all other screen modes, control characters appear on your screen as small uppercase letters with lines under it.

Entering any control character that's a letter is easy: first enter control-C and then press the appropriate letter. Since most of the one-key control characters, such as <CR> and the left and right arrow keys, have a control character assigned to them, you can enter them two ways in your text. For example, entering control-C and then pressing M has the same effect as pressing <CR>.

In printer manuals, you may encounter some strange looking characters like control-caret or control-shift-2 that you have to enter into your text as printer instructions. Just take these names as literal instructions: for control-shift-2, first enter control-C, then press the shift key and the digit 2 simultaneously. The result is the *break character*, also known as *null* or ASCII zero.

It's important to distinguish between control characters as Editor commands and control characters as items in your text. You use control character Editor commands to delete, insert, and move around in your text. These commands are not placed into your text. You enter control characters into your text to give instructions to BEX and your printer. These commands *are* placed into your text, and can be manipulated as normal characters.

- ***Control Characters on Voice and Braille Channels***

As we mentioned above, the Echo speaks the control characters that are in your text. BEX uses a special vocabulary for voice and braille output when you move character by character with the left and right arrow keys. When you enter a "control-shift-2" in your text and then arrow over it, your voice or braille device outputs "break." Some control characters have such long names that we chose to abbreviate them to "ASCII 31" or "ASCII 32."

- ***Control Characters on the Screen Channel***

As we mentioned above, every control character appears as the *delete* checkerboard in the two screen modes that use built-in character generators, modes W and N.

For users with large print and braille screen access devices such as a DP-10 or BDP, the command control-C D toggles the control characters in your text between the default checkerboard, and the at-sign. However, this is only visible in screen modes W and N. Part 6 explains more about this command. For all other screen modes, print and braille, BEX draws special characters. Most control characters appear as a smaller, underlined uppercase letter. As mentioned earlier, <CR> appears as tiny uppercase C and R jammed together. Control-J is the ASCII standard for *line feed*, so it appears as tiny uppercase L and F jammed together.

Part 2: Entering Editor Commands, Continued

At the Learner Level, we described in detail how to enter Editor commands correctly. Everything we said is true, but we omitted some potentially confusing details.

Remember that when a command is defined as two control characters such as control-A control-W you *must* type two control characters.

When a command is defined as a control character followed by one or more plain characters such as control-S A you *may* type the plain characters as control characters. After you enter control-S, BEX is laissez-faire about both shift and control keys: it interprets control-S control-S control-W exactly the same as control-S S W.

With standard versions of TEXTALKER you can change the Echo command character by entering control-E followed by another control character. For example, entering control-E control-A changes the Echo's command character from control-E to control-A. We have modified the TEXTALKER on your BEX disk so that this is not possible. However, if for some reason you have copied a different version of TEXTALKER onto your BEX disk, you should be aware that typing control-E control-A changes the Echo's command character to control-A. Control-A would no longer move the cursor forward. To recover, enter control-A control-E to change the command character back to control-E.

There are several editor commands that we document as being executed with the spacebar. An example is control-A <space> to move to the end of the page. As you may have discovered through experimentation, the spacebar is not the only key that executes the command. In fact, any key that is not specifically assigned to a command executes that command. In the case of advancing the cursor, control-A <space>, control-A W, control-A <CR>, control-A backslash and control-A control-J (or down-arrow) have identical effects.

If you're a "hot dog" typist, you can use the control character that starts one command as the "any key" character that executes the previous command (which is very handy for inserting). For example, you wish to insert five words of text and then delete the rest of the paragraph. Enter control-I, type the five words, then enter control-D control-P.

Part 3: Moving Around in the Editor

We introduced a few of these commands at the Learner Level. Now we'll give you the complete list. All the cursor move commands are available for all users; however, some cursor move commands also output text to the voice and braille channels.

Before we introduce the following command, we need to explain more about your text output. Because BEX is oriented around characters, internally it doesn't save data as *lines*. The *lines* that are centered around line-oriented movement commands (explained below) are not the same as *output lines*. The line-oriented movement commands work with lines which are created by the format commands you place in your text. *Output lines* are the result of the carriage width you specify when you configure a printer.

One unit you can specify for moving and deleting text is control-L for a line. This only refers to an explicit new line created by a hard <CR> or a new-line (\$l) indicator you type in your text. BEX only divides your text into output lines when it's printing. Control-L cannot refer to output lines in the Editor, since BEX doesn't know where they will be.

You can move by sentences in the Editor. BEX's definition of a sentence includes any series of letters ending with some punctuation such as period, question mark, exclamation point, and left parenthesis, then <space>. Because of this, BEX interprets most abbreviations as the end of a sentence. Keep this in mind when you use the sentence command, control-T.

When we introduce the commands, we use the number sign # to stand for number you enter. This number may consist of one or more digits. For example, when we say that control-A # <space> advances the cursor # characters, the # can be a digit from 1 to 4096, the limit of the page characters.

- ***Silent Cursor Movement***

The up arrow, control-K and the down arrow, control-J move the cursor up or down one line on the screen. When your cursor is located at the top or

bottom line on the screen, then control-K and control-J will scroll the screen, too.

Control-A starts many editor commands which Advance the cursor forward through your text: how far you advance depends on the subsequent numbers and *unit* character.

- Control-A # <space> - advances you cursor # characters; entering control-A <space> advances to the end of the page
- Control-A # control-W - advances # words; when you enter no value for , the cursor advances one word. Cursor lands on the space or <CR> that defines the start of a word
- Control-A # control-T - advances the cursor # sentences; when you enter no , advances one sentence. Cursor lands at the end of the sentence
- Control-A # control-L - advances the cursor # new lines; when you enter no , advances to the next new line. Cursor lands on the initial space of the # new-line (\$l) indicator or the next hard <CR>
- Control-A # control-P - advances the cursor # paragraphs; when you enter no , the cursor advances one paragraph. Cursor lands on the initial space of the paragraph (\$p) indicator
- Control-A control-S - advances the cursor forward one screen

You specify how many units to advance by entering control-A # then the unit's control character. This includes all the above commands except for control-A control-S. For example: control-A 200 <space> advances 200 characters, control-A 5 control-W advances five words, control-A 4 control-T advances the cursor to the space after the fourth sentence, control-A 3 control-L advances the cursor to the third new line, and control-A 2 control-P advances two paragraphs.

To move the cursor backward, use control-Z in place of control-A:

- Control-Z # <space> - zooms the cursor back # of characters; when you enter no value for , the cursor zooms back to the beginning of the page
- Control-Z # control-W - zooms back # words; when you enter no , the cursor zooms back one word. Cursor lands on the space or <CR> that defines the start of a word
- Control-Z # control-T - zooms back # sentences; when you enter no , the cursor zooms back one sentence. Cursor lands at the end of the sentence

- Control-Z # control-L - zooms the cursor back # lines; when you enter no , zooms the cursor back one line. Cursor lands on the initial space of the previous new-line (\$l) indicator or on the previous hard <CR>
- Control-Z # control-P - zooms back # paragraphs; when you enter no , the cursor zooms back one paragraph. Cursor lands on the initial space of previous paragraph (p) indicator
- Control-Z control-S - zooms back one screen

Just as with control-A, you use a number to specify how many units to move. You can use any unit command except control-S. Control-Z 5 control-W zooms back 5 words, control-Z 4 control-T zooms back the cursor to the space after the fourth previous sentence, control-Z 3 control-L zooms back the cursor to the third previous new line, control-Z 3 control-P zooms back three paragraphs, and control-Z 1400 <space> zooms back 1400 characters.

• ***Moving to Specific Characters: Locating Text***

You can locate text from your current cursor in two directions. Start out with control-L, then type the exact characters you wish to locate. Your search string can be up to 38 characters long. As you are typing in the characters you wish to locate, you can use the left arrow key to fix typing errors.

After you have typed your search string, execute the locate command with control-A to locate in advance of the cursor (towards the end of the page.) Or, execute the locate command with control-Z to locate backwards of the cursor (towards the beginning of the page). The next time you use control-L, you do not need to type the search string again. When BEX can't find an occurrence of your string, you get one high error beep and your cursor stays where it is. The locate command remembers the search string characters until you turn off the computer or enter different characters.

Control-L is very picky when it comes to searching for sequences of characters. You must type the search string exactly as it appears in your text. This includes case and punctuation. For example, you want to search for the word *Braille* and the word appears with both an uppercase and lowercase *B*. You must search twice: once with *Braille* and once with *braille*. A shortcut to the problem of initial letter case variation is to ignore the first letter. Typing

your search string as `raillle` finds both the uppercase and lowercase words.

Type initial and final spaces in your string when you want to find a word as oppose to a series of letters. Suppose you're trying to locate the word *the*. If you type `the` for your search string, your cursor lands on *other*, *weather*, and *theocratic*. When you type `<space>the<space>` as your search string, your cursor only lands on *the*.

To type a control character into your search string, you use the same method as typing control characters in your text. Enter control-C then type letter. For example, suppose you want to search for every occurrence of control-S followed by two dollar signs in your text. Control-L starts the locate command. Next type control-C `S $$` to define the search string, then enter control-A to start searching in front of the cursor.

- ***Talking Cursor Movement***

At the Learner Level, we documented many commands as talking movement commands. By this point, you're probably not surprised to discover that the same commands also output to the voice *and* braille channels.

You can move your cursor by words and output to the voice or braille channel with control-G and control-R. Control-G speaks and Goes forward a word. Control-R speaks and Reverses a word. You can move by words without output with control-A control-W and control-Z control-W.

The speed at which the cursor arrives at its new position is determined by the more sluggish device; when your cursor arrives at its new position, the Apple speaker makes a low boop. When you do not have a device attached to the voice or braille channel, these commands still move your cursor as described.

The left arrow key, control-H, and the right arrow key, control-U, move the cursor one character at a time to the left and right; *arrowing* speaks the character the cursor moves onto.

BEX uses a special vocabulary when outputting these characters. Regardless of how your output device handles capitalization and punctuation, BEX

outputs the exact status of the character: whether it is a plain or control character, etc. On the braille channel, you get the single screen braille cell corresponding to the character you arrowed to. For the voice channel, BEX uses a special vocabulary, with some creatively mangled spelling to improve pronunciation.

The next three commands allow you to review portions of text with voice and/or braille devices. The spacebar can be used to stop output; BEX boops when passing a hard <CR>; details are explained below.

- Control-T - talks the next sentence. BEX recognizes the end of a sentence by various combinations of punctuation and <space> or <CR>, which means that sometimes BEX doesn't *really* recognize a sentence. When your text contains abbreviations with periods, then BEX stops too soon.
- Control-Y - acts just like control-T except that your cursor doesn't move. Enter control-Y to output the next sentence as a means of discovering where you are in the text. You can shut up the Echo by pressing <space>; your cursor remains where it was when you entered control-Y. If you have no device attached to the voice or braille channel, then control-Y appears to do nothing.
- Control-O - the "output a bunch" command. Enter control-O, and BEX keeps speaking until the end of the current BEX page, or until you press <space>.

The spacebar stops output

For control-O, control-T, and control-Y, you can press <space> to stop output and cursor movement. If your serial voice device has any buffer, however, pressing <space> may not immediately stop the speech.

Using control-O and the spacebar is very handy for proofreading. When speaking, BEX pauses slightly between each word to check to see if you've pressed <space> to stop the speech output. BEX accomplishes this pause by inserting <CR> between each word right before it's sent to the voice and braille channel. When you set your voice device to pronounce <CR>s, then you hear "return" between each word when BEX reads with control-T, control-Y or control-O. Those <CR>s are not actually in your text.

The environmental command control-S J is the "jerky speech" toggle command. The default is jerky. Enter control-S J and BEX stops inserting

<CR>s, and the speech is much smoother. However, when you have toggled off jerky speech, pressing <space> no longer immediately stops output and cursor movement. See Part 6 for details.

BEX boops over a hard <CR>

When you enter control-G, control-R, control-T, control-Y, and control-O, BEX gives a low boop when it passes over a hard <CR>. When you arrow over it with the left and right arrow keys, hard <CR>s are spoken as "return."

- ***Moving from Page to Page***

Each BEX page can hold 4096 characters, but you don't need to fill up each page. To move from page 1 to page 2, enter control-P 2 <space>. The disk drive whirs as the characters in page 1 are saved, and you're at character position zero on page 2. When you enter control-P 0 <space>, BEX saves the current page and returns you to character position zero on the same page.

You can also move back and forth between pages without specifying the page number: control-P control-A advances to the next page. Control-P control-Z zooms back to the previous page. This also saves the information in the previous page to disk. Entering control-P <space> cancels any page move.

Control-C control-P cuts the page at the current cursor, leaving you at character position zero of the second page you've just created. You can use any of these features at any time to create more pages. BEX renumbers the pages automatically.

- ***Deleting Text***

At the Learner Level, we introduced several deletion commands. Now we present them all. The syntax for deletion commands is similar to control-A and control-Z: you combine control-D with a number and a unit command.

- Control-D <space> - deletes the character the cursor is on; Control-D # <space> deletes # number of characters

- Control-D control-W - deletes one word; control-D 15 control-W deletes 15 words
- Control-D control-T - deletes from the cursor to the end of the next sentence; control-D 6 control-T deletes six sentences
- Control-D control-L - deletes all text from the cursor up to, but not including, the initial space of the next new-line (\$l) indicator or to the next <CR>; control-D 6 control-L deletes all text to the sixth (\$l) or hard <CR>
- Control-D control-P - deletes from the cursor to the initial space of the next paragraph (\$p) indicator; control-D 6 control-P deletes six paragraphs
- Control-D control-A - deletes all the text from the character the cursor covers to the end of the page
- Control-D control-Z deletes from the beginning of the page up to, but not including, the character the cursor covers

Deleting the block is discussed under Part 4, *The Clipboard*.

Avoid pages with zero characters

As you are editing text, you may decide that all the characters in one page are useless, so you want to delete them. The best way to accomplish this is with option K - Kill pages on the Page Menu. To jog your memory about which page you want to kill, you can delete all but three characters. Then at the Page Menu, do File list. Any page with three characters is one you want to kill.

You might think that a simpler way to get rid of a page is to delete every character with control-D A, creating a BEX page with zero characters. We don't recommend this. If you delete every character in a page in the Editor, you end up with some orphaned page files cluttering up your disk. Understanding why requires a brief excursion into the bowels of BEX and DOS 3.3.

Whenever you ask BEX to manipulate a page file, BEX checks to see if that page has zero characters. When a page file is listed in the directory with zero characters, then BEX never asks DOS to BLOAD, BSAVE, DELETE, or RENAME that page. That's because DOS 3.3 can't handle a file with zero characters. If BEX asked DOS to work with an empty file, DOS would freak

out. So when you delete all the characters in the Editor, BEX only updates the directory file. BEX does not save a page file with zero characters.

Suppose you have a chapter named PSALM 25 that contains three pages. Before editing, option F - File list on the Page Menu yields this information:
Chapter PSALM 25
3 pages
Page 1 Size882 A
Page 2 Size 1764 B
Page 3 Size 3528 C

When you do a DOS catalog of this disk, you see these four files:
B005 PSALM 25.A
B008 PSALM 25.B
B015 PSALM 25.C
B003 PSALM 25

Suppose you edit the PSALM 25 chapter and delete all the characters in page 2, then quit. BEX updates the directory file to reflect the new size of page 2, but it does not save an empty PSALM 25.B page file. The older version of the page remains on disk.

Zip to the Page Menu now, and File list looks like this:
Chapter PSALM 25
3 pages
Page 1 Size882 A
Page 2 Size0 B
Page 3 Size 3528 C:

However, when you do a DOS catalog, the PSALM 25.B page file still has 8 sectors. You might think you could use option K - Kill pages to delete the PSALM 25.B page file, but that won't work. BEX checks the directory and discovers that page 2 holds zero characters, so it only changes the directory file. To get rid of the orphan PSALM 25.B page file, Quit BEX and use the DELETE command at the BASIC prompt.

- ***Inserting Text from the Keyboard***

You can use control-I to signal the start of *keyboard insert*. After you enter control-I, every character after your cursor turns into the underbar character. Each character you type is inserted in your text immediately before the cursor. While inserting, you can press <CR>, left arrow or <ESC>. Pressing control-N, the right arrow key, or any other control character except <CR>, the left arrow key or <ESC> finishes the keyboard insert and executes the command associated with the control character.

- ***Status Information***

The bottom line on the screen displays information about the current cursor position, the *size* (or total characters in current page), and the number of the current page. In 5-column screen, the display only shows the current cursor and total size.

Use the "Where am I?" command, control-W for status information output to screen, voice and braille channels. When you enter control-W, the text temporarily disappears. You're presented with a question mark prompt. You can now enter any of four characters:

- B - for block information: clipboard size, followed by marker position
- C - for character information: current cursor and total characters in page
- P - for page information: current page number and how many pages in the chapter
- A - for all information: all three of the above items in that order

You may press B, C, P and A as many times as you need to get information. After you digest the control-W information, press any key *except* B, C, P or A to return to the Editor.

Part 4: The Clipboard

The clipboard is an electronic scratchpad that helps you manipulate blocks of text. The clipboard features described here are available only with a 128K Apple.

64K Apple: | Users of a 64K Apple should read Appendix 4 for information on the limitations of the clipboard size and function.

The clipboard is a "floating" temporary page. It may contain up to 4096 characters, as large as a regular BEX page. However, it is not a "real" page in an actual chapter. You can place text on the clipboard from one page and insert it from the clipboard onto another page within the same chapter or in a different chapter. You can move text on and off the clipboard in various ways, which lets you accomplish many different tasks.

You cannot save the clipboard itself to disk. The contents of the clipboard remain until you copy more information on top of it, delete it, or turn the Apple's power off.

The idea of a *block* was introduced in the section covering deletion in Learner Level Section 4, Part 10. Using the same commands, you can copy a portion of text onto the clipboard. You can delete or insert text using the block commands we describe in this Part.

● **Block Marker Commands**

We first introduced these commands at the Learner Level.

- Control-B S - block set: set the block marker at current cursor position
- Control-B L - block locate: locate the marker: moves cursor to marker
- Control-B D - block delete: delete the text from the marker forward to your current cursor

To delete a block of text, you must have a marker set. The marker can define only the start of the block; the end of the block is the character immediately preceding your cursor. Control-B L moves your cursor to the marker either forward or backward, as required.

● **Clipboard Commands**

- Control-B A - board append: append text between block marker and cursor to the clipboard

- Control-B C - board copy: copy text between block marker and cursor to the clipboard and overwrite its previous contents
- Control-B I - board insert: insert contents of clipboard into the current page at the current cursor position
- control-B X - board exchange: exchange all the text in the current page with the contents of the clipboard
- control-B R - board restore: restores accidentally deleted clipboard. Restored clipboard has 4096 characters: text plus nonsense characters. Press Y to restore. Any other key returns you to the Editor

● ***Moving the Marker Around***

You can only have one marker set at any one time. Every time you enter control-B S, the old marker is erased as the current marker is set. Since the marker is an invisible pointer, erasing it does not alter your text.

The marker is automatically erased every time you execute any insert or delete command. This prevents BEX from becoming confused about where the marker is located. This means you should always set the block marker immediately before you append or copy text to the clipboard.

● ***Placing Text on the Clipboard***

There are three ways to copy text to the clipboard. First, you can exchange the contents of the page and the clipboard. Second, you can *append* text to the clipboard. When you *append* a block to the clipboard, you are placing the copied text after whatever text is already on the clipboard. Third, you can *copy* text to the clipboard. When you *copy* a block to the clipboard, then all the text on the clipboard is overwritten by the copied text.

Exchanging the page and the clipboard

You do not need to set a marker when you want to exchange the contents of the page and the clipboard. At any time, you may enter control-B X, and the clipboard and the current page are exchanged. Your cursor is located at character position zero. When you enter control-B X when the clipboard is empty, then you have a blank page. You haven't lost any of the text in your

page, it's just temporarily on the clipboard; it's like putting a phone call on hold.

Control-B X lets you examine and edit the contents of the clipboard. What was on the clipboard is now in the page. If you save the page to disk when the clipboard and you page are exchanged (either by control-Q or moving to another page) you won't save the text that's on the clipboard.

Enter control-B X again, and the original text from the page is back in the page, and the original text from the clipboard is back on the clipboard. It's important to develop the habit of checking to make sure that the text you wish to save is *in* the page and not on the clipboard before you save it.

Here's an example. You have a chapter with 3 pages: page 1 contains 1200 characters, page 2 has 2900 characters, and page 3 has 800 characters. You start on page 1. First, check to see the status of the clipboard: enter control-W B for block and clipboard information. After the question mark prompt, you get this message: Clipboard 0 No Marker so you know the clipboard is empty. Enter control-B X: all 1200 characters from page 1 are now on the clipboard. Use control-W B, and the response is: Clipboard 1200 No Marker . Now move to the next page with control-P 2 <space>. Enter control-B X to exchange the contents in page 2 with the clipboard. Control-W A yields:

```
Clipboard 2900 No Marker
Cursor 0 Size 1200
Page 2 of 3
```

You have moved all the text from page 1 in to page 2. All the text that was in page 2 is now on the clipboard. Move back to page 1, and enter control-B X one last time. Control-W A informs you:

```
Clipboard 0 No Marker
Cursor 0 Size 2900
Page 1 of 3
```

You've just switched the text in pages 1 and 2 by using the clipboard as an intermediary. You could accomplish the same thing by using option E - Exchange pages on the Page menu, but the clipboard is much faster!

Copying text to the clipboard

To copy text to the clipboard, you must first set the block marker at the beginning of the text you wish to copy. Move to the beginning of the text. Enter control-B S to set the marker. The marker defines the beginning of your text. Now move your cursor forward until you're at the end of the text you wish to copy. The marked text includes all the text from the marker up to the cursor; the character the cursor covers is not copied. Enter control-B C, and the marked text is copied onto the clipboard. Whatever text was on the clipboard is overwritten by the text you've just copied.

Copying text to the clipboard does not affect the text in your page. You must consciously remove this text by entering control-B D to delete text you marked in the page. Deleting text in the page with control-B D does not affect the contents of the clipboard.

Appending text to the clipboard

Appending text to the clipboard is a similar process. First establish the marker with control-B S, then move to the end of the text you want appended. Now enter control-B A to append the text onto the clipboard. The text from where you set the marker to the character before the cursor is appended after any existing text on the clipboard.

The text you just appended is still in your page; if you wish to delete it, then enter control-B D.

When you want to move text from the page to the end of the clipboard, you do two steps. First, append or copy the text onto the clipboard by setting the marker and using control-B A or control-B C. Second, use control-B D to delete the block of text you just appended.

Moving Text from the Clipboard to the Page

There are two ways to move text from the clipboard to the page. The first way we've already discussed: you can exchange text from the clipboard to the page by entering control-B X.

The second way is to enter control-B I. A copy of all the text on the clipboard is inserted in your page, immediately before the current cursor position. The contents of the clipboard is unchanged; you can enter control-B I somewhere else and insert the same text again.

- ***Clipboard-Related Error Beeps***

When you use control-B commands, you can receive three kinds of error beeps. A single high error beep signals that you've entered a command sequence BEX doesn't recognize. For example, if you enter control-B F, you get one high beep.

Two high beeps means there's not enough room to execute the control-B command. This can happen with control-B I (there's not enough room in your page) or control-B A (there's not enough room on the clipboard.)

Three high beeps indicates a marker error. When you place your cursor *before* the marker and then enter control-B A, control-B C, or control-B D, BEX rebels with three beeps. The marker always defines the start of your block, so it must precede your cursor.

When you enter a command that requires a set marker, but you have no marker set, you also get three beeps. Either you neglected to set the marker, or you executed an insert or delete command and your marker was erased. You can get three error beeps when you enter control-B A, control-B C, control-B D or control-B L.

- ***Some Examples of Using the Clipboard***

The ability to edit text in the clipboard (by entering control-B X, editing, and then exchanging back again) means that there are literally as many things you can do with the clipboard as you can in a BEX page. You'll find that many tasks you used to use the Page menu for can be accomplished more quickly with the clipboard.

The following examples are not exhaustive. For clipboard success, keep two things in mind. First, the contents of the clipboard are never saved to disk. When you want to hold on to the contents of the clipboard, put them in a page somewhere. Second, remember that control-B C overwrites all the text on the clipboard. You should be sure you don't need the text on the clipboard before you enter control-B C.

As you are developing familiarity with the clipboard, you can check on its status in two ways. Control-W B or control-W A lets you know how many characters are on the clipboard, and where your marker is in the page, if you've set it. Control-B X lets you directly examine the clipboard's contents. After you enter control-B X for a clipboard exchange, be careful not to overwrite the data on the clipboard by inadvertently using control-B C.

Inserting a paragraph

As you review your text, you see the need for a new paragraph. Insert a few unique characters in the text you can locate later to find your place: for example, ~~ (two tildes). Now enter control-B X to place all the text in your page on the clipboard. Your cursor is at character position zero of a blank page. Write your paragraph, but don't use control-B A or control-B C.

When you're finished writing your new text, enter control-B X to put the paragraph on the clipboard. Your cursor is at character position zero of your page. Enter control-L ~~ control-A. Your cursor is right where you want to insert the paragraph. First delete the tildes, then enter control-B I and a copy of the paragraph appears in your text.

Switching text

Place your cursor at the beginning of the text you wish to switch; enter control-B S to set the block marker. Move to the end of this text.

When you want to switch a sentence, enter control-T to get to the end of the sentence. To move to the next new line, enter control-A control-L. To move a paragraph, enter control-A P. For larger blocks of text, enter control-O to start outputting to the end of the page, and press <space> when you hear the end of the text. Since these commands always move your cursor to the space or <CR> that defines the end of a word or sentence, the complete *word* is included in the block, while the character the cursor covers is not.

Once you arrive at the end of the text, enter control-B C to copy the text to the clipboard (and overwrite anything that's already there). Enter control-B D to delete the text in your page that you just copied on to the clipboard. You have executed a delete command, so now your marker has been erased. Move the cursor to where you want the text to go. This can be in the same BEX page, in another page in the same chapter, or in a different chapter. To

move the text to a new chapter, simply quit the chapter you're in, and edit the chapter you want the text put into. Enter control-B I to insert it.

Erasing the clipboard

Enter control-B S to set the marker, then immediately enter control-B C. Since you haven't moved your cursor, you've copied zero characters to the clipboard, effectively erasing it.

When you want to make sure that you're not erasing anything important, then use this procedure: enter control-B X to exchange the clipboard and the page. Review the text for anything worth keeping. (You can append any worthwhile text to the end of the current clipboard, which is your original page). Enter control-Z <space> to position the cursor at the start of the page, then enter control-D A to delete all the text. Finally, enter control-B X again to get back to your original page.

Make a short-term backup of text

Enter control-B X to exchange the contents of your current page with the clipboard. Enter control-B I to insert the contents of the clipboard (your original text) into the page. Now type away, but don't use any clipboard commands. If you decide that you want to go back to your original text, just enter control-B X again and there it is.

Rearranging a series of paragraphs

Locate the paragraph you wish to have appear first. Copy it to the clipboard. Move to the next paragraph, and append it to clipboard. Continue appending until you've arranged the paragraphs in the right order on the clipboard, then enter control-B X to move the text in to the page.

Quick and dirty search and replace

Section 8 explains option R - Replace characters in great detail. At the User Level, however, Replace characters requires some disk access. You have to quit the Editor, choose option R - Replace characters, and read and write chapters to disk. At the Master Level, Replace characters is much faster. The following technique is good for quick changes.

In Part 3, we described the locate command: use control-L combined with control-A or control-Z to find particular characters in your text. We refer to the characters you're locating as the *find string*. The characters that take the place of the find string are called *change to string*.

The first step is entering control-B X to temporarily put your page "on hold" on the clipboard. Delete any characters that were in the clipboard with control-Z <space>, control-D A. Type the characters for your change to string. Enter control-B X again to return to your original page. Your cursor is now at character position zero. Enter control-L, then your find string, then control-A. At the first occurrence of these characters, enter control-B I to insert the characters from the clipboard. Your cursor is still on the first character of your find string.

When your find string is short, use control-D # control-W or control-D # control-T to get rid of it. When your find string is long, enter control-B S to set the marker; move forward with control-T or control-O; press <space> to stop your cursor, then enter control-B D to delete your find string. Enter control-L control-A again to locate forward, and repeat the process as needed.

Creating a table of contents

Tables of contents are made up of the major points in an essay or article. Usually, you use format commands \$\$h and \$\$c to make distinctive the headings and subheadings in your text. Following our recommendations in the Learner Level Section 6, Part 3, you concluded these headings with (\$p) indicators.

When your text follows these format guidelines, then creating a table of contents is a breeze. Use control-L to locate occurrences of \$\$h. At the first occurrence, enter control-B S to set the marker, then enter control-A control-P to advance to the next (\$p) marker. Now enter control-B C to clear the clipboard and copy the first entry to it. For every subsequent occurrence, enter control-B A to append the following heading or subheading to the contents of the clipboard. Advance through the entire chapter or chapters this way, harvesting all the items for the table of contents. Edit a new chapter and use control-B X to examine the entries for your table.

Restoring old text while keeping new text

You can use the clipboard to hold new text you've created while restoring old text you've changed. You save the new text by copying it to the clipboard, instead of leaving it in the page and saving the page. Then, you crash out of the Editor with Control-Reset so that you don't save the unwanted text, but still have the original text on disk.

For example, suppose you are editing an existing chapter. The page you are on has two paragraphs. You delete the first paragraph and create a new version. Then you realize you really want to use the original first paragraph plus the new paragraph you've just created.

First, copy the new text onto the clipboard: Move to the beginning of the text and enter control-B S to set the marker. Then move to the end of the text with control-A <space> or any of the other movement commands. Enter control-B C to copy the new text onto the clipboard. Append any other new text to the clipboard with control-B A.

Next, you need to restore your old text. When you have not saved since before you made changes in your text, you can enter Control-Reset to crash out of the Editor. Your clipboard text is saved, because it is not on any page in your chapter. Edit your chapter; your old text is back. You may then enter control-B I to insert the new text you saved into your chapter where you need it.

Part 5: Previewing Print Format in the Editor

Sighted users sometimes feel disoriented when entering data in BEX's Editor. That's because BEX fills each screen line, so words are sometimes broken between lines. One command allows you to preview format commands your text without quitting the Editor: control-V, the *View Mode* Editor command.

- ***Page Format for the View Mode***

When you enter control-V, the default page format is 80 characters wide by 24 lines long. All format commands on the page are executed. Format

commands such as `$$w#` and `$$f#` can easily override this format. Page breaks are shown by a line of slashes.

When you enter control-V, text scrolls up the screen quite fast. Use control-S to pause and restart the scrolling. Once the text is completely printed, you press any key, including control-S, to return to the Editor.

After all the text is printed with control-V, and before you return to the Editor, the cursor appears in the lower left-hand corner of the screen. You may also get some extraneous letters near the cursor; these are not in your text and should be ignored.

Your current cursor position determines how much of your text is printed when you enter View Mode. The text where the cursor appears becomes the middle screen line when you enter control-V: line *L* for Echo and SlotBuster users. When your cursor is at the end of the page, all the text on the page is printed to the screen with control-V. When your cursor is at the beginning of the page, only the first 11 lines of formatted text are printed. When your cursor position is in the middle of the page, all the text before the cursor, plus about ten lines after are printed to the screen.

Your cursor position is not affected when you enter control-V; you cursor is in the same place when you return to text.

- ***The Effect of Format Commands in View Mode***

When you enter control-V BEX prints the text of the *current* page to the 80 column screen. All format commands on your current page are executed, including line spacing and paragraphs. However, any long-term command on a different page is *not* executed. When you have a margin command or paragraph indent at the beginning of your chapter, those commands are only executed when you are viewing that page.

Clipboard format commands to the current page

You can temporarily copy your format commands from page 1 to your current page using the clipboard. Use control-B C to copy them onto the clipboard, and control-B I to insert them at the beginning of the page. The commands will then be executed when you enter control-V.

When you are finished editing the chapter, make sure to delete the format commands you've placed in your pages. If you do not, strange things may happen with your format. If you changed the margin using a plus or minus sign, and you did not delete all but the first instance of the command, your margin would increase or decrease by that amount every time BEX's formatter encounters the command.

For example, suppose you place the command `$$ml+5` at the beginning of your text. You clipboard it onto subsequent pages for use with control-V, and then forget to delete it before printing. Each time the formatter encounters `$$ml+5`, it moves the margin in five positions: so the text of your first page will have the correct format, but once the second page of your chapter is printed, its margin will be ten positions in, instead of five. The third page will be fifteen, and so on.

Part 6: Customize the Editor Environment

At the Learner Level we introduced three commands for customizing the environment in the Editor. At the User Level you have four classes of commands at your disposal, giving you control over the screen output, the keyboard input, the voice and braille channels, and special Editor features.

- ***Screen Modes***

You can choose from ten screen modes by entering control-S S followed by a screen mode letter. Any change lasts until you turn off your Apple or reboot. When you quit the Editor, the menu prompts are displayed with the screen mode you specified in your configuration. The next time you use the Editor, you are in the screen mode you had the last time you quit.

The ten screen mode letters are:

- Control-S S W - Wide (80 column print, control characters show as checkerboard)
- Control-S S H - HI-RES (40 column print)

- Control-S S N - Non-HI-RES (40 column print, control characters show as checkerboard)
- Control-S S L - Large (20-column print)
- Control-S S X - Roman numeral X (10-column print)
- Control-S S F - the word Five (5-column print)
- Control-S S B - Braille (40-column)
- Control-S S J - Jumbo (20-column braille)
- Control-S S Y - Yet larger (10-column braille)
- Control-S S G - Gigantic (5-column braille)

You use these same ten letters when printing to the screen--see Section 6 for details.

Screen channel affects the Editor's "feel"

The screen channel is independent of the voice and braille channels, so the screen mode you specify doesn't affect how your voice or braille device displays characters. The larger the screen mode, the longer it takes to draw the letters on the screen, and the more your keystrokes end up in the keyboard buffer. W and N screen modes are the fastest, because they use the Apple's built-in character generators. If you are partially sighted and a fast typist, you can use W or N mode while you are typing, then switch to L or X to carefully proofread your work. Sighted transcribers who are familiar with braille dot patterns can use one of the braille screen modes for proofreading.

● ***Braille Keyboard Mode***

As well as using the Apple keyboard in its time-honored fashion, you can also use six keys and the spacebar like a Perkins-style braille keyboard. Changing to braille keyboard is different from all other environmental commands. Every time you quit the Editor, the keyboard mode returns to normal; if it didn't, you would have to use the braille keyboard to type in chapter names.

Apple IIgs: | The braille keyboard mode does not work on the Apple IIgs.

Caution! | When you use the N or W screen modes along with the braille keyboard, you cannot see your cursor.

Caution! | The braille keyboard always creates lowercase letters.

The braille keyboard uses the "home" keyboard row: the keys S D F and J K L.

These are the codes you use to enter and exit braille keyboard mode:

- Control-S K B - Braille keyboard (uses S D F J K L, and the spacebar)
- Control-S K N - "Normal" keyboard

If you're familiar with the Perkins-style braille keyboard, then BEX's braille keyboard mode lets you do the same thing with less effort. You must depress the Caps Lock key to activate the seven keys: dots 1, 2, 3, are letters F, D, S, respectively; dots 4, 5, 6 are J, K, L; and the spacebar is the spacebar.

There are two ways to issue Editor commands in braille keyboard mode. You can use the same technique as with the regular keyboard: to issue commands, depress the control key, then press and release the appropriate letter key. You can also *chord* control characters, by brailing the letter and pressing the spacebar. Pressing F D J K and <space> simultaneously enters the Editor command control-G to go ahead one word.

There are two possible ways to type a control character into your text. The first is the same as with the regular keyboard: enter control-C then press the letter. The second way is to press F J and <space> and then the braille equivalent of the letter. For example, press F J and <space> then F J to enter the control-C character into your text.

To leave braille keyboard mode, depress the control key, then press S K N, then release the control key and the Caps Lock key. Braille keyboard mode is automatically canceled when you quit the Editor.

Caution! We do not recommend use of the braille keyboard mode for BDP (Braille Display Processor) or DP-10 users--you lose the cursor that your BDP or DP-10 can track. You can switch to a large print screen mode if you wish large print output.

• ***Sending Commands to the Voice and Braille Channels***

At the Learner Level, we detailed sending Echo commands inside the Editor. The only Echo command you can't use is control-L, because that's what BEX uses for locating. The syntax for Echo commands is the same in the Editor and at menus: control-E followed by the appropriate plain numbers and letters. (For example, control-E 12 V to set the volume medium loud.)

Commands to non-Echo devices

You can also send commands to other devices on the voice (and braille) channel. You have to find out the commands from that device's documentation. Use control-S O to start a command sequence, and <CR> to end it. For example, to send the Echo pitch command directly to the Echo itself, enter control-S O control-E # P <CR>. Control-S O tells BEX to send the next command on to the voice device. The <CR> at the end is mandatory--it tells BEX that you've finished your command sequence.

Your braille device may be controllable through commands sent from the host device. You may need to temporarily change command characters to prevent conflicts. The control-S O sequence sends the following commands to both the voice and braille channels. When you configure both an Echo GP and a VersaBraille as "devices for all the dialogue on the screen," then both devices receive the control-I 12 P sequence. Fortunately, the VersaBraille simply ignores it.

- ***Jerky Output Toggle***

Control-S J controls the quality of speech and the action of the spacebar during the execution of control-T, control-Y, and control-O. The default is jerky, and you can press <space> to immediately stop output and cursor movement. Toggle jerky speech off with control-S J and the speech will flow more smoothly. With jerky speech off, pressing <space> does not stop cursor movement and output immediately. Pressing <space> after entering control-O does stop cursor movement and output before the end of the text; how quickly your device responds depends on the device. Generally, it stops at the end of the sentence in which you pressed control-O. Try it and see! You may find that you want to use jerky speech (the default) when you're actively writing, and turn it off with control-S J when it comes time to review long sections of text.

When your jerky speech is toggled off, and you are searching for a specific sentence using control-O, use control-X to skip ahead to the next sentence. When you enter control-X in this situation, your voice device skips over the sentence it is currently speaking and reads the next sentence.

Pressing any key while the Echo's talking shuts it up and immediately passes that keystroke to the Apple. However, this feature may take some getting used to in BEX's Editor--see Learner Level 10 for details.

- ***The DP-10 Toggle***

The DP-10 large print screen display device cannot show the delete key checkerboard, which is used in screen modes W and N to show control characters in your text. Enter control-S D, D for display, to change the character that represents control characters on the screen. Control-S D changes it from the delete checkerboard to the at-sign, which the DP-10 can display. The DP-10 large print screen also cannot show BEX's HI-RES screen. Change your screen mode before using the DP-10 when you use the HI-RES screen mode as your default.

Part 7: Troubleshooting

A few problems are listed here. For further help, see Section 13, Advanced Data Recovery Techniques, and Learner Level Section 4, Part 14, Troubleshooting Editor Problems.

- Problem: I'm using Echo speech and things seem nutty. When I move my cursor with speech, I hear strange numbers and letters like "29 P" between each word.
- Cause: You've copied a different version of TEXTALKER onto your BEX disk than the version that came with it.
- Solution: Quit the Editor. You may have to use Control-Reset and the RUN 999 routine to save your data. If you still have it, copy our version of TEXTALKER back onto the BEX disk.
- Problem: I just accidentally deleted some text that I need.

- Solution: As long as you have not moved from the current page with control-P or quit BEX, it is possible to recover text deleted in error. Press Control-Reset and then type RUN <CR> and your data will be restored. See Section 13 for more help.

6 Print and Multi-Function Print

At the Learner Level, you used option P - Print to send formatted text to inkprint and large print printers, voice devices, and brailers. At the User Level, you have access to a new option on the Main menu: option M - Multi-function print. You can use either Print or Multi-function print to send formatted text to any printer you defined in your configuration. Section 3 describes how you define any of the eight printer classes in your configuration. There's also a lot of useful background information about printers in the BEX Interface Guide. This Section discusses using braille previewers, continues the discussion on the Review class printers, and introduces some concepts of printing without a format. Samples of using a Review class printer in Section 7.

For both Print and Multi-function print you use the chapter selection methods described in Section 4 to give BEX a list of chapters to print. Multi-function print allows you to do three things: specify which output page printing starts with; make multiple copies of chapter or chapters; and print the data in one BEX page.

The last step in both Print and Multi-function print is specifying where the text goes. You have many more choices here.

Part 1: Specifying Which Printer

At the User Level, you can direct text to more devices. As always, enter a question mark followed by <CR> at the `Which printer:` prompt to be reminded of the printers you configured:

Main Menu: P

Print

Drive or chapter: 2 <CR>
There are 2 chapters:
1 BLACK
2 WHITE
Use entire list? N <CR>
Select chapters by number
Chapter number: 1 <CR>
BLACK
Chapter number: <CR>
Which printer: ? <CR>
1 - Printer in slot 1 (80 by 58)
2 - Printer in slot 3 (80 by 58)
3 - Braille in slot 2 (41 by 25)
4 - Braille in slot 3 (41 by 25)
S - Screen output
N - New inkprinter parameters
L - Last printer parameters
Add +V for voice output
Which printer:

You can always cancel printing in progress by pressing <ESC>.

- **Numbered Printer**

Printers 1 through 4 are the printers you defined in your configuration. The first number is the printer number you enter at the `Which printer:` prompt. Braille embossers are listed as "brailleurs" when you chose them from the braille class. Review class printers are listed as "printers" and braille previewers are listed as "brailleurs." The slot for the printer is listed, followed by two numbers in parentheses, the carriage width by form length. You can tell inkprint printers and braille embossers apart from Review class printers and braille previewers by the slot numbers they are configured in: the Review class printers and braille previewers are always in slot 3. In the above example, printer 1 is the inkprint printer configured in slot 1, and printer 2 is the Review class printer configured in slot 3.

Hint! | The printer number and the slot number are *not* the same. You must use the printer number at the `Which printer:` prompt. You use

| the slot number to tell the types of printers you have configured.

- **Screen Options**

Answering S at the `Which printer:` prompt directs printing to the screen channel. Each screen has built-in values for carriage width and form length. BEX's formatter places soft <CR>s and page breaks where appropriate. S alone prints to the screen mode defined in your configuration. Add one of the 10 code letters to print to a different sized screen display. The code letters and their carriage widths and form lengths are:

- SW - 80 characters by 23 lines
- SH - 40 characters by 19 lines
- SB - 40 braille cells by 19 lines
- SN - 40 characters by 23 lines
- SL - 20 characters by 10 lines
- SJ - 20 braille cells by 10 lines
- SX - 10 characters by 5 lines
- SY - 10 braille cells by 5 lines
- SF - 5 characters by 3 lines
- SG - 5 braille cells by 3 lines

You can add voice output to any of these screen modes by entering the two letters followed by +V at the `Which printer:` prompt.

The screen is a printer with pause on form feed. When the screen is full, it pauses. Press <space> for the next screen. Press <ESC> as text is being printed to stop printing and return to the `Main menu:` prompt.

In wide screen mode, underlines are shown in reverse video. In all other screen modes, the underline is shown.

Echo: | You can only use Echo line review with the SW and SN screen displays.

When your carriage width or form length exceeds the built-in defaults for your screen display, the carriage width and form length in your text are ignored.

• **Defining New Inkprint Parameters Without Reconfiguring**

Answering N at the `Which printer:` prompt directs output to a new inkprint printer. You can use it for a new printer, or to change the parameters of an already configured printer. An N printer is always a generic inkprint printer. Just enter N <CR> at the `Which printer:` prompt and answer four questions, for which BEX supplies default values. The default values BEX uses are taken from the last printer you specified. When you define a new printer with N the first time you choose option P - Print or option M - Multi-function print after booting, BEX takes the values from printer 1 in your configuration and uses them as the default values.

Caution! | You cannot use N to define a new braille; N only works for inkprint printers.

Here's a sample:

```
Which printer: N <CR>
Old carriage width: 80
New width: 72 <CR>
Old form length: 58
New form length: 56 <CR>
Printer slot is 1
Enter new slot: 2 <CR>
No pause after form feed
Do you want to change? N <CR>
```

For these four questions, you can press <CR> to use the value listed, or enter a new value.

Last printer chosen

Enter L at the `Which printer:` prompt to use the same printer you used the last time you answered this question. When your previous printer was printer 1, for example, then L uses printer 1. The values used by the L choice are the same values that would appear as defaults for a new printer under N.

Add voice channel

At the Learner Level, you added the two characters +V to obtain simultaneous voice output with any printer choice. At the User Level, +V adds simultaneous output to the voice channel.

Caution! | The L (last printer) choice does not "remember" if you added voice output with plus V. If your last choice was printer 3+V you enter L+V for the same result.

Print to voice channel alone

When you have a serial voice device, you have three ways to direct output to it: You can describe your voice device as a class V printer in your configuration. Or you can enter +V alone at the `Which printer:` prompt. Or you can define a new printer with N.

Part 2: Option M - Multi-Function Print

This option adds three features to option P - Print. These new features are the ability to start printout on a specific page, to print multiple copies, and to print a single BEX page from a chapter. There are three more prompts after you specify the chapters to print. Which prompts you receive depends on how you answer them.

Hint! | Multi-function print adds a form feed at the end of printing, even when you make only one copy. When you want a document to roll out of your printer after printing, use Multi-function print instead of option P - Print.

- ***Restarting a Printout on a Specified Page***

This feature is handy if you have some mechanical trouble during a printout, for example you're printing a 15-page print document, and your ribbon breaks on page 9. After you choose Multi-function print, specify your chapter or chapters:

Main Menu: M
Multi-function print
Drive or Chapter: BLACK
Drive or Chapter: <CR>
Restart printout on output page 1 9 <CR>
Which printer:

We wanted printout from page 9, so we entered 9 <CR> instead of pressing return to accept the default 1 answer. BEX prompts: Which printer: and your choices are the same as with option P - Print.

BEX "pretends" to print the first 8 pages, and then actually sends data to your printer at the top of page 9. BEX behaves exactly like it's printing during the "pretend" portion: BEX reads data from disk and uses up time figuring out how the page should look. Restarting a printout in the middle of a very long document will be very time-consuming. When you have a braille book, for example, with 140 pages, *don't* use Multi-function print to reprint pages 75 through 140. Instead, divide your book among several BEX chapters. Start each chapter with the same format information, and use \$\$n# to set the page number to the correct digit at the start of each BEX chapter. If you don't wish to print all the pages of a document, press <ESC> to cancel the printout.

- ***Printing Multiple Copies***

This feature allows you to make more than one copy of one or more BEX chapters. When you accept the 1 default on the previous prompt, BEX asks for the number of copies, and offers an answer of one. When you change this answer to a higher number, BEX makes that number of copies of the list of chapters you've specified.

BEX automatically inserts a form feed at the end of each copy of the list, so each copy starts on a new page. For example, suppose you have a letter consisting of two BEX chapters: LETTERHEAD and JOAN and you want three copies of each. Here's how the dialogue looks:

Main Menu: M
Multi-function print
Drive or chapter: LETTERHEAD <CR>
Drive or chapter: JOAN <CR>

Drive or chapter: <CR>
Restart printout on output page 1 <CR>
How many copies? 1 3 <CR>
Which printer:

You want to make three copies of the letter, so you enter 3 <CR> and BEX prompts *Which printer:* Specify your printer, and BEX prints LETTERHEAD, JOAN, form feed, LETTERHEAD, JOAN, form feed, LETTERHEAD, JOAN, form feed.

Caution! BEX's formatter is reset each time you press P for Print. The formatter does not remember text format from the previous time you printed.

- ***Printing a Single BEX Page From One BEX Chapter***

When you only specify one chapter to print with option M - Multi-function print, BEX asks if you wish to print one BEX page from that chapter. To get one page only, enter Y in response to the question, and enter the page number you want printed:

Print single BEX page? N Y <CR>
Print BEX page: 2 <CR>
Which printer:

When you change the default to You, BEX reads the directory file of the single chapter you've specified. You have to know how many pages the chapter has. Enter a digit followed by <CR>, and BEX prints the text in that BEX page. Keep in mind that this text may occupy several output pages.

Part 3: Preview Brailers

Two preview brailers allow you to know exactly what the formatter is sending to a braille device. They function much like the inkprint Review class printer we discussed in Section 5 of the Learner Level. All three require that you have an 80-column card. The preview brailers are members of the B

- Braille class of printers. Braille code 1 is the *Braille Previewer*, code 2 is the *Braille Previewer with voice*.

- ***Braille Previewer with Voice***

Like the Review class printer, this mode is designed to work in conjunction with the Echo's line review feature. The output to both these modes is the same; the only difference are the default values used for underlining, paragraphs and page numbering.

Clearly the easiest way for a blind person to review braille format is with paper braille output, but you can't always depend on having a paper braille output device available. The braille previewer is a reasonable tool for checking the format of short sections of braille text, using the screen review functions of the Echo or SlotBuster.

For those of you who are unfamiliar with the Echo's line review functions, suggestions on which commands to use appear at the end of this section. For further help, Learner Level Section 5, Part 4 discusses Echo line review in detail.

Screen layout for the Braille Previewer with voice

The Apple 80-column screen is 80 characters across by 24 lines down. The page display starts at the left edge of the screen. The exact number of characters in the display depends on your carriage width. The first two characters are the line number, expressed in two digits: 01 through 24 at the start of each page. Then comes a vertical bar line which is the left margin *delimiter*. After the delimiter come the number of characters in your carriage width, and finally, another vertical bar line as the right margin delimiter.

The delete character appears immediately after the last non-space character on the line, and before the right vertical bar delimiter. On the screen, the looks like a square checkerboard. The Echo says "delete." This means that when you encounter a delete, the rest of the line contains nothing but more trailing spaces. Conversely, when you encounter a space character, you know that there are more real (non-space) characters on the line. When the first character in the line is a delete character, then that line is blank in your final output. For example, when line 09 is blank, and you have the Echo in

All punctuation mode, the Echo says: "zero nine vertical line, delete, space" (with "space" repeated 40 times or for the rest of your carriage width). Since the Apple's screen contains 80 characters, and since four characters are used up for the line numbers and vertical bar delimiters, the maximum carriage width that fits on one screen line is 75. Since most braille devices emboss at most 44 cells per line, this presents no problems for the Braille Previewer.

Printing to Braille Previewer with voice

When you wish to preview your braille material, print your chapters to the printer number you specified as class B, number 2 in your configuration. You don't add +V for simultaneous Echo output. If you did, you would hear every line as it is displayed. What you want to do is send a screenful of data and then examine it with line review.

As each line of text is printed to the screen, the Apple speaker makes a click. A short series of clicks alerts you to a short page. When the clicks stop, you can begin to use line review. To see any lines on your current page past line 24, you press the down arrow. To get the display of your next output page, press <space>.

It's important to keep in mind that after you enter line review, all of your keystrokes are interpreted as line review commands. See Learner Level, Section 5, Part 4 for a discussion of the Echo commands with line review for further help.

- ***Braille Previewer***

Braille code number 1 is designed for use by sighted transcribers. The screen display uses the delete key to make a border around the page. The screen also has a "cheat sheet" for screen braille and dot equivalents.

You use the Braille Previewer to proofread exactly what's sent to your braille device without having to emboss it. You configure the previewer with the same carriage width and form length as your braille device. It's more accurate than printing to the screen display SB, because it shows material at the true carriage width and form length.

The Braille Previewer is part of the TranscriBEX system. TranscriBEX uses a combination of BEX options to facilitate accurate braille transcription, following the translation and format rules as stated in *English Braille: American Edition* and the *Code of Braille Textbook Formats and Techniques*. It's a low cost add-on module to BEX. Contact us for more details.

Part 4: Printing with No Format

When you are creating inkprint or braille documents, you want your format commands executed. But when you are sending text to another computer, sometimes you *don't* want your commands executed. One of the printer classes is designed to transfer your text exactly as it appears in the Editor. This feature was originally designed to send text to the disk-based VersaBraille II. It has since proved useful in some other situations as well.

- ***Commands That Deactivate Format***

We introduce in Section 7, Part 9 a BEX format command that deactivates all subsequent format commands. When you enter \$\$z in your text, no single dollar sign format indicators nor double dollar sign format commands are executed. Your text is sent from the Apple *exactly* as it looks in BEX's Editor. The only <CR>s in your output are any *hard* <CR>s you typed; after you enter \$\$z, BEX no longer creates soft <CR>s. The effect of \$\$z lasts until the formatter encounters the \$\$d reset to default command or until you reload the formatter.

A second command provides a middle option between \$\$z and printing with full format. \$\$l0 (lowercase *l* digit zero) sets no soft <CR>s at the end of a printed line, and turns paragraph (\$p) indicators into hard <CR>s. Hard <CR>s are still executed. \$\$l0 is handy when you are transferring BEX chapters to other computer systems. See Section 7 for further discussion of format commands.

- ***The Paperless Brailer***

When you regularly send unformatted text to another computer, it's a pain to always remember to type `$$z` in your text. BEX has a shortcut for this situation: you configure one printer as a class P - Paperless braille. A Paperless braille prints without format--it's as if you typed `$$z` at the beginning of your chapter; only you can't turn formatting on again like you can with `$$z`. When you print with a Paperless braille, it strips out sticky spaces, touching tokens, discretionary hyphens and discretionary linebreaks--these commands are discussed in Master Level Section 5, Part 2. When you enter `? <CR>` at the `Which printer:` prompt, after configuring a Paperless braille, it is listed as a `Printer in slot # (40 by 0)` even though it prints with no format.

Section 11 discusses VersaBrailles in detail, and explains more about sending text to the VersaBraille II through a class P - Paperless braille. BEX's class P - Paperless braille provides you with a limited terminal feature. We combine this class printer with option I - Input through slot to transfer text between Apples, Macintoshes, and IBM'S here at Raised Dot Computing.

Part 5: Printing with a Software Driver

Many printers require special software, called *drivers*. BEX large print on dot-matrix printers, and printing to Review class printers and the Cranmer Braille are three common examples. When you tell BEX to print to one of these devices, it has to read the driver software from the program disk. When the program disk is not in drive 1, BEX hangs.

When you want to print two disks worth of chapters to a printer that requires a driver, you have to do some fancy disk swapping. Specify the chapters in drive 2 first, then the chapters in drive 1. You must have the data disk in drive 1 to specify the chapters. As soon as BEX prompts `Which printer:` replace the data disk in drive one with the BEX Main side. Now press `<CR>`; BEX reads the driver program from drive 1, then starts reading text from drive 2. At this point, you can remove BEX and insert the data disk back in drive 1.

7 Format Commands

In Section 5, Part 1 of the Learner Level, we introduced the *formatter*, which BEX uses to format the text you send to a printer, braille, or voice device. In Learner Level Section 6, we explained how you enter *format indicators* to instruct the formatter to break text into lines and paragraphs, as well as introduced ten fundamental *format commands* to instruct BEX to center, underline, and create margins and tabs. With this basic information under your belt, you're almost ready to learn about BEX's many format commands. First, you need to know some more detail about how the formatter works.

Part 1: The Formatter

The formatter determines output for three options: on the Main Menu, options P - Print chapters and M - Multi-function print; and on the Second Menu, option W - Write chapters to textfile. Option W is like printing to a file on disk--more details appear in Section 10. The formatter uses the carriage width to assemble one line of text at a time. When you print to SW, for example, the formatter uses a carriage width of 80 to build each line. When you print to SH, the formatter uses a carriage width of 40. At the Learner Level, you only used the carriage width and form length that you defined in your configuration. In Section 6, we mentioned that you can define carriage width and form length for generic inkprint printing without reconfiguration by answering N at the `Which printer:` prompt. In Parts 4 and 6 of this Section, we describe how you can change carriage width and form length within a document using format commands.

All format indicators and format commands are instructions to the formatter. The formatter is quite literal minded, and only recognizes the commands when you enter them correctly. To give you more flexibility, the formatter lets you instruct it to do some quite strange things. It's even possible to tell the formatter to do truly stupid things--but we try to warn you about these. Part of becoming skilled at formatting text with BEX is recognizing which format command to use in a particular situation. Because BEX has many

format commands, you frequently have more than one choice in a given situation.

The formatter's memory is persistent, which makes printing large amounts of text easy. You can create *set-up chapters* that contain most of the formatting instructions, and use the set-up chapters again and again. An example of this was the LETTERHEAD chapter M. J. Prude used to start off all his letters.

In this Section, we review a number of concepts and commands we talked about in the Learner Level, as well as introduce new format commands.

● ***Syntax for Format Indicators and Commands***

Format indicators are all four keystrokes long: space, dollar sign, lowercase letter, space.

Format commands vary in length. They start with two dollar signs, followed by a lowercase letter or letters, sometimes followed by a number--represented here with . When entering format commands with a braille keyboard, you must use the *ed* sign, dots 1-2-4-6, for the dollar sign, and Nemeth or "dropped" numbers for the value shown by # (the number sign).

The \$\$ command must be immediately preceded by one of five choices:

- 1. A space
- 2. The trailing space of a new-line indicator (\$l) or paragraph (\$p) indicator
- 3. A <CR>
- 4. The beginning of a BEX page
- 5. Another \$\$ command

● ***The Importance of Spaces***

We recommend that you precede and follow all format commands with a single space or hard <CR>. The single space *following* a format command is "thrown away" as BEX prints. When you enter format commands this way, each format command is a single BEX word, facilitating cursor movement in

the Editor. More importantly, the Grade 2 and Back from Grade 2 translators must be able to distinguish between format commands, which are never translated, and words, which are translated. Compare what happens to this text:

\$\$ccomputer literacy \$\$c<space>computer literacy

when it's sent through the Grade 2 translator. (Do it and see!). When you don't type a space between the \$\$c centering command and the word *computer*, the translator does not place the *com* sign, shown in screen braille by the hyphen.

- ***When Format Commands Take Effect***

All format commands must precede the text you wish them to influence. Some commands control format for one paragraph, while others influence the placement of all subsequent lines. Some commands affect placement of text on one line, while others control when the formatter moves to a new output page. Some commands create repeating elements that appear on every output page, and some commands' function is so bizarre that we've put them in the "miscellaneous" category.

The current line concept

The better you understand how BEX's formatter operates, the better your results. We mentioned earlier that the formatter assembles one entire line at a time. It uses the \$\$ commands you place in your text to format this current line, then hands the line to BEX to print. The line the formatter works with is defined in either of two ways: by new line commands you enter in your text, or by the soft <CR>s BEX places at the end of each line as it formats.

Because the formatter prepares each line before printing the line, commands such as \$\$p-# work. When the formatter encounters the command \$\$p-2, it places the next character two positions back on the line; whatever character was there before is gone. The formatter could not do this if the line were being printed and formatted at the same time. (See Part 5 for further discussion of \$\$p#).

However, it is precisely for this reason that you must be vigilant about putting new-line (\$l) or paragraph (\$p) indicators before commands that

take affect on a new line; commands such `$$r`, `$$h`, and `$$c`. You also must be vigilant about creating new lines *after* commands that need them, such as `$$ml#` and `$$mr#`. When the formatter encounters these commands, it makes note to execute them on the next line it formats. It does not reformat the current line it is working on, but waits for the next line.

You must also remember to place a new-line (`$l`) or paragraph (`$p`) indicator before a line which is centered. Otherwise, when the formatter encounters the `$$c`, it will go to the beginning of the entire line it is preparing, and center that text, instead of taking just the text that appears after the command.

Position zero

When we talk about the placement of the characters on a printed line, we will be referring to the character's *position*. A position is equivalent to the width of one printed character. *Position zero* is the leftmost point that the printhead on your printer can go to. Therefore, if your carriage width is 72, your printer will print characters on positions zero through 71. Position zero can move around on your paper; its exact placement depends on how you set your printer's internal margins. Perhaps you set them through the use of automatic set-up procedures (discussed in User Level Section 3, Part 3). However, in terms of how BEX's formatter view your text, position zero is always the leftmost position on the printed page.

You're about to learn about relative commands that use a numerical value that can alter the horizontal position of text on the current line. For example, suppose BEX's formatter prints a character on position 8. Next, the formatter encounters the command `$$p15`; it then prints the next character at position 15. Part 10 gives further examples of positioning characters at a specific point on a line.

Part 2: Carriage Width and Form Length

BEX allows you to define your carriage width and form length in three ways: in your configuration; when you answer N at the `Which printer:` for a new printer when printing; and within your document with format

commands. Whatever way you choose to specify these values, they provide the fundamental image of the page that the formatter uses. Many other format commands position text relative to the carriage width and form length. We draw your attention to this relationship in the detailed descriptions that follow.

- ***Establish Carriage Width and Form Length When You Configure***

We recommend that, where possible, you establish your carriage width and form length when you configure your printer. While you are answering the questions in the printer section, you can get suggested values for carriage width and form length by entering <CR> alone at these prompts. Since BEX has room to store four printer specifications, you can establish four printers that are all the same physical printer, but use four different sets of values for carriage width and form length. You can also establish carriage width and form length for an inkprint printer by entering N at the `Which printer:` prompt.

The advantage of these two approaches is that the carriage width and form length remain constant throughout your document. It makes your documents more "portable" between print and braille and between different printers.

Because many other format commands use your carriage width and form length as reference points, changing your carriage width or form length within a document can interfere with the execution of other format commands. When you do use format commands to establish carriage width and form length, always place them at the very beginning of your list of chapters to print. For predictable results, never change carriage width or form length in the middle of a document!

The formatter and carriage width

The formatter places *soft* <CR>s in your text automatically, when it recognizes that a word won't fit in the number of characters allowed on the line. The maximum number of characters on any line is determined by the interaction of the carriage width with the format commands that control paragraph

indent, and left and right margins. We recommend that you set your carriage width for the maximum number of characters you want to appear on any line, and then use margins to temporarily narrow the carriage width.

The formatter and form length

BEX's formatter uses the form length value to automatically break text into output pages. At the start of each page, the formatter starts counting every <CR> it sends out. When the number of <CR>s reaches the form length, the formatter sends out a *form feed* command, or control-L, to the printer. A form length of 55 establishes 55 possible <CR>s for the page. Since the formatter counts every <CR>, lines that do not contain text are still counted as part of your form length. With double spacing set by \$\$l2, (lowercase *l* digit two) text may appear on lines 1, 3, 5, ... 51, 53, 55. Exactly where line 1 appears on the output page depends where you set top-of-form on your printer. Remember that if you use the \$\$mt# top margin command, the formatter moves line 1 down # lines.

The formatter knows that a heading (begun with either \$\$c or \$\$h) all alone on the bottom of a page looks stupid, so it "looks ahead" to make sure that there's room for at least the first line of the next paragraph before executing the \$\$c or \$\$h command. When there isn't enough room, then the formatter sends the form feed message early. The amount BEX "looks ahead" depends how you set paragraph line spacing with \$\$s#.

A form length of zero tells the formatter to not bother counting <CR>s. If your printer is smart enough, it may be able to break pages itself, and you would want to enter \$\$f0 into your text. Some voice devices prefer a form length of zero. However, giving BEX's formatter a form length of zero disables some of BEX's features. For example, you cannot use BEX's page numbering or running headers or footers with a form length of zero. BEX will not prevent centered headings at the bottom of a page. Even when your printer has automatic page breaking, think twice before using a form length of zero.

When you do not use a form length of zero and your printer automatically breaks pages, you must prevent conflicts between BEX's page breaks and your printer's page breaks. To let BEX break your pages, make sure your printer is not set for a page length which is less than BEX's form length.

Part 7 discusses how to create running headers and footers in great detail. BEX can only execute these types of commands only when your form length is four or more lines.

- ***Carriage Width and Form Length in Printing to the Screen***

All screen modes have their own built-in defaults for carriage width and form length (listed in Section 6, Part 1). You enter `$$w#` and `$$f#` commands to establish carriage width and form length different than what you configured. It's quite possible that these values exceed the built-in defaults for a particular screen mode.

How BEX handles this situation depends on which screen mode you print to. (It doesn't matter whether you add voice output with *plus sign*, letter *V*.) When you print to the 80-column *W* screen or the 40-column *N* screen, BEX's formatter executes any `$$w#` and `$$f#` commands you have placed in your text. When lines of text are longer than the screen's width, one line of output is shown as two lines on the screen. For all other screen modes, BEX always uses the default carriage width and form length for that screen, and ignores any `$$w#` or `$$f#` commands in your chapter.

The screen modes behave as described whether you print to *S+V* (screen with voice) or to screen alone.

Part 3: Formatting Paragraphs

By this point, the paragraph (`$p`) indicator is quite familiar. We've stressed how useful it is, especially since the formatter formats paragraphs differently for print and braille.

- ***Paragraph Indent: \$\$i#***

As we documented in Learner Level Section 6, paragraph indent is controlled by `$$i#`. The value # always moves relative to the existing left margin. Default value for print is `$$i5`; for braille it is `$$i2`.

When you have the default left margin of zero: `$$i5` means the first character of every paragraph appears at position 5; `$$i-5` means the first five characters of every paragraph don't print; and `$$i0` means every paragraph starts at *position zero* (block style paragraphs).

Two reasons you may want to use `$$i#` are to create outdenting or to create block-style paragraphs. A negative indent combined with a positive margin makes "outdenting" (detailed in Learner Level, Section 9, Part 4). A zero indent makes "block style" paragraphs; usually you increase the paragraph line spacing to at least triple-spacing to make your paragraphs stand out.

When you establish a left margin at position 10: `$$i5` means every paragraph starts at position 15; `$$i-5` means every paragraph outdents, starting at position 5; and `$$i0` creates block-style paragraphs, with the first line of each paragraph even with the left margin.

- **Paragraph Spacing: `$$s#`**

As we said in the Learner Level, paragraph line spacing is controlled with `$$s#`. The default for print is double spacing, `$$s2`; for braille it's single spacing, `$$s1`. Place `$$s#` before the (\$p) indicator you wish to affect. Line spacing at paragraphs is independent of the line spacing set with `$$l#`. The three commands `$$l`, (lowercase *l* digit one) `$$s2` `$$i0` make block style paragraphs for business letters.

- **Centering: `$$c`**

As we said in the Learner Level, center text by placing a `$$c` before the first word you want centered. We advise you to make your centered text into paragraphs. Place (\$p) before and after centered text to make one blank line before and after the heading. Turn off centering with (\$p), (\$l), or <CR>. When you want to center more than one line, *and* you want to control where these lines break, you must format each line separately. To do this, you must

separate the lines with a new-line (\$l) or paragraph (\$p) indicator, or a hard <CR>, and type \$\$c at the start of each line:

\$l \$\$c The Story of Helen Keller \$l \$\$c and Anne Sullivan Macy \$l

When you have a heading longer than about 20 characters less than your carriage width, the formatter divides the heading between two or more lines. However, it tries to fit as many words as possible on the first centered line, and then centers the remaining words on the next line. The result may be heading with uneven line lengths. If this is unacceptable to you, divide the lines manually, as above.

- **Headings: \$\$h**

As we documented in the Learner Level, use \$\$h to center and underline a heading. Always precede and finish a heading with (\$p). When printed to a braille, \$\$h and \$\$c are executed exactly the same. When you do not want a paragraph (\$p) indicator after your heading, will turn off the underlining, and (\$l) will stop the centering. When you want to center and underline more than one line, *and* you want to control where these lines break, you must enter \$\$h at the start of each line.

- **Flush Right: \$\$r**

\$\$r places all the text on the current line against the right margin. Turn off flush right with a (\$p), (\$l), or <CR>. When you want to place more than one line against the right margin, *and* you want to control where these lines break, you must enter \$\$r at the start of each line.

- **Why BEX doesn't justify left and right**

Fully justified paragraphs are standard in typeset text. Typesetting software contains many routines that make fully justified text readable. Most microcomputer programs that perform left-and-right justification achieve this task by forcibly distributing the extra white space between words and letters. This type of justification coupled with a fixed-space printer is difficult

to read and frequently looks quite ugly. To make justification look good, you need three things: a printer that supports both proportional spacing and the ability to squeeze letters closer together in small increments, and excellent automatic hyphenation software. BEX does not do these things; we've chosen to make BEX powerful in other areas.

Part 4: Controlling Horizontal Format for Many Lines

For a discussion of tabs, see Part 5, Controlling Horizontal Format for One Line.

- ***Carriage Width: \$\$w#***

\$\$w# allows you to specify carriage width with a document. The value # is measured in characters; its range is 2 to 240. Entering a \$\$w# command clears any left or right margins you've previously established with any \$\$ml# or \$\$mr# commands. The \$\$w# command is suppressed when printing to any screen except SN and SW.

- ***Left Margin: \$\$ml#***

As we said in the Learner Level, \$\$ml# (lowercase *m* lowercase *l*) changes the left margin. The left margin establishes where every line starts, except for the first line of a paragraph. The first line of each paragraph is controlled by the interaction bet \$\$i# and \$\$ml#. The value # can be a number (with or without a plus or minus sign) or the asterisk character. The default value for left margin is \$\$ml0 for all printers.

When the number has a plus or minus sign, the value # is the *relative position* measured from the existing left margin. When the left margin is zero, \$\$ml+10 moves the left margin 10 to the right, to position 10. Enter \$\$ml+10 again, and the left margin is at position 20. Enter \$\$ml-8 now, and the left margin moves eight characters to the left to position 12.

When the number has no plus or minus sign, the value # sets the left margin at *position* . For example, when you enter `$$ml20` into your text, the left margin is set at position 20. You can establish a "standard" left margin with a plain number, for example, `$$ml10`, and then later make temporary changes with relative margins.

Use the asterisk character to establish a left margin at the current position. `$$ml*` can be a timesaver. Suppose you want to outdent a word from its definition. On a typewriter, you would establish a tab stop after you typed the word, then tab over to that tab stop for every line in the definition. With `$$ml*`, you can avoid that hassle:

Keyword - `$$ml*` Explanatory text that runs on without having to use tabs.
produces:

Keyword - Explanatory text that runs on without having to use tabs.

Every time you use a `$$ml#` command, you clear the previous left margin. The new left margin established by `$$ml#` takes effect at the beginning of the next new line after the command. This new line could be created by a soft `<CR>` from BEX's formatter, or an explicit new line triggered by (`$p`), (`$l`) or a hard `<CR>`. Place `$$ml#` commands immediately before (`$p`) or (`$l`) indicators, so you can control exactly where the new left margin begins. See more examples of left margin commands in Part 10.

- ***Right Margin: `$$mr#`***

The right margin command is the mirror image of the left margin command. The value # can be a number with or without a plus or minus sign. Every time you use a `$$mr#` command, you clear the previous right margin. The default value for right margin is `$$m0` for all printers. Place `$$mr#` commands after (`$p`) or (`$l`) indicators. The right margin command affects the formatter's behavior at the end of the line, so it takes effect immediately.

Measure the value # in positions *from* the right margin. When `$$mr#` has a plus sign, the right margin increases in *positions* to the left, and you have greater "white" or empty space along the right edge of the paper. When `$$mr#` has a minus sign, the right margin decreases to the right, and you have less empty space on the right edge of the paper.

For example, with a carriage width of 72 and a right margin of `$$mr10`, no characters appear between positions 62 and 72: so you have ten more positions of empty space along the right edge of the paper. Enter `$$mr+5`, and the right margin increases five positions to the left, so that no characters print between positions 57 and 72; you have an additional five positions of empty space along the right margin. Enter `$$mr-9` now, and the right margin decreases nine positions to the right, so no characters print between 66 and 72; the right margin has narrowed. Enter `$$mr0`, and the full carriage width is available for printing.

- ***Using left and right margins together***

Here's a quick example of using left and right margins. You are writing a term paper which is double spaced, with no margins. Any long quotation should be single spaced with left and right margins increased by 5. Immediately before the (`$p`) at the start of the quotation, enter: `$$l1` (lowercase *l* digit one) `$$ml+5 $$mr+5`. Immediately before the (`$p`) at the end of the quotation, enter: `$$l2` (lowercase *l* digit two) `$$ml-5 $$mr-5` to change the margins and line spacing back to their previous value.

- ***Top Margin: \$\$mt#***

`$$mt#` creates a top margin in your printout. The value `#` is equal to the number of soft `<CR>`s added after a form feed. The default value is zero. The `$$mt#` command is different from all other format commands: it does not change the number of lines of text printed per page. What changes is where the text starts on the page. You can't use a negative number with this command.

- ***Line Spacing: \$\$l#***

`$$l#` changes how far apart lines are on the page; it defines how many `<CR>`s BEX sends to the printer when moving to a new line. That new line could be one you explicitly requested one by typing (`$l`) or a hard `<CR>` in your chapter. The new line could also be a soft `<CR>` BEX creates as it fills up the

carriage width. The default value is `$$$l` (lowercase *l* digit one) for both print and braille. Line spacing `$$$#` and paragraph spacing `$$$s#` act independently. It's perfectly possible to instruct the printer to triple space all lines, but only single-space at paragraphs, by entering `$$$l $$$s3`. The result would be quite hard to read, however.

`$$$i0` (lowercase *l* digit zero) sets *no* soft `<CR>` at the end of a line, and turns a hard `<CR>` into a space. BEX still sends `<CR>`s for each paragraph (`$p`) indicator, unless you also set `$$$s0`. Most other word processors use hard `<CR>` to mark the end of a paragraph. When you type `$$$i0 $$$s1 $$$i0` in your document before printing to another computer, you create text that's easy for another word processor to use. `$$$i0` is the default for option W - Write chapters into a textfile on the Second Menu--more details are in Section 10.

Caution! | Do not use `$$$i0` when printing to the screen or to a Review class printer. `$$$i0` interrupts the default margins and causes BEX to crash. You may have to reboot.

Part 5: Controlling Horizontal Format for One Line

• *New Line*

There are three ways new lines are created in your output. The formatter places soft `<CR>`s in your text automatically, when it recognizes that a word won't fit in the number of characters left in your carriage width on a given line. The maximum number of characters on any line is determined by the interaction of the carriage width, margins, and paragraph indent commands.

You can force a new line two ways: the new-line indicator (`$l`) or the hard `<CR>`. While longer, the (`$l`) indicator is less ambiguous than a hard `<CR>` when encountered in paperless braille.

• *Tabs*

Set tab stops with `$$t#`. The value `#` can be a number (with or without a plus or minus sign) or the asterisk character. Examples of simple tabs appear in the Learner Level, Section 6, Part 3. Examples of using `$$t#` when the value `#` is positive or negative appear in this Section, Part 10.

You can set tab stops at the beginning of a chapter, or directly before where you need them in the chapter. Clear all tab stops at once with `$$tc`. It's a good idea to enter `$$tc` immediately before you establish new tabs; it ensures that only the latest tab values are in use.

Move to the next tab stop with `($$)` which is always four keystrokes. The leading and following spaces are integral to the command. Part 10 discusses using tabs.

- ***Immediate Horizontal Positioning: \$\$p#***

Use `$$p#` to place text at a specific position on the line. The `$$p#` commands are a much faster alternative to tabs or margins, and are handy for filling out forms. You can use `$$p#` to place text to the left outside of your current left margin; you can also create precise charts and diagrams. The use of `$$p#` and `$$t#` is extensively demonstrated in Part 10.

The value `#` can take two forms: it can be a number preceded by a plus or minus sign, or it can be a number from your carriage width, that is, from zero to one less than your carriage width. When the value `#` has a plus or minus sign, the formatter moves the text right or left `#` number of positions over from its current position on the line.

Determining exactly how many positions the formatter moves over is a bit tricky to understand. Suppose you want to place five blank spaces between the words *one* and *two*. You might think that typing `one $$p+5 two` would do it, but try it and see: you get six spaces between! That's because when BEX's formatter encounters the `$$p+5`, it has already printed the *e* in *one*. It is not until after it has printed the space *following* the *e* that it encounters the `$$p+5` command. Only then does it count the five positions over to place the *t* in *two*. The result is that you get six spaces between *one* and *two* instead of five:

Use `$$p#` to place text at a specific position on the line. The `$$p#` commands are a much faster alternative to tabs or margins, and are handy for filling out forms. You can use `$$p#` to place text to the left outside of your current left margin; you can also create precise charts and diagrams. The use of `$$p#` and `$$t#` is extensively demonstrated in Part 10.

The value `#` can take two forms: it can be a number preceded by a plus or minus sign, or it can be a number from your carriage width, that is, from zero to one less than your carriage width. When the value `#` has a plus or minus sign, the formatter moves the text right or left `#` number of positions over from its current position on the line.

Determining exactly how many positions the formatter moves over is a bit tricky to understand. Suppose you want to place five blank spaces between the words *one* and *two*. You might think that typing `one $$p+5 two` would do it, but try it and see: you get six spaces between! That's because when BEX's formatter encounters the `$$p+5`, it has already printed the *e* in *one*. It is not until after it has printed the space *following* the *e* that it encounters the `$$p+5` command. Only then does it count the five positions over to place the *t* in *two*. The result is that you get six spaces between *one* and *two* instead of five:

`one $$p+5 two`

produces:

`one<space><space><space><space><space><space><space>two`

The solution is to use `$$p+4` instead:

`one $$p+4 two`

produces the five spaces you want:

`one<space><space><space><space><space>two`

You must be careful when using a minus sign with the value `#`. When your number moves left over previous text, that text is not printed--see below for an example. `$$p-#` or `$$p+#` can be quite helpful when you are creating columns, and is faster than tabs--we show a neat trick in Part 10.

`$$p#` is a faster and more certain alternative to typing spaces. For example, when you wish to set off text with a number or heading to the left, use `$$p#` to position the heading. Suppose you have a left margin of 10. To set off your paragraphs, you type:

`$$p5 1. $$p10 A definition, perhaps. $l This text is at the left margin.`

This produces:

1. A definition, perhaps.

This text is at the left margin.

Because you set `$$p#` at the left margin (10), the rest of the paragraph aligns underneath the *A* on the first line.

When you don't use a plus or minus sign before the value # in `$$p#`, the formatter places the next character on that position on the line. In this case, the value # can be any number from zero to one less than your carriage width. You can use `$$p#` to place text outside your margins. In the example above, we used `$$pen` to place the number one outside the left margin, but within the carriage width.

`$$p#` overwrites your text under two circumstances. When you have a `$$p-#` command which places the next character of your text to the left of your current position, you lose # number of characters. For example, if you type `$$p-5`, you lose the five characters of your text previous to the `$$p-5` command.

As we mentioned in Part 1, BEX's formatter interprets the `$$` commands to build up the *current line*, then sends that line to the printer. It's possible for the value # in `$$p#` to be less than the number of your current position on the current line. If this is so, then the text between the current position and position # is overwritten as BEX builds the current line. For example, you type:

```
$l Educational Goal $$p10 Bachelors Degree by 1993
```

you get:

```
EducationaBachelors Degree by 1993
```

When BEX encountered the `$$p10` command, the current line contained the characters *Educational Goal*. The final *l* was at position 15. At this point, the value 10 in `$$p10` is less than the current position on the line, so the characters between position 10 and position 15 are overwritten.

Here's an example which illustrates the difference between the three types of values for `$$p#`: Suppose you have a sentence you want on a specific place in your printout. When you want to be absolutely sure of where it will be in the final printout, you start a new line, and use the `$$p#` command without a plus or minus sign:

```
$l $$p10 New words positioned.
```

places the words starting with the *N* in *New* on position 10 on the new line. When you use a plus sign, the formatter moves the words 10 positions to the right of the previous word:

Last of your text. `$$p+10` New words positioned.

produces:

Last of your text. New words positioned.

When you use a minus sign, the formatter moves the text to the left:

Last of your text. `$$p-10` New words positioned.

produces:

Last of yNew words positioned.

As you can see, some of the previous text has been overwritten!

Part 6: Moving to a New Page and Controlling Page Break

- ***Form Length: `$$f#`***

There are three ways to define form length: in your configuration; with a New printer when printing; and with the `$$f#` command in your text. The value # is measured in lines, and may range from 0 to 126. A `$$f#` command in your text overrides the form length from your configuration or from a new printer; additionally, a `$$f#` command cancels any of the commands that establish page numbers, and running headers and footers on the page (described in Part 7). The `$$f#` command is ignored when printing to any screen modes except SW or SN. When the value # is four or less, then none of these commands can be executed. A form length of zero tells the formatter to not bother counting <CR>s. Use the `$$f#` command very carefully: it should always be one of the first elements in your chapter. See Part 10 for examples.

- ***Advance to Specific Line: `$$a#`***

Use the `$$a#` to advance immediately to the start of line # on the output page. The value # must be greater than your current line and less than your form

length. The value # must be a plain number; it can't have a minus or plus sign. It's your responsibility to coordinate the value # with the lines where text normally appears, based on the interaction of your form length, line spacing, and line spacing at paragraphs.

- ***Form Feed Indicator: (\$f)***

The form feed indicator is four keystrokes: <space>, dollar sign, lowercase *f* <space>. Similar to the paragraph indicator, we show this as (\$f). When the formatter executes (\$f), the printer immediately advances to the top of the next page.

Caution! The (\$f) indicator can create blank pages! When (f) is the last command in your print-stream, *and* you have page numbering, running headers, or running footers, then the last page of your document is blank except for the page number, header, or footer. Only use (\$f) when you want a form feed, no matter what.

Option M - Multi-function print automatically adds one form feed to the *end* of the list of chapters to print. If you prefer to have the printer itself advance the sheet for you to tear off, you can print one copy using Multi-function print instead of Print, and let BEX give the printer the form feed. This is just the same as when you press the form feed button on your printer after printing text. Or you can add the commands (\$f) or \$\$vn (see below) at the end of your document. These steps, however, are not necessary when you have running footers or a bottom-line page number, since these always finish on the bottom line of the final page.

- ***Discretionary Page Breaks***

BEX's three page break commands give you three different levels of control over your page breaks.

Immediate: \$\$vn

- ***When you wish text to start at the top of a new page, enter `$$vn` at the beginning of the text. When the formatter executes , it sends out a form feed unless a form feed would create a blank page.***
-

Variable: `$$vl#`

`$$vl#` tells the formatter to execute a form feed only when there are less than # <CR>s remaining on the page. When there are at least # <CR>s left, then the formatter doesn't do anything with the `$$vl#` command. When there are fewer than # <CR>s left, then the formatter moves to a new output page.

The formatter takes into account *every* line in your form length, whether or not text is printed on that line. When your text is double-spaced, or when you have skipped lines with `$$vs#`, you need to count the blank spaces between the lines as well as the lines with text. When you have a running footer, you need to count the lines involved with that as well.

Use the `$$vl#` command to ensure that a segment of text appears entirely on the same printed page. `$$vl#` is useful for charts, tables, and multiple-line headings. For example, when you want to prevent a 15-line chart from being broken between two output pages, type `$$vl15` (lowercase *v* lowercase *l* digits one five) before the first characters in the chart. When you print the same chart with lines double spaced, then you type `$$vl30`, since the value # is the number of <CR>s remaining in the output page.

We mentioned that the formatter automatically "looks ahead" when executing a `$$c` or `$$h` heading. However, the formatter is no genius, and it doesn't check to see if the *next* line also contains `$$c` or `$$h`. This means that multi-line headings that you format manually can sometimes appear alone at the bottom of the page. Use the `$$vl#` command to get around this problem-- see Part 10 for a sample.

When BEX executes `$$vn` and `$$vl#`, the formatter creates the new output page no matter how many characters are in the current line. For example, when you have:

The bureau's `$$vn` ten criteria for grading papers ...

BEX prints the two words *The bureau's* followed by one <CR>, then moves to a new page to start printing *ten criteria* at the left margin of the new page's line one. This looks pretty stupid, so you don't want to enter `<CR>` or `$$vl#` in the middle of a line. Usually, you want to enter `$$vn` or `$$vl#` right before an explicit new-line or new paragraph.

- ***Variable at end of line: \$\$vi#***

The `$$vi#` command is not as immediate as `$$vn` or `$$vl#`. The `$$vi#` command first finishes the current output line, then moves to a new page when there are fewer than # <CR>s left on the current output page. Use `$$vi#` when you want to keep the end of one paragraph together with the start of the next paragraph. Similar to `$$vl#`, the value # in `$$vi#` must include every possible <CR> left on the page. When your text is double-spaced, you must include the number of blank lines as well as the number of printed lines in calculating `$$vi#`.

But unlike `$$vl#`, you don't have to put `$$vi#` *exactly* where you want the new page break. A good place to use `$$vi#` is at the end of a letter. You don't want the second page of a letter to contain just the *Sincerely, Macalaster J. Prude* complementary close. When the complementary close adds up to 5 <CR>s, then place `$$vi?` in the middle of the last paragraph. When your letter is short, then there are at least eight <CR>s left on the page, and the formatter doesn't execute `$$vi0`. When your letter is longer, the second page contains the last half of the paragraph plus the complementary close. You can use `$$vi#` whenever you want to keep text together, even when there are no explicit new lines or new paragraphs.

Part 7: Creating Headers and Footers

BEX allows you to specify particular treatment of individual lines on the page. To allow the same document to work with varying form lengths and carriage widths, BEX labels the lines on the output page with values that are *relative* to the top and bottom of the output page. The top lines are numbered 1, 2, 3, 4; the bottom lines are numbered -0, -1, -2, etc. You start with the

bottom line as minus zero. With a form length of 58, the line called *-1* could also be called *line 57*. All of the commands covered in this Part are canceled when you change the form length with a `$$$f#` command. None of the commands can work when your form length is less than or equal to four.

- ***Simple Page Numbering: \$\$\$np***

As introduced at the Learner Level, the `$$$np` command establishes print or braille format page numbering depending on whether you are printing to a printer or a braille. For printers, the page number default is the word *Page* followed by the appropriate page number centered on the bottom line of each printed page. The line above the page number does not contain text. In addition to any device you configured as a printer, BEX uses the print format when printing to the screen. A printer defined with N defaults to print format when you use `$$$np`, as does any device you configured as a printer.

When you print to a braille screen, as well as any printer configured as a braille, the page number default is braille format: the page number appears on the right-hand margin of line one, with at least three preceding spaces. As always, you can type ? <CR> at the `Which printer:` prompt to get a list of configured printers.

- ***Setting Page Number: \$\$\$n#***

When you enter `$$$np` in your text, BEX starts numbering with *Page 1* or *#a* on the first output page. When you wish to start with a different page number, use `$$$n#`, where the value of `#` is the page number you want to start with. The `$$$n#` command must *follow* the `$$$np` command. The formatter won't pay attention to `$$$n#` unless it knows that it has the responsibility of page numbering. To start page numbering at page 25, enter `$$$np`

- ***Roman Numeral Page Numbering: \$\$\$vg***

\$\$vg allows you to substitute lowercase Roman numerals for regular page numbering, up to xc (90). Enter \$\$vg immediately af \$\$np, or after a sophisticated page numbering command (explained below).

- **Skipping Lines: \$\$vs#**

Prevent BEX from printing on line number # on each page with \$\$vs#. # varies from 1 to form length. The value # may be absolute or negative (with minus sign). If absolute, then # equals the line number, starting with the top line as 1. If # is negative, you count *up* from the bottom, starting with the bottom line as -0. With a form length of 58, \$\$vs-2 skips line 56. \$\$vs-0 skips the bottom line.

As you use \$\$vs#, keep in mind how the form length and line spacing interact. With a 55-line page, \$\$vs-1 stops text from appearing on line 54. If you establish double spacing with \$\$l2 (lowercase *l* digit two), text won't appear on line 54 anyway. If you wish to establish page numbering in the running footer, you may also wish to enter \$\$vs-1 and \$\$vs-2, so that there's a second blank line between the text and the running footer.

When you use \$\$mt# instead of \$\$vs# to create blank lines at the top of a page, keep in mind that \$\$mt# does *not* use up lines from your form length, while \$\$vs# does.

- **Running Headers: \$\$vh# [TEXT] <CR>**

A *running header* is text that appears on the top of every output page. Use \$\$vh# to signal the start of text. The value of # defines which line on the output page contains the running header text. Running headers may appear on any line, but you are limited to a maximum of four running header lines: line 1 plus three other lines. For example, you can define \$\$vh1, \$\$vh2, \$\$vh15, and \$\$vh16. When you don't have a running header on line 1, then you are limited to three running header lines. You could h \$\$vh2, \$\$vh15, and \$\$vh16.

The text of the running header must be the only text in the formatter's current line. When the formatter encounters \$\$vh#, it actually says:

"remember all the text in the current output line and print it again on line number # of every output page." While we show the header definition as ending with <CR>, you can also use (\$p) or (\$l). When the running header definition is not at the very beginning of the first chapter you print, then you must precede the definition with (\$p), (\$l), or <CR>. This ensures that text before the running header definition is not "captured" and printed along with your running header. The only time you don't have to explicitly move to a new line before the header definition is when your running header definition is the first information in a list of chapters to print.

You position the running header text horizontally by including other format commands such as \$\$c, \$\$h, \$\$r, or \$\$p# in the header definition:

```
$$vh1 $$c FINAL REPORT<CR>
```

centers the words *FINAL REPORT* on line 1 of each output page. You must check to be sure that there's enough room for your header. For example, when you're printing to an 18-point large print printer, this running header definition:

```
$$vh1 $$p0 MAY MINUTES $$p40 FINAL DRAFT $l
```

won't work. BEX recommends a carriage width of 40 for 18 point large print printers. If you printed a chapter with that running header definition, you would get the words *MAY MINUTES* on the line 1 of the first page, then *FINAL DRAFT* on line 2 of the first page. Subsequent pages would just contain the words *MAY MINUTES* on line 1.

Running headers are sometimes useful in braille as well as print. BEX allows a running header to coexist with braille page numbering. To center the words *Spring Catalog* at the top of every braille page, enter:

```
$$np $$vh1 $$c Spring Catalog<CR>
```

at the start of your text before you translate it. Make sure that the translated result is short enough to fit on a 41-cell line, and does not interfere with a long page number.

In just a few paragraphs, we explain how to use the page number token to get page numbers in a running header or footer.

- **Running Footer: \$\$vf [TEXT]<CR>**
-

A *running footer* is text that appears on the bottom line of every output page. Use `$$vf` to signal the start of the footer text, which may be formatted with other format commands. The same rules apply to footers as headers: the running footer definition must be the only text in the formatter's current line. Precede `$$vf` with `<CR>`, `($p)`, `($l)`, or make sure it's the first text in the list of chapters you print. End the running footer definition with `<CR>`, `($p)`, or `($l)`.

- ***Canceling Numbered Line Commands: \$\$vo#***

Clear all commands affecting line number # with `$$vo#` (lowercase *v*, lowercase letter *o*, digit.) The value # may be absolute or negative (with minus sign). If # is absolute, then # is the line number, counting from the top line as one. If # is negative, then you count *up* from the bottom, starting with the bottom line as minus zero. With form length of 25, `$$vo-2` is line 23. The footer is `$$vo-0`.

- ***Sophisticated Page Numbering: The Page Number Token***

The following commands are designed for formatting print. RDC'S TranscriBEX software supports the complex running heads and page numbering required by the Code of Braille Textbook Format.

BEX uses the `` character for several special applications, one of which is the *page number token*. Establish a repeating line on the page (running header or running footer) that contains the page number token, the `` character. Use other format commands to place the token horizontally. When the formatter executes the repeating line containing the page number token, it substitutes the appropriate page number for the `` character. You can still use `$$n#` to set the page number to any number , and `$$vg` to make lowercase Roman numerals.

Using BEX's own "simple print page numbering" as an example, this is how you'd define it with "sophisticated" commands:

```
$$vs-1 $$vf $$c Page <DEL><CR>
```

The `$$vs-1` ensures that the line above the page number does not contain text. The `$$vf` defines a running footer, text appearing on the bottom line of every page. The `$$c` centers the text that follows: the word *Page* and then the page number itself, shown with the `` page number token. The new line command (here, a hard `<CR>`) is crucial to indicate the end of the running footer text.

You can combine more text with the sophisticated page numbering scheme, as in the following example:

```
$$vs2 $$vs3 $$vh1 $$p0 7-<DEL> $$p54 History of Poland<CR>
```

The `$$vs2` and `$$vs3` ensure that lines 2 and 3 on every page do not contain text, making blank lines that visually separate the running header from the rest of the text. The `$$vh`, establishes the beginning of a running header on line one. The `$$p0` and `$$p5`. position text horizontally on the carriage width of 72, defined in the configuration. For each page, line one contains `7-#` jammed to the left margin, and *History of Poland* jammed to the right margin. Since you can integrate the page number token into your text, it's easy to make compound page numbers, like the ones in this manual. If you include the page number token in more than one running header or footer definition, you get more than one page number on each output page. If you typed:
`$$vh1 $$p0 -- $l $$vf $$c -- $l`

at the start of a chapter, then output page 1 would contain `-1-` starting at position 0 on line 1 plus `-1-` centered on the bottom line.

Warning!

BEX assumes a page number is two digits long. When your output page number will exceed 99, don't use `$$r` to format the running header or footer line containing the `` page number token. If you did, BEX miscounts how many spaces to put in the line to make the header or footer flush right. The last character on the header or footer creeps over to the second output line. When the last character is the page number, then the last digit of the page number creeps over to the second output line. To avoid losing portions of your page number, use `$$p#` commands to position the page number token or other words in the header on the right. When your page number changes from two to three digits, decrease the `#` value in `$$p#` by one.

• **Changing Running Headers and Footers Mid-Document**

You can redefine a header or footer in the middle of your document. The only limit is that you have no more than three headers total which are on line numbers greater than one. For example, if you have running headers on lines 1, 3, 25, and 27, you may not add an additional header, but you may redefine the text for these headers.

The tricky part is putting the new header or footer definition in the right spot. When you want the old header text on output page 7, and the new header text on output page 8, you must type the new header definition after the text that appears on output line 1, page 7 and before output line 1, page 8. You can redefine a header or footer that includes the page number token. Changing the header definition does not interfere with sequential page numbering.

Example: Changing running headers

The following comments about running headers are equally true for running footers.

Suppose you're writing a report with two sections: *Overview* and *Recommendations*. You'd like a centered running header containing the section name and page number. You want the entire report numbered consecutively. Additionally, you want to omit the running header for the first page of each section.

You can define a new running head that includes the page number token in the middle of your document, which changes the text without interrupting the page numbers. This is because the *simple* page numbering command, `$$np`, restarts page numbering at 1, but sophisticated page numbering with the character does not. Here's what you enter:

```
$$vs1 $$vs2 $$vs3 $$h Section 1: Overview $l [First paragraph of Overview text] $l $$vh1 $$c Section 1: Overview - <DEL> $l [Remaining Overview paragraphs]
```

The first three commands make BEX skip lines 1, 2, and 3. Then comes the heading for the first section, centered and underlined by `$$h`. Lines 2 and 3 are skipped throughout the document. Line 1 is sometimes skipped, and sometimes contains a running header. This first running header is defined *after* the first paragraph of text, so that there will be no running header on the first output page. The `($p)` at the end of the first paragraph *is* executed (which also ensures that the running head definition begins on a new line). The `($p)` preceding the remaining *Overview* paragraphs only serves to delimit the header definition. The result is one paragraph in the output, plus one running head definition. The second and subsequent output pages contain the section title and page number on line 1. To change running header definitions mid-document, this is what you type:

```
[Final Overview paragraph] $$vs1 $$vn $$h Section 2: Recommendations $l  
[First paragraph of Recommendations text] $l $$vh1 $$c Section 2:  
Recommendations - <DEL> $l [Remainder of Recommendations text]
```

After all the *Overview* text, there are two commands: `$$vs`, `$$vn`. The command `$$vs`, suppresses the *Overview* running header for the next output page. The discretionary page break command `$$vn` creates that next output page, unless BEX was about to create a new page already. The order of these two commands is very important: entering `$$vn $$vs1` would not work. That's because when BEX is placing a running header on line 1, it does it *first thing* after moving to a new page--before BEX even notices the `$$vs1` command. When you tell BEX to skip line 1 *before* you move to a new page, then BEX knows what to do at the top of the new page, so the running head doesn't print.

After the first paragraph of *Recommendations* text, use the same trick to change the text of the running header as you did with the *Overview* section: the new running head definition is preceded and followed with `($p)` paragraph indicators. The running header definition still contains the `` page number token. The second output page of *Recommendations* contains the modified running head: the word *Overview* is replaced with *Recommendations* and the page number continues sequentially.

Part 8: Underlining

- **Basic Underlining: $\$sub$ and $\$uf$**

$\$sub$ signals underline begin, and $\$uf$ signals underline finish. You can also turn off underlining with a paragraph ($\$p$) indicator. Always enter these commands as five keystrokes: `<space>` or hard `<CR>`, dollar sign, dollar sign, lowercase *u* lowercase *f* or *b*, `<space>`. That way, the Grade 2 translator recognizes the underlining and place the italics signs where appropriate. The space or `<CR>` before and after all format commands ensure that the translator knows when the commands end and your words begin.

- **Preventing Conflict Between Underlining and Punctuation: $\$sp$**

The command $\$sp$ tells the formatter: "Stop underlining at some punctuation." $\$sp$ changes how is executed at the end of clauses and sentences. Once $\$sp$ is in effect, the final punctuation is *not* underlined in the four cases: In the following four examples, the CBC emphasis indicator shows which letters are underlined

- 1. Any characters, period, `<space>`, $\$uf$ Any characters. New words.
- 2. Any characters, comma, `<space>`, $\$uf$ Any characters, new word.
- 3. Any characters, colon, `<space>`, $\$uf$ Any characters: new words.
- 4. Any characters, semicolon, `<space>`, $\$uf$ Any characters; new words.

You turn off the effects of $\$sp$ with the $\$d$ command. At the Master Level, We introduce the *touching token* which lets you embed underlining and all other format commands within words.

Part 9: Miscellaneous Nifty Commands

- **Set Uppercase Lock: $\$su$**

When you want all caps output of your text, enter \$\$su. The formatter changes all subsequent lowercase letters in your text to uppercase as the text is sent to the printer. \$\$su does not change the case of your text itself, so your format commands still work. Some braille devices and word processors prefer all uppercase text.

- ***Stop and Beep: \$\$b***

When the formatter encounters \$\$b1 it stops output and makes the Apple speaker beep once. The printing stops before printing the line where the \$\$b appears. To resume output, press <space>. This command comes in handy when you want to change typewheels, or when you want to be alerted to the end of a printout.

Option M - Multi-function print allows you to print a range of pages, but the range always involves printing to the final page of your output. Using \$\$b1 you can print from any page to a page before the final one. For example, your text is 12 pages long, and you only want reprints of pages 2 through 5. Use Multi-function print with a Review class printer to restart printing on output page 6. Note the first few words on line 1 of output page 6, then press <ESC> to cancel printing. Edit the chapter and type \$\$b at the beginning of that text. Print your text with Multi-function print to the actual printer, restarting printing on output page 3. The printout stops before output page 6, and the Apple beeps. Simply press <ESC> to stop the printout.

- ***Zap Format Commands: \$\$z***

You can turn off the formatter entirely by entering \$\$z in your text. BEX simply sends every character in the Editor to the printer number you specify. All format commands become lifeless dollar signs, characters and numbers, and are printed along with your text. No (\$p), (\$l), or (\$f) indicators are executed. The formatter no longer breaks text between lines or pages. Hard <CR>s still reach the printer, as do any other control characters. Wake the formatter up again by entering \$\$d. Use \$\$z when you want an exact copy of your BEX pages on another device or in a textfile. You get slightly different results when you configure a device as a class P - Paperless braille; the formatter filters out some control characters. See Section 5, Part 4 for details.

- ***Reset to Default: \$\$d***

Get in the habit of placing `$$d` at the start of each document you format. (Not at the start of every BEX chapter!) Each document is contained in just one chapter.) `$$d` reestablishes the print or braille defaults: no margins; no tabs; line spacing at single space; paragraph indent and line space as appropriate for print or braille; clear any repeating element from the page (page numbering, running heads, or footer); clear zap format; clear selective punctuation underlining mode; clear set uppercase all caps output mode. `$$d` does not change the carriage width or form length, however.

Every time you choose options P - Print, M - Multi-function print, or W - Write chapters to textfile, the formatter is also reset to the default values. When you want the format commands from one chapter to also control output for subsequent chapters, create a print-stream containing all those chapters. You can scan more than one disk drive by preceding the drive number with the plus sign--see Section 4 for a full explanation of chapter selection methods.

- ***Repeat a Character: \$\$vrX***

The format command `$$vrX` repeats the character `X` to the end of the current output line as defined by whatever margins are in effect. `X` may be an underline, dash, or even the space character. Only one `$$vrX` command works on any one line. Make sure you use `($p)`, `(1)`, or `<CR>` to terminate the line properly. You may place other characters on the line using tab or positioning commands.

For example, you want to fold a braille letter in thirds to fit a standard business envelope. A line of dots 2-5, the digit 3 in screen braille, creates a good place to crease the letter. The following commands establish two running headers that divide the page vertically in thirds; the narrower carriage width fits 11 inch paper:

```
$$w32 $$vh8 $$vr3 $l $$vh18 $$vr3 $l
```

Or suppose Macalester J Prude wanted to create a decorative border on his letterhead. His inkprint printer's carriage width is 72 and his name is 18

characters long. Use half the difference between the carriage width and his name for the # in \$\$p# and place equals signs on the rest of the line like this:
\$l \$\$vr= \$\$p27 Macalester J Prude \$l

Part 10: Putting Format Commands to Work

BEX's variety of format commands let you do many things. The three examples in this Part try to give you a flavor of what's possible. We've tried to give you an understanding of how the formatter executes these commands so that you can predict their influence on your text.

- ***General Principles***

Here are some general pointers for success. The paragraph (\$p) indicator is your friend; use it often. You can easily manipulate blocks of text defined as "paragraphs" by the (\$p) indicator. You can easily control how it's executed with \$\$s# and \$\$i#. Option R - Replace characters is also a powerful tool for formatting and re-formatting text. We don't want to get ahead of ourselves-- Option R is explained in detail in Section 8--but Replace characters lets you define your *own* format commands, using the mnemonics that you're most comfortable with. The third tool that allows you to format with confidence is printing to the Review class printer. In this Part, we'll use this tool a lot.

- ***Using Tabs***

This example provides insight as to how the \$\$t# and \$\$p# commands work in BEX. It's a little tricky to understand at first, because it's not at all like you'd format text when typing or brailleing: you can only do it with a computer.

Tabs in BEX can work for you in ways that are not possible on an ordinary typewriter. You can establish tabs *relative* to a word's position on the output line, both forward and backward from the word. This is particularly useful

when you want to produce both print and braille versions of a document. Since you use *relative* `$$t#` and `$$p#` commands, BEX does some calculations for you, simplifying a tricky format situation for both print and braille.

Suppose you want to create a print version *and* a braille version of a class roster. This class roster contains two columns: the Student Name and the Seat Number.

You want to establish a tab to position the digits in the *Seat Number* column. You want to make sure this tab allows enough room for the longest student name, but the length of that name is different in print and braille. Here's how you proceed.

Type all the students' names, entering each on separate lines divided with either the new-line (`$l`) indicator, or a hard `<CR>`. Here's a list of names we cooked up:

Jane McCloskey
Andrea Higgins
Henry Mukwanago
Marvin Gardens
Victoria Bright Feather
Wanita Coombs
Daniel Rosenblatt

To see which name is longest, use control-V or print it to a Review class printer and examine it with line review.

Move to the top of your list, and press control-I to insert the format commands. Whenever you set up new tabs, you should begin with `$$tc` to clear all previous tabs. Type the command `$$tc`. Then begin a new line and type the student's name that is the longest. Here, our choice is Victoria Bright Feather:

```
$$tc $l Victoria Bright Feather
```

Do not delete this name from the list; this name is used as a reference for your column headings which will be on line one.

Establish a *relative* tab stop after this name for the second column heading. Make sure it is enough positions from the last letter in the name so that your

second column will be set off. For print, we use eight positions:
\$\$tc \$l Victoria Bright Feather \$\$t+8

Because there's a plus sign in the tab command, BEX sets a tab eight positions to the right of the last letter of Victoria's name. In print, this tab stop is at position 32.

But of course, the first line of the class roster doesn't contain a student's name, it begins with the column headings. You use \$\$p0 to position the column heading *Student Name* right on top of the *Victoria Bright Feather*. Like this:

```
$$tc $l Victoria Bright Feather $$t+8 $$p0 Student Name
```

Remember that any text positioned to the left on the line overwrites preceding text. BEX replaces the first 12 characters of *Victoria Bright Feather* with *Student Name*. When you have a margin set with \$\$ml#, use the number # to position your first column heading. For example, when your left margin is eight, type \$\$p8 instead of \$\$p0.

Since your column heading is shorter than the name it overwrites, use the repeat any character command \$\$vr to place spaces to fill up the rest of the line:

```
$$tc $l Victoria Bright Feather $$t+8 $$p0 Student Name $$vr<space>
```

Now you want to position the *Seat Number* column heading. Move to the tab stop, then type the second column heading:

```
$$tc $l Victoria Bright Feather $$t+8 $$p0 Student Name $$vr<space> $$ Seat No.<CR>
```

For your print copy, you want to use a larger number for \$\$t#. Here, we'll use +8 for print and +2 for braille. You probably want to center the second column heading over the data; use a \$\$p-# command to do it:

```
$$tc $l Victoria Bright Feather $$t+8 $$p0 Student Name $$vr<space> $$ $$p-3 Seat No.<CR>
```

This first line accomplishes two tasks: you define the tab stop for the second column *and* you format the column headings. This strategy is possible due to the way BEX prints. BEX creates each line of text by executing the format commands it encounters, in order. When BEX builds up the column heading line, it creates several versions. First, it puts all the letters in *Victoria Bright*

Feather at the start of the line. Next it encounters the `$$t+8` command. At this point, the formatter notes the current position on the line, adds 8, and stores this number in its list of valid tab stops.

Next, the formatter encounters `$$p0`, so it moves the current position back to the start of the line. The letters in *Student Name* are placed right on top of the letters of *Victoria Bright Feather*. The `$$vr<space>` command tells the formatter to put a space in any position that does not contain any other explicit character. When the formatter encounters the `($$)` move-to-the-next-tab command, it checks its list of valid tab stops. Since the relative tab stop is already established, the formatter places the words *Seat No.* at the start of the second column.

All of your lines after the first line look like this:

```
$l Jane McCloskey $$ 1 $l Andrea Higgins $$ 2 $l Henry Mukwanago $$ 3 $l
Marvin Gardens $$ 4
```

Use control-V or a Review class printer to see your handiwork.

The next step is to translate the chapter into braille with option G - Grade 2 translator. Be sure to use a different name for your target chapter, as you want both print and braille versions of this list. Use control-V or a Review class printer and line review to examine the translated chapter to see which name is longest in braille (it might be different). Here, the same name, *Victoria Bright Feather*, is still the longest:

```
,/ud5t , "n s1t ,no4
,jane ,mc,closkey #a
,&rea ,hi79s #b
,h5ry ,mukwanago #c
,m>v9 ,g>d5s #d
,victoria ,b"r ,f1!r #e
,wanita ,coombs #f
,daniel ,ros5blatz #g
```

When the longest name is different, use the clipboard to move it to the first line, in place of the longest print name.

Here's what the final print copy of your list looks like in the Editor:

```
$l Victoria Bright Feather $$t+8 $$p0 Student Name $$vr<space> $$ $$p-3
Seat No.$l Jane McCloskey $$ 1 $l Andrea Higgins $$ 2 $l Henry Mukwanago
```

3 \$1 Marvin Gardens 4 \$1 Victoria Bright Feather 5 \$1 Wanita Coombs
6 \$1 Daniel Rosenblatt 7

And your braille chapter:

```
,$tc ,victoria ,b"r ,f1!r $t+2 $p0 ,/ud5t ,"n $vr<space> $s1t ,no4 $l ,jane  
,mc,closkey $a $l ,&rea ,hi79s $b $l ,h5ry ,mukwanago $c $l ,m>v9  
,g>d5s $d $l ,victoria ,b"r ,f1!r $e $l ,wanita ,coombs $f $l ,daniel  
,ros5blatt $g
```

Here's what they look like when printed:

Student Name Seat No.

Jane McCloskey 1

Andrea Higgins 2

Henry Mukwanago 3

Marvin Gardens 4

Victoria Bright Feather 5

Wanita Coombs 6

Daniel Rosenblatt 7

And when embossed:

```
,/ud5t ,"n s1t ,no4
```

```
,jane ,mc,closkey #a
```

```
,&rea ,hi79s #b
```

```
,h5ry ,mukwanago #c
```

```
,m>v9 ,g>d5s #d
```

```
,victoria ,b"r ,f1!r #e
```

```
,wanita ,coombs #f
```

```
,daniel ,ros5blatt #g
```

Now you see how \$t# and \$p# can work together, to make the same text printable in both inkprint and braille. You only need to change two format commands to make nice print from braille (or vice versa).

- ***Examining the Q FORM Chapter***

There's a chapter called Q FORM on the BEXtras disk, which contains some format commands to add "sophisticated page numbering" to the by-now familiar QUANDARY article. Edit the Q FORM chapter to see these commands.

The first command is `$$d`, resetting the formatter to default. Next come three hard `<CR>`s. The format] sends out three `<CR>`s before any text is printed. A few blank lines at the start of an inkprint paper makes the first page more noticeable. Three new-line (`$l`) indicators would have the same effect, as would the command `$$a3`. The next command is `$$sp`, establishing selective punctuation for underlining.

Next comes `$$vh1`, signalling the start of the running head on line 1. The `$$r` immediately after places the header all the way to the right. The header contains the page number token, ``. The (`$p`) after the `` does not create an extra blank line, but does indent the next line for the beginning of a paragraph. The (`$p`) signals the end of the text for the running header.

Caution! You can use `$$r` to position running header or footer text flush right, but only when your output page number is 99 or less. When you have three or more digits in your output page number, use `$$p#` commands instead.

The page number for page 1 is commonly omitted in print documents, since the first page is readily identifiable by the extra room at the top of the page and the headings that usually appear there. This is also true for letters printed on letterhead: the first page doesn't have a page number, but subsequent pages contain "continuation lines" with the addressee and author's name and the page number.

The last two format commands in the Q FORM chapter instruct the formatter to skip lines 2 and 3 on all output pages, which separates the running head text from the bulk of the article.

Before you can print Q FORM and QUANDARY to see what the running head looks like, you have to edit QUANDARY to make sure there aren't any conflicting format commands. In fact, the first two commands do conflict: delete the `$$D` and `$$np` commands so that the commands in Q FORM can work. Also delete the hard `<CR>`, since Q FORM provides three blank lines at its start. Now, print chapters Q FORM and QUANDARY.

Use a Review class printer to examine the results

Tell BEX to print Q-FORM and then QUANDARY to a Review class printer. When the clicks stop, enter control-L to start line review. Press A, G and sit

back and listen. Lines 1 through 3 are blank. Line 4 is in all uppercase, which is how a Review class printer shows underlined text. When you're done exploring the first 24 lines, press <ESC> to exit line review, then press the down arrow to get more text. How many screens comprise each output page depends on the form length you defined for this printer.

When you press <space> for the second output page, enter line review again. Press A to hear all of line 1--there's the running head, page number 2 and all. Use the right and left arrows to get to Lauer, the first word in the running head. You can press the spacebar now to hear the horizontal position of the audio cursor. The exact position of the cursor depends on the carriage width, the number of characters in the header, and the number of characters on the screen that show format for the Review class printer.

Print to the screen

You can print Q FORM and QUANDARY to the screen in any of the screen modes. The pages break differently depending on whether you print to SW, SH, or SL. If you put the format commands \$\$w# and \$\$f# into the chapter, they do not take effect when you print to any screen but SW and SN. These two commands are filtered out to avoid conflicts with the built-in screen carriage widths and form lengths.

With 20-column screen, the text of the running head can't fit on one line. The \$\$vh, command is quite strict: if the text is too long for one line, then it's cut short. Print these chapters to a screen mode or printer with a carriage width greater than 38 characters, and the running head fits perfectly.

- ***Examining the RESUME Chapter***

The one-page chapter named RESUME on the BEXtras disk contains fairly involved format commands. Print it to a Review class printer first, then edit it to see how it's done.

This chapter starts out with two format commands that set carriage width and form length: \$\$w72 and \$\$f50. The next three commands set "block style" paragraphs. Following \$\$s2 there's a hard <CR>. This suppresses the printing of the running head on line 1 on the first page of the resume. The text of the running head is next: it starts with \$\$p0 to place it at *position zero*. Next comes

a running footer, to automatically place the *continued overleaf* message at the bottom of the first page. After the <CR> that delimits the footer, there are two "skip line" commands.

The resume has two columns. The left margin establishes the starting point of text in the second column. The major headings for the resume: *Name*, *Employment history*, *Education*, etc., appear in the first column.

The `$$$p0` precedes the actual beginning of text. So far, there has been only one hard <CR> that appears in the output, between `$$$s2` and `$$$vh1`, and this means line 1 is blank. Lines 2 and 3 are also blank, because of the `$$$vs2` and `$$$vs3` commands. So Nevin's name starts at position zero on line 4. After *Olson*, there are three commands that accomplish a lot of work without a lot of calculation. The `$$$p+5` tells the formatter to place 5 spaces between the *n* at the end of Olson and the start of the street address. The `$$$t*` sets a tab at this position, and the `$$$ml*` sets the left margin at that same position. The street address ends with (`$$$p`). The next two lines start at the left margin, so they line up right under the street address.

A (`$$$p`) indicator follows the phone number. With the paragraph indent set to zero, the text would normally start at the left margin. The `$$$p0` command positions the word *Employment* in the first column, to the left of the current left margin. The (`$$$`) moves the first employer name to the tab stop, which is set at the same position as the left margin. This line ends with (`$$$p`), so the job title is positioned at the left margin, lining up underneath the employer's name. After the date of employment, there are two margin commands, separated by a new-line indicator. Always change the left margin before a new line, and the right margin at the start of a new line. The margin commands affect behavior at the end of a line, whether the line is begun with a new-line (`$$$l`) indicator, a hard <CR>, or with a soft <CR> which BEX places while printing. If don't you enter a margin command before a command which begins a new line, you wouldn't know exactly on which line the new margin begins.

The six paragraphs in the *Employment history* section all use this same format: the narrower margins for the text underneath make the employer's name, job title, and dates distinctive on the page. At the start of the fifth paragraph, there's a `$$$vl10` command, to prevent any paragraph from being broken between the first and second output page. The value 10 is a guess. Another approach would be to print the text to a Review class printer, and see where

it breaks without any `$$vl#` commands. Then place a `$$vn` command right before the paragraph where it breaks, so that the formatter advances to the next page before starting to print it. In either case, the next page starts with running head copy and is numbered.

The section that starts with *Volunteer* shows two approaches to creating tabs. The desired result is to indent the dates five spaces. The first paragraph uses the `$$+5` command. In the second paragraph, `$$t+5` sets a tab at the same position. Note that you need the (`$$`) immediately after to move to the tab just set. That's a lot of keystrokes, it's true, but for subsequent paragraphs, it's quite economical.

Right before the last paragraph, there's a `$$vo-0` command--this clears the running footer, so that *continued overleaf* doesn't show up on the second page. While *minus zero* sounds loony, it is consistent with numbering from the bottom of the page. The line above the running footer is -1, so the footer itself is -0.

8 Replace characters

Option R - Replace characters on the Main Menu makes BEX a powerful word processor. You can automatically reformat information from other word processing programs or computers. You can define your own keyboard shortcuts for frequently typed words, phrases, or format commands. Replace characters lets you find any series of characters and change it to a different series of characters. For example, you can find all occurrences of `:::` (three colons) and change it to `Department of Public Instruction`. You can find every `<CR>` and change it to a space. You can find every appearance of `^*` (caret, asterisk) and change it to `$$ml+4 $p $$mr+4`

Part 1: The Tools of the Trade

Before we can explain the process of using Replace characters, there are five terms we must define. A *transformation chapter* is a BEX chapter containing the instructions Replace characters uses in making changes. It tells option R

what to look for and what to replace it with. You already have direct experience with a transformation chapter. At the Learner Level, you used the transformation chapter called FIX TEXT to create BEX formatting commands from the <CR>s and spaces in a chapter read from a textfile. The BEXtras disk contains several transformation chapters to help you reformat text. We analyze how one works in Part 4.

You can create a transformation chapter two ways: by writing it from scratch in the Editor, or by typing in response to the prompts Replace characters supplies. We show you first how you can directly type in the changes you want Replace characters to make, which provides insight into the structure of transformation chapters.

No matter how it's created, the transformation chapter must have a particular structure. It must contain at least one transformation rule. A *transformation rule* consists of the *find string* and the *change to string*. The *find string* is what you instruct BEX to look for in your original chapter. It's the group of characters you wish to change. When BEX locates the find string, the characters in the find string are deleted, or depart. The departing characters are replaced by the characters in the *change to string*.

You can enter any character in your find or change to strings that you can type in the Editor--any of the 128 ASCII characters. However, for each transformation chapter, you must reserve one character as the *terminator* character. The *terminator* character marks the end of your find and change to strings. The terminator character can't be a part of your find or change to strings. You use the terminator character much as you use it when you type <CR> after a chapter name. The <CR>, and the terminator character tell BEX, "OK, pay attention to what I have typed." The terminator character can be <CR>. However, at times you want to include <CR> in your find or change to strings. In this situation, specify another character as the terminator. In the examples that follow, we use # (number sign) as the terminator.

Part 2: Typing Find and Change to Strings Directly

Replace characters is on the Main Menu. When you press R, BEX prompts for the chapters you want to work with. Once you enter chapter names, BEX

prompts for target chapters. You can choose to use Replace characters in two ways: you can make modified *copies* of your original chapters, or you can overwrite your original chapters with the changed data. When you supply different names for the target chapters, you end up with transformed *copies* of your original chapters. When you use the S naming method, your improved target chapters overwrite your source chapters, saving you the effort of killing the source chapters later.

Once you finish telling BEX the source and target chapter names, BEX asks for the name of the transformation chapter. When you want to type in the changes directly, press <CR> at the Transformation chapter name : prompt. BEX then asks you to declare one character as your terminator. Now BEX starts asking for the first find string. Type in the string that contains the exact characters you want to find. Signal the end of the string by pressing your terminator character. BEX then prompts for the change to string. Here you enter the new characters you want in your target chapter. End the string with your terminator character. BEX then prompts again for a find string. Continue writing transformation rules by alternating between the Find: and Change to: prompts. When you are done with your list changes, enter your terminator alone at the Find: prompt. BEX then asks if you want to continue. When your disks are in the right places, press <CR> to accept the Y default and start replacing. When you don't want to go through the list of changes, enter N <CR> and you return to the Main Menu.

Here's what the dialogue looks like when you press R for Replace characters:

```
Main menu: R <CR>
Replace
Drive or Chapter: BLACK
Drive or Chapter: <CR>
Target chapter name: BLACK-R
Transformation chapter name: <CR>
Enter terminator: #
Find: <CR><space><space><space><space><space><space> ;#
Change to: <space>$p<space>#
Find: <CR><space>#
Change to: <space>$l<space>#
Find: #
Continue? Y <CR>
```

BEX loads the first page of the chapter into the page buffer, and executes the first transformation rule. BEX searches through the text character by character, until it finds an exact copy of the find string, then replaces it with an exact copy of the change to string.

For example, suppose you have a string of text *ABCDEFGH* in a chapter, and you want to change *EFG* to *FEG*. BEX's find string is *EFG*. It compares each group of three letters until it finds a match with the find string. It searches *ABC*, then *BCD*, then *CDE*, until it comes across the string *EFG*. Once it finds *EFG*, it replaces it with the change to string, *FEG*. The string of text now looks like *ABCDFEGH*. BEX then continues to search through the rest of the chapter for further occurrences of the find string.

BEX searches for each find string in the order they are entered into your transformation chapter. When BEX finds no more occurrences of the first rule's find string, BEX executes the second transformation rule, and so on through the list of rules.

The Apple speaker sounds for every change BEX makes. When there are a lot of changes, you hear interesting beeps, tones, and crackles. As you gain experience, the sounds tell you whether Replace characters is working correctly.

BEX goes through the entire list of transformation rules, then saves the first page of transformed text to the target chapter on disk, and moves on to the next page. BEX lists the name of each source chapter as it's completed, and then the total number of replacements that occurred.

When the changes are complete, BEX prompts: `Save transformation chapter:` providing you with an opportunity to save the list of rules on disk for later use. When you don't want to save the transformation chapter, just press <CR>. (Do not press the terminator you've used in your transformation chapter.) To save it as a BEX chapter, type a name followed by <CR>. Later you can use the same transformation chapter with other chapters. For example, you can create a transformation chapter that fixes your most common typing errors and save it on your program disk.

- ***Sample: Typing Changes Directly***

Suppose you have a chapter a friend typed into your computer for you. However, she didn't know much about BEX, and she entered a lot of <control-R> characters. She didn't use the paragraph (\$p) indicator; for each paragraph she typed two <CR>s and five spaces. She always misspelled the word *precede*, and insult of insults, she spelled BEX wrong.

Here's what you type to correct the mistakes:

Main menu: R

Replace

Drive or chapter: GOOF <CR>

Drive or chapter: <CR>

Target chapter name: ALLFIX <CR>

Transformation chapter name: <CR>

Enter terminator: #

Find: <control-R>#

Change to: #

Find: <CR><CR><space><space><space><space>&l t;space>#

Change to: <space>\$p<space>#

Find: precede#

Change to: precede#

Find: Becks#

Change to: BEX#

Find: #

Continue? Y <CR>

Chapter GOOF done

Replaced 28 times

Save transformation chapter: 1SALLY <CR>

Main Menu:

After you press return at the Continue? Y prompt, you hear the Apple speaker click for each change. When you hear no more clicks, all the errors are fixed.

You type a control character in your find or change to string the same way you enter an Editor command: depress the control key, then press the appropriate letter. This is different than entering a control character in your text in the Editor. (You don't press control-C, then a plain letter.) To delete the entire find string, enter only your terminator at the Change to: prompt. To enter space characters into a string, press <space>. To signal BEX to start replacing, enter your terminator at the Find: prompt. The next time

you want to transform a chapter that Sally typed, simply enter SALLY at the Transformation chapter name: prompt.

When you are directly typing in a transformation chapter, you can fix typing errors by using the left and right arrow keys. In some situations, you may want to include <control-U> (right arrow) and <control-H> (left arrow) as data in your find and change to strings. To do this, you give BEX a special signal: press the left arrow key at the Enter terminator: prompt. BEX responds with the message: Warning: left and right arrow keys used as data then prompts again for the actual terminator you want to use.

Caution! | When left and right arrow keys are entered into your find and change to strings, you can't use them to correct typing mistakes.

Let's examine the contents of the transformation chapter SALLY. It contains every keystroke entered between pressing <CR> at the Transformation chapter name: prompt and pressing <CR> at the Continue? Y prompt. Here's what the transformation chapter looks like:

```
#<control-R>##<CR><CR><SPACE><SPACE>&l  
t;SPACE><SPACE><SPACE># $p  
#Preceed#Precede#Becks#BEX##
```

The first character in a transformation chapter is always your terminator: here it is # (the number sign). Next comes the first rule: the find string is the one character, <control-R>, finished by the number sign terminator. The change to string consists of just the terminator, so <control-R> is deleted by this rule; we call this an empty change to string. The find string for the next rule is two returns and five spaces, finished by the number sign. The change to string is <space> dollar sign, lowercase *p* <space> number sign. There are two more rules: *preceed* changes to precede and *Becks* changes to BEX. The very last character is the terminator you supplied to the Find: prompt to start the replacement.

Part 3: Creating Transformation Chapters in the Editor

You have two choices for creating transformation chapters: type changes directly following the menu prompts, or write your chapter in the Editor. When you type changes directly, it's easy to keep track of where you are. BEX continually gives you the `Find:` and `Change to:` prompts. You can use line review to look back on previous find and change to strings.

Creating a transformation chapter from scratch in the Editor gives you much more control. You can get the Echo to voice everything simply by moving around with the left and right arrow keys. But you do have to keep track of where you are in the rule without prompting.

When your replacing task is short, then type your changes directly. When you are working with control characters, escape codes, or other complex strings, create your transformation chapter in the Editor.

Caution!

When you're first learning about Replace characters, there is some risk involved when using the same names for source and target chapters. If you don't structure your transformation rules correctly, it's possible to end up with a chapter with only some of the changes made. You can use `<ESC>` to cancel Replace characters in progress. When you cancel replacing, and you have entered the same name for the source and target chapters, some transformation rules are executed and some aren't--it depends when you pressed `<ESC>`. When you press it before BEX saves the first page, no changes are made. When you press `<ESC>` after BEX has saved the first page, then the first page has the changes. But your directory file for that chapter won't reflect the change in size. To recover fully, you have to use Fix chapters, and check your data to make sure all is well. Use different source and chapter names for the first few times you use Replace characters, and anytime you feel doubtful about the outcome after replacing.

- ***Keeping Track in the Editor***

Every transformation chapter must contain an even number of terminators: two for each rule, plus two more terminators. One of the "extra" terminators appears as the first character. When you tell BEX to use a chapter as a transformation chapter, BEX interprets the first character as your terminator.

The characters between the first terminator and the next terminator are interpreted as the find string of the first rule. The characters between the second terminator and the third terminator are used as the change to string for the first rule. Every transformation rule contains two terminators; one defines the find string, the other defines the change to string.

The other "extra" terminator is the last character, which signals the end of the list of rules, as opposed to the end of the change to string. This means that the final two characters must both be terminators. (When your last rule deletes the find string (with an empty change to string), then three terminators appear at the end of the chapter. The transformation chapter SALLY we discussed in Part 2 contains ten terminators: two for each of the four rules, plus the beginning and ending terminators.

You have two terminators next to each other in only two situations. These two situations are quite different. First, you have two terminators in a row when your change to string is empty (when the find string is deleted). In this case, the pair of terminators appears between the find and change to strings. The first of the pair marks the end of the find string, and the second marks the end of the empty change to string. In this situation, all the find characters depart with no replacement. For example, the first rule in the transformation chapter SALLY deletes <control-R>s--so there are two terminator characters after the <control-R> character: one to mark the end of the find string, and one which is the entire change to string.

The second situation when you have two terminators in a row occurs at the end of your transformation chapter. The first of the pair signals the end of a change to string, and the second marks the end of the transformation rules.

Remember this when you use option M - Merge chapters on the Second Menu to combine two transformation chapters. At least three terminators appear together where your first source chapter and second source chapter meet. Delete the extra terminators from your new, larger transformation chapter before you use it. If you don't, then Replace characters would only execute the changes up to the "seam," since the two terminators stop the process.

Counting terminators in a transformation chapter

When you Merge chapters, or when you create a transformation chapter in the Editor, you may want to count the terminators in the chapter to make sure you have the right amount. However, counting with the right arrow key can be tedious. Use option R - Replace characters to count your terminators for you.

Here's how: Specify your transformation chapter as the source and target chapter in Replace characters. Key in changes directly, writing just one rule: replace your terminator character with itself. This does not change your transformation chapter. When Replace is finished, BEX announces `Replaced # times:` where # is the number of terminators in your chapter. This number must be even.

Remember that the number of terminators should be two more than twice the number of rules in your chapter.

- ***Transformation Chapter Limits***

A find or change to string cannot exceed 100 characters. A transformation chapter cannot exceed two BEX pages--a total of 8192 characters. During the Replace characters process, BEX refuses to execute a replacement that would make your target page contain more than 4096 characters. If you try to do this, you hear a grotesque noise, like sound effects for a flying saucer. Press <ESC> to stop the Replace characters process, and give your Apple a reassuring pat to calm it down.

- ***Order of the Rules***

The changes you specify, whether typed in directly or created in the Editor, are executed exactly in the order they appear in the transformation chapter. That means that you can have a rule which wipes out a new change, if you are not careful. Or you can create a two-step rule, where the first part changes one part, and the second another. For example, the transformation chapter KRM that is discussed in Part 4, has a two-step rule to insert paragraph (\$p) indicators at the end of paragraphs. The first of the two rule deletes all spaces in front of <CR>s (the number sign here is the terminator):

#<space><CR>#<CR>#

The second rule replaces <CR>s with periods in front of them with (\$p) indicators:

#.<CR>#.<space>\$p<space>#

A two-step rule ensures that all the changes are made: if the KRM rule just replaced all <CR>s with spaces before them to paragraph (\$p) indicators, any instance of <CR>s with periods before them would go unchanged.

Another way of ensuring that only the characters you want replaced are changed, is to use the intermediate rule method. For example, suppose you want to change two <CR>s to the paragraph (\$p) indicator, but you want to leave three <CR>s alone. For your first rule, change the occurrences of three <CR>s to a distinctive set of characters like \$\$ (three percent signs). Next, find the two <CR>s and change them to (\$p). Then, find the \$\$ strings and change them back to <CR><CR><CR>.

- ***Uppercase and Lowercase***

Replace characters is "case-sensitive". Replacing every *A* (uppercase *A*) with \ (backslash) does not affect any *a* (lowercase *a*). Sometimes the word you want to change appears at the start of a sentence, with an initial uppercase letter, and sometimes the same word appears all lowercase, in the middle of the sentence. In this situation, you must write two rules. (At the Master Level, you learn how to use contextual Replace to specify a letter regardless of case.) Another way around the problem of initial letter case variation is to ignore the first letter. For example, in the transformation chapter for our Newsletter, we tried to create consistency in naming contracted braille using a number instead of Roman numerals. To avoid uppercase vs. lowercase problems, we don't search for *Grade II* and *grade II*. We search for *rade II* and change it to rade 2. This sometimes has unintended results: an article that discussed *Super Bowl Parade III* got changed to Super Bowl Parade 2I.

- ***Words vs. Parts of Words***

In the sample shown in Part 2, Sally misspelled the word *precede*; she spelled it *preceed*. The third rule found *preceed* and changed it to *precede*. This works fine in a sentence like *We preceed the action with deliberation*. This rule won't fix the sentence *He preceeded the action with deliberation*, because the second word changes to *preceeded*. To find any occurrence of a misspelled word, you must ponder the possible variations in prefix and suffix, and create rules for each of them.

Another example relates to keyboard shortcuts. While writing this manual, we developed many abbreviations that were expanded after the text was complete. One writer always reverses the *a* and *o* in *keyboard*, so she developed a transformation chapter that expanded every appearance of *k-b* into *keyboard*. Imagine her surprise when she saw a sentence about the *diskeyboardased VersaBraille*. You can place spaces before and after the abbreviation: replace `<space>k-b<space>` with `<space>keyboard<space>` But, that doesn't work very well when you want to place punctuation next to the expanded word. To get around this problem, there are two complementary strategies. First, make your abbreviations so distinctive that they can't mistakenly appear embedded in words you don't want to change. For example, instead of using *k-b* use *#kb*. The second approach requires you to write enough transformation rules to deal with all possible surrounding punctuation.

At the Master Level, you learn about contextual Replace, which allows you to write one transformation rule to handle many possible surrounding contexts.

Part 4: Supplied Transformation Chapters

The transformation chapters on the BEXtras disk can help you tidy up your data in many situations. When you want to use one of these transformation chapters, just insert your BEXtras disk in one of your drives. Specify the chapter by name, preceded with drive number if necessary. BEX reads the transformation rules into memory, then gives you the `Continue? Y` prompt. Remove the BEXtras disk and get your data disks in the correct drives, then press `<CR>` to start Replacing.

Most of the supplied transformation chapters are regular Replace chapters, so you can edit them to see how they work. You should copy them before you edit them, just in case you inadvertently change something.

KRM

This transformation chapter cleans up general text material imported from the Kurzweil Reading Machine (KRM). The Kurzweil doesn't provide any information about where paragraphs are divided in the print original. The KRM transformation chapter tries to guess at these divisions. This chapter contains five rules; its terminator is | (vertical bar). The first rule deletes any linefeeds, or <control-J>s. The second and third rules work together: the assumption is that the last line of a paragraph ends with a period, a space, and a <CR>. The second rule deletes the space, so that this end-of-paragraph situation is reduced to <period><CR> which is the find string of the third rule. This is then changed to the (\$p) indicator. The only use for <CR>s is in their role as possible indicators of paragraph endings. Because BEX executes the rules in order, after the third rule is executed, you can get rid of the <CR>s. The fourth rule changes any remaining <CR>s to spaces, and the last rule deletes any extra spaces.

UCLC and LCUC

The "UpperCase to LowerCase" transformation chapter transforms entirely uppercase files to entirely lowercase files. Use this to work with files downloaded from all uppercase systems.

LCUC does just the reverse of UCLC, changing every lowercase letter to uppercase. Your format commands and format indicators are also changed--so they become powerless. When you want your format commands executed *and* all uppercase text, use the set uppercase format command \$\$su--see Section 7, Part 9.

SP2

This chapter is a contextual Replace chapter. In Master Level Section 6 we explain in gory detail how this contextual Replace chapter works. You can examine the chapter in the Editor if you want, but you will see many strange things!

This chapter places two spaces after the end of a sentence, and deletes two spaces anywhere else. (Since it's a Contextual Replace chapter, it's clever enough to not place two spaces after abbreviations.) Use SP2 whenever you want your print text to follow this common inkprint convention. (For both braille and typeset print, you never end sentences with two spaces.)

FIX TEXT

This chapter is also a contextual replace chapter. FIX TEXT is designed to reformat textfiles transferred with option R - Read textfiles to chapters (documented in Section 10, Part 5). It untangles the mess of control characters and text you may get as the result of underlining in some textfiles. It changes the many instances of <control-H> to BEX's own *and* underlining commands, deletes any line feeds, and places \$\$c commands and paragraph (\$p) indicators where needed.

Part 5: Troubleshooting Problems with Replace Characters

Here are some problems you may encounter when using Replace characters:

- Problem: I am in the middle of typing changes in directly, and I realize that I have made a big mistake.
- Solution 1: Keep entering your terminator until you reach the Continue? Y prompt, then enter N <CR>
- Solution 2: Enter Control-Reset, then type RUN to get back to the Main Menu: prompt.
- Problem: Option R - Replace characters seems to be proceeding fine, then all of a sudden I hear a terrible noise, like a *TARDIS* taking off or the Martians landing.
- Cause: You have an overflow error. You've instructed Replace characters to increase the number of characters in your page beyond the 4096 limit. This could happen if you asked to expand many abbreviations to long phrases. Rather than overfilling your pages and causing havoc, Replace characters alerts you with that sound.

- Solution: Press <ESC>. Use option A - Adjust pages on the Second menu to reduce the number of characters in each page and increase the number of pages in your chapter, then try again.
- Problem: BEX prompts Starting to replace and then stops. I hear no clicks; I never get the Chapter NAME done message, and the data drive does not whir. OR I hear many clicks at the beginning, but they start slowing down and never end.
- Cause: You have a problem with your Editor-created transformation chapter. You are probably missing two terminators at the very end.
- Solution: Enter Control-Reset, then type RUN to restart BEX. Control-Reset stops Replace characters. Because you didn't hear the disk drive whir, no erroneous data has been saved to disk. If you typed changes in directly, either repeat this process, or create the transformation chapter in the Editor. Depending on what your terminator is and the length of your transformation chapter, it is possible that you may need to reboot to fix this.
- Problem: Replace characters seems to work, but when I inspect the target chapter, only a portion of the changes were executed.
- Cause: The terminators are "out of phase", so all the rules were not executed. This happens when a terminator appears at a find string before the end of your transformation chapter.
- Solution: Edit your transformation chapter. Use control-L control-A to locate every appearance of two terminators next to each other. When a terminator pair follows a find string you want to delete, it's fine. But wherever two terminators appear after the change to string, replacing stops.

9 Translating Print and Braille

At the User Level, you gain access to several more braille translation features. You can translate inkprint source chapters to grade 2 *and* grade 1 braille target chapters. You can also switch between grade 2 braille and *no* translation within a chapter. A new option is available on the Main Menu: option H - Heading test. It allows you to do almost instantaneous braille translation. It's a quick way to find out how BEX translates a particular word.

Also new at the User Level is option B - Back translate from Grade 2. You supply BEX with grade 2 braille source chapters, and BEX creates inkprint

target chapters. You can create grade 2 source chapters on a braille-oriented device like the VersaBraille or Cranmer--see Sections 11 and 12 for information on how you transfer braille text to the Apple. You can also input grade 2 braille using BEX's braille keyboard mode in the Editor--details in Section 5, Part 7.

Whether you translate inkprint to grade 2 braille chapters, or whether you create braille chapters yourself, you must configure a braille device before you can emboss them. See Learner Level Section 8 for details.

Part 1: Option G - Grade 2 Translator

In Section 7 of the Learner Level, we described how to use option G - Grade 2 translator. You supply BEX with the names of source chapters. BEX creates correctly translated braille *target* chapters. We stressed the importance of consistently naming braille and print chapters so you can tell them apart. We suggested that you name your braille chapters by adding the digit 2 to the end of the print chapter name. In Section 4 of the User Level, we explain selective drive scanning. You can enter 1/2 <CR> or 2/2 <CR> at the Drive or chapter: prompt and get a list of all chapters that end with the digit 2. In this Part we detail how the translator handles some tricky translating situations. Part 3 discusses the *translator controls* that allow you to switch between grade 2, grade 1, and no translation within a chapter.

- ***Embossing Braille Chapters***

You *must* configure your braille embosser appropriately or you won't get correct format. Suppose you have a Cranmer Braille in slot 2, but you configured it as a class G - Generic inkprint printer, with carriage width 40 and form length 25. Print a correctly translated grade 2 chapter to the Cranmer, and things look all wrong.

Because BEX thinks it's formatting text for a printer, BEX executes the underlining commands $\$sub$ and $\$h$. For any underlined word, you see character, dots 4-5-6, character, dots 4-5-6, character, etc. BEX executes the

\$\$np number pages format command by centering the word *Page* followed by Nemeth digits on line 25. There would be a blank line between paragraphs. Only when you accurately reconfigure the Cranmer as a class B - Braille embosser, brailleur code 3, can you get the correct format.

And as mentioned in Section 6, when you define a New printer without reconfiguring, BEX always uses inkprint defaults. So if you attempted to send text to a brailleur by answering N at the `Which printer:` prompt, you would get the same incorrect format.

Some basic hints for braille format are supplied in the Learner Level, Section 9. BEX's format commands are designed to make the transition easy. The page numbering command \$\$np and paragraph (\$p) indicator are executed appropriately for print and braille format, depending on how the printer is defined in your configuration. When you send chapters to a device configured as a brailleur, then BEX suppresses the execution of the \$\$ub and \$\$h underlining commands. The relative format commands l \$\$tin or \$\$ting; automatically adjust for the differences in word length between print and braille.

One aspect of BEX's \$\$ commands can prove confusing to experienced transcribers. In all the reference works on braille format, the first cell on the line is called *cell one*. In terms of BEX's \$\$ commands, that same cell appears at *position zero*. In other words, the numbers in BEX's \$\$ commands are one less than the cell numbers transcribers are accustomed to.

A plug for TranscriBEX

You can do a variety of braille formats using BEX's \$\$ commands. However, some braille formats require extensive, involved strings of \$\$ commands. RDC has developed an add-on module for BEX named *TranscriBEX*. TranscriBEX fully supports the exacting braille translation and page formats mandated by English Braille--American Edition, the *Code of Braille Textbook Formats and Techniques*, and the *Code for Computer Braille Notation*. But you don't have to enter complex strings of \$\$ commands to create these complex page formats--TranscriBEX does it for you. When you are doing a lot of braille transcribing, TranscriBEX is the right tool for the job. TranscriBEX formats include general text, textbook format, line-numbered poetry and plays, and more. TranscriBEX also includes special symbols that let you switch between grade 1 and grade 2 translation in the middle of a word;

control placement of braille boldface indicators and the braille letter sign; and a host of other subtle translation issues--contact us for further details.

- ***Notes on Translation***

BEX's Grade 2 translator is quite accurate. You can improve the accuracy through some manual intervention based on an understanding of how it works. When you find that the translator makes a systematic error, you can create a transformation chapter to fix it. Section 8 discusses transformation chapters in detail. RDC always appreciates written comments about how we can improve the accuracy of the translator.

Print underlining and braille italics

Braille italics are shown with the italics sign, dots 4-6 or the period in screen braille. The Grade 2 translator uses the presence of the format commands `$$sub` and `$$uf` to place the italics sign as appropriate. When three or less words are underlined, a single italics sign is placed before each word. When more than three words are underlined, a double italics sign (two dots 4-6, two periods in screen braille) is placed before the first underlined word and a single italics sign is placed before the last underlined word. For more details on the braille italics, see Learner Level Section 9.

Hyphens and dashes

In Master Level Section 5, we explain the discretionary linebreak. When BEX's formatter encounters the `<ASCII 30>` control character it's treated as an acceptable place to break the line, just like a `<space>`. As the Grade 2 translator translates inkprint, it places `<ASCII 30>` after a hyphen and both before and after two hyphens (a braille dash). The discretionary line break means more characters can fit on a line, especially useful with the relatively narrow 41-cell carriage width of most braille. Please note that the Grade 2 translator cannot place discretionary hyphens, only discretionary linebreaks.

Accent sign

The braille accent sign is dot four, the at-sign in screen braille. When you want to show an accented letter in grade 2, precede that letter with at-sign @

in your inkprint chapter. For example, enter `cl ich@e` in inkprint when referring to a stale expression. The translator creates `cli*@e` in the grade 2 target chapter, where the asterisk stands for the two letters *ch*.

Less-than and greater-than symbols

These two symbols are used in various contexts. They can be symbols of enclosure, a function they serve in this manual. For example, the single linefeed control character is shown as `<control-J>`. In that context, `<` and `>` are sometimes called *left* and *right angle brackets*. At other times, these symbols are used mathematically to compare two values. BEX's Grade 2 translator always places a braille accent, dot 4, before these symbols, to distinguish them from the *gh* and *ar* contractions. `<` is translated to dot 4, dots 1-2-6 or `@<` in screen braille. `>` is translated to dot 4, dots 3-4-5, or `@>` in screen braille.

Backslash

The Grade 2 translator places a dot 4, or at-sign in screen braille, before a backslash. This distinguishes between a backslash in print and the *ou* sign.

Equals sign

The translator violates the rules for literary braille when translating the equals sign. The rules require the insertion of the word *equals*. Instead, the translator changes the inkprint `=` equals sign into the braille Nemeth Code equals sign: dots 4-6, dots 1-3, or `.k` in screen braille.

Percent

In inkprint, the percent sign `%` follows a number: `80%` for example. The grade 2 braille rules require that the percent sign (dots 2-5, dots 1-2-3-4) immediately precede the number: `3p#hj` is how *80 per cent* should be translated. The translator has two ways of coping. When you place the percent sign *before* the digits in your inkprint, the grade 2 result follows the rules. Inkprint `%80` becomes `3p#hj` in grade 2. When the percent sign *follows* the digits in inkprint, then the grade 2 version use the properly contracted words *per cent* with a space before and after. Inkprint `80%` becomes `#hj p] c5t` in grade 2.

Single quote marks and apostrophe

On most typewriter keyboards, and in the ASCII Code, opening single quote, closing single quote and apostrophe are the same symbol. However, in grade 2 braille, the opening quote, the closing single quote, and the apostrophe are three different symbols. The opening single quote is dot 6, dots 2-3-6 or , 8 in screen braille; the closing single quote is dots 3-5-6, dot 3, or 0 ' in screen braille; and the apostrophe is dot 3, or ' (apostrophe) in screen braille. The Grade 2 translator guesses which symbol to use depending on context, and it does a pretty good job. One situation requires manual intervention: when an apostrophe appears at the beginning of a word, usually in poetry or dialectical speech. The translator assumes that the apostrophe symbol at the beginning of a word should be translated as an opening single quote. You force this symbol to be translated as an apostrophe by preceding it with the greater-than sign. When your inkprint contains something like *'twas the night 'fore Christmas*, enter

>'twas the night >'fore Christmas

The translator then creates

'twas ! ni<t '=e ,*ri/mas

instead of creating

,8twas ! ni<t ,8=e ,*ri/mas

- ***Limitations of the Grade 2 Translator***

Sometimes the rules for grade 2 braille require that you braille a word in more than one way, depending on the word's context or pronunciation. BEX's Grade 2 translator always chooses the more common usage. For example, the word *do* can be a verb of action, or it can be the first note of the musical scale. The translator assumes *do* is a verb; it becomes the single letter *do*. When you talk about the first note of the musical scale, however, the rules say spell it out in full, but the translator still contracts.

Foreign words, names, and made-up words are not always reliably translated. When you have questions about how a word translates, use option H - Heading test to get instant answers. When you have a lot of questions, enter the words in a separate BEX chapter. Translate that chapter and examine the results. Then write a transformation chapter to fix the errors with option R - Replace characters--details in Section 8.

For example, suppose you're writing a story about a law firm named Hallshurst and Granthorn. Those names are pronounced *halls hurst* and *grant horn*, so they should use neither the *sh* nor the *th* contractions. But the translator doesn't know any better, so it uses these contractions. You can create a transformation chapter like this, where the terminator is vertical bar:

```
|,hall%ur/|,hallshur/|,gran?orn|,granthorn||
```

and then use this transformation chapter on the *final* grade 2 braille chapters.

We know there will always be some idiosyncrasies in the translators. We are grateful for the written reports from users which have enabled us to continually upgrade the translators' performance.

Part 2: Braille to Print with the Back Translator

Option B - Back translate from grade 2 on the Main Menu is the opposite of option G - Grade 2 translator. Your source chapters are grade 2 braille, your target chapters are inkprint. Braille chapters can be typed in through the braille keyboard or imported from a remote braille device. As far as specifying chapters, both braille translation options are the same. However, the *action* of the back-translator is not a mirror image of the "forward" translator. While the Grade 2 translator can also create grade 1 braille, the Back from Grade 2 translator cannot back-translate grade 1 to inkprint. Part 3 explains how you can turn off back-translation within a chapter.

Grade 2 braille is a highly context-dependent code. The same character can have different meanings depending on where it appears in a word. When you carefully follow the rules for grade 2 braille, you get more accurate results with the Back from Grade 2 translator.

Back translate from grade 2 allows you to prepare material entirely in braille. When you enter BEX's format commands, use the *ed* sign (dots 1-2-4-6) for the dollar sign, and Nemeth numbers (dropped *a* through *just*) for any numbers. Parts of this manual were written in grade 2 braille on the VersaBraille, and then back translated to be merged with the rest of the text.

Prevent overflow

There is a crucial space consideration when back-translating. Grade 2 braille uses many one and two cell contractions to represent long inkprint words. An extreme example is this silly sentence: *According to her, tomorrow's question is about knowledge.* In print this is 57 characters (counting spaces): in grade 2 braille, its equivalent is 25 characters. Of course, that sentence uses a high proportion of contractions, but a full BEX page of grade 2 braille would probably overflow when it's expanded by translation back to inkprint.

The rule of thumb is, expect a 22 per cent increase. For grade 2 chapters that you intend to back-translate, keep your pages under 3200 characters. You can use option A - Adjust pages on the Second menu to create chapters that back-translate successfully.

If you did not follow this rule, you would get a spirited audio reminder. If there's too much text and BEX gets an overflow error, you hear the overflow shriek, which sounds like a spaceship taking off. Press <ESC> to cancel back-translation. In the unfortunate situation where you are using the same name for source and target chapters, *and* you have an overflow error after the first page is saved, you have data salad. You won't be able to recover the information in the original first page.

In addition to entering grade 2 through BEX's braille keyboard mode, there are two other ways to create grade 2 braille chapters. Braille text can be transferred from the tape-based VersaBraille with option F - From VersaBraille, or it can be imported from any serial braille device by option I - Input through slot. (Section 11 discusses Input through slot and the disk-based VersaBraille II.) Recognizing the need to prevent overflow, BEX tries to limit page size to under 3200 characters in these situations.

VersaBraille:

Take advantage of option A - Autoprint from VersaBraille on the Main Menu. This one option transfers a grade 2 file from the tape-based VersaBraille to the Apple, performs a back translation, and sends the result to an inkprint printer. See Section 11 for details. Raised Dot Computing also distributes a program called NUMBERS which back translates both literary braille and Nemeth mathematics braille.

- ***Increasing Accuracy of Back-translation***

As mentioned above, careful data entry on your part results in more accurate back-translation. The following suggestions are a general guide. Whenever your grade 2 text contains possibly ambiguous abbreviations or technical material with many symbols, it's a good idea to back-translate a sample and check it out in the Editor. Back-translation can create some pretty hilarious errors, which make perfect sense to braille readers but are quite opaque to those who only read print.

Avoid grade 2 slang

The back-translator simply cannot cope with grade 2 slang. For example, many experienced braille users write the word *income* as *in sign, com sign, e,* or *9-e* in screen braille. The rules state that dots 3-6 only stands for *com* when it appears at the beginning of word. When dots 3-6 appears in the middle of a word, the back translator considers this a hyphen. When you enter *9-e* it becomes *in-every* in inkprint, because the back-translator is faithfully following the grade 2 rules. Similarly, the back translator won't translate an en-sign directly before punctuation as the word *enough*. Experienced braille users often forget the grade 2 rule that forbids any dropped-letter word contraction before punctuation.

Use the letter sign

The letter sign (dots 5-6) distinguishes between an isolated letter and a grade 2 whole-word contraction. When you want an isolated letter in your inkprint, you *must* alert the back translator with the letter sign. This prevents back translating an isolated *b* into *but*, an isolated *c* into *can*, and so forth.

The Back from Grade 2 translator requires the letter sign in some contexts where good grade 2 does not. For example, in good grade 2 you do not use a letter sign before the *E* in *E. Washington Avenue*. But, you must use the letter sign for correct back-translation, or you get: *E. Washington Avenue*. Similarly, when you letter each element in an outline, you must precede the letter with the letter sign--even when you place punctuation after the letter. There's one exception: when you enclosed a single letter within parentheses, the back translator recognizes that you couldn't mean a single-cell contraction. So *7c7* back-translates to (c) and *7,e7* back-translates to (E) as they should.

The back translator just can't understand contextual clues as well as the human braille reader. When your text includes many isolated letters, write a test grade 2 chapter and back-translate it. Examine the inkprint results, and discover where you need to use the letter sign.

The slash character

The B - Back from Grade 2 translator does not exactly reverse the G - Grade 2 translator. The inkprint slash character is unchanged when translated to grade 2. The abbreviation *I/O* becomes , i / , o when translated. However, the Back from Grade 2 translator always interprets the slash character (dots 3-4) as an st-sign. When you want to have a slash in inkprint, precede the slash with the letter sign, dots 5-6 or ; in screen braille. Enter , i / , o in braille and you get I st O in inkprint; enter , i ; / , o in braille and you get I / O in inkprint.

Capitalization

Most braille embossers respond the same way to uppercase and lowercase letters. Send either an *a* or an *A* and you get dot 1. Some braille input devices allow you to choose between lowercase and uppercase data entry. (BEX's braille keyboard mode limits you to lowercase data entry.) As it processes grade 2 text, the Back from Grade 2 translator changes any uppercase letters to lowercase letters.

When you want uppercase letters in your inkprint chapter, use the caps sign in your grade 2 chapters. (Part 3 explains how you can use the no-translation *translator control* to exempt some text from back-translation.) The back-translator recognizes a mid-word change in capitalization signalled by the termination sign, dot 6, dot 3 or , ' in screen braille.

This second half of the 20th century seems to have brought with it an uncontrollable urge to coin new words, making the most remarkable assemblages of letters, numbers, and punctuation. In response, we've provided the back translator with some techniques to handle mid-word change of capitalization.

For example, dot 6, dots 1-3-5-6 generally stands for the suffix *ation*. Yet those two cells also appear in the word *VersaNews*. As a mark of respect to that fine publication, when the back-translator encounters , v] sa , news it creates

the right result. In fact, the back translator assumes that any double dot 6 in the middle of a word is signalling a change in capitalization. So when you braille `9, , l>ge & , transcri, , bex` in your chapter, the back translator creates `inLARGE` and `TranscriBEX` using the double dot 6 to create uppercase letters.

However, there are some patterns of punctuation that the translator *can't* cope with. When a grade 2 word ends with dots 3-4-6, or + in screen braille, the translator creates *ing*. You must make a conscious effort to get a word that ends with the plus sign in inkprint: you can either turn the translator off or use Replace characters.

Develop transformation chapters to help you

Section 8 discusses Replace characters in detail. Replace characters is the perfect companion to the back-translator. Create a grade 2 chapter containing all your ambiguous text, back-translate it, then edit it. You will quickly see where the back-translator has difficulty. It's usually easiest to fix problems after they have occurred, in the inkprint target chapters. Words like *in-every* or *I/O* in an inkprint chapter are unambiguously wrong. Part 3 discusses turning off the back-translator entirely within a document. When you know the back-translator would mangle some text, use this feature to ensure that your inkprint contains what you want.

Part 3: Controlling Translation within a Document

You can enter special symbols in your text that control how it's translated. Not surprisingly, these symbols are called *translator controls*, or *TC'S* for short. When using option G - Grade 2 translator, you can switch between four translation modes: grade 2, grade 1, grade 2 with "British style" capitalization, and no translation. When using option B - Back translate from grade 2 you can switch between two translation modes: grade 2 or no translation.

All TC'S are four keystrokes long; the first and fourth character are always spaces. Just like BEX's paragraph (\$p) indicator, the leading and trailing

spaces must be present for the TC to work; we enclose the full TC in parentheses to emphasize the leading and trailing space.

Disappearing translator controls

- (_l) - Start grade 2 translation; <space>, underbar, lowercase *l*, <space>
- (_o) - Start grade 1 translation; <space>, underbar, lowercase *o*, <space>
- (_b) - Start "British style" grade 2; <space> underbar, lowercase *b*, <space>
- (_-) - Start no translation; <space>, underbar, hyphen, <space>

Residual translator controls

- (@l) - Start grade 2 translation; <space>, at-sign, lowercase *l*, <space>
- (@o) - Start grade 1 translation; <space>, at-sign, lowercase *o*, <space>
- (@b) - Start "British style" grade 2; <space> at-sign, lowercase *b*, <space>
- (@-) - Start no translation; <space>, at-sign, hyphen, <space>

The third character in the TC stands for the translation mode you're choosing. Grade 2 is shown with the lowercase letter *l* for *literary* braille. Grade 1 is shown with the lowercase letter *o* for grade *one* braille. "British style" suppressing of caps sign is shown with the lowercase letter *b* for British. No translation is shown with the hyphen character; think of a minus sign.

The second character in the TC determines whether the TC appears in your target chapter. A switch in translation is usually obvious to the reader; you don't need an explicit symbol to alert the reader to a change in modes. In some rare instances, however, you *do* want an explicit symbol. For these two situations, BEX provides two types of TC'S: the *disappearing* TC uses the underbar as the second character; the *residual* TC uses the at-sign as the second character.

Ninety-nine percent of the time, you use the disappearing TC: the four characters change the translation mode and then become one space in the target chapter. For those rare occasions when you want evidence of a change in translation, use the residual TC'S. The residual TC can be useful as *on* or *off strings* in Contextual Replace, discussed in Master Level Section 6. Each constant TC changes the translation mode and remains in the target chapter.

- **Using Grade 1 TC with the Grade 2 Translator**

The Grade 1 translator is built in to the Grade 2 translator. Grade 1 braille requires much more space than grade 2 braille, because grade 1 does not use any contractions. But grade 1 braille is not the same thing as untranslated braille: inkprint and grade 1 use different symbols for numbers and punctuation. Grade 1 uses the same composition signs as grade 2 to show capitalization and italics. Grade 1 is appropriate when you need to have a letter-for-letter representation of inkprint material. Grade 1 braille may be preferred for adults just learning braille, and is used to represent foreign languages.

Caution! | Option B - Back translate from grade 2 cannot back-translate grade 1 to inkprint.

Once the Grade 2 translator encounters the grade 1 TC it continues to do grade 1 translation until you explicitly tell it to return to grade 2 translation with the grade 2 TC.

Sample: Grade 1 and grade 2 TC'S

In most situations, you use the disappearing TC'S. When a word is intentionally misspelled, you should transcribe it in grade 1 instead of grade 2. Here's how you would place grade 1 and grade 2 TC'S in a hypothetical child's letter to Santa Claus:

Dear Santa: \$p I am _o riting_lyou because my mother told me I _o shud. _l I have been a good girl this year and Mommy says you will bring me a _o Kemistry _l Set. I would love it! \$p Regards, your friend Francie

When this chapter is processed through the Grade 2 translator, the result is:
,de> ,santa3 \$p ,i am riting y 2c my "m told me ,i shud4 ,i h be5 a gd girl ? ye>
& ,mommy says y w br+ me a ,kemistry ,set4 ,i wd love x6 \$p ,reg>ds1 yr fr
,francie

Every four character disappearing TC has become one space. While in grade 1 translation mode, the translator still places italics signs when it encounters \$\$sub and \$\$uf.

- ***Using no translation***

When using either Grade 2 translator or Back translate from grade 2, you may wish to take control of translation yourself. You accomplish this with the no-translation translator control. The residual (-) TC still appears in your text once it's translated; the disappearing (-) TC becomes a single space in the translated text.

Once the translator encounters (-) or (-), it stops translating or back-translating until you explicitly return to grade 2 translation with the Grade 2 translator control.

No translation with option G - Grade 2 translator

When you are doing inkprint data entry, it's easiest to use the full Apple keyboard. When you want to include occasional Nemeth or other braille codes in your text, you can switch off translation and enter the braille directly. Once you do, you probably want to use BEX's braille keyboard mode. (This feature is not available on the Apple IIgs keyboard.) In the Editor, depress the Caps Lock key, and enter control-S K B. Now you can use the S-D-F J-K-L keys and the spacebar for braille data entry.

When you return to regular inkprint data entry, type the grade 2 TC, and then enter control-S control-K control-N to return to normal keyboard mode. Finally, release your Caps Lock key.

No translation with option B - Back translate from grade 2

Turning off the back-translator can be a handy way to enter tricky letter and number combinations. For example, you're writing some advertising copy for a new diet soda called *Cherry77*. Back-translation of , *] r y # g g results in Cherryblegg which does not sound very appetizing. When you enter _ - Cherry77 _ 1 in your grade 2 text, you turn off back-translation, so you get *Cherry77* in your inkprint.

You cannot enter uppercase letters with BEX's braille keyboard mode; the braille keyboard always creates *lowercase* letters. When you are doing braille

data entry in BEX, you must switch to full Apple keyboard to type `_ - Cherry77 _ 1` and then switch back to braille keyboard. (Alternatively, you can enter `_ -cherry77 _ 1` on the braille keyboard, then use Replace characters to change the initial *c* from lowercase to uppercase.

Most other braille devices allow you to distinguish between upper- and lowercase inkprint letters as you enter them. On the disk-based VersaBraille II, for example, you type an uppercase *C* by brailing chord-K dots 1-4.

10 Textfiles

BEX stores information as *chapters*, a collection of binary files on disk. Many other programs store information as textfiles. In Section 12 of the Learner Level, we demonstrated option R - Read textfiles to BEX chapters on the Second Menu. Read textfile lets you *import* data from other Apple programs, both DOS 3.3 and ProDOS. At the User Level, there are three options on the Second Menu concerned with textfiles: option R - Read textfiles to chapters; option W - Write chapters into a textfile, and option T - Textfile kill.

Option W - Write chapters into a textfile is the opposite of option R - Read textfiles to chapters. Write chapters into a textfile lets you *export* BEX data to other DOS 3.3 Apple programs. RDC distributes a ProDOS utility called QTC that copies BEX chapter directly to ProDOS textfiles. When you want to export BEX data to ProDOS programs, use QTC.

Part 1: Background on Textfiles

BEX operates in the computer environment called *Apple DOS 3.3*. DOS 3.3 controls how information moves between the computer and the user, in particular, how characters are read from and written to disk. Within DOS 3.3, there are four ways of saving information on disk--four *file types*. Each type of file has its uses, and is identifiable by the letter in the second column of a DOS disk catalog.

The letter *A* identifies Applesoft BASIC programs. You can LOAD, RUN, SAVE, and LIST type *A* files. While you can include text in a BASIC program, it's not very efficient. On the Main side of BEX, an Applesoft program named SECOND contains the Second Menu. It includes text like *There are # chapters on this disk*.

Binary files store information for use by application programs: type *B* files can contain either programs or data, and your BEX disk contains samples of both. On the Boot side, for example, there's a type *B* file named *FID*, the File Developer utility. You can BLOAD, BRUN or BSAVE type *B* files, but you can't LIST them.

Also on the Boot side are these files:

B017MESSAGES.A

B015MESSAGES.B

B013MESSAGES.C

B003MESSAGES

which together make up the BEX chapter named *MESSAGES*. The *MESSAGES* chapter contains most of BEX's prompts, and at the Master Level, you learn how to modify it. More details about BEX chapter structure appear in Section 13 at both Learner and User Levels.

The third file type is *T* for textfiles, and that's what we concentrate on in this Section. The fourth file type is *I* for *Integer BASIC*, a slightly different dialect from Applesoft BASIC. BEX doesn't use it, so we'll leave it at that.

- ***Textfiles vs. Binary files***

You don't need a word processing program to use textfiles. In fact, all you need is Apple's DOS 3.3 or ProDOS. Commands to create, read and modify textfiles are built in to these operating systems. This has two consequences: almost any Apple program can handle textfiles, and, if a program is lacking this facility, a fairly inexperienced programmer can incorporate textfile-handling into the program.

But there are some disadvantages to storing data in textfiles with DOS 3.3, and that's why BEX stores data in binary files. One big minus is speed. Most textfile-related DOS 3.3 commands are very slow. BEX's Read textfile option

is very fast, but only because it totally bypasses DOS 3.3. The other minus relates to a textfile's structure: it's basically one long string of characters. Imagine a textfile containing 8000 characters. If you decide to delete 200 characters somewhere in the middle, then you have to rewrite all of the characters after the deleted ones--and this process is doubly slow.

On the other hand, as you create BEX chapters, each BEX page is stored as one binary file. BEX adds extensions to your chapter name to make the binary filename. Every BEX chapter also has one three-sector binary file with the same name as the chapter name--that's the *directory* file that serves as the internal table of contents for the chapter. Since each BEX page is one file, it's easy to swap pages around. You only have to change the information in that chapter's directory file. If you were working with a textfile, you would have to rewrite all the data to disk.

Part 2: Working with Textfiles

All the chapter selection and naming features we've described in Section 4 also apply to selecting and naming textfiles. For example, when BEX presents the `Textfile:` prompt, you can type `1/Q <CR>` to scan drive 1 for all the textfile names ending in the letter *Q*.

There's one difference between target naming methods for chapters and textfiles. *Never* use the same name for a textfile and a BEX chapter, even when the two files are on different disks. This means you can't use the *S* naming method. Follow the same rules for textfile names as for BEX chapters: the first character must be a letter; never use a period, comma, semicolon, or colon; and limit your textfile names to 25 characters. When your disks get cluttered with textfiles, use option *T* - Textfile kill to delete them.

Textfiles do not appear in the list of chapters supplied by option *D* - Disk catalog. In a DOS catalog, you identify textfiles by the letter *T* in the second column. The quickest way to locate textfiles on a disk is to scan the disk drive with either option *R* - Read textfiles to chapters or option *T* - Textfile kill.

Option *R* - Read textfiles to chapters can copy ProDOS textfiles to DOS 3.3 BEX chapters. However, BEX can neither catalog nor delete a file from a

ProDOS disk. The only way to see a ProDOS textfile on disk is to scan the ProDOS disk with option R - Read textfiles.

Part 3: Using Option W - Write Chapters into a Textfile

After you press *W* at the Second Menu, you supply BEX with a list of one or more chapters to print to disk. When you finish the list of source chapters, BEX gives you the `Target textfile:` prompt. All your source chapters are combined and written to one DOS 3.3 textfile. You can specify source chapters from both drive 1 and 2, but you must make sure there's enough room on the disk for your target textfile. A textfile uses slightly fewer sectors than its BEX chapter counterpart.

- ***Format Defaults for Write chapters into a textfile***

BEX's formatter controls what characters appear in your textfile. Every time you press *W* at the Second Menu, the default format is `$$$f0 $$$w0 by $$$l0 $$$s2 $$$i5`. These values are different from the defaults for printing to the screen, a printer, or a braille; let's see what each means in detail:

- **Line Spacing** - default value is `$$$l0`, lowercase *l*, digit zero. This suppresses BEX's soft `<CR>`s at the end of each line. The `$$$l0` also suppresses any `<CR>`s for a new-line (`$l`) indicator. When you want to enable 1 `<CR>` at the end of each line, use the format command `$$$l1`, (lowercase *l*, digit one) at the start of your chapter, for two `<CR>`s, enter `$$$lbb`, and so forth. No matter what value you use for `$$$l#`, hard `<CR>`s you type in your text are just characters, so they always appear in your target textfile.
- **Paragraphs** - default values are `$$$i5 $$$s2`. Every (`$p`) indicator in your chapter is executed with an indent of five spaces and two `<CR>`s in your textfile. When you want to have three `<CR>`s and no indented spaces in your textfile, then type the format commands `$$$s3 $$$i0` at the start of your chapter.
- **Control Characters** - Control characters in BEX chapters are just like any other character. They pass through the formatter and are written to disk.

Some other word processors use <control-I> to move to a new tab. You can use Replace characters to change (\$\$) to <control-I> for these programs.

- Underlining - BEX does not suppress underlining when you write a textfile. The and \$\$h commands are executed as always: BEX prints a character, then <control-H> to backspace, then the underbar. When you don't want underlining in your target textfile, use Replace characters to remove \$\$h or \$\$sub commands from your text.
- Carriage Width - default value is \$\$w80. The carriage width determines where BEX places *soft* <CR>s at the end of each line. When you use the default linespacing of \$\$l10, the carriage width has no effect. When you do enable <CR>s at the end of each line with \$\$l10, then \$\$w80 means that BEX automatically inserts a <CR> when the next word won't fit in before hitting 80 characters.
- Form Length - default value is \$\$f0. A zero form length suppresses any page-oriented commands, like page numbering, running heads and footers, or form feeds. You must increase the form length to greater than \$\$f4 when you want the textfile to contain running heads, page numbers, and running footers.

At any point in your chapters, you can enter the *reset to default* format command \$\$d. This reestablishes these defaults: \$\$l10 \$\$s; \$\$i5. When you have established new values for carriage width or form length with \$\$w# or \$\$f#, the new values continue; \$\$d does not change them.

Part 4: Format Considerations

Here at RDC, we have experience transferring files between BEX and a few, but by no means all, other programs. The details of file transfer are the kind of topic covered in the RDC Newsletter. What follows is a spotty summary of a rich and complex field. When the target software has a feature like BEX's Replace characters, you can basically use any format you want. Some programs are less flexible, so you want to create just the right format with BEX.

Transfer to other word processors

Most word processors use <CR> to mark the end of a paragraph. The default value \$\$1" for linespacing suppresses soft <CR>s at the end of each line. The only <CR>s in the target textfile are created by (\$p) or hard <CR>s you have typed in your source chapter.

Some word processors have *filtering* features that, though not quite as sophisticated as Replace characters, can automatically get rid of <CR>s at the end of each line. AppleWorks does this automatically. In the Macintosh environment, MacWrite, Microsoft Word, and JustText ask you a question like *Do you want every <CR> interpreted as a new paragraph?* or *Does your source textfile use <CR>s to mark the end of every line?* When you use the default \$\$1" value for line spacing, you would answer Y to the first question and N to the second.

Most ProDOS programs can't directly read DOS 3.3 textfiles. Programs like AppleWorks and ProWORDS *can* read ProDOS textfiles. RDC'S QTC utility can copy either a BEX chapter or a DOS 3.3 textfile to a ProDOS textfile. AppleWorks won't know what to do with BEX's \$\$ format commands. You can globally delete them from your BEX chapter, then use QTC to copy the chapter to a textfile. Or you can use Write chapters to textfile to print the data to disk, then use QTC to copy the formatted DOS 3.3 textfile to a ProDOS textfile.

Proofreading with textfiles

You can use textfiles to proofread exactly how your output appears. This can be particularly handy when you're creating forms and other line-oriented material. At the start of your chapter, establish the form length, carriage width, etc. as it should appear in the final output. It's important to establish the form length first, and then the carriage width. If you want to see any page-oriented commands, your form length must be 4 or greater. Write your chapters to a textfile, then Read the textfile back to a chapter. Review this chapter to see exactly where all the <CR>s and spaces appear in your final output. A new page is shown by the <control-L> character in your text.

Textfiles for telecommunications

When you send plain text over the phone with a terminal program, you do want a <CR> every 70 to 80 characters. You can create this format when

writing chapters to textfile by inserting `$$l`, (lowercase *l*, digit one) at the start of your chapter.

Textfiles for programming

When writing Applesoft BASIC programs you definitely don't want BEX to automatically generate `<CR>`s every 80 characters, since the `<CR>` signals the end of a BASIC statement. Enter `$$w240 $$l1 $$s1 $$i0` at the start of your chapter. The formatter then executes (`$p`) indicators just like `<CR>`s. You also may wish to use `$$su` to create all uppercase output.

Textfiles with NO format

Sometimes you don't want any format commands executed in your textfile. A good example is when you're using a DOS 3.3 spell-checker program. (When you want to use a ProDOS spell-checker, then it's quicker to use QTC.) You write a textfile and then let the spell-checker make changes in that. Then you read the improved textfile back to a BEX chapter. Insert `$$z` at the start of your chapter. The `$$z` command turns off the formatter entirely, so the textfile contains exactly the same characters as those in your BEX page. All BEX's format commands become lifeless dollar signs, letters and numbers in your textfile, while hard `<CR>`s you've typed in your text still appear in your textfile.

- ***Things That Can Go Wrong***

Because textfiles store information in a fairly standard pattern, you can use textfiles as a trade language for moving data between programs. We've tried to make BEX write very correct textfiles, but sometimes another application program may rebel at reading them. Often, this rebellion is traceable to weaknesses in the other software's read textfile features. In particular, some programs need textfiles with frequent `<CR>`s--at least one `<CR>` every 250 characters. BEX's default values for writing textfiles only place `<CR>`s at paragraph (`$p`) indicators, but it's easy to change that default by using the `$$l`, command. Some terminal programs won't let you transmit many control characters, so control characters that are really there in the textfile BEX wrote seem to disappear.

Part 5: Option R - Read Textfiles to Chapters

This option copies the information in both DOS 3.3 and ProDOS textfiles into a BEX chapter. You can use this option to Read more than one textfile to the same number of BEX chapters. To name the BEX chapters you create, you can use all of BEX's *target chapter naming methods* except S.

Caution!

ProDOS allows you to establish *subdirectories* on any storage medium. Generally you use this feature to organize the large amount of storage on a hard disk. However, you *can* create a subdirectory on a ProDOS floppy disk. When you write a ProDOS textfile in a subdirectory, BEX can't find it. BEX can only read ProDOS textfiles at the *root* level directory. For example, /LETTERS is the ProDOS volume name. When you create a /LETTERS/SANDY textfile, BEX can find it and copy it. But if you create a /LETTERS/APRIL/ subdirectory, and write the SANDY textfile there, BEX won't be able to find it.

Textfiles can be formatted in two fashions. When a word processing program always saves data as textfiles, then the textfile probably contains format commands and data. The format commands may be plain text, like BEX's \$\$ commands. For example, ProWORDS and AppleWriter use "dot" commands. It's also possible that the format commands are control characters.

Other programs use a unique format to save data--AppleWorks is the prime example. AppleWorks and other software like this creates textfiles by printing to disk. (In the next Part, we show you exactly how to do this.) In this situation, when you edit the new chapters, they contain no BEX format commands. There's just plain text, <CR>s, and spaces. As we demonstrated at the Learner Level, use option R - Replace characters with the BEXtras disk transformation chapter `FIX TEXT` to change the spaces and <CR>s to BEX format commands.

When a textfile contains underlining, it can be messy. When underlining the word *dog*, many word processors (including BEX), create:
d<control-H>_ o<control-H>_ g<control-H>_

FIX TEXT'S transformation rules change this tangle of <control-H> and underbars to BEX's *AND* underlining commands.

After the transformation, there is only one space at the end of each sentence in the target chapters. When you want two spaces at the end of sentences, use Replace characters with the SP2 transformation chapter, also on your BEXtras disk.

At the User Level, you are limited to 5.25-inch disk drives. This means you can only read DOS 3.3 and ProDOS textfiles on 5.25-inch floppy disks. At the Master Level, you can read textfiles from 3.5-inch disks (both DOS 3.3 and ProDOS), as well as DOS 3.3 textfiles on the Sider hard disk and on RAM drives. Also at the Master Level, we explain how FIX TEXT and SP2 work.

Importing data from ED-IT

The ED-IT software is a line-oriented braille word processor. It saves data as all uppercase DOS 3.3 textfiles. When you do data entry in ED-IT, all your formatting is manual; you press <CR> at the end of each line. You can copy this information to a BEX chapter with option R - Read textfile.

However, *don't* use FIX TEXT to reformat data from ED-IT; FIX TEXT assumes that paragraphs are marked by a blank line. In braille, a standard paragraph is shown with one <CR> and two spaces. So you can replace <CR><space><space> with (\$p). But, there's an additional complication. Transcribers doing ED-IT data entry probably create other paragraph formats, for example an outdented list with indent to cell 1, runover to cell 3. In this case, you must reformat information manually.

Part 6: Creating ProDOS Textfiles in AppleWorks

The following information focuses on AppleWorks' Word Processor. The June 1986 RDC Newsletter details exporting textfiles from AppleWorks' Database. Back issues are available--contact RDC for details.

AppleWorks, like BEX, saves information in a unique binary file format. BEX cannot read AppleWorks files directly. The general procedure for exporting information from AppleWorks to BEX requires four steps.

- 1. You load the AppleWorks Word Processor file onto the Desktop.
- 2. You tell AppleWorks to print. When AppleWorks offers you a list of printer destinations, you choose `A text (ASCII) file on disk`. This creates a ProDOS textfile.
- 3. You boot BEX, go to the Second Menu, and use Read textfile to copy the data from the ProDOS textfile into a BEX chapter.
- 4. Finally, you use Replace characters to insert paragraph (\$p) indicators where they should go.

AppleWorks has gone through four versions. At this writing, it's at Version 2.0. The big difference between AppleWorks 2.0 and all earlier versions is where <CR>s appear in the ProDOS textfile. For AppleWorks 1.1, 1.2, and 1.3, the only <CR>s in the textfile are those in the Word Processor file--usually, only at the end of every paragraph. But AppleWorks 2.0 places at least one <CR> at the end of *every* line when you create the textfile.

When you do data entry with AppleWorks 2.0, you must take special care. While regular print output contains between 70 and 80 characters per line, braille is usually limited to 41 cells. When you bring rigidly line-oriented data from a print to a braille environment, the resulting braille is not properly formatted.

It's crucial to differentiate between a new paragraph, a *meaningful* new line (like a new poetic line in verse), and a new line that's only pertinent in the print copy. Because AppleWorks 2.0 places a <CR> at the end of every line, the AppleWorks data entry must clearly distinguish between <CR>s that matter and those that don't.

After you Read the ProDOS textfile to a BEX chapter, Replace characters must be able to find patterns of <CR>s from the AppleWorks file and change these to the appropriate (\$p) and (\$l) indicators.

- ***Tailoring AppleWorks Data Entry for BEX***

First, we discuss two strategies for distinctive data entry. At the end, we summarize the step by step procedure.

Tabs and blank lines

When you open a new Word Processor document, it has preset tab stops every five characters. The first strategy for distinctive data entry builds on a habit that's well-developed in most typists; this works best for straight textual material. At the start of each paragraph, you type two <CR>s and then press the Tab key. By adding a blank line before each paragraph and indenting five spaces at its start, you create the kind of distinctive pattern that Replace characters can easily find.

By the way, AppleWorks doesn't have "real" tabs: when you press the Tab key, AppleWorks simply inserts enough spaces to move to the next tab stop. Because of this, you can't insert tabs *after* you've typed your text, only as you go.

For a meaningful new line, enter two <CR>s but *don't* press Tab; the line begins at the margin. Once you make a ProDOS textfile and Read this to a BEX chapter, you want to use Replace characters. Here's what the dialogue looks like the first time you do this.

Replace characters

Drive or chapter: 2 <CR>

There are 3 chapters:

1QUIZ

2TAKEHOME

3FINAL

Use entire list? Y <CR>

Target chapter naming method: S <CR>

Use transformation chapter: <CR>

Enter terminator: |

Find: <CR><CR><space><space><space><space>&l t;space>|

Change to: <space>\$p<space>|

Find: <CR><CR>|

Change to: <space>\$l<space>|

Find: <CR>|

Change to: <space>|

Find: <space><space>|

Change to: <space>|

Find: |
Continue? Y <CR>
Starting to replace ...
Replaced (big number) times
Save transformation chapter: 1AWTRANS <CR>

In this sample, the target chapter naming method is S, which means that the transformed chapters overwrite the source chapters. You only have to go through this dialogue once, because you save the AWTRANS transformation chapter on your BEX disk. Next time you bring data from AppleWorks, enter 1AWTRANS <CR> at the Transformation chapter name: prompt. BEX reads the transformation rules into memory, then prompts, Continue? Y Press <CR>, then sit back and listen to a symphony of clicks.

Use unambiguous indicators

The other strategy for distinctive data entry is a universal solution for BEX data entry in *any* "foreign" word processor--AppleWorks, PC Write, WordPerfect, or what have you. An essential quality of BEX is that every format indicator can be entered with plain printing characters; BEX's formatter does not require you to use any control characters at all. Instead of using patterns of <CR>s and tabs, enter <space>, dollar sign, lowercase *p*, <space> to mark the beginning of every paragraph. To unambiguously define the start of a meaningful new line, enter <space>, dollar sign, lowercase *l*, <space>. The BEX (\$p) and (\$l) format indicators are treated just like text in a foreign word processor.

This method, although it requires a little more training for the person doing data entry, is guaranteed to work. As long as they faithfully enter paragraph and new-line indicators where appropriate, they can place <CR>s wherever they feel like it. Once you bring this data into BEX, the Replace characters task is very simple: Replace every <CR> with one space.

- ***Step-by-step Summary: Exporting ProDOS Textfiles from AppleWorks***

Whether you use <CR> plus space patterns or enter unambiguous indicators, the export procedure is basically the same. The only difference lies in which transformation chapter you use as the last step. Here's what you do.

- 1. Prepare a disk - Get AppleWorks up and running. Note the AppleWorks version number on the first screen. At the Main Menu folder, press 5 - Other Activities. At the Other Activities folder, press 5 - Format a blank disk. Enter a disk name of your choosing; we'll use Transfer in this sample. Press space when ready, then listen to the gronking sound. When the noise is finished, remove the newly-formatted disk from the drive and *label it with its name!*
- 2. Load the Word Processor file - Press <ESC> twice to get to the Main Menu. Find the disk with the Word Processor file that you plan to transfer and add it to AppleWorks' Desktop. We'll call it 3.Week.Quiz for this sample.
- 3. Check for single and double spacing (AppleWorks 2.0 only)
 - Now you need to make sure the entire document is single-spaced. Enter open-Apple-1 to move to the top of the file. Enter open-Apple-F to Find, then press O to find "Options for printer". Enter DS <CR> to search for occurrences of Double Space. When you find one, press <CR> to stop finding. Then delete the -----Double Space characters. Repeat this Find-then-Delete routine until you've searched the entire document. Then, go through the same procedure to check for any occurrences of Triple Spacing. Finally, press open-Apple-1 again to get to the top of the file. Press open-Apple-O followed by SS <CR> to place the -----Single Space printer option at the start.
- 4. Print to disk - At this point, you're ready to create the ProDOS textfile. Press open-Apple-P to Print. When AppleWorks prompts you Print from? Beginning accept this default by pressing <CR>. You're presented with the list of printers you have defined. The last choice is A text (ASCII) file on disk. Arrow down to this choice and press <CR>.
- 5. Provide full pathname for textfile - Insert the disk you formatted in step 1 in your data drive. AppleWorks is now prompting Pathname? in the lower left-hand corner. You must supply the *complete* pathname as follows: slash, name of the disk, slash, name of the file, <CR>. In this case, type: /transfer/quiz3 <CR> Notice that the textfile name is different from the AppleWorks Word Processor filename: 3.weekddquiz becomes quiz3. When you don't use periods in the textfile's name, then

you don't need to manually rename the chapter in BEX. AppleWorks should then start "printing" the file to disk. If AppleWorks says, Unable to begin writing this file or Unable to continue writing this file don't try again. Ninety-nine percent of the time, this error message means you have mistyped the disk's name. Press <ESC> and AppleWorks reprompts Pathname? Make sure you're following the /DiskNameestFileName pattern.

- 6. Repeat steps 2 through 5 for every AppleWorks Word Processor file you want to export.
- 7. Boot BEX on a *two-drive* system. - Place the ProDOS /Transfer disk in drive 2. Move to the Second Menu and press R. When BEX prompts: Textfile: remove the BEX program disk, insert a DOS 3.3 data disk, and type 2 <CR> You get a numbered list of ProDOS textfiles. Use the standard BEX chapter selection methods to choose which files to work with. Use 11-P for target chapter naming method, which writes the chapters on drive 1, adding the two characters *hyphen, P* to make the chapter names.
- 8. Use Replace characters - When the AppleWorks data uses patterns of <CR>s and spaces, follow the sample given previously. When the AppleWorks data contains BEX's own (\$p) and (\$l) indicators, just Replace every <CR> with one space.

11 BEX and the VersaBrailles

Telesensory Systems Inc. has marketed two very different devices. The first device was the tape-based *VersaBraille*. TSI now refers to this machine as the *classic VersaBraille*. Beginning in 1986, TSI started selling the disk-based *VersaBraille II*, and in 1987, the disk-based *VersaBraille II Plus*. From here on in, we use a shorthand: *VB* for the tape-based VersaBraille, and *VB II* for the disk-based VersaBraille II and II Plus. Before you attempt to use either VB with BEX, please read the appropriate section in the BEX Interface Guide: Section 8 for the classic VB, and Section 9 for the disk-based VB II.

Part 1: Are You Ready?

This Section assumes that you understand the differences between the two machines and know which one you have. Having read the Interface Guide, you know how to interface your machine to the Apple. For the tape-based VB, you have set up the appropriate overlay chapters. For either VB, you have connected and tested communication between the Apple and the VB or VB II.

Additionally, you must supply appropriate answers in your configuration. For the tape-based VB, you must supply your model letter and slot number when asked. For the disk-based VB II, you must answer Y to Do you have a remote serial device to input text through slot? and define one of your printers as a class P - Paperless brailer.

Back in 1981, the first device supported by BRAILLE-EDIT (the precursor of BEX) was the classic VersaBraille. Thanks to this heritage, many BEX features facilitate transferring information between the Apple and the VB or VB II. BEX and the VB'S use the same symbol to mark a paragraph: *ed* sign, *p*. Dots 1-2-4-6, dots 1-2-3-4 is dollar sign *p* in computer braille--the same as BEX's paragraph (\$p) indicator.

Like BEX, the tape-based VB stores data in units called *chapters*, which consist of some number of *pages*. But the VB page is much smaller than a BEX page: a totally full VB page is 1000 characters compared with 4096 for a BEX page. When BEX transfers information to and from the VB, it automatically preserves the VB page breaks. You can also control this manually--more details in Part 3. The disk-based VB II does not use a chapter and page structure, but it does use (\$p) to mark paragraphs.

Part 2: Three BEX Options for the Tape-Based VB

When you transfer information from the VB to the Apple, you use option F - From VB. When you move information in the opposite direction, you use option T - To VB. Option A - Auto print from VB combines three Main Menu options. It transfers text from the VB to the Apple, then back-translates it, then prints the resulting inkprint chapter to the printer you've defined as number 1 in your configuration.

- **Using Option F - From VB**

Cable the VB to the Apple. Turn on the Apple, then turn on the VB. On the VB, load the FROM VB overlay, and press chord-R R to put the VB in remote. Make sure you have enough room on the data disk in drive 2. To start the transfer, press F at the Main Menu. BEX asks you two questions with supplied defaults.

Main Menu: F

From VB

Do you want control characters? N

When you are transferring a grade 2 chapter, press <CR> to accept the N default. Part 3 discusses when you would answer Y. After you answer the first question, BEX continues:

Do you want VB page indicators? N

Most of the time, you want to press <CR> to accept the N default. Once you answer these questions, BEX prompts: enter chord X H at VB chapter name and then BEX waits for around two minutes.

Now pay attention to the VB. After you load the FROM VB overlay, eject it and load the tape containing the chapter you want to send to the Apple. Find the chapter title in the table of contents, and stay there--do not press the advance bar to enter the chapter. Press chord-X H on the VB and the rest is automatic. You hear the VB tape moving the moment you press the H. When the transfer is finished, the BEX chapters are on disk with the same names as the VB chapters.

- **Using Option T - To VB**

Cable the VB to the Apple. Turn on the power for both devices. On the VB, load the TO VB overlay. When it is loaded, press chord-R R to put the VB in remote. Eject the tape, and load a tape with enough room to contain the text you're sending.

On BEX's Main Menu, press T. Specify a list of chapters to transfer (details on BEX chapter selection appear in Section 4). Once the list is complete, BEX

asks two questions and supplies defaults.

Main Menu: T

To VB

Drive or chapter: 2 <CR>

There are 3 chapters

1SAMPLE1

2SAMPLE2

3SAMPLE3

Use entire list? N Y <CR>

Have you defined all VB page breaks? N

Unless you are *positive* no VB page would end up with more than 1000 characters, press <CR> to accept this default. BEX continues:

Maximum VB page size is: 950

Enter new page size:

Use this default value for material you'll be reading; simply press <CR>.

When you intend to do extensive editing, supply a lower number in the range of 500 to 700.

Once you answer the page size question, BEX starts sending material to the VB. When the transfer is finished, the VB is in local. Before doing another transfer, you need to press chord-R R again.

- ***Using Option A - Auto print from VB***

Auto print from VB combines three Main Menu options: From VB, Back translate from Grade 2, and Print chapters. This one option transfers your grade 2 chapter from the VB to the Apple, runs the Back from Grade 2 translator, and prints the translated text to printer number 1.

Caution! | You must have at least two disk drives to use option A - Auto print from VB.

Before you use this option, you must specify your inkprint printer as printer 1 in your configuration; set top of form and make sure it's on-line. Set up the VB as described above for option F - From VB. Make sure you have enough room on the data disk in drive 2. On the Apple, press A at the Main Menu.

BEX uses the N defaults for control characters and page breaks; the only prompt you get is `Enter chord-X H` to send chapters

Enter the `chord-X H` command on the VB, and the rest is automatic. Isn't technology fun? The Auto print option creates a BEX chapter named XXX containing the back-translated text.

Part 3: Finer Control of VB Transfers

In Part 2, we recommended you use the default values for the questions BEX asks for option T - To VB and option F - From VB. Here we explore the cases when you may wish to use different values.

VB and BEX chapter naming

When a VB chapter is transferred to the Apple, BEX tries to use the same chapter name as the VB'S chapter name. When the VB chapter is named DIARY then the BEX chapter is also named DIARY. However, it's possible for a VB chapter to begin with a character that's not a letter. When this happens BEX inserts `] right bracket` at the beginning of the BEX chapter name. This becomes part of the name, which you must type when specifying BEX chapters individually. When the VB chapter is named `-PUT]` then BEX names it `] -PUT]` once it arrives in the Apple. When this chapter is transferred back to the VB, the `]` is stripped off.

VB page breaks

For option F - From VB, BEX asks you `Do you want VB page indicators?` and supplies a N default. When you answer Y, then `<control-P>` appears in your BEX chapter wherever there's a new VB page. When transferring back to the VB, a `<control-P>` within your text forces a new VB page. You can enter `<control-P>` characters in BEX text that you create for the VB to make better VB format. When you need to reserve ten blank VB pages in a row, just type ten `<control-P>` characters in your BEX page.

When transferring to the VB, BEX asks Have you defined all VB page breaks? with a N default. When you are *positive* that your BEX text contains 1000 or less character between each <control-P> character, then you can answer Y. This way, you can transfer the same text back to the VB with the same page boundaries.

Warning! Only answer Y if you have been very careful about your <control-P> characters. A VB page cannot contain more than 1000 characters. If there are more than 1000 characters between <control-P>s, then some text is overwritten in the VB.

In most situations, let the computer keep track of VB page breaks. During option T - To VB, BEX asks
Maximum VB page size is: 950
Enter new page size:

When you enter a lower value, BEX creates VB pages within the limits you provide, making a smaller page when a new paragraph starts within 100 characters of the page size. BEX tries to divide at the end of a line or a word. If there is a very long word at the end of the page, however, BEX would divide that word. During the transfer, a <control-P> still forces a VB page break.

Control characters

For option F - From VB, BEX asks Do you want control characters? and supplies a N default. Depending on your VB model, some common grade 2 contractions are identical to ASCII control characters. When you choose to answer Y, you do want control characters, make sure that your text does not contain ambiguities. For example, dots 4-5-6, dots 1-3-4 could mean either *many* or <CR> on a model B VB. For a model C, dot 4, dots 1-5 could mean an accented letter or <control-E>.

When using option T - To VB, BEX strips out four control characters that could appear in your text: <control-S>, <control-T>, <ASCII 30>, and <ASCII 31>. The Grade 2 translator places <ASCII 30>, the discretionary linebreak, as it translates. The other control characters would be there because you typed them in.

Part 4: Two BEX Options for the Disk-based VersaBraille II

Please be aware that the tape-based, classic VB and the disk-based VB II are *very different machines*. You can't use the three options described in Part 2 to move text between the Apple and the disk-based VB II. However, two options make transferring data between them quite straightforward. Section 9 in the BEX Interface Guide describes getting the two machines communicating.

- ***Input through slot: from the VB II to the Apple***

On the Second Menu, option I - Input through slot allows you to import data from another computer cabled to the Apple. Complete instructions on Input through slot appear in Section 12. To use Input through slot, you must provide BEX with the slot number of the interface card in your configuration. Check Section 9 in the BEX Interface Guide for what cable to use and how to set the CCP'S on the VersaBraille II.

Once you're ready, press I at the Second Menu. BEX requests a target chapter name. After you supply it, BEX tells you to start sending text from your remote device.

Now pay attention to the VB II. To send text to the Apple, enter F for files, P for print, and S for serial. You're asked if you want to format text. Answer Y when you want the VersaBraille to format text. Answer N when you want BEX to format text.

Next, answer the VersaBraille questions about the device (internal, drive 1 or drive 2), and the file name appropriately. Finally, you're asked if you want to pause; answer N. The file is now moving to the Apple; you hear a steady series of clicks. When the transmission is over, press Q on the Apple keyboard; your chapter is saved to disk.

When the VersaBraille is formatting text, you can reduce blank lines and spaces by entering the appropriate format commands in your VB II file. The

commands .pk to pack text and .op to omit page numbers, as well as the commands that zero the margins, should prove useful.

- ***Printing: from the Apple to the VB II***

To send a file to the VB II, you must define one of the four printers in your configuration as a *class P - Paperless brailier*. This turns off BEX's formatter, so that your entire text is transferred as is, format commands intact. The class P - Paperless brailier also strips out four control characters that could appear in your text: <control-S>, <control-T>, <ASCII 30>, and <ASCII 31>. The Grade 2 translator places <ASCII 30>, the discretionary linebreak, as it translates. The other control characters would be there because you typed them in.

Before you can specify a list of chapters to print, you must tell the VB II to expect some text. When the VB II is receiving text, you can set *VersaBraille emulation* either on or off. We recommend turning VersaBraille emulation off.

To tell the VB to accept a file, enter F for files, P for print, and I for serial in. You are asked for the file name and device (internal, drive 1, or drive 2); supply the appropriate answers. Finally, you are asked Send commands? Answer N by pressing any key except execute. Now specify a list of chapters to print, and direct it to the printer number you defined as a class P printer. When the transmission is over, press chord-Z on the VB II keyboard.

When the file is less than 10,000 characters and you want to examine the text as it's being absorbed, you can turn *VersaBraille emulation* on, by using the communications submenu of the parameters menu. When VersaBraille emulation is on, the procedure is slightly different. Once you specify F, P, and I, commence printing with BEX. Once BEX is back at the Main Menu prompt, press chord-R on the VB II keyboard. Once it arrives in the VB II, the file VERSABRL.SAV contains the text from the Apple. You should immediately rename the file, or it would get clobbered the next time you transfer a file.

12 Importing Text with Input through Slot

There are several ways to get text into BEX. The most common way is to enter text in the Editor, either in print or in braille. A second way is to copy the information from a DOS 3.3 or ProDOS textfile into a BEX chapter. Option R - Read textfiles to chapters is on the Second Menu; it's explained in detail in Section 10.

Section 11 discusses transferring files between the tape-based VersaBraille and BEX. Section 11 also discusses the disk-based VersaBraille: you use option I - Input through slot to send text from the VersaBraille II to the Apple.

Part 1: How Input through Slot Works

Option I - Input through slot on the Second Menu allows you to import information from a remote serial computer device. You establish communication between the remote device and the Apple, and then provide BEX with a new chapter name. From this point on, BEX accepts the information through the interface card in a slot instead of through the Apple keyboard.

BEX pages are limited to 4096 characters, so BEX must be able to send a message to the remote device to pause occasionally. While the device pauses, BEX writes the current page to disk and opens a new page. Finally, BEX must be able to tell the remote device to start sending again. This kind of dialogue between computers is called *handshaking* in the interfacing subculture.

To capture the data, BEX uses option I - Input through slot. How the remote computer *sends* the data can vary from device to device. We have successfully interfaced a wide variety of devices, including the Kurzweil Reading Machine, the IBM-PC, the Macintosh, and the Keynote and SmallTalk notebook computers.

Hint! | The KRM Interface is tricky. Please check out Section 10 in the Interface Guide.

Part 2: Equipment Requirements

There are two basic requirements for the remote device: It must support handshaking and it must be controllable externally. Input through slot supports two handshaking methods or *protocols*: software handshaking and hardware handshaking.

Handshaking issues

Software handshaking is known by several other names: *Xon/Xoff*, *control-S control-Q*, and *flow control* are all terms you may encounter. Software handshaking is universally provided by a *terminal* or *communications* program. This means that one way to send text to the Apple is to run a terminal program on the remote device. Software handshaking is also one way a computer can communicate with a serial printer. You may be able to define the Apple using BEX as a *printer*. Then you can use a word processing program on the remote device to send the data to BEX.

Hardware handshaking is sometimes called *Data Terminal Ready* or simply *DTR* handshaking. Hardware handshaking is the most common way that a computer communicates with a serial printer. Chances are very good that you can tell your remote device to print and capture the data with BEX using Input through slot. Sometimes this requires a word processing program. No other software is required for the MS-DOS and PC-DOS operating systems, as printing is built in to the DOS.

Controlling communications on remote device

During Input through slot, BEX is concentrating on ingesting information, so you can't use the Apple keyboard to tell your remote device anything. This means that you must be able to control the device externally through its keyboard or other buttons.

If you cannot push buttons on the remote device to specify which file to send and when to start sending it, then Input through slot won't work. (That's one of the things you *can* do with a true terminal program. Terminal software lets you run another computer remotely.)

Apple interface cards

Input through slot is very picky. You can only establish communications with three interfaces: an Apple Super Serial Card, an Apple IIc port, or the modem port on a SlotBuster II. You *can't* use an Apple IIgs port for Input through slot.

- ***Getting the Right Connections***

When you get two computers to talk to each other successfully, you've *interfaced* them. Establishing a serial interface requires some patience and experimentation. Once you have interfaced two devices, however, actually sending data is a snap.

The first issue is getting the correct cable. RDC sells a wide variety of cables that work with different interfaces; see Section 14 in the Interface Guide for details.

Once you have the correct cable, you must match the values for four parameters on the sending device and the Apple. RDC'S *standard parameters* for the Super Serial Card, IIc ports, and SlotBuster modem port are:

- Baud rate: 9600
- Parity: None
- Data bits: Eight
- Stop bits: Two

The simplest interface is when you can set these same parameters on your remote device. If you insist on using other parameters, Section 6 of the Interface Guide tells you how to change these values for the Super Serial Card and IIc.

Part 3: Using Input through slot

When you establish a configuration at the User Level, BEX asks Do you have a remote serial device to input text through slot? When you answer Y, you specify the interface card and an optional

set-up sequence. Before you can use Input through slot, you must establish a configuration where you answer Y to this question. Next, you must test the connection to make sure that data is transferring correctly--Part 4 offers some advice. Once the Apple and your remote device are on speaking terms, using Input through slot is simple.

On the Second Menu, press I and BEX asks for a target chapter name. Once you provide this, BEX tells you to start sending data. You then move to your remote device and type the commands or push the buttons that initiate sending text.

BEX clicks once for every character as it enters the BEX page. When the remote device sends characters in a steady stream, you hear a steady tone. When the remote device thinks it's printing, then you hear bursts of clicks for each line. Your baud rate determines the pitch: 2400 baud is a low moan; 19,200 baud is a high shriek.

BEX page size

A BEX page can contain from zero to 4096 characters. When you are back-translating grade 2 text, it's important to limit your pages to around 3200 characters. You can fill your pages more when you're not translating the data. Knowing this, BEX uses two different limits when creating pages during Input through slot.

When the first sector of data is all lowercase or all uppercase, BEX limits the pages to about 3072 characters. BEX assumes this data is braille text, and the 3072 limit minimizes overflow errors when back-translating. When the first sector of data is a mixture of upper- and lowercase, then BEX uses a higher limit for pages; BEX moves to a new page at about 3840 characters. If neither limit suits your preferences, you can modify the chapter when the session's over. Once the target chapter is saved to disk, you can use option A - Adjust pages on the Second Menu to establish page boundaries you prefer.

When the first BEX page is full, BEX tells the remote device to pause. The program tries software handshakes first. If the remote device doesn't respond, then BEX sends a hardware handshake signal.

Controlling Input through slot

During the Input through slot process, you can press three letters on the Apple keyboard:

- Q - Quit the Input through slot process. BEX tells the remote device to stop sending data. Then BEX saves the current page, and builds a directory file for the chapter you've just created.
- P - Force new BEX page. Press P whenever you want a new BEX page in the chapter. BEX tells the remote device to pause, writes the current page to disk, opens a new page, and tells the device to continue sending data.
- T - Type comments in the chapter. Press T and BEX tells the remote device to pause then beeps. Now every key you press boops as the character is placed in the current page. Use this to identify documents and to enter print page indicators. When you've finished typing text, press <CR>, and BEX tells the remote device to send again.

Part 4: Tips and Techniques

When you first work out the interface between two devices, allow yourself enough time. A successful serial interface requires matching a number of parameters on both sending and receiving devices. It's easy to become flustered; when you do, it's unlikely you'll get all the parameters to match.

When you start changing the serial parameters, it's crucial that you keep track of what you're doing. Never change more than one parameter at a time, or you won't know what the correct combination is.

Step-by-step interfacing instructions for Input through slot are frequently published in the RDC Newsletter. Call us to see if you can purchase a back issue that has the details on a specific interface. In addition, keep Section 6 of the Interface Guide close at hand, as it lists the command sequences for the Super Serial Card and IIC ports.

Testing different parameters without reconfiguring

When you're experimenting with different interface parameters, you don't want to turn off your Apple just to change switch settings on the Super Serial Card. As an alternative to flipping switches, you can write a BEX chapter

containing the command sequences to change any parameter. Suppose your Input through slot Super Serial Card is in slot 2. You want to lower the baud rate from the "standard" 9600 to 4800. Edit a new BEX chapter and press control-C I then type 12 B <CR> and control-Q to quit. This chapter contains <control-I> 12 B <CR> which sets the SSC to 4800 baud. When you print this chapter to the SSC, then you change its parameters.

Chances are that you have not configured any of your printers in slot 2. But you can still print the chapter, by defining it as a N printer:

Main Menu: P

Print

Drive or chapter: BAUD <CR>

Drive or chapter: <CR>

Which printer: N <CR>

Old carriage width: 80

New width: <CR>

Old form length 23

New form length: <CR>

Printer slot is 0

Enter new slot: 2 <CR>

No pause on form feed

Change it? N <CR>

The important question is the slot number; respond with the interface card for Input through slot. Your answers for the carriage width, form length, and pause on form feed questions don't matter.

- ***Troubleshooting***

Before you start interfacing the two devices, create some test data on both devices, if you can. The data should follow a recognizable pattern so that you instantly know when characters are missing. The data should include long and short words and long and short lines. Your test data must be large enough to assess whether handshaking is working correctly.

12,000 characters is a good number; BEX must handshake with the remote device for three BEX pages. An ideal test file would be the numbers from 0 to

9 repeated again and again, with different clusters of digits simulating the distribution of letters in words and sentences.

Missing characters

Symptoms of inadequate handshaking usually pop up at BEX page breaks. When the last character of one BEX page and the first character of the next are not sequential in the source data, then you know that BEX is not able to tell the remote device to pause. Some characters are being lost as BEX saves the current page and opens a new page. As mentioned earlier, BEX tries software handshaking first, and then uses hardware handshaking. If your remote device is unusually sluggish to acknowledge software handshaking, BEX may use hardware handshaking instead. The result is no handshaking. RDC may be able to offer advice on how to cope--call the Technical Hotline for assistance.

Garbage characters

When the data you receive is totally gar4, then the first suspect is baud rate. Check to see that both devices are set at the same rate. The next suspect is stop bits: you get garbage characters when the remote device is set for one stop bit while the BEX standard parameter is two stop bits. If you can't change the remote device to two stop bits, then use command sequences to change the Apple interface card.

A rarer cause for totally gar4 data is the number of data bits. Plain text only uses seven data bits; the eighth data bit is used for different functions on different computers. The RDC standard parameter is eight data bits. During Input through slot, BEX always "strips off" the eighth data bit.

When you know that the baud rate and stop bits match yet your data is totally gar4, try changing the remote device to seven data bits while keeping the Apple at eight data bits. While this advice baldly contradicts our earlier statements about matching parameters, it has worked in a number of situations--the VersaBraille II and the Macintosh in particular. Call it voodoo interfacing!

Rare and random garbage characters

When you examine the BEX chapter, you may encounter *occasional* garbage control characters, as well as random punctuation or unexpected capital letters. These characters are the BEX result of special characters like accented letters and copyright or registered trademark symbols. In the Apple II environment, there are 128 possible characters (96 printable ones plus 32 control characters). In many other computers, there are 256 possible characters. These extra 128 characters are known as *high bit set* characters. Input through slot strips off the high bit, so the characters seem like random garbage. When your source file contains a lot of accented letters, you should globally change them before you send the file to the Apple.

- ***IBM-PC Example***

Use a 9F cable to connect an IBM serial port to an Apple Super Serial Card. The male end of the cable connects to the Super Serial Card. The female end of the cable plugs into the IBM serial port. For an Apple IIc, use a 106 cable. When you configure, supply the slot number for the Super Serial Card or IIc port.

Get BEX up and running on the Apple, and get DOS up and running on the IBM-PC. Connect the two devices. Move to BEX's Second Menu and press I, then provide a target chapter name.

The A> prompt appears on the IBM. Type the following:

```
A> MODE LPT1:=COM1: <CR>
A> MODE COM1:96,N,8,2,P <CR>
```

If you're using COM1 for a voice device, you can substitute COM2 in place of COM1. When you are using more than one serial interface on the IBM, you may have to modify the DOS commands slightly. See the DOS manual for details.

You can establish a list of files to print, or *print queue*, using this formula:

```
print <space> /p <space> [disk drive]:[filenames] <CR>
```

To print one file named REPORT . TXT on drive B, you enter:

```
A> print p/b:report.txt <CR>
```

To print this file plus a MEMO . FMT file on drive A, enter:

```
A> print p/b:report.txt memo.fmt <CR>
```

To print every file on drive B, enter:

```
A> print p/b:*. * <CR>
```

Once the queue is defined, you're ready to print. The first time you print on the IBM, you must change the printing device to COM1. DOS tells you:
NAME OF LIST DEVICE PRN=:

If you pressed <CR> here, PRN would become the printing device. But you want to print out the serial port, so you type:
com1 <CR>

and the IBM tells you which files are printed as it sends them. As the IBM begins to transmit, you should hear the Apple squeal. Once the chapter is captured to disk, you can use the FIX TEXT transformation chapter on your BEXtras disk to strip out the <CR><linefeed> at the end of each line and place paragraph (\$p) indicators.

13 Advanced Data Recovery Techniques

In Learner Level Section 13, we discussed how to cope when things went wrong. This Section provides you with three tools to help you cope with most of the problems you could encounter using BEX. One important tool is an understanding of the DOS commands that BEX itself issues. Next, we provide the full details of how option F - Fix chapters works, and present several hints for coping with problem chapters. Finally, we demonstrate the FID utility on the Starting Menu.

Most of the time BEX handles routine housekeeping tasks, like making sure a chapter's directory file accurately describes its page files, perfectly well. However, there are times when BEX becomes befuddled. Fortunately, your human problem-solving skills are a thousand times superior to any computer, so with the right tools, you can almost always put things right.

Every menu has option Q - Quit. When you press Q, the Apple responds with the *BASIC prompt*, the single] right bracket character. The Echo pronounces this as "Ready." You also get the BASIC prompt when you press Control-Reset to stop a BEX option in progress. In the following samples, we show the BASIC prompt where it will appear. Don't type the right bracket itself, just the commands after it.

A complete discussion of Apple DOS 3.3 commands is outside the scope of this manual. Older editions of the *Apple IIe Owner's Guide* contain an in-depth discussion of these commands. However, starting in 1987, all references to DOS 3.3 have been expunged from Apple's Owner's Guides. Fortunately, many excellent books about Apple DOS 3.3 and BASIC programming have been written by third parties. See Appendix 5 for publishers of accessible manuals.

Here are some fundamental facts: Always depress the Caps Lock key when you type DOS commands. DOS commands *always* end with <CR>. In a DOS filename, every character, including punctuation and spaces, is very important; DOS won't be able to find a file on disk unless you type its name *exactly*. If you make a typing mistake in a DOS command, DOS lets you know with the words SYNTAX ERROR and a high beep. If you mistype a filename, DOS responds with FILE NOT FOUND and a high beep. In the following samples, we enclosed the word *filename* in brackets to show where you type the filename of your choosing.

When you have more than one disk drive, you must tell DOS which drive you want to operate on. When you press Control-Reset or Quit from any BEX Menu, DOS'S default drive is the BEX program drive, usually slot 6, drive 1. To direct DOS'S attention to a different drive, you add the slot and drive number to the end of any DOS command. The syntax for this is typing S6, D1 for *slot 6, drive 1* or S6, D2 for *slot 6, drive 2*. Once you supply a slot and drive, DOS assumes all subsequent commands are directed to that drive until you give a new instruction. (At the Master Level, you learn about RAM drives, portions of the Apple's memory that you can configure to hold the BEX program and data.)

\$\$\$vl: Speech and large print at the BASIC prompt

Quitting allows you to use Apple DOS 3.3 while maintaining BEX's control of input and output--speech and large print display. When you press Control-Reset, you temporarily lose BEX's input and output control. To restore large print display, type RUN <CR> and then Quit again. To relink speech at this point, you can type either] PR#0 <CR> to restore speech with 40-column screen or] PR#3 <CR> to restore speech with 80-column screen. When BEX is sending output to the Echo or SlotBuster, it sets the combination of speech and screen output itself. When you type PR#0 or PR#3 at the BASIC prompt, you reset the speech software to "speech only" mode. If you want output to the synthesizer plus the screen, issue the appropriate command.

To relink a serial voice or braille device in slot 5, you type:] PR#5 <CR>
When a serial device is in a different slot, change the digit 5 to that slot number.

- ***The Most Important DOS Command: RUN***

When you are at the BASIC prompt, you return to BEX by typing
] RUN <CR>

It's possible to Quit BEX, type NEW and then create your own Applesoft BASIC program. When you do this, you erase the BEX Menu program from the Apple's memory. To return to a BEX Menu, you need to type RUN followed by the filename of the BASIC Menu program, followed by the slot and drive numbers for the BEX program disk. For the Starting Menu, you type: cher/ RUN START, S6, D1 <CR>. For the Main Menu, you type: cher/ RUN MAIN, S6, D1 <CR>. For the Second Menu, you type: cher/ RUN SECOND, S6, D1 <CR>. For the Page Menu, you type: cher/ RUN PAGEMENU, S6, D1 <CR>

When you press Control-Reset or Quit, you can move directly to a different menu from the one you left by typing cher/ RUN [menuname] , S6, D1 <CR>

- ***Six More DOS Commands***

The BEX Menu programs issue a number of DOS commands as they copy, merge, delete, rearrange chapters and pages within chapters. When you understand the commands that BEX uses, you are better able to issue these commands yourself when BEX is befuddled.

CATALOG

Shows the contents of a DOS 3.3 disk. BEX issues this command when you press <space> after the list of chapters with a BEX D - Disk catalog. When you are having problems with a chapter, it's a good idea to perform a DOS catalog. It shows you every file on the disk. When you Quit BEX and type CATALOG you see every file on the disk in BEX drive 1, usually slot 6, drive 1. To see the files on the disk in drive 2, type:

```
] CATALOG,S6,D2 <CR>
```

The catalog information is shown in four columns. The first column is one character wide: when a file is locked, position 1 contains an asterisk. The second column is also one character wide: it contains a single letter that corresponds to the type of file. For binary files, the letter is *B*, for Applesoft BASIC files the letter is *A*, for textfiles the letter is *T*, and for Integer BASIC files the letter is *I*. The third column occupies positions four through six; it always contains a three-digit number. This number is the sector count for the file. From position eight on to the end of the line is the fourth column; it contains the filename.

INIT HELLO and CLOSE

Initializes a disk, preparing it to save files. This sequence of *two* commands does the same thing as option I - Initialize disks on the Starting Menu. Be cautious when using this command: any information on the disk is wiped out when it's initialized. While Initialize disks on the Starting Menu prevents you from initializing your BEX program disk by mistake, no such protection exists when you type INIT HELLO at the BASIC prompt. Assuming the disk to initialize is in drive 2, this is the safest way to proceed.

```
] CATALOG,S6,D2 <CR>
```

DOS gives you a listing of all files on the disk, allowing you to be sure you don't need them.

```
] INIT HELLO <CR> %%%ms0%%m10
```

Actually starts the initialization process. If DOS reports I/O ERROR then you should throw the disk straight in the trash. Finally, type] CLOSE <CR> to prepare the Apple's memory to run another program.

DELETE [filename]

Permanently erases a file from the disk. When you kill a chapter, BEX issues one more DELETE command than the number of pages in the chapter. One DELETE command is required for each page file, and then one more for the chapter directory file. BEX uses DELETE whenever you Kill chapters, textfiles, configuration files, or pages. In Part 3, we demonstrate when you might wish to DELETE files instead of letting BEX do it for you.

RENAME [old filename] [new filename]#

Changes a file's name on disk. Second Menu option N - Name change for chapters uses the RENAME command to change the names of the page and directory files. As it renames the page files, it reletters them to follow their numerical order. This requires a fresh directory file, which BEX saves on top of the old one. A situation where you'd use RENAME is described in Part 3.

Caution!

DOS 3.3 unfortunately has a bug regarding file names: This operating system allows you to have *more than one* file with the same name on a given disk. This is why you should never copy a chapter to the same disk with the same name, or rename a chapter to a name that already is on a disk. We discuss ways of coping with this problem in Part 3.

LOAD SAVE BLOAD and BSAVE

Warning!

The following information is here to whet your curiosity. When you use these commands, it's quite possible that you could damage some of the programs on your BEX disk. It's also possible that you can learn a lot about Applesoft BASIC, by exploring the Apple's power in great detail. Only you can guarantee the safety of your BEX disk by always using a back-up copy.

The LOAD command copies information from disk into the Apple's memory. The SAVE command is its opposite; it copies information from the Apple's

memory to a file on disk. You can only LOAD and SAVE Applesoft programs, a type of file identified by the letter *A* in the second column of a DOS catalog. All of BEX's menu programs are Applesoft BASIC. Once it's loaded into memory, you can look at an Applesoft program by typing LIST at the BASIC prompt.

Binary files, (identified by the letter *B* in the second column of a DOS catalog), use a variation on the LOAD and SAVE commands. BLOAD copies information from disk into the Apple's memory, and BSAVE copies from memory to disk. Many of BEX's smaller programs are binary files, and all BEX chapters are composed of binary files. Whenever BEX is reading and writing chapters between memory and disk, it's issuing BLOAD and BSAVE commands. BLOAD and BSAVE commands are frequently accompanied by a specific *address* in memory. For example, when BEX copies a 2795-character page into the page buffer, BEX asks DOS to BLOAD beginning at the address A\$9500 a file of length L\$2795

Eavesdropping on BEX with MON C

Perhaps all this talk about DOS commands has piqued your curiosity, and you want to know more. There's an interesting Apple feature that lets you watch a program as it's running. One way to understand DOS housekeeping is to look over BEX's shoulder. At the BASIC prompt, type:

```
] MON C <CR>
```

and all the DOS commands that BEX issues become visible to you. For example, when you Quit from the Second Menu, type MON C and then RUN you eavesdrop on BEX telling DOS to CLOSE any open textfile, then allocate a lot of memory for the program with MAXFILES 2

To stop eavesdropping on BEX, type:

```
] NOMON C <CR>
```

Using MON C doesn't fix anything by itself; it just lets you see how BEX works when things are going smoothly. Before using MON C make sure that you've made lasting copies of anything you want to keep.

Warning! | Monitoring the computer while it's running BEX can easily overwhelm the Apple, resulting in a "deep" crash. In a "deep" crash, the DOS in the Apple's memory is incorrect, and even

simple DOS commands like CATALOG yield ?SYNTAX ERROR. In particular, using MON C in the Editor or during Fix chapters is guaranteed to create havoc. You probably will have to turn off the Apple and reboot. But a "deep" crash doesn't really *harm* anything. As long as all essential data is saved to disk, sit back and enjoy the show!

- ***DOS Error Messages***

When BEX gets overwhelmed, it usually crashes with an error message. These error messages can help you diagnose the problem and point the way to a solution.

BREAK IN LINE four digit number

As we said in Learner Level 13, you may encounter the BREAK IN LINE followed by a four-digit number error message. From that discussion, you know that when BEX says BREAK your BEX disk is not broken. Rather, it means that the BASIC program crashed at a particular line number. When BEX crashes at line 7410, it means there's some problem writing to disk. Perhaps you specified an illegal chapter name? Maybe the disk drive door is open? When BEX crashes at line 7105, then there's a problem reading from disk. You want to remove and insert the disk a few times to make sure it's seated correctly in the drive.

OUT OF MEMORY error

Whenever BEX presents a list of chapters, it's doing a lot of analytical work behind the scenes. So when you press D for a BEX catalog, or enter a drive number to scan the disk, BEX uses blocks of the Apple's memory to build up the chapter list for your perusal. When you ask BEX to create chapter lists several times in a row, BEX can run out of room in the Apple's memory. The result is that you crash with an OUT OF MEMORY error message. Typing RUN <CR> solves this problem.

SYNTAX ERROR IN 65239

When BEX crashes with this error message, it's a symptom of major disorganization in the Apple's memory. The actual number can vary; it's always five digits long and begins with 65. There's a chance that you can recover, so type `RUN MAIN, S6, D1 <CR>` at the BASIC prompt. When you try this two times in a row and just get the same `SYNTAX ERROR IN 65239` message, you must reboot.

The Asterisk Prompt: Crashing into the Monitor

The *Monitor* is a unique feature deep inside the Apple. It allows programmers to directly examine the contents of the Apple's memory. When BEX encounters problems, it usually crashes to the right-bracket BASIC prompt. When BEX gets *really* confused, you could crash directly into the Monitor. The Monitor's prompt is an asterisk (the Echo calls this character star). When you crash into the Monitor, you get a star and then at least one line of what looks like totally incomprehensible garbage. (It's actually useful information about the contents of the Apple's memory, expressed as base 16 numbers.)

To leave the Monitor and get back to the BASIC prompt, press control-C <CR>. Try this twice; if this doesn't give you the right-bracket prompt, it's time to reboot.

Part 2: Advanced Techniques for Recovering Chapters and Pages

In Learner Level Section 13, we gave an example of using option F - Fix chapter directory on the Second Menu. You use Fix chapters when one or more pages of data seem to have disappeared from a BEX chapter, and you know you haven't killed those page files. A thorough understanding of how Fix chapters works can help you cope when Fix chapters doesn't solve the problem.

- ***Fix Chapters in Detail***

BEX adds a two-character extension to the chapter name to make a unique identifier for each page file. The first character of this extension is always period, the second character is usually a letter. (More details in Learner Level Section 12, part 3.) There are only 26 letters in the alphabet, and a BEX chapter can contain up to 30 pages. For the last four page file extensions, BEX uses the left bracket, the backslash, the right bracket, and the caret, in that order.

Suppose you've created a chapter named `TERM PAPER` that's 20 pages long. When you print it, some of the data is missing, yet you know you did not kill any pages. Time to try Fix chapters. After you press F at the Second Menu, you must type in the exact name of the chapter you want to fix. BEX then attempts to find any page file that *could* exist for a chapter with that name.

First, BEX tries to BLOAD a file named `TERM PAPER . A` into the page buffer. When `TERM PAPER . A` exists on disk, then this load is successful, and BEX counts the characters in the page buffer and creates an entry in the new directory file for that page. When a file named `TERM PAPER . A` does not exist, then BEX tries again with the next possible page extension. When `TERM PAPER . B` exists, then BEX creates an entry in the new directory file. BEX repeats this process for every possible page.

Normally, the directory file for a chapter keeps track of which file on disk corresponds with which BEX page number. When you cut pages in the Editor, or change the order of pages with the Page Menu, it's quite possible that page 5 corresponds to `TERM PAPER . A` and `TERM PAPER . G` corresponds to page 1. Because Fix chapters searches for page files in alphabetical order, the directory file it creates always makes `TERM PAPER . A` page 1, `TERM PAPER . B` page 2, and so forth.

After you use Fix chapters, you may have to rearrange the pages with the Page Menu or the Clipboard. If you Zip to the Page Menu and check a newly-fixed chapter with File list, you may find that the chapter has some pages with zero characters. As we mentioned in Section 5, Part 3, you should avoid creating a BEX page with zero characters by deleting text in the Editor. That's because you'll end up with residual files on disk. But when Fix chapters creates a zero page, however, it doesn't create an empty disk file-- just an entry in the chapter directory. If the `TERM PAPER` chapter contained zero characters in page 1, for example, it means that BEX couldn't locate `TERM PAPER . A`

When Fix chapters can't find the page files

Fix chapters moves through the alphabet as it tries to BLOAD page files. Fix chapters gives up if it tries and fails 20 times in a row. Suppose your TERM PAPER chapter started out at 25 BEX pages. You deleted the first sixteen pages. Then you copied this chapter to a too-full disk, and BEX crashed before it created the directory file. Next time, press # to check the free sector count!

When you ask Fix chapters to recover the TERM PAPER chapter, it tells you that no page files could be located. Fix chapters has tried to BLOAD the page files named TERM PAPER . A through TERM PAPER . P and those files don't exist. Since Fix chapters has failed 20 times in a row, it has given up. The solution is to change the page file names manually to earlier letters in the alphabet. Quit BEX and catalog the full disk with the problem TERM PAPER chapter. Just to make typing easier, the following sample uses the shorter name TP Here's what you type:

```
] RENAME TERM PAPER.V,TP.A <CR>  
] RENAME TERM PAPER.W,TP.B <CR>  
] RENAME TERM PAPER.X,TP.C <CR>  
] RENAME TERM PAPER.Y,TP.D <CR>
```

Now you can go through the process outlined above to make room for the directory file then use Fix chapters with target chapter TP.

- ***Coping with DISK FULL Errors***

The directory file is always the *last* file of a chapter written to disk. When some event interrupts the saving of a BEX chapter to disk, then it's likely that the disk contains some of the page files, but no directory file. This can easily happen when the disk you copy to is close to full. Before you create a target chapter by translating, replacing, copying, adjusting, or whatever, you should check that there's enough room on the target disk. You can obtain a free sector count by pressing # at any menu prompt.

When a disk fills up before the directory file is created, Fix chapters alone won't help you. BEX can find all the page files without problem, but there

won't be room to write the new directory file to disk. Before you can use Fix chapters, you must make room for the directory file.

First, Copy any *other* chapters from the problem disk to another disk. Then use Kill chapters on the problem disk to delete the chapters you just copied. Now there's enough room for Fix chapters to write a new directory file.

If you already have copies of every chapter on the disk except the problem chapter, you can use a shortcut. Quit BEX, CATALOG the disk, and then erase just one page file from the target disk by typing DELETE [filename] <CR> at the BASIC prompt. This makes enough room to save a directory file, so you can use Fix chapters. After Fix chapters is successful, you copy the just-fixed chapter to another data disk, and initialize the problem disk.

- ***When a Disk Contains More than One File with the Same Name***

As mentioned in Part 1, DOS 3.3 does not prevent you from renaming one file to a name that's already taken. We're not talking here about *overwriting* an old chapter with a new chapter. When your target disk already contains a chapter named INVOICES and you copy a chapter to the name INVOICES on this disk, the new INVOICES information replaces the old. And usually, that's exactly what you want to happen.

You only encounter the duplicate filename problem when you are *renaming* chapters, or when you try to create a BEX chapter with the same name as a textfile (or vice versa). Suppose your target disk contains two chapters: a two-page chapter named INVOICES and a BILLS chapter containing three pages. A DOS catalog of this disk shows:

```
B 010 INVOICES.A
B 015 INVOICES.B
B 003 INVOICES
B 016 BILLS.A
B 008 BILLS.B
B 008 BILLS.C
B 003 BILLS
```

Further suppose you move to the Second Menu and use option N - Name change for chapters. You specify BILLS as your source chapter and INVOICES as your target chapter. After BEX has renamed the chapter, a DOS catalog of this disk reveals:

B 010 INVOICES.A
B 015 INVOICES.B
B 003 INVOICES
B 016 INVOICES.A
B 008 INVOICES.B
B 008 INVOICES.C

BEX uses the RENAME command to change BILLS.A to INVOICES.A, BILLSDDDB to INVOICES.B, and BILLSDDC to INVOICES.C. While it isn't evident in this sample, Name change for chapters reletters the page files so they are in alphabetical order. BEX's final step in Name change is saving the new directory file. In this case, the new INVOICES overwrites the old INVOICES. When you ask to print INVOICES, you get the contents of the chapter that used to be named BILL.

We hope this problem remains completely hypothetical for you. But if you should encounter it, you can use Fix chapters to recover. In our sample, copy INVOICES to BILLS. Then, Kill INVOICES. Now, you have the two page files left from the original INVOICES chapter. Create a new directory file by using Fix chapters with target chapter INVOICES.

FILE TYPE MISMATCH error

Textfiles are type *T* files and BEX chapters are type *B* binary files. There are some DOS commands that only work for textfiles, and some others that only apply to binary files. If you try to do a binary file operation on a textfile, or vice versa, DOS 3.3 reports a FILE TYPE MISMATCH error.

When a disk contains a BEX chapter named APPENDIX one binary file named APPENDIX is its directory file. If you told BEX to Write a textfile named APPENDIX on that same disk, DOS 3.3 would complain with the FILE TYPE MISMATCH error. That's because BEX has asked DOS to OPEN a type T file named APPENDIX, but DOS has found a type B file named APPENDIX that can't be OPENed.

You would encounter the same error message if a textfile named APPENDIX existed on disk and you tried to copy a chapter to the name APPENDIX. BEX asks DOS to BSAVE the directory file named APPENDIX, and DOS is upset because you can't BSAVE a textfile. If you tried to edit a new chapter named APPENDIX, you would get a `Disk read error` message from BEX.

Part 3: FID

FID stands for *File Developer*. The FID utility program helps you manage DOS 3.3 files. FID lets you copy, delete, lock, and unlock files, as well as cataloging disks. Some FID functions can also be done manually, but FID lets you act upon many files at once.

FID is a binary file on BEX's Boot side. When you press F at the Starting Menu, BEX BRUN'S the FID program. (You don't need to have booted BEX to use FID, however. Once you have booted *any* DOS 3.3 program, you can insert the BEX boot side and start the program by typing `BRUN FID` at the BASIC prompt.) The FID software is incompatible with BEX large print. In fact, once FID is loaded, many of BEX's software pointers are erased. That's why you must reboot BEX after using FID.

We've modified the original FID to make it a little easier to use with speech. As with BEX, FID lists its numbered options when you press <CR> at its `FILE DEVELOPER ENTER OPTION` prompt. You must depress the Caps Lock key for FID to correctly interpret your responses. To make a choice, you type a number followed by <CR>. The options are: Since FID'S output is all uppercase, the remaining samples are transcribed in reverse capitalization mode

- 1 - copy files
- 2 - catalog
- 3 - space on disk
- 4 - unlock files
- 5 - lock files
- 6 - delete files
- 7 - reset slot & drive

8 - verify files

9 - quit

FID allows you to search for files on a disk without knowing their names. When FID prompts `filename` you can specify the exact file, or you can use `=` (equals sign) for any portion of a filename you don't know or want to specify. The equals sign functions as a wildcard. When you type `m=d` FID finds any file that begins with `M` and ends with `D`, no matter how many characters come between these two letters. Whenever you use the wildcard, FID asks you if you want *prompting*. When you answer no, then FID performs the requested action without further intervention. When you answer yes, FID presents each filename that matches the wildcard, and waits for your Y or N decision.

For example, suppose you want to copy some of your configurations on to another disk. The four character suffix `.con` identifies BEX configuration files. Here's how that dialogue goes:

enter option 1 <CR>

copy files

source slot? 6 <CR>

drive? 1 <CR>

destination slot? 6 <CR>

drive? 2 <CR>

filename `=.con` <CR>

do you want prompting? y <CR>

insert disks. press <ESC> to return to main menu or any other key to begin

filename: `=.con`

filename default.con y <CR>

done

filename dm.con n <CR>

cancelled

filename nj.con y <CR>

done

filename thiel.con n <CR>

cancelled

filename teach.con y <CR>

done

You can use the wildcard for all FID'S options. Since deleting a file by mistake can be tragic, it's a good idea to take advantage of the prompting

feature illustrated above when deleting files. For less destructive operations like LOCK and UNLOCK, you can answer n <CR> to the do you want prompting? question. At the Master Level, we recommend that you lock all the files on the BEX Main side before you load the software on to a RAM drive. You can use the equals sign wildcard with FID'S option 5.
