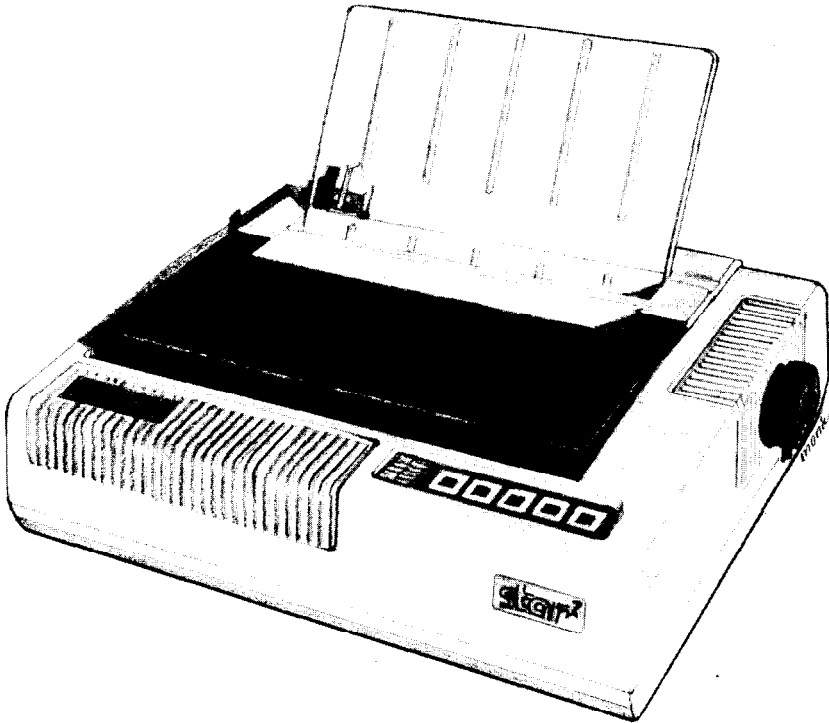


# **Radix**

## **USER'S MANUAL**



NOT INTENDED FOR SALE

**Federal Communications Commission  
Radio Frequency Interference  
Statement**

This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the U.S. Government Printing Office, Washington, D.C., 20402, Stock No. 004-000-00345-4.

For compliance with Federal Noise Interference Standard, this equipment requires a shielded cable.

**A note about the programs in this manual:**

This manual contains several programs that help to demonstrate the versatility of the Radix printers. Star Micronics has made every effort to insure that the programs are functional and accurate. However, Star Micronics cannot guarantee their accuracy or suitability to any particular application.

**Trademark Acknowledgement**

**Radix-10, Radix-15, grafstar:** Star Micronics

**Apple, Apple II, Apple II + , Apple IIe, Applesoft:** Apple Computer Inc.

**Compaq:** Compaq Computer Corporation

**CP/M:** Digital Research

**EasyWriter II:** Information Unlimited Software, Inc.

**IBM Personal Computer, IBM PC, IBM XT:** International Business Machines Corp.

**Kaypro:** Kaypro Computer Corporation

**Microsoft BASIC:** Microsoft Corporation

**Osborne 1:** Osborne Computer Corporation

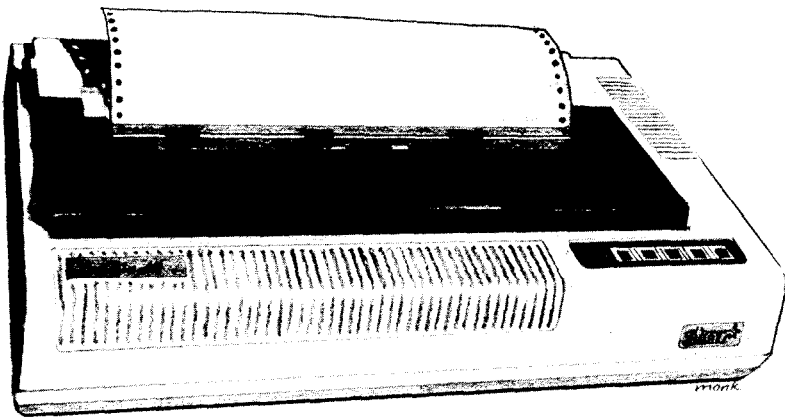
**PeachText:** Peachtree Software Incorporated, an MSA Company

**SuperCalc:** Sorcim Corporation

**Scriptit, TRS-80:** Radio Shack, a division of Tandy Corporation

**WordStar:** MicroPro International Corporation

©Copyright 1984 Star Micronics, Inc.



## ***A Special Message to the New Owner***

You're to be congratulated on selecting the printer of choice for both the sophisticated as well as the first-time user/owner — the new Radix!

Right now, before you even start readying your Radix for action, we'd like to impress you with these two thoughts:

1. In as few words as possible, we'll highlight the several special features that Radix offers you, and
2. We'll show you how this manual can help you get the most from your Radix, while saving you time, effort, and money.

Taking up the special features first, so they'll be fresh in your mind as you ramble through this manual . . . specifically . . .

**Speed** — At 200 characters per second top printing speed, it's one

of the fastest in its class. And Radix is smart too: when printing blank spaces, Radix speeds up to a blistering 240 CPS!

**16K Memory** — Also called *print buffer*. Radix has a mind of its own! Buffer memory holds over 16,000 characters or about 8 printed pages, thus allowing your printer to accept information as fast as your computer can send it. Since computer speed is faster than printer speed, this feature frees your computer to do other things while your printer continues to print.

**472 Characters** — Allows printing in no less than nine different fonts or type faces, including a brand new face which we call . . .

**Near Letter Quality** — A solid black dot-free, high-resolution type face that looks more like typewriter than computer-generated printing. Perfect for correspondence.

**Faster Paper Handling** — More economical, too. Automatic feeding for both single sheets and sprocket paper. And the unique built-in tractor design — behind the platen — avoids wasting a sheet each time you start printing, as in conventional loading. It also permits “reverse paper feed,” for multiple column printing or other special applications, with a neater appearance, too.

**Graphics** — If you're designing your own, you'll be delighted at finding three different dot graphic densities with varying degrees of resolution or sharpness. There's even a quadruple density, with 240 dots per inch horizontal by 72 dots per inch vertical! And, you can print double density graphics at *double speed*!

**Macro Instruction** — A real timesaver on the keyboard. This feature allows you to define a sequence of codes and call (transmit) that entire sequence with a single code.

**Easy Interfaces** — Both parallel and serial interface capabilities are built into Radix — there's nothing extra to buy.

**Easy Everything!** — All the DIP switches are quickly accessible for ease in connecting your computer and changing print parameters; the ink ribbon comes in its own enclosed cartridge, ready to snap into place; paper is machine-fed, not cranked into place manually. Easy is the word for Radix!

We think you'll also find this manual easy and pleasant to use. We've gone to great lengths to make it so. As a first example, look over the table of contents and you'll see what we mean. Whether greenhorn or wizard, everybody will find what they need to know to fulfill their expectations. We suggest that each new user/owner, before you even unpack the box, read or at least scan Chapters 1 and 2 — “Getting to Know Your Radix” and “Getting Started with Radix” — as well as Appendix A, “Setting Up Radix.” Now you can unpack the box and start putting things together.

When you're ready to connect your computer to your Radix, look at Appendices B through E for directions applying to your

make of computer. Remember, Radix has both serial and parallel interfaces, so there's nothing extra to buy!

If you're not a programmer, you'll be most interested in Chapters 3, 4, and 5. They explain — in non-programmer's language — how to get the most from Radix using some of the most popular software packages on the market today. You won't need to know a word of BASIC!

For you who wish to design your own characters, do your own plotting, your own infinite variety of dot graphic patterns and densities, you'll have a ball! For you, Chapters 7 through 12 are a must, and of course everybody should look at Chapter 14, which tells how to maintain your Radix for a long and carefree life.

In this manual there are plenty of example programs to demonstrate and show off all of Radix's features. There are even two utility programs included: one which allows you to design your own printing characters on your computer screen and one to set up Radix the way you want it with just a few keystrokes. Since many Radix users have IBM Personal Computers (or the equivalent) all the example programs are written in Microsoft BASIC for the IBM. But throughout the manual, users of other computers will find hints on how to make Radix work with their computer. And in the appendix, complete translations of the utility programs for several popular computers are included.

So, gentle reader, with this manual we hand you the key to the wonderful world of Radix. May you enjoy years of handsome, fast, and carefree printing!



# Table of Contents

<b>Chapter 1</b>	<b>Getting to Know Your Radix</b>	<b>1</b>
	Components and Controls	
	Paper Selection and Loading	
	Loading single sheets	
	Loading sprocket-feed paper	
	Ribbon Installation	
	Adjusting the Gap	
	Self-Test	
	Some Tips for Smoother Operation	
<b>Chapter 2</b>	<b>Getting Started With Radix</b>	<b>15</b>
	Using Commercial Software	
	First, some terminology	
	The escape code	
	Using this book without learning BASIC	
<b>Chapter 3</b>	<b>Word Processing With Radix</b>	<b>21</b>
	General Concepts	
	The escape code	
	The master reset code	
	Using Near Letter Quality (NLQ)	
	Getting the most from your print choices	
	Using Radix With EasyWriter II	
	Redefining pitch settings and print control codes	
	A sample printout with EasyWriter II	
	Redefining your own print pitches	
	Redefining your own print control keys	
	Using Radix With PeachText	
	Entering special function codes	
	Using boldface print	
	Underlining with Radix	
	A sample printout with PeachText	
	Storing documents with function codes	
	Using Radix with WordStar	
	User-defined print commands	
	Using Radix with Scripsit	
	Defining user print codes	
	A sample printout with Scripsit	
	Redefining your own print codes	
	Using Radix With AppleWriter II	

---

<b>Chapter 4</b>	<b>Using Radix With Spreadsheet Programs</b>	<b>39</b>
	Using Radix with VisiCalc	
	Using Radix with SuperCalc	
	Sending control codes from SuperCalc	
	Lotus 1-2-3	
<b>Chapter 5</b>	<b>Using Radix With Graphics Programs</b>	<b>47</b>
	SuperCalc <sup>3</sup>	
	BPS Business Graphics	
	Making pie graphs round	
<b>Chapter 6</b>	<b>Controlling Radix With BASIC</b>	<b>53</b>
	Some Basics About BASIC	
	Establishing communications	
	The CHR\$ function	
	Control Codes	
	The escape code	
	Some problem codes	
<b>Chapter 7</b>	<b>Printing Text With Radix</b>	<b>59</b>
	Some Special Kinds of Text	
	Near Letter Quality characters	
	Italic printing	
	Underlining	
	Superscripts and subscripts	
	Changing the Print Pitch	
	Expanded print	
	Making Radix Print Darker	
	Mixing Modes	
	Summary	
<b>Chapter 8</b>	<b>Line Spacing and Forms Control</b>	<b>75</b>
	Starting New Lines	
	Reverse line feeds	
	Changing Line Spacing	
	Moving down the page without a carriage return	
	Forms Controls	
	Form feed	
	Reverse form feed	
	Changing the Page Length	
	Top and Bottom Margins	
	Summary	
<b>Chapter 9</b>	<b>Formatting Your Output</b>	<b>89</b>
	Using Horizontal Tabs	
	A one-shot tab command	
	Setting Left and Right Margins	
	Using Vertical Tabs	
	A one-shot vertical tab command	
	Summary	



---

<b>Chapter 10</b>	<b>Special Features of the Radix Printer</b>	<b>97</b>
	Now hear this	
	Initializing Radix	
	Putting Radix to sleep	
	Printing to the bottom of the sheet	
	Backspace and delete	
	Unidirectional printing	
	The seven bit dilemma	
	International character sets	
	The macro control code	
	Summary	
<b>Chapter 11</b>	<b>Creating Your Own Characters</b>	<b>109</b>
	Dot Matrix Printing	
	The Print Matrix	
	Defining Your Own Characters	
	Rule 1: Download characters are seven dots high	
	Rule 2: Dots cannot overlap	
	Add up each column of dots	
	Assigning a value to your character	
	Download character definition command	
	Printing Download Characters	
	Proportional Characters	
	Defining proportional characters	
	Printing proportional characters	
	Connecting characters	
	Mixing Print Modes With Download Characters	
	A Utility Program	
	Instructions for using DLEDIT.BAS	
	Summary	
<b>Chapter 12</b>	<b>Printing With Dot Graphics</b>	<b>139</b>
	Comparing Dot Graphics With Download Characters	
	Using the Dot Graphics Commands	
	Specifying the number of columns of dots	
	Specifying the graphics data	
	Combining text and graphics	
	Printing a Design or Logo	
	Plotting With Radix	
	How the program works	
	Using Radix for business graphics	
	High Resolution Graphics	
	If You Have Problems With BASIC	
	Summary	
<b>Chapter 13</b>	<b>Putting Radix to Work For You</b>	<b>161</b>
	Radix Setup Program	

---

<b>Chapter 14</b>	<b>Basic Maintenance</b>	<b>171</b>
	Cleaning Radix	
	Replacing the Ink Ribbon	
	Replacing a Fuse	
	Replacing the Print Head	
<b>Appendix A</b>	<b>Setting Up Radix</b>	<b>181</b>
	Where Shall We Put It?	
	What Have We Here?	
	Removing the printer covers	
	Removing packing and shipping screws	
	Installing the platen knob	
	Installing the ribbon cartridge	
	Connecting Radix to Your Computer	
<b>Appendix B</b>	<b>IBM Personal Computer and Compaq Computer</b>	<b>189</b>
	Connecting Radix to an IBM	
	Connecting with the parallel interface	
	Connecting to the serial interface	
	BASIC Programming	
	Listing programs	
	Printing Graphics Screens	
	Program Listings	
<b>Appendix C</b>	<b>Apple II Computers</b>	<b>195</b>
	Setting the Switches	
	Applesoft BASIC	
	Listing programs	
	Program Listings	
	Download character editing utility	
	Piechart program	
	Printer setup utility	
<b>Appendix D</b>	<b>TRS-80 Computers</b>	<b>213</b>
	Setting the Switches	
	TRS-80 BASIC	
	Listing programs	
	Program Listings	
	Download character editing utility	
	Piechart program	
	Printer setup utility	

---

<b>Appendix E</b>	<b>Kaypro, Osborne and Other CP/M Computers</b>	<b>231</b>
	Setting the Switches	
	Using MBASIC	
	Listing programs	
	Program Listings	
	Download character editing utility	
	Piechart program	
	Printer setup utility	
<b>Appendix F</b>	<b>The Parallel Interface</b>	<b>249</b>
	Functions of the Connector Signals	
<b>Appendix G</b>	<b>Serial Interface Specifications</b>	<b>253</b>
	Configuring the Serial Interface	
	Radix's Serial Protocols	
	Serial busy protocols	
	XON/XOFF protocol	
	ACK protocol	
<b>Appendix H</b>	<b>DIP Switch Settings</b>	<b>259</b>
	Switch Functions	
<b>Appendix I</b>	<b>ASCII Codes</b>	<b>263</b>
<b>Appendix J</b>	<b>Character Style Charts</b>	<b>269</b>
<b>Appendix K</b>	<b>Function Code Reference</b>	<b>281</b>
	Commands to Control Print Style	
	Font style controls	
	Font pitch controls	
	Special print modes	
	Commands to Control Vertical Position of Print Head	
	Line feed controls	
	Form feed controls	
	Commands to Control Horizontal Position of Print Head	
	Download Character Commands	
	Commands to Control Graphics	
	Macro Instruction Commands	
	Other Commands	
<b>Appendix L</b>	<b>Command Summary in Numeric Order</b>	<b>313</b>
<b>Appendix M</b>	<b>ASCII Code Conversion Chart</b>	<b>315</b>
<b>Appendix N</b>	<b>Technical Specifications</b>	<b>323</b>
<b>Index</b>		<b>327</b>
<b>Consumer Response</b>		<b>331</b>

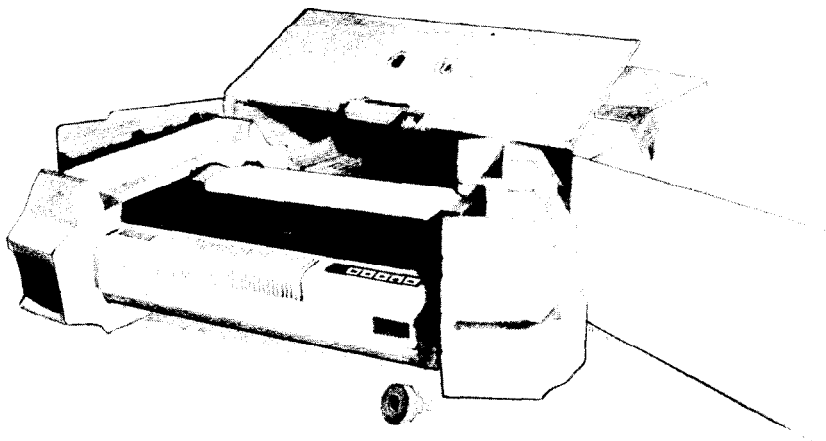
<b>Star Service Centers</b>	<b>332</b>
<b>Warranty</b>	<b>333</b>
<b>DIP Switch Quick Reference</b>	<b>334</b>
<b>Command Quick Reference</b>	<b>Inside back cover</b>

# Table of Tables

Table 1-1	Left margin on the single sheet guide	7
Table 3-1	Easywriter II control keys	27
Table 3-2	PeachText print functions	32
Table 4-1	Visicalc control codes	40
Table 4-2	Print columns on a page with Visicalc	41
Table 4-3	Print columns on a page with SuperCalc	42
Table 4-4	Print columns on a page with 1-2-3	45
Table 5-1	Rounding factors for console modes	51
Table 7-1	Near letter quality commands	61
Table 7-2	Italic commands	61
Table 7-3	Underline commands	62
Table 7-4	Superscript and subscript commands	63
Table 7-5	Print pitch commands	64
Table 7-6	Expanded print commands	66
Table 7-7	Print emphasis commands	67
Table 8-1	Line feed commands	77
Table 8-2	Line spacing commands	80
Table 8-3	Form feed commands	83
Table 8-4	Form length commands	83
Table 8-5	Top and bottom margin commands	84
Table 9-1	Horizontal tab commands	91
Table 9-2	Left and right margin commands	92
Table 9-3	Vertical tab commands	95
Table 10-1	Bell commands	98
Table 10-2	Some miscellaneous commands	99
Table 10-3	Printing direction commands	101
Table 10-4	Eighth bit control commands	102
Table 10-5	International character set commands	104
Table 10-6	International character sets	105
Table 10-7	Macro instruction commands	106
Table 11-1	Download character definition commands	122
Table 11-2	Download character printing commands	126
Table 11-3	Mixing download characters with various print modes	130
Table 12-1	Calculating n1 and n2	141
Table 12-2	Dot graphics commands	156
Table 13-1	Menus of Radix setup program	162
Table B-1	Recommended DIP switch settings for IBM-PC	190
Table B-2	IBM-PC parallel cable	190
Table B-3	Serial switch settings	191
Table B-4	IBM-PC serial cable	191

---

Table C-1	Recommended DIP switch settings for Apple	195
Table C-2	Apple parallel cable	196
Table D-1	Recommended DIP switch settings for TRS-80	213
Table D-2	TRS-80 Model I parallel cable	214
Table D-3	TRS-80 Model II parallel cable	214
Table E-1	Recommended DIP switch settings for Kaypro	231
Table E-2	Kaypro parallel cable	232
Table E-3	Osborne 1 parallel cable	232
Table F-1	Parallel interface pin functions	251
Table G-1	Serial interface pin functions	254
Table G-2	DIP switch B	255
Table G-3	Handshaking protocols	255
Table G-4	Data transfer rates	255
Table H-1	DIP switch settings	260
Table H-2	International character sets	262



## Chapter 1

# Getting to Know Your Radix

The more you learn about Radix and its sophisticated features, old and new, the better Radix is going to perform for you. Remember, it's not just what you know — it's what you know how to use! So, let's start getting acquainted!

**Subjects we'll cover in this chapter include:**

- **Components and controls**
- **Paper-out and front-cover-open detectors**
- **Paper selection and loading**
- **Adjusting the gap — for different paper thickness**
- **Self-test — printout of available characters**
- **Some tips for smoother operation**

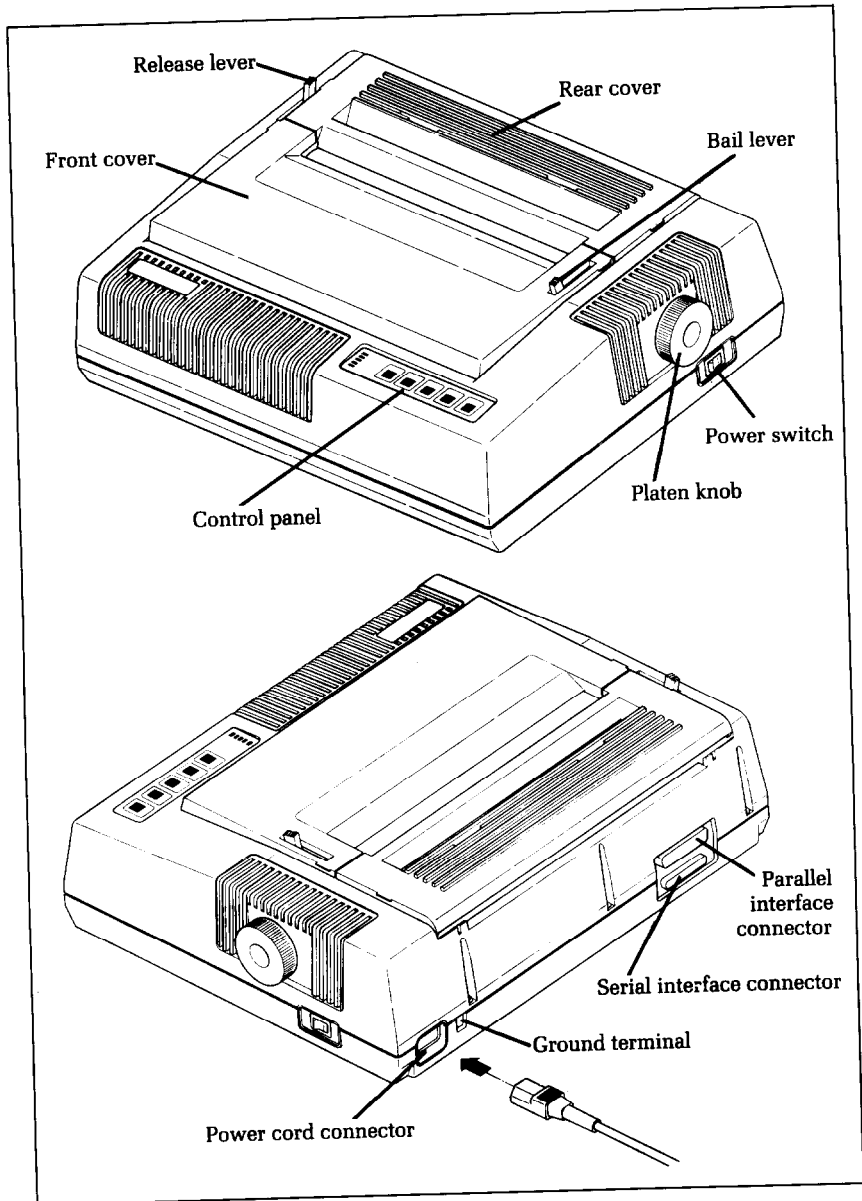


Figure 1-1. Front and rear views of Radix-10.

## Components and Controls

First, the components. You saw most of these when you unpacked your printer. Now we'll give you a condensed run-



down on what they do. (For details on your initial set-up of Radix, with all components in place, see Appendix A.)

**Printer covers** — There are two, front and rear. Their function is to protect the ribbon and print head from dust and dirt, and also to reduce the sound level.

**Single sheet guide** — As you've guessed, this plastic rack is used to support and guide the single sheets during printing.

**Sprocket paper guide** — This wire rack serves the same function, but for sprocket paper.

**Ink ribbon cartridge** — A neat and tidy timesaver, which snaps into place within a few seconds.

**Power cord** — Connects the printer to its power source, usually a wall outlet. It's located at the right rear.

**Print head** — This is the unit which does the actual printing. Like a typewriter, the print head prints through an ink ribbon.

**Tractor** — This built-in unit sits in the rear of your printer, under the rear cover. Its sprocket wheels carry the sprocket-feed paper on its pathway through the printer.

**Platen** — This is the rubber cylinder that carries paper to the print head.

**Parallel interface connector** — Around on the back, this is the place where you connect your computer to Radix, so that they are able to communicate with each other. It's for computers that use parallel communications.

**Serial interface connector** — This interface allows you to connect your Radix with a computer that uses serial communications.

Now let's take a tour around the controls, starting with the control panel board, located at the right front. There are 5 lamps and 5 buttons on the panel:

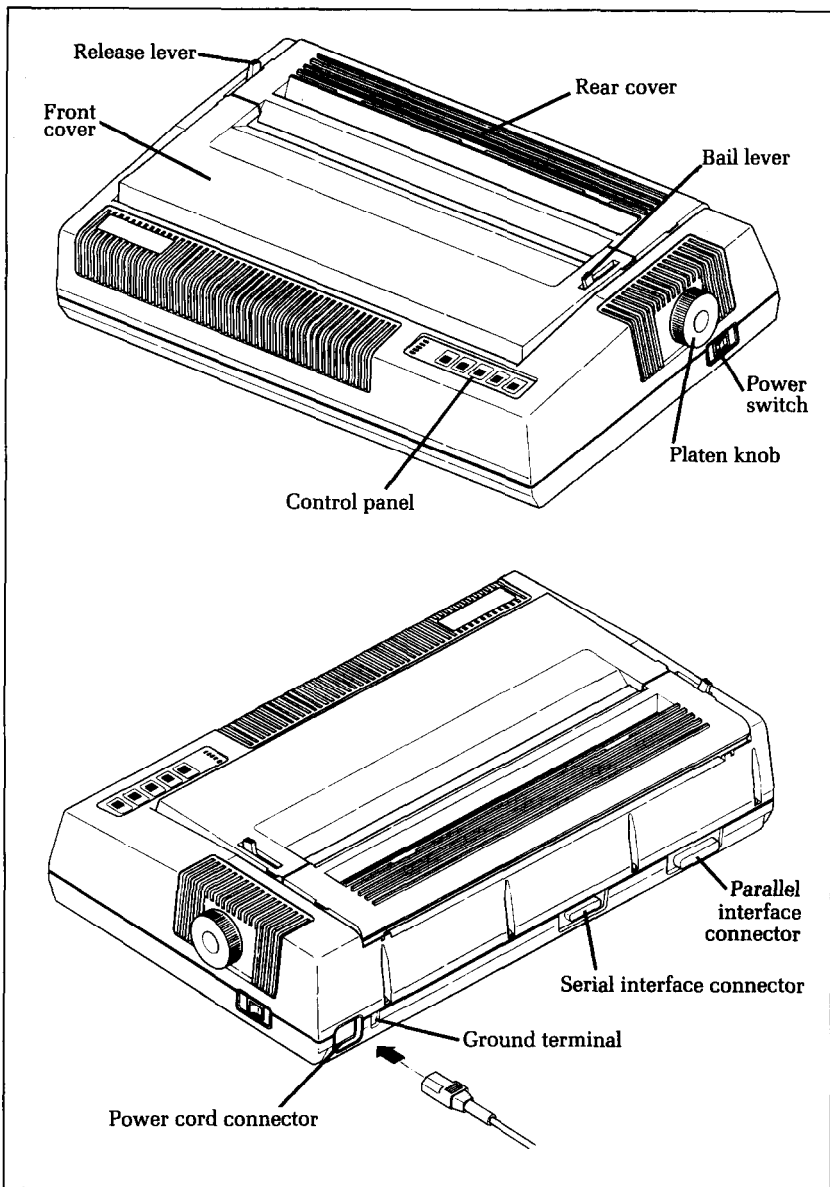
**Power lamp** — Glows green when the power is on.

**Ready lamp** — Glows green when the printer is ready to accept data. This light flickers during transmission. Don't worry about the flicker; it's normal!

**On Line lamp** — Glows green when the communication lines to your computer are open.

**Paper-Out lamp** — Glows red when the printer is out of paper and stops printing. It works only when you're using sprocket paper.

**Pause lamp** — A very important control! It glows green when the pause button has been pressed or when the front cover has been opened. When the pause lamp is on, you can feed paper with the LF, FF, or Feed buttons — but there's no printing possible. When the pause lamp is off, the printer will print — but you can't feed paper.



**Figure 1-2.** Front and rear views of Radix-15.

**Pause button** — Basically, this button allows you to change the printer status from “printing” to “not printing” or vice versa, with the results stated above under the Pause Lamp heading. This

allows you to stop printing to advance the paper — a few lines or to the top of the next page.

**Feed button** — This is used for automatic feeding of single sheets, which is described in detail later in this chapter.

**LF button** — Stands for “Line Feed,” and allows you to advance the paper one line at a time when the pause lamp is on. If you hold the button down, you’ll get consecutive line feeds, one after the other.

**FF button** — Stands for “Form Feed.” When you tap this button while the pause lamp is on, you advance the paper to the top of a new page or “form.”

**On Line button** — Lets you change the printer status between “off line,” and “on line.” When it’s on line, the printer can receive data from the computer. When it’s off line, the printer sends a signal to the computer indicating that it cannot accept data. When you turn the power switch on, you are automatically on line.

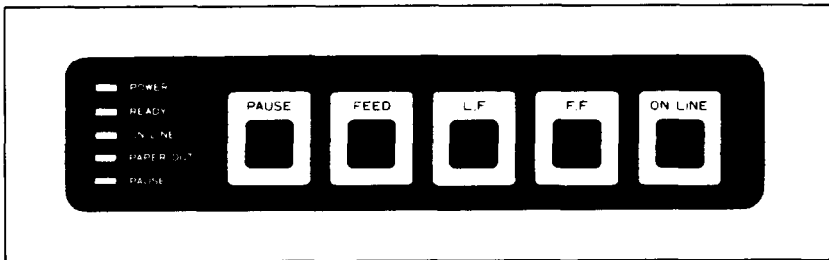


Figure 1-3. Radix's controls.

There are other kinds of controls, not connected to the control panel board. Some of the more important ones are:

**Power switch** — Towards the back, on the right side. This turns on the electricity to your machine.

**Platen knob** — Middle, right side. Lets you manually turn the platen, just like a typewriter. CAUTION: Turn this knob only with the power switch off. Turning it with the power on could damage the platen drive gears.

**Release lever** — On top, near the left rear corner. You'll be using this particular control often. What it does is control the pressure of the paper against the platen. Its position is crucial to feeding the different paper types — sprocket and single sheets. It has three settings: “Friction,” “Set,” and “Tractor.” The first two are used for single sheet printing, and the Tractor position for sprocket paper. This will be fully explained in the section describing paper

loading procedures.

**Bail lever** — The bail is the movable bar that presses the paper against the platen during printing, and when moved away from the platen, allows the paper to reach its proper position during the loading operation. The lever which controls it is on the right side of the platen.

**Paper-out detector** — This sensor automatically stops printing and tells you when the printer runs out of sprocket paper. The paper-out lamp glows red and a beep tone alerts you when the printer runs out of paper. The pause lamp also glows, so you are ready to load more paper. The lamp also glows if the release lever is not set in the tractor position for sprocket paper loading.

**Front-cover-open detector** — When the front cover is not fully closed, this magnetic detector causes the pause lamp to glow, and printing is interrupted (or won't begin). If this happens, printing may be re-started by securely closing the cover and pressing the pause button.

**DIP switches** — Primarily, these switches are used in interfacing Radix to your particular brand of computer. But there are also switches to set the power-on default settings for print style, line spacing, and page size. See the appendix for a complete explanation.

## ***Paper Selection and Loading***

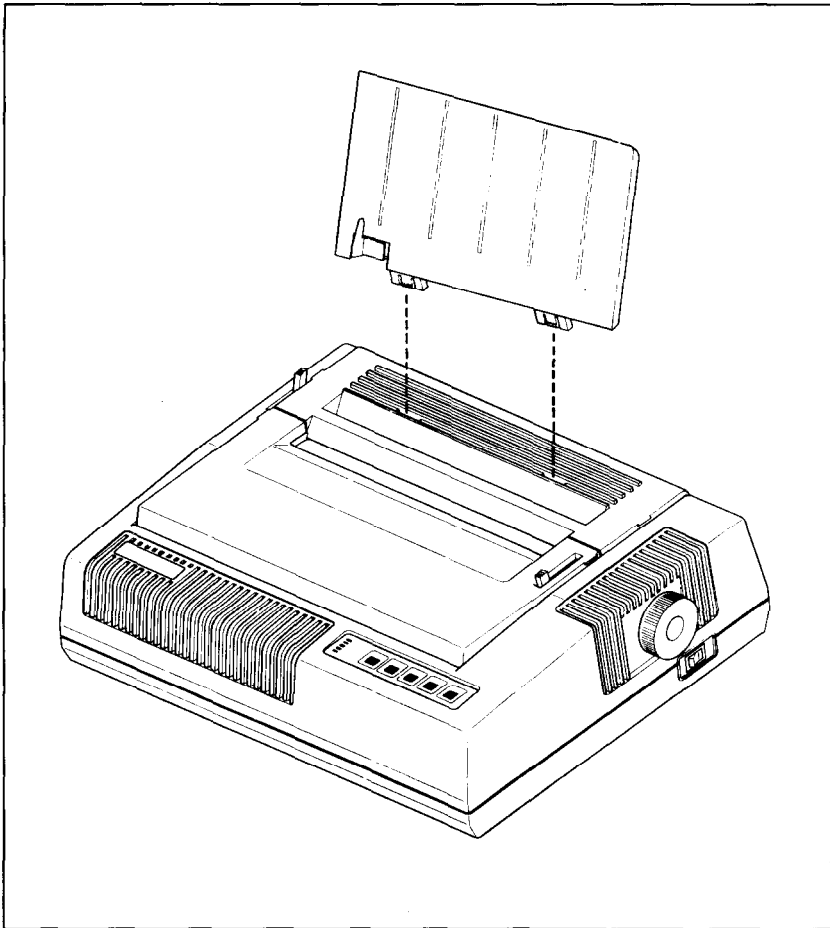
Now we'll look at paper. Your Radix can handle single sheets — standard-size stationery, multi-part carbonless business forms, or almost any other kind of cut sheet. You can also print on "computer paper" with the holes along the sides, which is also called sprocket, punched, or perforated fan-fold. The loading procedures are quite different for single sheet and sprocket paper. We'll try to keep it short and sweet, but without sacrificing clarity and preciseness in our explanations.

### ***Loading single sheets***

Start with the proper paper. Paper width must be between 5½ and 8½ inches (5½ and 14½ inches for the Radix-15), and paper thickness between .07 mm and .10 mm (16 pound to 24 pound bond falls in this range). Loading is done automatically and instantly by pushing the Feed button. Here's the correct sequence:

1. Attach the single sheet guide to the printer (Figure 1-4).

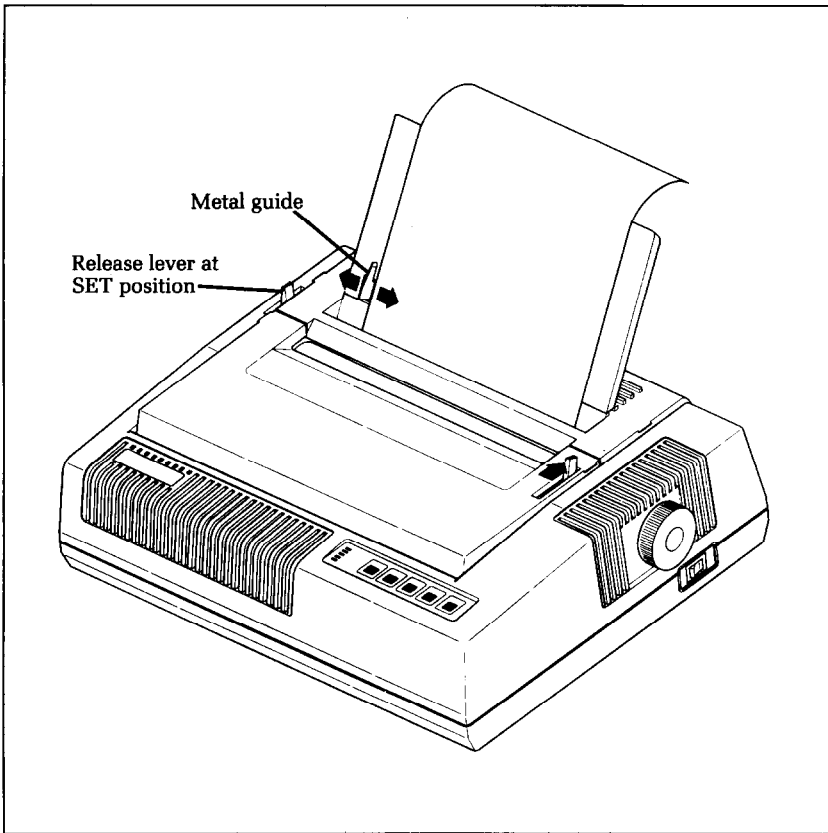
2. To set the margin, use the little metal guide (shown in Figure 1-5) in one of its 3 positions.
3. Put the release lever in the "set" position. This step is very important for proper sheet alignment.



**Figure 1-4.** Use the single sheet guide for loading cut paper.

**Table 1-1**  
**Left margin on the single sheet guide**

Position of Guide	Distance from Left-Hand Edge of Paper	
	For Radix-10	For Radix-15
Left	Approx. .6 inch	Approx. .8 inch
Middle	Approx. .3 inch	Approx. .5 inch
Right	Approx. .1 inch	Approx. .3 inch



**Figure 1-5.** The metal guide is used to align the left margin.

4. Putting the left edge of the sheet against the metal guide, insert a sheet into the paper chute until the bottom edge of the paper touches the paper stopper. (The set position of the release lever permits you to get the paper in straight.)
5. Now, push the release lever away from you to the "friction" position. This grips the paper securely for proper feeding.
6. Make sure that the bail is resting against the platen (you should push the bail lever away from the front of the printer). Radix will automatically lift it out of the way at the proper time!
7. With the power on, press the Feed button, and the paper automatically moves around the platen to the correct position to start printing, just one inch from the top edge of the sheet!

**Note:** If you'd like to start the first line of printing lower down on the sheet, as for letter correspondence for example, just press the

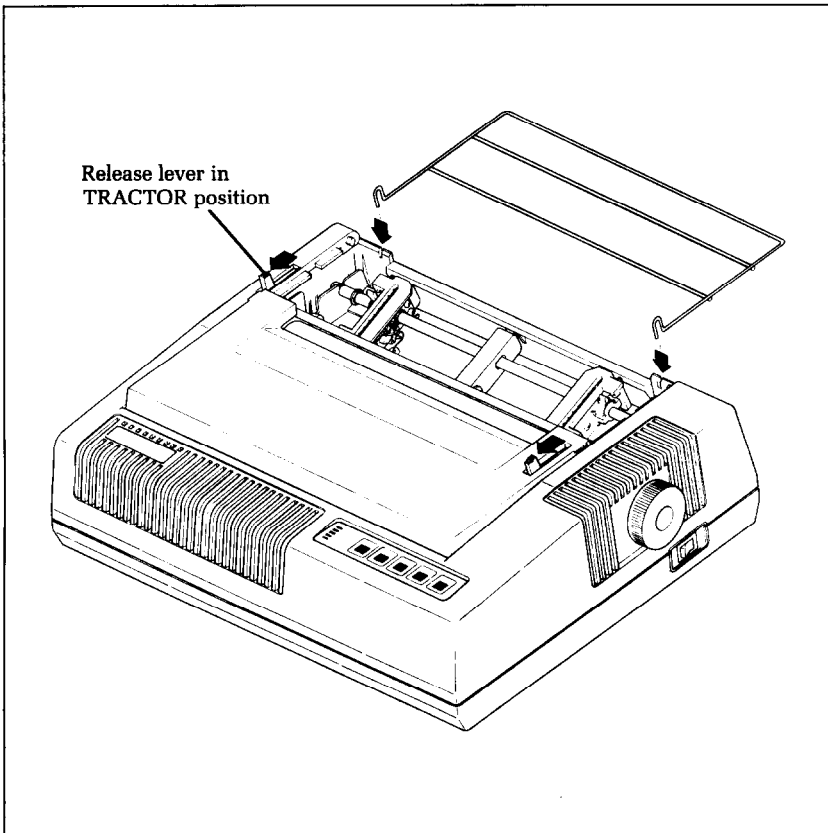
Pause, then the LF (line feed) button to move the paper to the desired starting point. Hold down the LF button for multiple line feeds.

### **Loading sprocket-feed paper**

Continuous paper feeds into the printer from the rear. So, the paper should be stacked directly back of the printer, either on the same surface, if there's room, or on the floor.

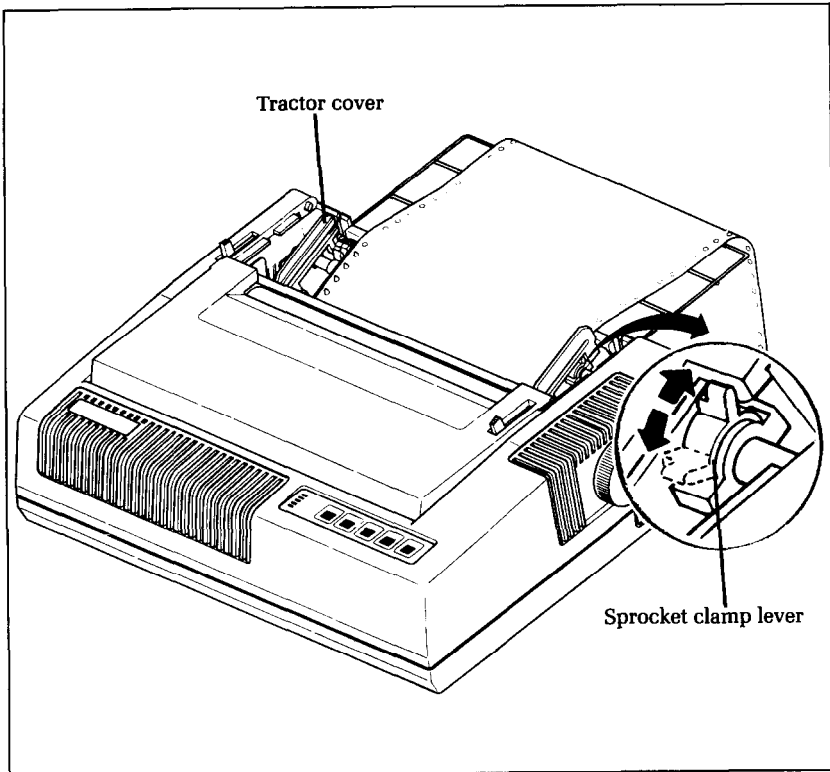
Here's the proper sequence for loading:

1. Turn off the power and remove the rear cover. (After you've practiced a few times, you'll find it easy to load paper by just opening the cover.)
2. Attach the wire paper guide to the rear of the upper case, as shown in Figure 1-6.



**Figure 1-6.** The wire paper guide keeps continuous paper away from the cables.

3. Pull the release lever towards you to put it in the "tractor" position.
4. Pull the bail lever towards you to the open position.
5. Open the tractor covers, located on top of the left- and right-hand sprocket units (Figure 1-7).

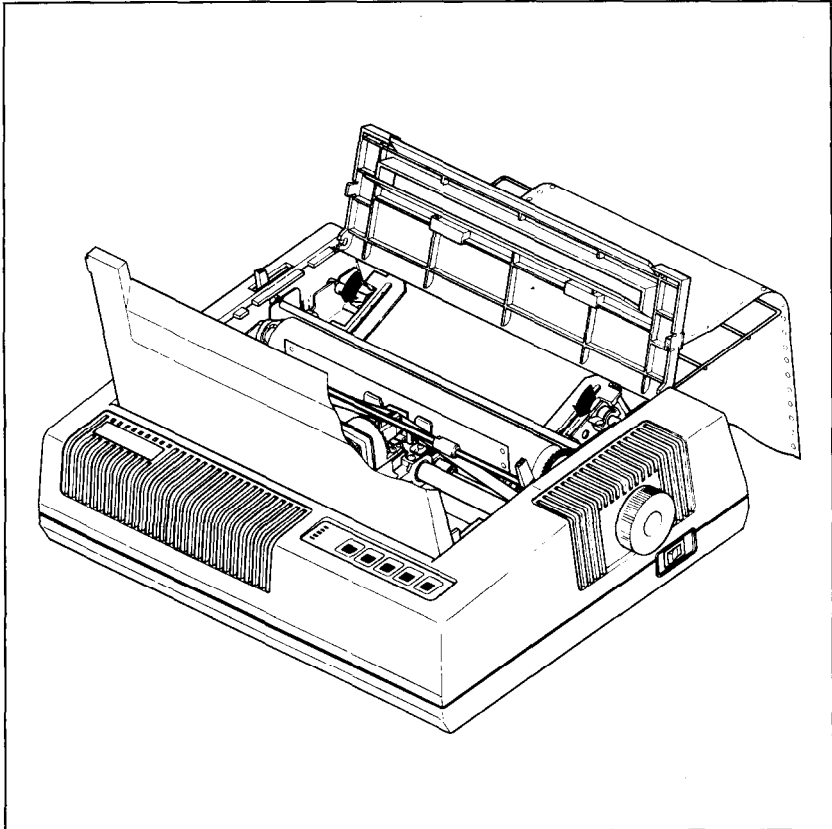


**Figure 1-7.** The tractors, which guide the paper, are underneath the rear cover.

6. Flip the sprocket clamp levers towards the rear. This unlocks the sprocket wheels to move left and right so you can align them with the holes in the paper.
7. Bring the paper up from the back, over the wire guide, and into the back of the printer. When the holes in the paper fit snugly over the nubby teeth in both sprockets, close the tractor covers and snap the clamp levers back into their locked positions (Figure 1-8).
8. Now we'll feed the paper around the platen automatically. To do this, close the rear cover, turn on the power, then push the



Pause button and hold down the LF button until the paper moves smoothly into position.



**Figure 1-8.** With the tractors in place, you're ready to close the covers and advance the paper.

9. Close the bail lever (push away from you). The top edge of the paper should line up with the cutter edge of the front cover so that printing will start one inch from the top edge.

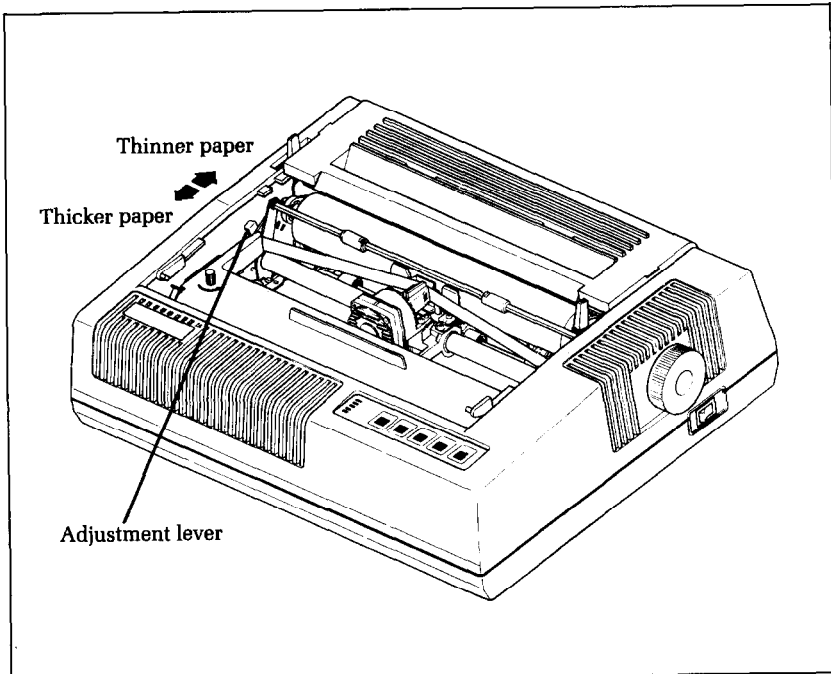
## **Ribbon Installation**

This is described in two places: installation of the ribbon cartridge is explained in Appendix A; replacing the ink ribbon inside the ribbon cartridge casing is described in Chapter 14 ("Maintenance").

## Adjusting the Gap

The gap is the space between the print head and the platen. Adjusting the gap is simply adjusting the printer to accommodate different thicknesses of paper.

To make this adjustment, move the adjustment lever which is under the front cover, immediately in front of the release lever shown in Figure 1-9. Pulling the adjustment lever towards you will widen the gap; pushing it away from you will narrow the gap.



**Figure 1-9.** The adjustment lever allows for different thicknesses of paper.

Five positions are available; you can feel the lever clicking into the various notches. The second step (illustrated) is the one most commonly used for single sheets of paper. The lever is nearly straight up in this position.

You shouldn't encounter any difficulty in finding the right gap setting to fit your paper. If necessary, experiment; you'll soon find the best position for the paper you're using.

## Self-Test

The “self-test” is a trial run of your beautiful new machine. Radix carries a built-in program that prints out sample lines of letters, numbers, and other characters — to show you that everything’s in good working order. It also serves as a display of the characters available in the Radix. And finally, it’s a “warm-up” that permits you to check your installation of ribbon and paper, and the adjustment of the print head gap.

Best of all, you don’t have to wait another minute — you can print the self-test without hooking up the Radix to your computer! It’s as simple as 1, 2, 3 . . .

1. Plug the printer’s power cord into a 120 VAC outlet.
2. Insert a sheet of paper (or sprocket paper, either one).
3. While holding down the LF button, turn the power switch on.

Were you surprised? It’s speedy, isn’t it? 200 characters a second, to be exact (when printing normal pica type).

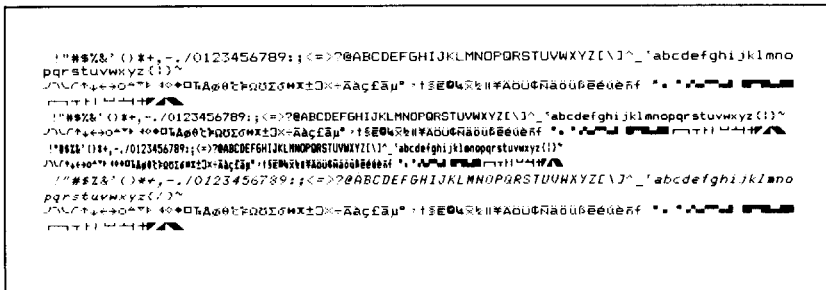


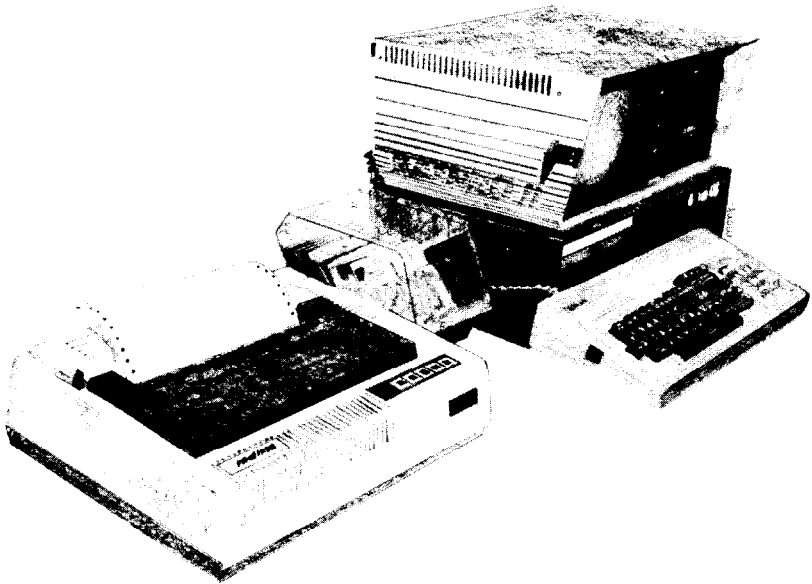
Figure 1-10. Radix’s self-test gives a preview of its capabilities.

## Some Tips for Smoother Operation

Here are some ideas that might save time and trouble with Radix.

- When setting the left-hand margin on sprocket paper, you’ll find the bail bar is marked with pica size unit measurements, so it’s a handy reference. (There are 10 pica characters to the inch, so the markings 10, 20, 30 and so on also correspond exactly to inches, 1, 2, 3, etc.)
- The sprocket paper is perforated in page size units, to facilitate easy folding (that’s the way it comes, in a stack). It is this edge that you should align with the front cover cutter edge so that printing will start just one inch below that point.

- When loading sprocket paper, never place the release lever in either the “set” or “friction” position. You’ll know when this happens by the beep tone and the paper-out lamp glowing red. Use the “tractor” setting at all times when loading or running sprocket paper.
- When you use multi-layer paper, such as a 3-part carbonless form, you should adjust the print head gap to fit the greater paper thickness, as explained earlier in this chapter.
- If paper should jam when loading sprocket paper, it’s usually because you forgot to put the bail lever in the open position (by pulling it towards you). Best thing to do then is to turn the power off, open the front cover, and roll the paper *backwards* by turning the platen knob.
- If the printing is faint, first check the thickness adjustment lever, then try a new ribbon. If it’s *still* too faint, perhaps it’s *finally* time for a new print head.



## Chapter 2

# Getting Started With Radix

**In this chapter you'll learn about:**

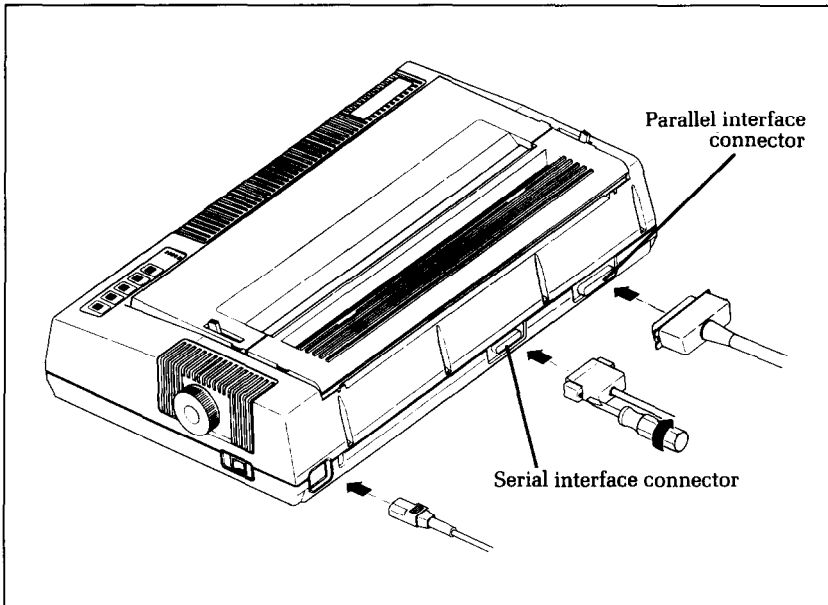
- **Connecting Radix to your computer**
- **Using Radix with commercial software**
- **ASCII codes**

You have assembled and tested your printer, and seen a quick sample of Radix's capabilities in the self-test. Now it's time to do what you bought Radix to do: print information from your computer.

But first you need to connect Radix to your computer. Figure 2-1 shows where the cable connects, but there's more that you need to know. Complete instructions for connecting Radix to many popular computers are given in the appendix. Find the appendix that covers your computer and follow the instructions

for connecting Radix and for setting the DIP switches. If your computer isn't listed in the appendix, then ask your Star dealer which computer that is listed is most like yours. If none of the listed computers are similar to yours, then your Star dealer will give you advice on connecting Radix to your computer.

When everything is connected, come back here and we will check it out!



**Figure 2-1.** Radix has both serial and parallel interfaces.

## Using Commercial Software

Many of you purchased Radix to use with commercial software. You made a good choice because Radix is compatible with most commercial programs, from word processing programs to spreadsheet programs to accounting programs.

Many of these programs have a routine for describing your printer. These routines are often in "installation programs". They typically give you a choice of printers or printer types to pick from. Some typical descriptions that you might pick for Radix are: "TTY type printer with backspace", "IBM-dot matrix printer", "Centronics-type printer", "Dot matrix ASCII printer". Radix should work fine with any of these descriptions.

Some printer lists are not very clear, and may not include any-

thing that you think describes Radix. If you can't decide which description best fits Radix, we recommend that you narrow the list to two or three choices (you can quickly eliminate all the daisy-wheel printer types) and then experiment. You won't hurt anything if you guess wrong; it just won't work right. This should quickly tell you if your guess is right. If all else fails, though, your Star dealer will be happy to give you some advice.

Some programs don't ask you what kind of printer you have, but instead they ask some questions about what your printer can do. Here are the answers to the "most asked" questions. Radix can do a "backspace". Radix can do a "hardware form feed".

With these questions answered, you are ready to start printing. Read the manual that came with your commercial software and Chapters 3 through 5 of this manual to see how to make it send information for Radix to print. This is all you need to know to use Radix as a regular printer. But Radix isn't just a regular printer. Radix has many capabilities that your commercial software isn't aware of. A little later we will see what it takes to use some of Radix's advanced features with commercial software.

### **First, some terminology**

Radix knows what to print because it knows how to interpret the codes that the computer sends to it. These codes are numbers that the computer sends to Radix. Both the computer and Radix know the meaning of these codes because they are a set of standard codes used by almost all microcomputers. This set of codes is the *American Standard Code for Information Interchange*, which is usually referred to as ASCII (pronounced ask-key). There are ASCII codes for all the letters of the alphabet, both lower case and capital, the numbers from 0 to 9, most punctuation marks, and some (but not all) of Radix's functions.

ASCII codes are referred to in several different ways, depending on the way they are used. Some times these codes are treated as regular numbers. For example, the letter "A" is represented by the number 65 in ASCII. Appendix M shows all of the ASCII codes.

In BASIC, ASCII codes are used in the CHR\$ function. This function is used to print the character that is represented by the number in the CHR\$ function. The BASIC statement PRINT CHR\$(65) will print an "A" on the terminal.

In some other programming languages, ASCII codes are referred to by their hex value. "Hex" is short for *hexadecimal* which is a base-16 number system (our usual numbers are base-10) Since hex needs 16 digits, it uses the numbers 0 through 9 and

then it uses the letters A through F for digits. The ASCII code for the letter "A" is 41 in hex.

Of course, most of the time we don't even need to think about this code system. Our computers are smart enough to know that when we press the "A" key on our keyboard we want to print the letter "A". The computer takes care of all the rest.

But there are a number of ASCII codes that don't have keys on the keyboard. The most important of these codes are the codes that have ASCII values below 32. These codes control many of Radix's functions. Even though there aren't keys for these codes, most keyboards can send these codes. It's done by holding down the "control" key (many times marked CTRL) and simultaneously pressing a letter key. The particular letter key that is pressed determines what code is sent. Control and A sends ASCII code 1, control and B sends ASCII code 2, and so on. Because of the way they are created, these codes are often referred to as "control-A" etc.

So there are four common ways of referring to the same set of codes: the character or name of the code, the decimal ASCII value, the hexadecimal ASCII value, and the "control-" value.

For example, the code that causes Radix to advance the paper one line is ASCII 10 (decimal). This code is commonly referred to by all the following names:

line feed	— its name
<LF>	— the abbreviation of its name
ASCII 10	— its decimal value
ASCII 0AH	— its hexadecimal value (the H signifies hex)
CHR\$(10)	— the way it's used in BASIC
control-J	— the way you send it from a keyboard.

There's a chart in Appendix M that shows these side-by-side so that you can convert back and forth.

The reason that we are telling you all this about ASCII codes is that people are not very consistent about how they describe ASCII codes. We are going to help you use Radix with commercial software, but we don't know what its documentation is going to call the various codes. So if you know all the different things that the codes might be called, it will be easier to figure out what it is trying to tell you.

Now, armed with the knowledge of what to look for, you can delve into the manuals of your commercial software and dig out the secrets of how to send "control codes" to your printer. When you find the method that your program uses, then you can shop



through this manual to find the function that you want to use. By translating the codes from the system that we use, to the system that your commercial software uses, you should be able to use many of Radix's advanced features. It may help, however if we look at a couple of examples. In the next three chapters, we'll give examples of many of Radix's codes using many popular programs.

### **The escape code**

There's one particular ASCII code that we are going to be using more than all the rest. This is ASCII 27, which is called *escape*. With all of Radix's advanced features, there weren't enough single ASCII codes to go around. So *escape* is used to start sequences of control codes that open a wider range of functions to us.

While you must call this code CHR\$(27) in BASIC, we are going to refer to it as <ESC> in this book. This will make it much easier to recognize when we use it.

A typical *escape code sequence* starts with <ESC> which is followed by one or more codes. As an example, the *escape code sequence* to turn on italic print is:

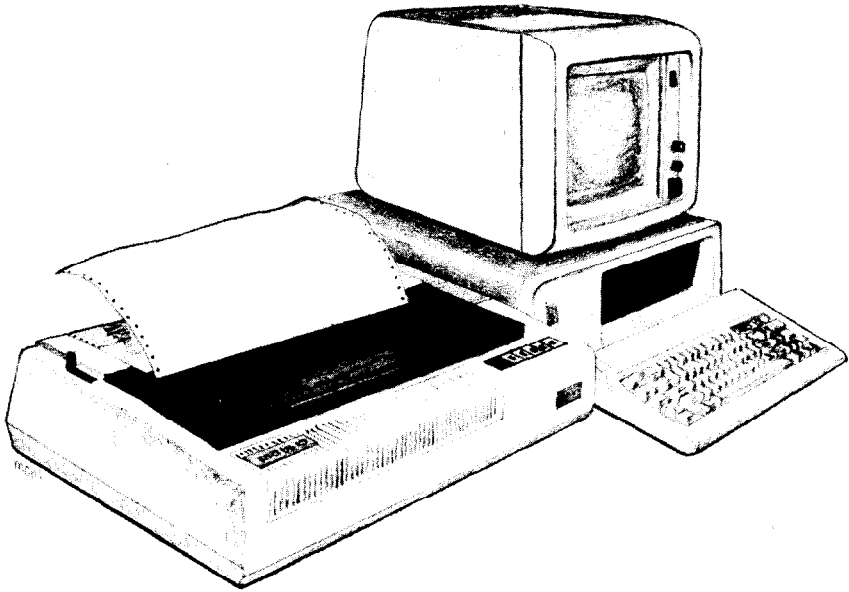
```
<ESC> "4"
```

We'll learn more about these *escape code sequences* and how to use them in the chapters that follow.

### **Using this book without learning BASIC**

Throughout the latter part of this book we will be teaching you how to use all of Radix's features using the BASIC programming language in our examples. This is because it is easy to communicate with Radix from BASIC and because, despite its shortcomings, BASIC is the nearest thing to a universal language among users of personal computers. But it's not the only way to communicate with Radix, as you will see in the next chapters. Even if you don't know BASIC, you can learn how to use Radix's features by reading on. When you find a function that you want to use, just apply what you already know about translating from one name for codes to another. The examples will still show you how the commands are used, even if you are not using BASIC.





### Chapter 3

## Word Processing with Radix

Not many word processing programs directly support the advanced features of printers like Radix. They usually provide a method for using a few of the more common print features such as boldface and underlining. But as you are probably beginning to see from this manual, Radix can do much more than that.

As a result, most word processing programs provide a way of sending special codes to a printer. The actual codes used (as well as the method of entering them) can be different. The theory behind these methods, however, is basically the same.

**This chapter discusses four word processing programs most used by Radix owners. The programs also provide a variety of ways to enter the codes necessary to use the advanced features of Radix. These concepts can be applied to many other**

programs besides those detailed here. The four programs are:

- Easywriter II
- PeachText
- WordStar
- Scripsit

If your word processing program is not included in this chapter, you should still study the different techniques used. Then, with the help of your program manual and the supporting chapters in this manual, you should be able to figure out how yours works.

## General Concepts

Each word processing program has a way to get out of the standard text entry mode in order to accept the special printer function codes. PeachText uses an `\OUT\` statement. Wordstar uses the CONTROL key in different ways to define the print function codes.

Easywriter II has a system function which allows you to define print pitches and special print functions for use with the ALT key. Scripsit has a similar process in which user-defined codes are used as recognition characters to select and cancel print functions.

Your word processing User Manual (if it supports this process) will have a section describing how to get out of the standard program. You will probably have to figure out on your own which codes are used. The general concepts and details of the four sample programs should be enough to help you be successful.

### The escape code

Most of Radix's special print functions start with a code called the escape code. It can be entered in decimal or hexadecimal values, by an ASCII character, or by using the control keys on your keyboard. It depends on which program and which computer you are using.

This escape code tells the printer to interpret the values (or characters) following it as printer functions. The codes used to describe the functions are also entered in the same method as the escape code. In this chapter, we will show you the format each word processor uses as well as the general rules to correctly enter the function codes.

Chapter 2, "Getting Started with Radix," covers how to convert forms of ASCII codes. You should review Chapter 2, if you have not already done so, before working with the function codes.

### **The master reset code**

There is one function code which turns off all the print functions currently being used by the printer. It is called the master reset code and resets the printer to its DIP switch settings. These print characteristics are the same as the ones used by the printer when it is first turned on.

The code sequence for master reset is `<ESC> "@"`. By checking the ASCII equivalents in Appendix M, you can see that the decimal expression is 27 64. You'll see these numbers several times in this chapter.

Technically speaking, initializing the printer clears the print buffer and the form length, character pitch, character set, line feed pitch and international character set are all reset to the values defined by their respective DIP switch settings.

We suggest you get in the habit of using the master reset code in any document where you use function codes. If you do not, the printer will keep the characteristics most recently defined and print any following documents the same way.

You could turn Radix off each time (which also resets the default settings) but that would be hard on the printer circuits. Also, you'll save time and paper by letting the printer automatically reset with this code. (If you need more information on DIP switch settings for Radix, please refer to Appendix H.)

### **Using Near Letter Quality (NLQ)**

With near letter quality, Radix prints more dots for each character than with the draft printing. This process results in a higher quality look to your text. Draft quality characters print much faster, so use them for your first drafts and use near letter quality for a professional looking finished manuscript.

The escape code sequence to turn the NLQ set on is `<ESC> "B" 4` and the code sequence to select draft quality is `<ESC> "B" 5`. The decimal equivalents are 27 66 4 and 27 66 5, respectively.

Near letter quality printing can be printed in pica width (10 characters per inch) and underlined if you wish. It cannot, however, be mixed with Radix's other print widths, italics, superscripts, subscripts, double-strike, or emphasized printing.

### **Getting the most from your print choices**

After working with Radix for a while, you may find that you want to add to or change some of the print functions we have described in this chapter.

We suggest you do three things. First, you should review Chapter 2 and Appendix K to become as familiar as possible with ASCII codes and the Radix function codes.

Second, read Chapter 7 which describes them in greater detail and shows examples of how they are used in BASIC programming. The functions will, for the most part, act the same in your word processing program. Understanding what's available and how they perform will help you use them correctly in your documents.

And third, follow the procedures in this chapter and your program User's Manual.

You may want to experiment with expanded text in combination with other print types. You can create some great-looking results with these functions. Also, try applying what you have learned in this chapter to your own work while it is still fresh in your mind. If you are unsure of any functions, review them first, then try some of your own samples.

## **Using Radix with Easywriter II**

**(Note:** If you have not read the "General Concepts" section at the beginning of this chapter, you should do so before continuing.)

The Radix printer can be used with most of the standard print functions available with your Easywriter II word processing program. These functions require no special adjustments to the printer or your program. They include:

1. Printing from the Print List Form screen.
2. Setting margins, tabs and lines per inch in the ruler line of your document. (The pitch settings, however, should be adjusted to obtain maximum use. They will be discussed later in this chapter.)
3. Print settings in the System Parameter function which are either default settings or new settings edited by you.

You can also redefine print functions of Easywriter II to take advantage of many of the printing capabilities of Radix. You may already be familiar with reconfiguring the printer driver from Appendix B of your Easywriter II User's Manual. If not, don't be nervous; it's not as hard as it sounds. We will show you how to

make changes in your program specifically to help you print with Radix.

By changing the pitch settings, you can use the document ruler line to print pica, elite and condensed width pitches. In addition, you can use the same theory to print in near letter quality pica.

The print control codes can be redefined to enhance the final product of your document. The boldface, underline, superscript and subscript functions require only a slight "recoding" of information in the printer driver. And we have some suggestions for changing the characteristics of the other print control codes to use italic, expanded, emphasized and italic-underline print. With these options, you will have even more printing flexibility with Radix.

### ***Redefining pitch settings and print control codes***

In order to change the settings used in the document ruler line and the print control codes, it is necessary to edit ASCII code decimal values in the System Functions portion of your Easywriter II program. (For more details on ASCII codes, please refer to Chapter 2.)

Your Radix printer is considered a Type B printer by the Easywriter II program. Before making any changes in the printer driver, you should first check to be sure the printer selection is set for printer Type B (Option 7 on the System Functions menu).

Then follow the instructions in Appendix B of your Easywriter II User's Manual to reconfigure Type B printers. To become more familiar with the reconfiguration process and its terms and to make the instructions in this section easier to understand, we suggest you read through Appendix B first.

In these next few paragraphs, we'll show you the ASCII decimal values we feel provide a good flexibility in printing with Radix. You should follow the instructions hands-on with your own Easywriter II program.

The changes you will make are for pitch settings and print control codes (also called font support). However, all the screens involved will be explained as you see them displayed.

From the System Functions Menu, choose Option 9 (Reconfigure Printer Type B) and the printer name will be displayed. Type over the present printer name as follows:

1. Printer Name [*Radix Dot Matrix Printer* ]

Press RETURN and the Edit Global Sequences screen will be displayed. These codes control the print functions for form feed, line feed, margin settings and automatic justification. We do not recommend that you edit any of these codes.

Press RETURN and the Edit Pitch Table screen will be displayed. On this screen, you will enter the ASCII decimal values to define the print pitches. The first two fields in each line define the pitch range (which in this case are both the same number). They should be assigned as follows:

10 = Draft Pica      1 = Near Letter Quality Pica  
 12 = Draft Elite    2 = Master Reset Code  
 17 = Draft Condensed

On this screen, the column labeled "Sequence" is used to define the print functions in their ASCII decimal values. For these print pitches, we will use a combination of codes to turn near letter quality print on and off and to choose the function code for each pitch. (For more details on function codes, please refer to Appendix K.)

Follow the sample and enter the (*italic*) codes for lines 17-21.

17.	[10 ]	[10 ]	[	27	66	5	27	66	1	]
18.	[12 ]	[12 ]	[	27	66	5	27	66	2	]
19.	[17 ]	[17 ]	[	27	66	5	27	66	3	]
20.	[1 ]	[1 ]	[	27	66	4				]
21.	[2 ]	[2 ]	[	27	64					]
22.	[120]	[120]	[							]
23.	[120]	[120]	[							]
24.	[120]	[120]	[							]
25.	[120]	[120]	[							]
26.	[120]	[120]	[							]

The codes 120 in lines 22 through 26 can be changed to reflect more pitch settings. We recommend that, until you are more familiar with using special function codes, you use just the five we have defined.

When you have finished, press RETURN. You will be transferred to the Edit Line Spacing screen. Do not change these codes.



They define how many lines per inch the printer uses. Press RETURN to transfer to the Edit Font Support screen.

Change all of the entry fields to Option 2 (Control Code Support) on the Edit Font Support screen. Also, make changes in the other fields to look like the figure shown below. Enter the (italic) codes for lines 41-50.

- 41. Bold/Shadow Face Support [2]
- 42. Single Underline Support [2] Using Character [95 ]
- 43. Double Underline Support [2] Using Character [0 ]
- 44. Overstrike Support [2] Using Character from
- 45. Special (Color) Support [2] System Parameters
- 46. Sub/Superscript Support [2]
- 47. Will underline retain font (Y) or be normal font (N)?  
[N]
- 49. Start double underline [ ]
- 50. After double underline [ ]

When you have finished, press RETURN and the Edit Font Sequences screen will be displayed. Here you will define print control codes for use in your documents. As with the pitch settings, ASCII decimal values are used that correspond to the print function assigned to each control key. Table 3-1 shows the current control function, the print function we will assign to it and the keyboard keys used.

**Table 3-1**  
**Easywriter II control keys**

Easywriter II function	New function	Keys used
Boldface	Boldface	ALT & B
Shadow	Italic	ALT & S
Underline	Underline	ALT & _
Double	Expanded	ALT & =
Overstrike	Emphasized	ALT & O
Special	Italic Underlined	ALT & *
Subscript	Subscript	ALT & D
Superscript	Superscript	ALT & U

Enter the (*italic*) codes for lines 51-66.

51. Normal to Bold	[	<i>27</i>	<i>71</i>	<i>27</i>	<i>69</i>	]
52. Bold to Normal	[	<i>27</i>	<i>72</i>	<i>27</i>	<i>70</i>	]
53. Normal to Shadow	[	<i>27</i>	<i>52</i>			]
54. Shadow to Normal	[	<i>27</i>	<i>53</i>			]
55. Normal to Underline	[	<i>27</i>	<i>45</i>	<i>1</i>		]
56. Underline to Normal	[	<i>27</i>	<i>45</i>	<i>0</i>		]
57. Normal to Double	[	<i>27</i>	<i>87</i>	<i>1</i>		]
58. Double to Normal	[	<i>27</i>	<i>87</i>	<i>0</i>		]
59. Normal to Overstrike	[	<i>27</i>	<i>69</i>			]
60. Overstrike to Normal	[	<i>27</i>	<i>70</i>			]
61. Normal to Special	[	<i>27</i>	<i>52</i>	<i>27</i>	<i>45</i>	<i>1</i> ]
62. Special to Normal	[	<i>27</i>	<i>53</i>	<i>27</i>	<i>45</i>	<i>0</i> ]
63. Normal to Subscript	[	<i>27</i>	<i>83</i>	<i>1</i>		]
64. Subscript to Normal	[	<i>27</i>	<i>84</i>			]
65. Normal to Superscript	[	<i>27</i>	<i>83</i>	<i>0</i>		]
66. Superscript to Normal	[	<i>27</i>	<i>84</i>			]

When you have finished, press RETURN. You're done! You will be transferred out of the Reconfigure Type B Printer function and back to the System Functions Menu.

### ***A sample printout with Easywriter II***

Let's look at a short example to demonstrate how pitch settings and print control keys can be used in a document. The example below shows the use of expanded and italic prints used in combination with condensed and pica pitch settings. Use your Easywriter II program hands-on and type the example below.

SUBJECT: ORDERING STATIONERY SUPPLIES

I would like to place an order for stationery supplies from your mail order catalog. Enclosed is my order form and a check for \$247.67. Please process this order as soon as possible. Thank you.

With the cursor under the "S" in "SUBJECT", set the print pitch in the ruler line to condensed width pitch. Name the ruler

line "condensd" (without the quotes) and change the character pitch to 17 and the line spacing to 6. To make the subject title expanded, use the ALT and = keys (in the line mode) to highlight the line.

Now, change the pitch setting in the next line to pica by setting a new ruler line: Ruler Name - pica; Character Pitch - 10. Use the print control key S (for italic) to highlight the second sentence in the paragraph. Move the cursor to the "E" in "Enclosed" and (in the sentence mode) use the ALT and S keys to highlight the sentence. (You'll have to press S twice to get the .67.)

At the end of the document, reinitialize the printer to its default settings with a new ruler line using the Master Reset code. Ruler Name - reset; Character Pitch - 2.

Print the document. Your printout should look like this:

**SUBJECT: ORDERING STATIONERY SUPPLIES**

I would like to place an order for stationery supplies from your mail order catalog. *Enclosed is my order form and a check for \$247.67.* Please process this order as soon as possible. Thank you.

The subject title will print in expanded condensed characters which are twice the width as standard condensed characters. The sentence in the paragraph is printed in italic pica print. The last ruler line will reinitialize the printer. (See the general concepts section of this chapter for more details on master reset.) This is just one example, however, you should be able to apply most of the function codes to the setup used here.

### **Redefining your own print pitches**

If you want to define a new print pitch (Edit Pitch Table), be sure to start the sequence with the code 27 66 5. This code tells the printer to turn off near letter quality print. Then enter your function code to choose the print you want. By not using the function code 27 66 5 first, the printer will continue to print near letter quality.

For example, if you found yourself frequently using italic print for large blocks of text in pica width pitch, you can combine italic and pica pitch to define italic pica and use it in the ruler line of your document. The ASCII code sequence would be 27 66 5 27 66 1 27 52 which would print italic pica pitch.

### ***Redefining your own print control keys***

The ASCII codes to redefine the print control keys (Edit Font Sequences) are pretty straight forward. There are individual ASCII decimal values to turn on and off different prints. You want to affect that aspect but not the print pitch itself. Leave that for your document ruler line. Remember, all the codes can be found in Appendix K of this manual.

Also, keep in mind that print control keys can be combined in your document such as boldface and underline. Easywriter II uses three methods of highlighting on the display screen. It highlights, underlines and shows reverse image characters. You cannot combine print control functions that use the same method of highlighting.

For example, in our definitions, underline and expanded prints are both displayed as underlined on the screen. Whichever function you use last will cancel out any previous modes.

### ***Using Radix with PeachText***

**(Note:** If you have not read the "General Concepts" section at the beginning of this chapter, you should do so before continuing.)

Radix can be used with PeachText for a wide range of different print functions. Radix automatically supports many of the standard printing capabilities as well as the method of sending special codes to use all of its printing features. As a result, you can really customize the final look of your documents.

With Radix, you can perform all the following print functions without making any special changes to your PeachText program.

1. Print documents from the Text Edit screen.
2. Print documents from the Print Status screen.
3. Use print commands and recognition characters in your document.
4. Select and print variable information for merge letters, etc.

All of these functions are fully described in the PeachText User's Manual and will not be discussed in this chapter. You should refer to the manual if you need help in successfully performing any of these functions.

With the print capabilities described above, however, some individual functions will not work without using special codes. They include changing lines per inch and characters per inch,

using superscript and subscript recognition characters, backspacing, and using horizontal and vertical tabs commands.

### **Entering special function codes**

When you want to enter a code to perform a special print function, you must first enter the PeachText command, `\OUT`. Then, enter the decimal values relating to the ASCII code for each function you want performed. The statement is closed by using the `\` symbol.

Each value must be separated by a comma. For example, a valid statement would be `\OUT27,69\`. If the function code you want is expressed in more than one ASCII character, you must use a comma between each decimal value. To select elite pitch, for example, the ASCII code is `<ESC> "B" 2`. The PeachText statement would read `\OUT27,66,2\`.

Different print combinations can be combined within one statement. In these cases, you need to use the escape code for each function. For example, to select double-strike and emphasized print at once, the PeachText statement is `\OUT27,71,27,69\` where 27,71 selects double-strike print and 27,69 selects emphasized print.

When you enter these codes in your document, they will appear on the display screen, but they will not print out on paper. The characters do not take up any hard space when printed. Your first character of text will actually print in the first space where the command begins (on the screen).

When you start using these codes on your own, you should turn to Appendix K in this manual to look up the printer function you want. To enter the codes into PeachText, simply use the decimal values in the format previously described. Let's take an example.

Say you want to change to near letter quality characters. The function code as shown in Appendix K is `<ESC> "B" 4`. The decimal equivalent is 27 66 4. Your PeachText command would be entered as `\OUT27,66,4\`. By following this format, you can use any of the function codes applicable to Radix as described in Appendix K.

Table 3-2 references the most commonly used print functions and the PeachText statements you should enter in your document.

### **Using boldface print**

There are several different ways to highlight text as boldface

print. The easiest way is to use the boldface function with the PeachText recognition character (@) as described in the PeachText User's Manual. You can also use function codes to select double-strike, emphasized or combine both to highlight in boldface fashion.

Each type will give you a slightly different look. You may want to experiment with them and use the one you like the best. Once you have chosen the method you feel is best for your needs, try to stick with it as much as possible. Switching back and forth will be confusing and ultimately detract from the look of your documents.

**Table 3-2**  
**PeachText print functions**

Print Function	Select Code	Deselect Code
Pica Pitch	\OUT27,66,1\ 	-
Elite Pitch	\OUT27,66,2\ 	-
Condensed Pitch	\OUT27,66,3\ 	-
Near Letter Quality	\OUT27,66,4\ 	\OUT27,66,5\ 
Double-Strike	\OUT27,71\ 	\OUT27,72\ 
Emphasized	\OUT27,69\ 	\OUT27,70\ 
Italic Print	\OUT27,52\ 	\OUT27,53\ 
Single-Line Expanded	\OUT27,14\ 	see note below
Continuous Expanded	\OUT27,87,1\ 	\OUT27,87,0\ 
Underline	\OUT27,45,1\ 	\OUT27,45,0\ 
Superscripts	\OUT27,83,0\ 	\OUT27,84\ 
Subscripts	\OUT27,83,1\ 	\OUT27,84\ 

**Note:** A carriage return will automatically turn off single-line expanded text.

### **Underlining with Radix**

There are two different ways to underline text. The easiest way is to use the underline function with the PeachText recognition character (—) as described in the PeachText User's Manual. With Radix, this will print as a series of dashes under the text (as opposed to a solid line).

The special printer function codes can also be used to select and cancel underlining. As shown in the Table 3-2, the code \OUT27,45,1\  
selects underlining and \OUT27,45,0\  
cancels it. With these codes, Radix will print a solid line under the text (including spaces) rather than dashes.

The choice is yours as to which you use depending on your application. The flexibility, however, is nice to have.

### A sample printout with PeachText

Let's look at a short example to demonstrate how function codes are used in a PeachText document. The following example shows how italic and italic-underline print are used. Use your PeachText program hands-on and type the example exactly as you see it below.

SUBJECT: \OUT27,52\ORDERING STATIONERY  
SUPPLIES\OUT27,53\

I would like to place an order for stationery supplies from your mail order catalog.

\OUT27,52,27,45,1\Enclosed is my order form and a check for \$247.67.\OUT27,53,27,45,0\ Please process this order as soon as possible. Thank you. \OUT27,64\

In the subject title of this example, the first code \OUT27,52\ selects italic print; the second code \OUT27,53\ cancels it. In the paragraph of the example, the first function code combines italic with underline print. The decimal values 27,52 select italic and 27,45,1 select underline.

Also in the paragraph, the function code \OUT27,53,27,45,0\ cancels italic-underline print. The decimal values 27,53 cancel italic and 27,45,0 cancel underline.

The final function code \OUT27,64\ resets the default settings on the printer. Now, print the document. Your printout should look like this:

SUBJECT: ORDERING STATIONERY SUPPLIES

I would like to place an order for stationery supplies from your mail order catalog. Enclosed is my order form and a check for \$247.67. Please process this order as soon as possible. Thank you.

This is just one example, but you should be able to apply most of the function codes to the setup used here.

### **Storing documents with function codes**

When you create combinations of function codes you like, store them as skeleton documents. To access the documents later, go to the Main Menu and use the copy document function (CO) to duplicate the skeleton document. Then rename the new document. This will save you time and reduce your chance of entering incorrect codes.

## **Using Radix with WordStar**

(**Note:** If you have not read the "General Concepts" section at the beginning of this chapter, you should do so before continuing.)

Radix supports many of the standard WordStar printing capabilities without requiring any changes. You can:

1. Print documents from the No-File Menu.
2. Use the dot commands except for lines per inch, characters per inch and microjustification.
3. Print boldface, underline, double-strike, strikeout, superscript and subscript characters as well as use print pause.
4. Select and print variable information for merge letters, etc.

### **User-defined print commands**

There are several CONTROL-P (^P) commands that automatically work with Radix and require no changes. They include:

^PS Underscore	^PB Boldface
^PD Double-strike	^PX Strikeout
^PT Superscript	^PV Subscript
^PC Print Pause	

It is also possible to define the ^PA (alternate pitch) command to change the print pitch of your document. The WordStar User's Manual fully describes the use of these ^P functions. You should refer to your manual if you need help with them.

There are four alternative ^P codes that can be defined during the installation of your WordStar program to perform other printer functions. They are ^PQ, ^PW, ^PE, and ^PR. The process of defining ^P commands is called "patching" and is a fairly complicated process. Once you have successfully defined these codes, they are inserted in your text exactly like the other ^P commands. If you wish to use them, refer to the WordStar User's Manual for instructions or contact your dealer for assistance.



Perhaps the most useful user-defined ^P command is ^PE. If you define this as an escape (ASCII code 27), you can then access nearly all of Radix's advanced features. Without this patch, you cannot place an escape in the WordStar document and subsequently, you are limited to using WordStar's repertoire of print functions. A shame when you have a powerful Radix!

## Using Radix with Scripsit

**(Note:** If you have not read the "General Concepts" section at the beginning of this chapter, you should do so before continuing.)

Radix can be used with most of the Scripsit print functions. You can use most of the basic functions on the Open Document Options screen as well as on the Print Text Options screen. Two minor adjustments you need to make. On the Open Document Options screen the printer type is LPN4. Also, DIP switch C-4 must be OFF to cause a line feed at the end of each line.

There are some print functions, however, that do not work without redefining the printer driver or by continually changing the DIP switch settings. That could be an awkward, time-consuming task.

Instead, we recommend the User Print Code Facility to define several recognition characters and control your print functions. With the use of the CLEAR key and each recognition character, Scripsit can perform any print function you define.

Print Pause and Form Feed automatically work with Radix. Double underscore only works with daisy wheel printers and strikeout does not work without redefining the printer driver. (Since strikeout is seldom used, we do not suggest you try it.)

Boldface, underline, superscripts and subscripts do not work with the recognition characters described in the Scripsit User's Manual but can be defined in the Print Code Facility along with some other print functions (emphasized, italic and expanded).

The print pitch settings on the Open Document Options screen will not work with Radix. Therefore, we will show you how to use recognition characters to define pica, elite, condensed and near letter quality pitches.

### Defining user print codes

Your first step is to enter the Edit Printer Control screen. If you are not familiar with this function, review your Scripsit User's Manual for details. Each character on this screen will be

assigned an ASCII decimal value corresponding to the function you want. If you have not read Chapter 2 in this manual, we suggest you do so before continuing.

The decimal equivalents of each function can be found in Appendix K. A space should be left between each decimal value. The sample below shows each code as it should be entered into the Edit Printer Control screen. Type the (*italic*) codes as they appear here.

Code	Units	Sequence:	up to 11 codes will	be counted	Comments
0	0—	27 64	_____		<i>mstr reset</i>
1	0—	27 66 5	27 66 1 _____		<i>pica</i>
2	0—	27 66 5	27 66 2 _____		<i>elite</i>
3	0—	27 66 5	27 66 3 _____		<i>condensed</i>
4	0—	27 66 4	_____		<i>near lettr</i>
5	0—	27 71	27 69 _____		<i>bold on</i>
6	0—	27 72	27 70 _____		<i>bold off</i>
7	0—	27 69	_____		<i>emphasz on</i>
8	0—	27 70	_____		<i>emphas off</i>
9	0—	27 45 1	_____		<i>under on</i>
		Press <ENTER> to edit next screen			
!	0—	27 45 0	_____		<i>under off</i>
"	0—	27 52	_____		<i>italic on</i>
#	0—	27 53	_____		<i>italic off</i>
\$	0—	27 87 1	_____		<i>expand on</i>
%	0—	27 87 0	_____		<i>expand off</i>
&	0—	27 83 0	_____		<i>supscrp on</i>
'	0—	27 83 1	_____		<i>subscrp on</i>
(	0—	27 84	_____		<i>script off</i>
)	0—	_____	_____		_____
@	0—	_____	_____		_____
		Press <ENTER> to return to System Setup menu			

When you have finished entering the codes, press ENTER to return to the System Setup menu. Then, press BREAK to return to the Main menu.

Each function code is pretty straight forward and they are all included in Appendix K of this manual. But there is a certain way to enter them.

Say you want to enter the codes for double-strike print. The function code as shown in Appendix K is <ESC> "G". The deci-

mal equivalent is 27 71. Your Scripsit command would be entered as 27 71. By following this format, you can use any of the function codes applicable to Radix as described in Appendix K.

The function codes for pica, elite and condensed width pitches are expressed a bit differently than the rest. If you want to request one of these three prints and you are changing from Near Letter Quality (NLQ), the first codes 27 66 5 will turn NLQ off before printing draft pica, elite or condensed 27 66 1 (2 or 3).

Notice that boldface is actually a combination of function codes for defining double-strike and emphasized print. If you want to use just double-strike as boldface, you can delete the codes 27 69 and 27 70 from sequences 5 and 6.

### **A sample printout with Scripsit**

Let's look at a short example to demonstrate how pitch settings and print control keys can be used in a document. The example below shows the use of boldface and italic prints used in combination with pica and elite pitch settings. Use your Scripsit program hands-on and type the example below. (To get the "@" symbol, use the CLEAR key on your keyboard).

```
@1SUBJECT: @5ORDERING STATIONERY SUPPLIES@6
@2I would like to place an order for stationery supplies
from your mail order catalog. @'Enclosed is my order
form and a check for $247.67.@# Please process this
order as soon as possible. Thank you.
@0
```

The first recognition character @1 will start pica width pitch. The second character @5 will turn on boldface print until @6 turns it off. At the beginning of the paragraph, @2 starts elite pitch. The second sentence is bracketed by the recognition characters @' and @# which will turn italic print on and off. The last code @0 is the master reset code which will reinitialize the printer to its default characteristics.

Now print the document. Your printout should look like this:

SUBJECT: **ORDERING STATIONERY SUPPLIES**

I would like to place an order for stationery supplies from your mail order catalog. *Enclosed is my order form and a check for \$247.67.* Please process this order as soon as possible. Thank you.

The subject title prints in boldface pica. The entire paragraph prints in elite pitch but with the second sentence as italic characters. And although you do not see it happen, the printer is reset by the code @0. (See the general concepts section of this chapter for more details on Master Reset.)

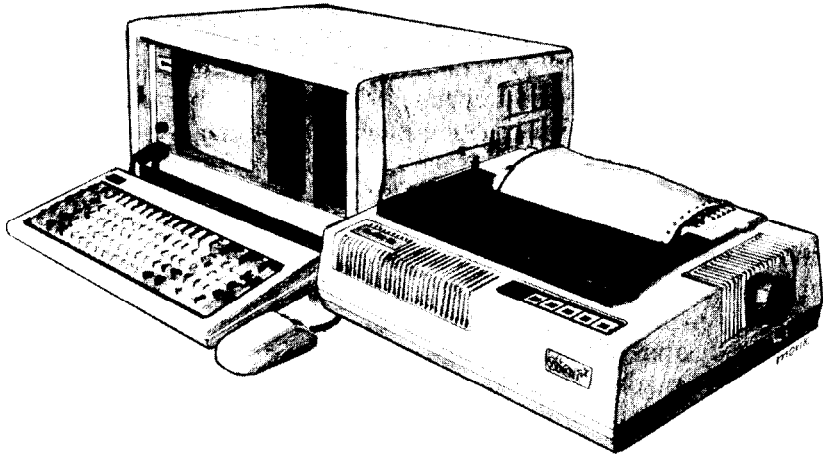
This is just one example, but you should be able to apply most of the function codes to the setup used here. A note about underlining: with Scripsit, Radix will underline the spaces designated for the left margins of each line. You will have to enter the control codes at the beginning and end of each line to turn underlining on and off.

### **Redefining your own print codes**

If you want to define a new print pitch, be sure to start the sequence with the code 27 66 5. This code tells the printer to turn off near letter quality print. Then enter your function code to choose the pitch you want. Otherwise, if you change pitches in a document from near letter quality to draft, the function code alone (in the printer driver) will continue to print near letter quality.

For example, if you found yourself frequently using italic print for large blocks of text in pica pitch, you can combine italic and pica pitch to define italic pica and use it in the ruler line of your document. The ASCII code sequence would be 27 66 5 27 66 1 27 52 which would print italic pica pitch. Remember, you can enter up to 11 codes for each User Print Code.

Also, keep in mind that recognition characters can be combined in your document to print, for instance, italic and underline. Simply enter the two codes before the text. With italic-underline, the codes would be @"@9 to turn it on and @#@! to turn it off.



## Chapter 4

# Using Radix With Spreadsheet Programs

Radix is a good printer to use with spreadsheet programs because its capabilities match the requirements for printing spreadsheets. It can print large spreadsheets fast and it can also print good looking final reports with the Near Letter Quality (NLQ) character set.

**We will look at how to use Radix with three popular spreadsheet programs:**

- VisiCalc
- SuperCalc
- Lotus 1-2-3

## Using Radix With VisiCalc

VisiCalc was the first spreadsheet program. Although it is now widely imitated, it is still one of the most popular. The descriptions here are for VisiCalc as implemented on the IBM-PC; it may work slightly differently on your computer.

VisiCalc spreadsheets are printed with the */Print* command. The */Print* command prints the area of the spreadsheet between the active cell, at the upper left, and a cell that you enter in the command, at the lower right.

First, move the active cell to the cell in the upper left corner of the area that you want to print. Then, to start the print command, enter */P*, and then *P* to direct the output to the printer. Now you must specify the lower right corner of the part of the worksheet that you want to print, and press return to start printing.

This system works fine if the area of the worksheet that you want to print all fits on one page. But if you want to print a larger area and break it into pages, then you must figure out the different areas that you want to print and use separate */Print* commands to print them, moving the active cell to the upper left cell of each area before you print it. You also must use the printer controls to advance the paper to the next page so that you don't print over the perforations on sprocket feed paper.

Radix gives you another way to print large spreadsheets. You can change the width of the characters that Radix prints, and thus print more characters per line. The "Setup" option allows you to send function codes to the printer from VisiCalc. To change the character pitch to condensed, enter "**^CO<RET>**". The following table shows what the shorthand that VisiCalc uses when sending function codes to the printer. The **^** character (the caret on the

**Table 4-1**  
**VisiCalc control codes**

<b>^C</b>	This marks the next character as a control code. For example, if you want to send control-O to the printer, you must enter <b>^CO</b> .
<b>^E</b>	This sends an escape to the printer. Since escape is so widely used, they made a special code for it.
<b>^R</b>	This sends a carriage return to the printer.
<b>^L</b>	This sends a line feed to the printer. You can use this to put blank lines between sections of the worksheet as you print it out.
<b>^H</b>	This code says to treat the next two characters as hexadecimal digits. For example, if you entered <b>^HOF</b> , then VisiCalc would send hex OF (decimal 15, or control-O) to the printer.
<b>^^</b>	This sends one caret character (^) to the printer.

keyboard) signifies the beginning of one of the special codes to VisiCalc.

With these codes you can send any of Radix's function codes through VisiCalc. If you select a different printing style than normal, you will have to enter it before printing each section of the spreadsheet because VisiCalc resets the printer as it starts each /Print command.

The most common codes are to change the print pitch, so the following table shows how many print columns will fit on a page with the various print pitches possible, and the codes to use to get them. If you have a Radix-15, but are using 8½ inch wide paper, use the values for a Radix-10.

**Table 4-2**  
**Print columns on a page with VisiCalc**

Pitch	Radix-10	Radix-15	Setup codes
Pica	80	136	^EB^CA
Elite	96	163	^EB^CB
Condensed	136	233	^EB^CC
Pica Expanded	40	68	^EB^CA ^EW^CA
Elite Expanded	48	81	^EB^CB ^EW^CA
Condensed Expanded	68	116	^EB^CC ^EW^CA
NLQ on	80	136	^EB^CD
NLQ off			^EB^CE

One more thing that you may wish to do is to switch to NLQ printing for a final report. To switch to NLQ, enter “^EB^CD. This will turn on NLQ printing. As you can see, you can use any of Radix's features with VisiCalc, just by entering the proper codes in the “Setup option to the /Print command.

## Using Radix with SuperCalc

SuperCalc is a popular spreadsheet program. It has a lot of flexibility and can utilize many of Radix's advanced features.

The /Output command is used to print SuperCalc spreadsheets. This command allows you great variation in the way you print your spreadsheets.

The simplest way to print a spreadsheet is to enter **/O** to start the */Output* command, type **D** to print the spreadsheet as it is displayed, and then type **ALL**, to specify printing the entire spreadsheet, and finally press **P** to direct the output to the printer.

If your spreadsheet is too wide to fit onto a single sheet of paper, then SuperCalc will automatically split the worksheet into strips (you'll see how to tell SuperCalc how wide your printer is in a moment). First, SuperCalc will print as much of the spreadsheet as will fit on a page, and then it will print additional page-wide strips until the entire worksheet is printed.

To make this automatic system of dividing the spreadsheet into strips work you need to tell SuperCalc how many character columns that you want to print on each page. (Note that we are now talking about character columns, and not spreadsheet columns.) The */Output* command has a Setup option that, among other things, allows you to specify the number of character columns that will fit on a line. To use this option, enter **/O D ALL, S**. This will present you with a menu of setup choices. The selection that we are interested in is "W = Change page width". Select this option and enter the appropriate number of columns for the printing pitch and paper width that you want to use. SuperCalc will remember this setting until you exit the program. (You can make a setting permanent by using the INSTALL program that comes with SuperCalc.)

**Table 4-3**  
**Print columns on a page with SuperCalc**

Pitch	Radix-10	Radix-15	Setup codes
Pica	80	136	<ESC> B Ctrl-A
Elite	96	163	<ESC> B Ctrl-B
Condensed	136	233	<ESC> B Ctrl-C
Pica Expanded	40	68	<ESC> B Ctrl-A <ESC> W Ctrl-A
Elite Expanded	48	81	<ESC> B Ctrl-B <ESC> W Ctrl-A
Condensed Expanded	68	116	<ESC> B Ctrl-C <ESC> W Ctrl-A
NLQ on	80	136	<ESC> B Ctrl-D
NLQ off			<ESC> B Ctrl-E

### **Sending control codes from SuperCalc**

The Setup option of the */Output* command also allows you to send control codes to the printer. The menu item "S = Manual



setup codes" lets you send any type of code to the printer that you wish. When you select "S", a prompt appears that says "Enter codes (CR when done:)." You can then enter any codes that you wish, and when you are done you press return to signal the end of the setup codes. Control codes are sent in the normal manner, by holding down the control key and pressing a letter key. On some computers the <ESC> key will not send an escape code (ASCII 27). If this is a problem, try using control-[. Many times this will send an escape code to the printer.

As an example, to turn on NLQ printing, enter <ESC> **B control-D return**, and then press **P** to start printing the report. Table 4-3 shows the codes required to change the print width to various sizes.

## Lotus 1-2-3

Lotus is one of the new integrated software packages that includes a spreadsheet, a database manager and graphics. We will see how to print Lotus 1-2-3 spreadsheets in this chapter.

Lotus 1-2-3 uses the */Print* command to print spreadsheets. When you enter */P*, a menu appears that presents you with a number of choices. Lotus 1-2-3 gives you a lot of flexibility in printing spreadsheets through this menu, but the only thing you have to do is to define a range to print. All the other items have default values that make getting started easy.

If you do change several of the things listed on the */Print* menu, Lotus 1-2-3 will remember the selections that you have made and use them each time you print the spreadsheet. They are even saved with the spreadsheet so that they will be the same the next time that you use the spreadsheet.

You can specify the range to print in all the normal ways: by pointing, by typing the cell addresses of the endpoints, by entering a range name, or by using the F3 key to point to a range name.

After you have specified a range to print, and changing any of the other options that you wish, begin to print the spreadsheet by selecting the Go option. Lotus 1-2-3 will split the spreadsheet into sections to fit onto pages if it won't all fit on one page.

Let's look at some of the other options on the */Print* menu, and see how they add to the flexibility of printing spreadsheets.

The *Line* option advances the paper one line. Use this to put space between different sections of your spreadsheets when you print them. The *Page* option advances the paper to the top of a

new page. Use this option to start on a new page.

Selecting the *Align* option tells Lotus 1-2-3 that you have moved the paper to the top of a new page. Use this option after using the F.F. button to move the paper or after inserting a new single sheet of paper.

The *Clear* option allows you to clear any or all of the other options that you have selected. The *Quit* option ends the */Print* command and returns you to Ready Mode.

Selecting *Options* from the */Print* menu presents you with some additional page format selections.

You can add *Headers* or *Footers* to each page of your output. A header is a line that prints at the top of each page, while a footer is a line that prints at the bottom of each page.

Lotus 1-2-3 has three characters that perform special functions when they are included in a header or a footer. You can include sequential page numbers on each page by including the # character where you want the page number to print (For example: Page #).

The current date will be printed if you include the @ character in a header or footer. (For example: As of @.)

You can direct sections of headers and footers to the left, right, or center by using the | character. Each header or footer is divided into three sections; Left, center, and right. The | character shows the limits of these sections. So to print a header with the date to the left, a title in the center, and a page number to the right, the header might look like this:

```
@|Spreadsheet Title|Page #
```

And, on January 12, 1984, the results might look like this:

```
12-Jan-84      Spreadsheet Title      Page 1
```

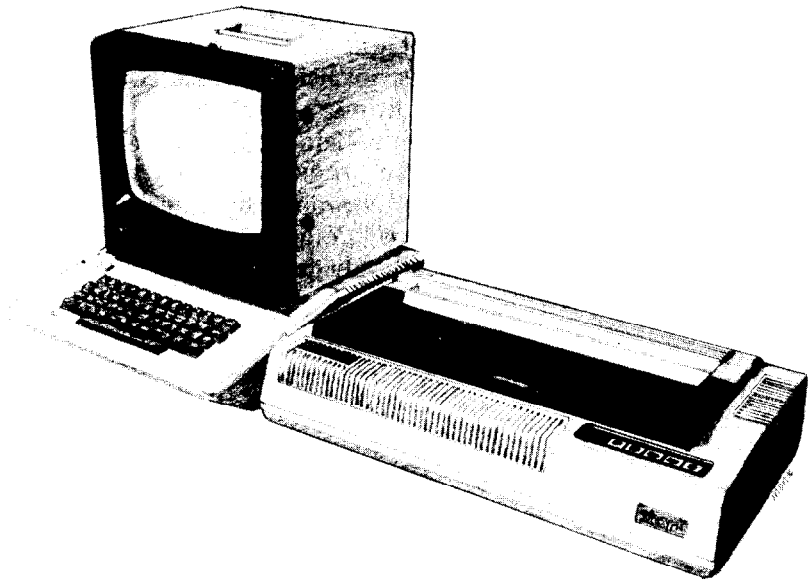
Another of the selections under *Options* is *Setup*. This selection allows you to create a *setup string* that will be sent to the printer before each section of a spreadsheet is printed. You can include non-printing codes in the setup string by using a backslash (\) followed by a three digit number that consists of the decimal ASCII value for the code that you wish to send (with lead-

ing zeros if required). For example, to print a worksheet in condensed print, use the setup string \015. This sends ASCII 15 which is the code for condensed printing. The following table shows how many character columns will fit with different printing widths, and the setup string to get each width.

**Table 4-4**  
**Print columns on a page with Lotus 1-2-3**

Pitch	Radix-10	Radix-15	Setup codes
Pica	80	136	\027B\001
Elite	96	163	\027B\002
Condensed	136	233	\027B\003
Pica Expanded	40	68	\027B\001 \027W\001
Elite Expanded	48	81	\027B\002 \027W\001
Condensed Expanded	68	116	\027B\003 \027W\001
NLQ on	80	136	\027B\004
NLQ off			\027B\005





## Chapter 5

# Using Radix With Graphics Programs

There are now many business graphics programs on the market. Radix is a good printer to use with these programs because of its advanced graphics capabilities. We will look at two graphics programs in some detail. If you have a different graphics program, then hopefully you can use some of the concepts in the programs that we cover to help you with your program.

**In this chapter you'll learn how to use Radix with:**

- **SuperCalc<sup>2</sup>**
- **BPS Business Graphics**

## SuperCalc<sup>3</sup>

SuperCalc<sup>3</sup> can produce 7 different kinds of graphs. Using information contained on the worksheet you can create and print a wide variety of graphs.

Before you start printing graphs you must tell SuperCalc<sup>3</sup> what kind of printer you are using. This is done with the STARTUP program that is furnished on the SuperCalc<sup>3</sup> disk. To use this, type STARTUP at the A> prompt and follow the on-screen directions. If the Star Radix printer is not listed, please consult with your Star dealer for an alternate printer selection.

Once you have started the SuperCalc<sup>3</sup> program there are still some things that you must tell the program about your printer. Use the */Global Graphics Options Command* by entering */G G O*. You will see a menu like that shown in Figure 5-1. There are many things on this screen but we need only concern ourselves with a few.

GRAPHICS AND DEVICE OPTIONS			
Appearance Features:		For graphics printers:	
Grids	N	Resolution	D
Axes	Y	For pen plotters:	
Ticks	Y	Num. pens	2
Graph Box	N	Transparency	N
For Pie, Bar and Stacked-Bar:		Plotter Interface:	
Fill Type	S	Use	S
For Line, Hi-Lo, Area and X-Y:		Parallel Options:	
Point Markers	Y	Printer number	1
Lines	Y	Serial Options:	
GRAPHIC DEVICE SETTINGS:		Com number	1
Console:		Baud Rate	4800
Monitor	C	Parity	N
		Data bits	8
		Stop bits	1
H(orizontal), V(ertical), N(either) or B(oth)?			
25>/Global,Graphics,Options			
F1 = Help; F2 = Erase Line/Return to Spreadsheet; F9 = Plot; F10 = View			

Figure 5-1. SuperCalc<sup>3</sup> /Global Graphics Options menu.

The first selection to make is at the top of the second column. Use the Tab key to quickly move to the "Resolution:" entry. This choice affects the density and quality of the graphs that you print. There are four choices, single through quad density, but if you did not find the Radix listed in the STARTUP program only single and

double will work. We suggest that you select double until you are more familiar with the program.

The next thing that you must specify is the "Plotter Interface" that you are using. Move to this selection and enter P if you are using the parallel interface on the Radix, or S if you are using the serial interface.

Your entry of S or P will affect your next entry also. Depending on which you have selected, a different section of the menu below will be highlighted. Here you need to specify which printer number the Radix is. Generally, printers are connected as printer number 1, so if you don't know any differently, enter 1. Then, if you are using a serial interface you must enter the values that reflect the way that you set the DIP switches on Radix. If these settings are not consistent with the way the switches are set then it will not work.

Once these selections are made, enter /G G S to save your selections to disk so that you won't have to make these selections every time that you want to print a graph.

Now you are ready to print a graph. Create a graph using the /View command, and when it is in the form that you like, press the F9 key to print the graph on Radix.

By using the /Global Graphics Fonts command you can change the style of letters that are on your graphs. SuperCalc<sup>3</sup> has eight different fonts available and you can use any or all of them on any graph.

You can change the size and placement of the graph on the page by using the /Global Graphics Layout command. This command allows you to put up to four graphs on a page.

## **BPS Business Graphics**

BPS Business Graphics is a program that changes groups of numbers into graphs. It is a very flexible program and can produce many different kinds of graphs.

BPS Business Graphics prints graphs by first creating the graph on the screen and then making a copy of the screen on the printer. This means that your computer must have the ability to do graphics on the screen.

Before you can print a graph you must have the proper device driver installed. Use the INSTALL DEVICES B: command with the device installation disk in drive B:. If you do not find RADIX in the list of available devices, please ask your Star dealer for an alternate printer selection. You only need to do this installation once;

BPS Business Graphics will use the same device driver until you install another one.

Now, when you have your points loaded and are ready to print a graph, use the following commands:

```
DRAW (Plus the appropriate arguments)
SET OUTPUT UNIT PARALLEL
WRITE SCREEN (Plus the appropriate device name)
```

Your graph will first be displayed on the screen and then printed on Radix.

There are several ways to vary the way graphs are printed. Obviously you can use the many features of BPS Business Graphics to add elements to your graph. But you can also change the quality of the way the graph is printed.

One way to change the printing is to use the SET FILL CYCLE command. This will change the pattern of each element of the graph as it is drawn. In a pie graph it will change the pattern in each section of the graph.

Another way to improve the quality of the printed graph is to use the DET DEVICE CONSOLE HIBW command. This command tells the program that you have a high resolution monitor. BPS Business Graphics then plots in its highest quality, which may not look very good on your screen if you don't have a high resolution monitor. But when the image is printed it will be in the highest possible quality.

### **Making pie graphs round**

BPS Business Graphics uses what it calls a *rounding factor* to make pie graphs round. Many output devices produce oval pie graphs because their *aspect ratios* don't match. The aspect ratio of an object is the ratio of its height to its width. The rounding factor is the third argument to the DRAW PIE command, so with a rounding factor of 1, the command looks like this: DRAW PIE 0,0,1.

The following table shows the rounding factors to use with the different console modes.

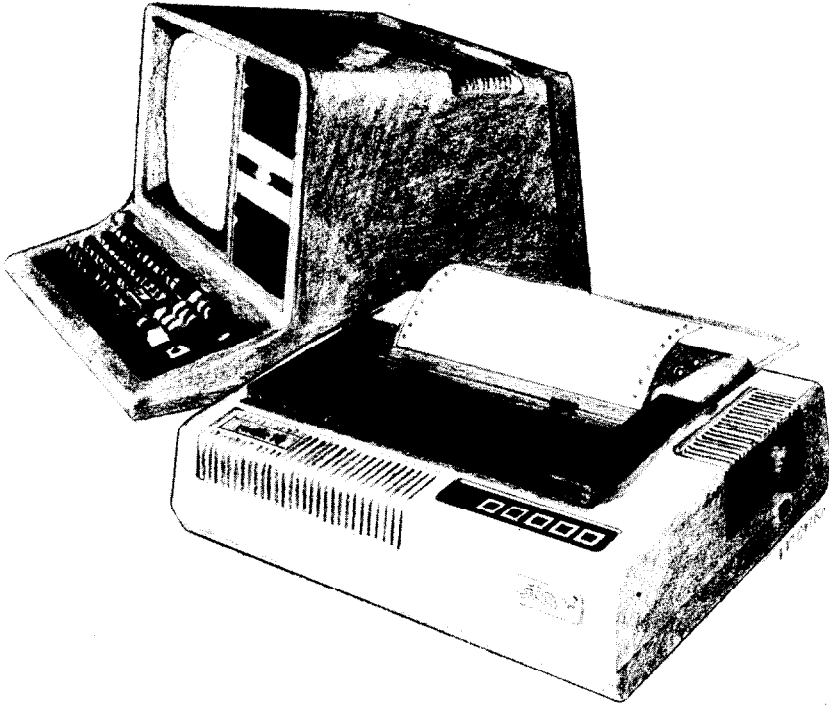
BPS Business Graphics can create many types of graphs, and we have just touched the surface of its capabilities, but with these hints you should be able to get started printing graphs quickly.



**Table 5-1**  
**Rounding factors for console modes**

<b>Console Mode</b>	<b>Rounding Factor</b>
COLOR	1.21
RGB	1.21
TV	0.61
HIBW	2.43
LOBW	1.21





## Chapter 6

# Controlling Radix With BASIC

Throughout the rest of this book we will be teaching you how to use Radix's features using the BASIC programming language in our examples. It is easy to communicate with Radix from BASIC and, though it has its detractors, BASIC is the nearest thing to a universal language among users of personal computers. But remember that it's not the only way to communicate with Radix, as we have already seen.

**Subjects covered in this chapter include:**

- **Listing BASIC programs on the printer**
- **Printing from BASIC**
- **CHR\$ function**
- **Problem codes**

All of the examples in this manual are written in Microsoft

BASIC (specifically, Microsoft BASIC for the IBM Personal Computer). With minor modifications, the examples and utility programs can be adapted to run in any version of BASIC. In this chapter and in the appendix for your computer, we'll tell you what modifications need to be made and how to do it. In this chapter we assume that you have some familiarity with BASIC.

## Some Basics About BASIC

Probably the simplest thing to do with your printer in BASIC is to list a program on the printer. But in this world of proliferating microcomputers even this presents a problem. It seems that every computer uses a different system of communicating with the printer. We are going to tell you about some of the more common ways, and hope that between this and your computer's BASIC manual you will be able to stay with us.

First on our list is Microsoft BASIC's way of communicating with the printer. They just add an "L" to the beginning of the LIST and PRINT commands, making them LLIST and LPRINT. This method is used by more computers than any other and so we will use it throughout this book, after telling the rest of you how to follow along.

Microsoft BASIC is used by TRS-80 computers, IBM-PC computers, many CP/M computers, and many other computers. (Look in your BASIC manual; it will probably say if it's Microsoft BASIC.)

Next we need to talk about Apple II computers. They have a real simple system. To list a program that you have loaded into memory, just type:

```
PR#1  
LIST  
PR#0
```

The PR#1 says "send everything to the printer," the LIST sends it, and the PR#0 says "Ok, back to the screen now."

Some other computers require you to open the printer as a numbered device, and then direct the output to that device. For

example, to list a program on the printer with a Commodore C-64 computer you type the following:

```
OPEN4,4  
CMD4  
LIST  
CLOSE4
```

This says that the printer is device 4, directs the output to it, lists the program, and finally closes device 4.

The appendix gives more information about listing programs on various computers. Find the appendix that tells how your computer works, and try it.

Now that we all know how our computers address the printer, let's try listing a BASIC program. Load a BASIC program and LLIST it (or however your computer does it). We've crossed the first major hurdle—learning how to list programs on Radix. Now we are ready to jump into the world of programming with Radix. But first, there are a few fundamentals that we need to cover.

### ***Establishing communications***

We've learned something about communicating with our printer. Now we need to adapt what we know to printing in a BASIC program. Generally, computers use about the same procedure for printing in a program as they do to list a program. Again take a few moments to look at the appendix that relates to your computer. We'll continue when you have it all figured out.

Welcome back. Let's try what we learned. Type the following:

```
NEW  
10 LPRINT "TESTING"  
RUN
```

Remember—we use LPRINT; you may have to use something else!

At any rate, you should have the word "TESTING" on your printer. Quite an achievement, isn't it? Let's get done with this simple stuff so that we can go on to something interesting.

### **The CHR\$ function**

We mentioned CHR\$ in Chapter 2 as one way to express ASCII codes. We are going to use it a lot in communicating with Radix. Radix uses many of the ASCII codes that don't represent letters and numbers. The CHR\$ function gives us an easy way to send these codes to the printer. Try this to see how the CHR\$ function works:

```
NEW
10 LPRINT CHR$(82)
RUN
```

That should print an "R" for Radix. If you check the chart in Appendix I you will see that 82 is the ASCII code for "R".

### **Control codes**

Radix uses many of the non-printing ASCII codes for control codes. These codes perform a function rather than printing a character. Let's try an easy one right now:

```
NEW
10 LPRINT CHR$(7)
RUN
```

Where did that noise come from? That's Radix's bell. We will learn more about it in Chapter 10. We just wanted to illustrate a code that causes Radix to perform a function.

### **The escape code**

There's one ASCII code that we are going to be using more than all the rest. This is ASCII 27, which is called *escape*. In BASIC it is CHR\$(27). With all of Radix's advanced features, there weren't enough single ASCII codes to access all of them. So escape is used to start sequences of control codes that open a wider range of functions to us.

While you must call this code CHR\$(27) in BASIC, we are going to refer to it as <ESC> in this book. This will make it much easier to recognize when we use it.

A typical escape code sequence starts with <ESC> which is

followed by one or more CHR\$ codes. As an example, the escape code sequence to turn on italic print is:

```
<ESC> CHR$(52)
```

In a program, this would look like this:

```
NEW  
10 LPRINT CHR$(27) CHR$(52);  
20 LPRINT "TESTING"  
RUN
```

Try this program. It will print the word TESTING in italic.

Some of you fast students may have noticed that CHR\$(52) is the same as "4". That's right, the program will work just as well if line 10 is changed like this:

```
10 LPRINT CHR$(27) "4";
```

That's just another form of the same ASCII code, and it's all the same to Radix.

Here's another shortcut for BASIC programmers: since <ESC> is used so often, assign it to a variable. In a long program, typing ESC\$ is much easier than typing CHR\$(27) each time! Now our program looks like this:

```
5 ESC$=CHR$(27)  
10 LPRINT ESC$ "4";
```

Turn your printer off and back on now, or you will be printing in italic for quite a while!

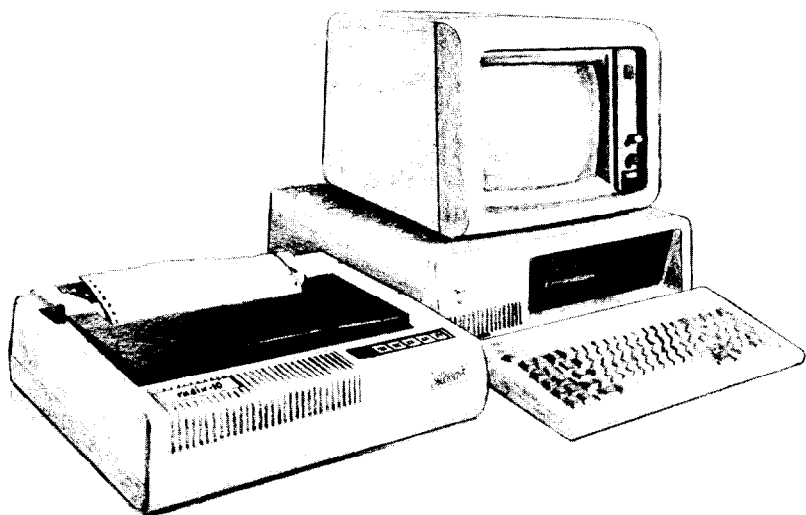
### **Some problem codes**

Before we go too far we need to mention some codes that may cause you problems. Like most of the subjects in this chapter, we have to be a little vague because of the differences in computers.

Nearly all BASICs change some of the ASCII codes between your BASIC program and your printer. Some turn CHR\$(10) (a line feed) into a CHR\$(13) (a carriage return) before sending it on. Some other problem codes are 0, 7, and 9 through 13. Once again we refer you to the appendix about your computer, where some more specific information awaits.

That's it for the basics. You are ready to learn how to use the many features of Radix.





## *Chapter 7*

# **Printing Text With Radix**

Beginning with this chapter we will be exploring all the features of Radix.

**In this chapter we'll cover:**

- **Near letter quality characters**
- **Italics**
- **Underlining**
- **Superscript and subscripts**
- **Print pitch**
- **Print emphasis**

All our examples will be given in Microsoft BASIC as used by the IBM Personal Computer, but remember that you don't need to know BASIC to use Radix's features. Just use the same ASCII codes as we do in our examples.

If your computer doesn't use Microsoft BASIC, look in the appendix to see what changes you need to make for your BASIC. The appendix tells you how to change the short example programs, and gives complete listings of the longer programs, already converted for your computer.

You have already printed a few lines on your Radix printer. Now it's time to start looking at the many variations of printing style that you have available to you.

## **Some Special Kinds of Text**

If you looked carefully at Radix's self test, you noticed that it can print in italics. But there's more! Radix can underline characters, print superscripts and subscripts, and perhaps most exciting, print near letter quality characters.

### **Near Letter Quality characters**

Radix's Near Letter Quality (sometimes abbreviated as NLQ) character set is ideal for correspondence and other important printing, for it takes a keen eye to detect that it is from a dot matrix printer. Normally (unless you have turned DIP switch A-4 off), Radix prints draft quality characters. This is adequate for most work and it prints fastest. But for the final printout, try NLQ. The program below shows how.

```
10 'Demo near letter quality character set.
20 LPRINT CHR$(27) "B" CHR$(4) ; 'Select NLQ.
30 LPRINT "This line shows Radix's NEAR LETTER QUALITY!"
40 LPRINT CHR$(27) "B" CHR$(5) ; 'Select draft.
50 LPRINT "This line shows Radix's standard print."
```

In this program, line 20 selects NLQ characters with the <ESC> "B" CHR\$(4) command. Line 30 prints a sample before line 40 switches Radix back to draft printing with an <ESC> "B" CHR\$(5). When you run the program you should get this:

```
This line shows Radix's NEAR LETTER QUALITY!
This line shows Radix's standard print.
```

**Table 7-1**  
**Near letter quality commands**

Function	Control code
Near letter quality ON	<ESC> "B" CHR\$(4)
Near letter quality OFF	<ESC> "B" CHR\$(5)

### ***Italic printing***

*Italic* letters are letters that are slanted to the right. Radix can print all of its letters except NLQ characters in *italic* as well as the roman (standard) letters you are accustomed to. Italics can be used to give extra emphasis to certain words. The command codes to turn italic on and off are shown in Table 7-2.

**Table 7-2**  
***Italic commands***

Function	Control code
Italic ON	<ESC> "4"
Italic OFF	<ESC> "5"

Use this program to see italic characters:

```

10 'Demo italic and roman.
20 LPRINT CHR$(27) "4" ; 'Italic on.
30 LPRINT "This line is in ITALIC characters."
40 LPRINT CHR$(27) "5" ; 'Italic off.
50 LPRINT "This line is in ROMAN (normal) characters."

```

Here is what you should get:

```

This line is in ITALIC characters.
This line is in ROMAN (normal) characters.

```

This program is easy; line 20 turns italic on with <ESC> "4", and line 40 turns it off with <ESC> "5".

### ***Underlining***

Not only can Radix print all styles of printing in both roman

and italic, but it can underline them too. The control codes are shown in Table 7-3.

**Table 7-3**  
**Underline commands**

Function	Control code
Underline ON	<ESC> “-” CHR\$(1)
Underline OFF	<ESC> “-” CHR\$(0)

Again, that's simple. Let's try it with this program:

```
10 'Demo underlining.
20 LPRINT CHR$(27) "-" CHR$(1) ; 'Underline on.
30 LPRINT "This phrase is UNDERLINED;" ;
40 LPRINT CHR$(27) "-" CHR$(0) ; 'Underline off.
50 LPRINT " this is not."
```

It should come out like this:

```
This phrase is UNDERLINED; this is not.
```

In this program underline is turned on in line 20 with <ESC> “-” CHR\$(1), and then off in line 40 with <ESC> “-” CHR\$(0). There's a new little wrinkle in this program, though. It all printed on one line. The semicolons at the end of the first three lines told BASIC that those lines were to be continued. Therefore, BASIC didn't send a carriage return and line feed at the end of those lines. We just did this to illustrate that all these control codes can be used in the middle of a line. It's easy to underline or *italicize* only part of a line.

### **Superscripts and subscripts**

Radix can print in two different heights of characters. The smaller characters are called *superscripts* and *subscripts* and are half the height of normal characters. *Superscripts* print even with the tops of regular printing while *subscripts* print even with the bottom of regular printing. They are frequently used to reference footnotes, and in mathematical formulas.

Table 7-4 has the codes for using superscripts and subscripts.

**Table 7-4**  
**Superscript and subscript commands**

Function	Control code
Superscript ON	<ESC> "S" CHR\$(0)
Subscript ON	<ESC> "S" CHR\$(1)
Super & subscript OFF	<ESC> "T"

Try this program to see them work:

```

10 'Demo subscripts and superscripts.
20 LPRINT "Look! " ;
30 LPRINT CHR$(27) "S" CHR$(0) ; 'Superscript on.
40 LPRINT "Superscripts " ;
50 LPRINT CHR$(27) "T" ; 'Cancel superscripts.
60 LPRINT "& " ;
70 LPRINT CHR$(27) "S" CHR$(1) ; 'Subscripts on.
80 LPRINT "subscripts " ;
90 LPRINT CHR$(27) "T" ; 'Cancel subscripts.
100 LPRINT "on one line."

```

Look! <sup>superscripts</sup> & <sub>subscripts</sub> on one line.

Here line 30 turns on superscripts with <ESC> "S" CHR\$(0). It's turned off in line 50 with <ESC> "T". Then, between printing text, subscripts are turned on in line 70 with <ESC> "S" CHR\$(1), and finally off in line 90. Again, everything prints on one line because of the semicolons.

## Changing the Print Pitch

In "printer talk," character width is called *pitch*. Normally, Radix prints 10 characters per inch. This is called *pica* pitch because it's the same spacing as a standard pica typewriter.

Radix can also print 12 characters per inch. This is called *elite* pitch because it is the same spacing as an elite typewriter.

Condensed print is approximately 17 characters per inch.

Condensed pitch allows you to get 136 columns of printing on an 8½ inch page.

You tell Radix which pitch you want to use with the <ESC> "B" command. The table below shows three options of this command.

**Table 7-5**  
**Print pitch commands**

Pitch	Characters/inch	Control code
Pica	10	<ESC> "B" CHR\$(1) or CHR\$(18)
Elite	12	<ESC> "B" CHR\$(2)
Condensed	17	<ESC> "B" CHR\$(3) or CHR\$(15)

Let's see how these three pitches look. Try this program:

```

10 'Demo all pitches.
20 LPRINT CHR$(27) "B" CHR$(3) ; 'Select condensed
   pitch.
30 LPRINT "This line is CONDENSED pitch."
40 LPRINT CHR$(27) "B" CHR$(2) ; 'Select elite pitch.
50 LPRINT "This line is ELITE pitch."
60 LPRINT CHR$(27) "B" CHR$(1) ; 'Select pica pitch.
70 LPRINT "This line is PICA pitch (normal)."
```

When you run this program you should get this:

```

This line is CONDENSED pitch.
This line is ELITE pitch.
This line is PICA pitch (normal).
```

Line 20 turns on condensed pitch with <ESC> "B" CHR\$(3). Line 30 prints a line at 17 characters per inch. The <ESC> "B" CHR\$(2) in line 40 changes Radix to elite pitch and line 50 prints a line in elite pitch. Line 60 resets Radix to pica pitch and line 70 prints a line in pica pitch.

Pica pitch and condensed pitch can be set with "shortcut" codes. Instead of using <ESC> "B" CHR\$(n), you can set them with a single code. CHR\$(18) sets pica pitch and CHR\$(15) sets condensed pitch. You can not set elite pitch with a single code.

### Expanded print

Each of Radix's three print pitches can be enlarged to twice its normal width. This is called expanded print. Try this program to see how it works:

```
10 'Demo expanded mode.
20 LPRINT "Demonstration of " ;
30 LPRINT CHR$(14) ; 'Expanded mode on.
40 LPRINT "EXPANDED" ;
50 LPRINT CHR$(20) ; 'Expanded mode off.
60 LPRINT " printing."
70 LPRINT "Notice that " ;
80 LPRINT CHR$(14) ; 'Expanded mode on.
90 LPRINT "EXPANDED mode"
100 LPRINT "automatically turns off at end of a line."
```

Demonstration of **EXPANDED** printing.  
Notice that **EXPANDED mode**  
automatically turns off at end of a line.

Expanded print set with CHR\$(14) is automatically canceled at the end of the line. This is convenient in many applications, such as for one line titles. Note that you don't need to put an <ESC> in front of the CHR\$(14), although <ESC> CHR\$(14) works just the same.

You can also cancel one line expanded print *before* a carriage return with CHR\$(20), as done in line 50.

Sometimes you may wish to stay in expanded print for more than one line. Change your program to this:

```
10 'Demo permanent expanded mode.
20 LPRINT CHR$(27) "W" CHR$(1) ; 'Expanded mode on
   permanently.
30 LPRINT "Permanent expanded"
40 LPRINT "mode stays on until"
50 LPRINT "it is ";
60 LPRINT CHR$(27) "W" CHR$(0) ; 'Expanded mode off.
70 LPRINT "turned off."
```

Now the results look like this:

```
Permanent expanded
mode stays on until
it is turned off.
```

When you turn on expanded print with  $\langle \text{ESC} \rangle$  "W" CHR\$(1) it stays on until you turn it off with  $\langle \text{ESC} \rangle$  "W" CHR\$(0).

**Table 7-6**  
**Expanded print commands**

Function	Control code
One line expanded ON	CHR\$(14) or $\langle \text{ESC} \rangle$ CHR\$(14)
One line expanded OFF	CHR\$(20)
Expanded ON	$\langle \text{ESC} \rangle$ "W" CHR\$(1)
Expanded OFF	$\langle \text{ESC} \rangle$ "W" CHR\$(0)

By combining expanded print with the three pitches, Radix has six different character widths available.

Enter this program to see how the print pitches and expanded print can be combined:

```
10 'Demo pitches in combination with expanded mode.
20 LPRINT CHR$(27) "W" CHR$(1) ; 'Permanent expanded
   mode on.
30 LPRINT CHR$(27) "B" CHR$(3) ; 'Select condensed
   pitch.
40 LPRINT "This line is EXPANDED CONDENSED pitch."
50 LPRINT CHR$(27) "B" CHR$(2) ; 'Select elite pitch.
60 LPRINT "This is EXPANDED ELITE."
70 LPRINT CHR$(27) "B" CHR$(1) ; 'Select pica pitch.
80 LPRINT "This is EXPANDED PICA."
90 LPRINT CHR$(27) "W" CHR$(0) ; 'Permanent expanded
   mode off.
100 LPRINT "This is UNEXPANDED PICA pitch (default)."
```



Here's what you should get from this program:

```

This line is EXPANDED CONDENSED pitch.
This is EXPANDED ELITE.
This is EXPANDED PICA.
This is UNEXPANDED PICA pitch (default).

```

## Making Radix Print Darker

Radix has very good print density when it's just printing regularly. But sometimes you may want something to stand out from the rest of the page. Radix provides two ways to do this: double-strike and emphasized print. Both of these go over the characters twice, but they use slightly different methods to darken the characters. Let's try them and see what the difference is.

The following table shows the control codes for getting into and out of double-strike and emphasized modes.

**Table 7-7**  
**Print emphasis commands**

Function	Control code
Double-strike ON	<ESC> "G"
Double-strike OFF	<ESC> "H"
Emphasized ON	<ESC> "E"
Emphasized OFF	<ESC> "F"

Try them now with this little program:

```

10 'Demo double-strike and emphasized.
20 LPRINT CHR$(27) "G" ; 'Double strike on.
30 LPRINT "This line is DOUBLE-STRIKE printing."
40 LPRINT CHR$(27) "E" ; 'Emphasized on.
50 LPRINT "This line is DOUBLE-STRIKE and EMPHASIZED."
60 LPRINT CHR$(27) "H" ; 'Double strike off.
70 LPRINT "This line is EMPHASIZED printing."
80 LPRINT CHR$(27) "F" ; 'Emphasized off.
90 LPRINT "This line is normal printing."

```

Run this program. The results will look like this:

```

This line is DOUBLE-STRIKE printing.
This line is DOUBLE-STRIKE and EMPHASIZED.
This line is EMPHASIZED printing.
This line is normal printing.

```

Line 20 turns on double-strike with `<ESC> "G"` and line 30 prints a line of text. In line 40 emphasized is turned on with `<ESC> "E"`. Line 50 prints a line of text in double-strike and emphasized. Line 60 then turns double-strike off with `<ESC> "H"` so that line 70 can print in emphasized only. Finally, line 80 turns emphasized off, so that Radix is set for normal printing.

Look closely at the different lines of printing. In the line of double-strike printing each character has been printed twice, and they are moved down just slightly the second time they are printed. In emphasized printing, they are moved slightly to the right the second time Radix prints. The last line combined both of these so that each character was printed 4 times. Now that's pretty nice printing, isn't it?

## Mixing Modes

We have learned how to use Radix's many different printing modes individually. Now let's see how we can combine these modes for even more printing effects. Condensed, italic, double-strike, underlined subscripts are something that you are probably just itching to print!

There are 336 "theoretical" combinations of the modes that we have learned. Of these, a mere 114 will work! (Some combinations, like expanded superscripts, just don't work.) Instead of trying to list all the combinations that work, we have a program that prints a chart showing all the combinations. There is a sample of each of the 114 possible combinations on the chart. (The dots just indicate the few combinations that don't work.) Enter the following program and run it to make your own chart.

```

10 'Prints a chart of all RADIX print styles.
20 WIDTH "LPT1:", 255 'Cancel auto CR & LF after 80
   chars.
30 '

```

```

40 'Initialize constants.
50 ITALIC$ = CHR$(27) + "4"
60 ROMAN$ = CHR$(27) + "5"
70 EXPANDED$ = CHR$(27) + "W" + CHR$(1)
80 NOT.EXPANDED$ = CHR$(27) + "W" + CHR$(0)
90 PICA$ = CHR$(27) + "B" + CHR$(1)
100 ELITE$ = CHR$(27) + "B" + CHR$(2)
110 CONDENSED$ = CHR$(27) + "B" + CHR$(3)
120 NLQ$ = CHR$(27) + "B" + CHR$(4)
130 NOT.NLQ$ = CHR$(27) + "B" + CHR$(5)
140 EMPHASIZED$ = CHR$(27) + "E"
150 NOT.EMPHASIZED$ = CHR$(27) + "F"
160 DOUBLE.STRIKE$ = CHR$(27) + "G"
170 NOT.DOUBLE.STRIKE$ = CHR$(27) + "H"
180 UNDERLINED$ = CHR$(27) + "-" + CHR$(1)
190 NOT.UNDERLINED$ = CHR$(27) + "-" + CHR$(0)
200 SUPERSCRIPT$ = CHR$(27) + "S" + CHR$(0)
210 SUBSCRIPT$ = CHR$(27) + "S" + CHR$(1)
220 NOT.SCRIPTED$ = CHR$(27) + "T"
230 RESET.ALL$ = NOT.EMPHASIZED$ + NOT.UNDERLINED$ +
    NOT.DOUBLE.STRIKE$
240 RESET.ALL$ = RESET.ALL$ + ROMAN$ + PICA$ +
    NOT.EXPANDED$ + NOT.NLQ$
250 TRUE = 1 :FALSE = 0
260 '
270 '
280 'Print heading.
290 LPRINT RESET.ALL$
300 LPRINT EXPANDED$ " NORMAL EXPANDED "
310 LPRINT RESET.ALL$;
320 LPRINT UNDERLINED$ ;
330 LPRINT NLQ$ " NLQ " NOT.NLQ$ ;
340 LPRINT CONDENSED$ "CONDENSED " ;
350 LPRINT ELITE$ " ELITE " ;
360 LPRINT PICA$ " PICA " ;
370 LPRINT CONDENSED$ "CONDENSED " ;
380 LPRINT ELITE$ " ELITE " ;
390 LPRINT PICA$ " PICA "
400 LPRINT RESET.ALL$
410 LPRINT "*REGULAR*"
420 GOSUB 540 'Print four lines regular.
430 LPRINT "*DOUBLE STRIKE*"
440 LPRINT DOUBLE.STRIKE$;
450 DS.OR.EMP = TRUE

```

```

460 GOSUB 540 'Print four lines double strike.
470 LPRINT "*EMPHASIZED*"
480 EMPHASIZED = TRUE
490 GOSUB 540 'Print four lines emphasized.
500 LPRINT "*DOUBLE STRIKE & EMPHASIZED*"
510 LPRINT DOUBLE.STRIKE$ EMPHASIZED$;
520 GOSUB 540 'Print double strike & emphasized.
530 END
540 '
550 'Subroutine to print four lines.
560 'Each shows NLQ, also the three different pitches
570 'are shown in normal and expanded.
580 'Roman, roman underlined, italic, and italic
  underlined.
590 '
600 ITALICS      = FALSE           :LPRINT ROMAN$ ;
610 UNDERLINED = FALSE           :LPRINT NOT.UNDERLINED$ ;
620 EXPANDED    = FALSE           :LPRINT NOT.EXPANDED$ ;
630 PICA        = FALSE
640 '
650 'Produce a line in four different pitches.
660 IF EXPANDED THEN 720
670 IF ITALICS THEN LPRINT "... " ; : GOTO 720
680 IF DS.OR.EMP THEN LPRINT "... " ; : GOTO 720
690 LPRINT NLQ$ ; : NLQ = TRUE
700 GOSUB 860           'Print near-letter-
  quality.
710 LPRINT NOT.NLQ$ ; : NLQ = FALSE
720 LPRINT CONDENSED$ ;
730 GOSUB 860           'Print condensed.
740 LPRINT ELITE$ ;
750 GOSUB 860           'Print elite.
760 LPRINT PICA$ ; : PICA = TRUE
770 GOSUB 860           'Print pica.
780 '
790 'See what has just been done and prepare for next
  line.
800 IF EXPANDED = TRUE THEN LPRINT :GOTO 820
810 LPRINT EXPANDED$; :EXPANDED = TRUE :GOTO 630
820 IF UNDERLINED = TRUE THEN LPRINT :GOTO 840
830 LPRINT UNDERLINED$; :UNDERLINED = TRUE :GOTO 620
840 IF ITALICS = TRUE THEN LPRINT RESET.ALL$ :RETURN
850 LPRINT ITALIC$; :ITALICS = TRUE :GOTO 610
860 '

```

```
870 'Print a small sample showing upper case, lower
    case.
880 'Also show subscripts and superscripts if
    appropriate.
890 '
900 BLANK$ = STRING$(6,32) :FOUR.DOT$ = "...."
910 IF EMPHASIZED = FALSE THEN LPRINT "ABcd"; :GOTO 970
920 IF PICA = FALSE THEN LPRINT FOUR.DOT$; :GOTO 940
930 LPRINT EMPHASIZED$ "ABcd" ;
940 IF EXPANDED = TRUE THEN LPRINT " "; :ELSE
    LPRINT BLANK$;
950 RETURN
960 '
970 'Handle non-emphasized cases.
980 IF EXPANDED OR NLQ THEN LPRINT " "; :RETURN
990 LPRINT SUPERSCRIPT$; "Xx";
1000 LPRINT SUBSCRIPT$; "Yy ";
1010 LPRINT NOT.SCRIPTED$;
1020 RETURN
```

Here is the chart it produces:

NLQ	NORMAL			EXPANDED		
	CONDENSED	ELITE	PICA	CONDENSED	ELITE	PICA
<b>*REGULAR*</b>						
ABcd	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
ABcd	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
<b>*DOUBLE STRIKE*</b>						
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
....	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd <sup>xy</sup>	ABcd	ABcd	ABcd
<b>*EMPHASIZED*</b>						
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd
<b>*DOUBLE STRIKE &amp; EMPHASIZED*</b>						
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd
....	....	....	ABcd	....	....	ABcd

## Summary

### Control code

<ESC> "B" CHR\$(4)  
 <ESC> "B" CHR\$(5)  
 <ESC> "4"  
 <ESC> "5"  
 <ESC> "-" CHR\$(1)  
 <ESC> "-" CHR\$(0)  
 <ESC> "S" CHR\$(0)

### Function

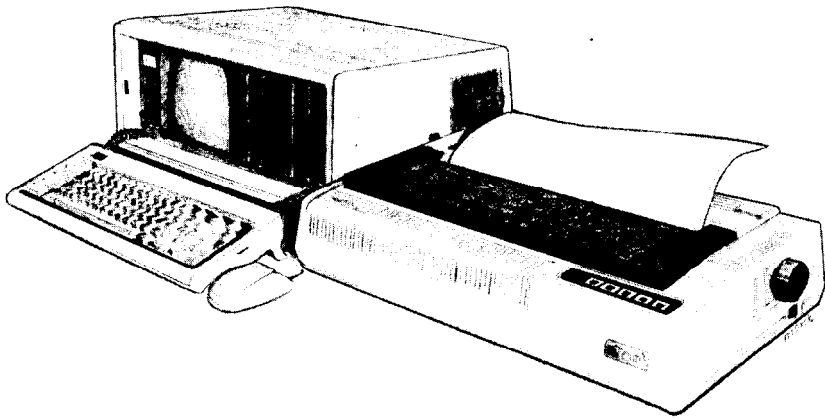
Near letter quality on  
 Near letter quality off  
 Italic on  
 Italic off  
 Underline on  
 Underline off  
 Superscript on

---

<ESC> "S" CHR\$(1)	Subscript on
<ESC> "T"	Super & subscript off
<ESC> "B" CHR\$(1)	Sets pica pitch
<ESC> "B" CHR\$(2)	Sets elite pitch
<ESC> "B" CHR\$(3)	Sets condensed pitch
CHR\$(18)	Sets pica pitch
CHR\$(15)	Sets condensed pitch
CHR\$(14)	One line expanded
<ESC> CHR\$(14)	One line expanded
CHR\$(20)	One line expanded off
<ESC> "W" CHR\$(1)	Expanded on
<ESC> "W" CHR\$(0)	Expanded off
<ESC> "G"	Double-strike on
<ESC> "H"	Double-strike off
<ESC> "E"	Emphasized on
<ESC> "F"	Emphasized off







## *Chapter 8*

# *Line Spacing and Forms Control*

We have learned how to print in many different ways, but so far we haven't looked at how to position the printing on the page.

**In this chapter we will learn how to:**

- **Change the vertical spacing**
- **Change the length of the page**
- **Set top and bottom margins**

## *Starting New Lines*

Up until now the only time we have thought about printing on a new line is when we *didn't* want it to happen. We learned that

putting a semicolon (;) at the end of a BASIC line will not end the line of printing. So somehow, the computer is telling the printer when to end one line and start another.

There are two codes that are used to end one line and start another. They are *carriage return* (CHR\$(13)) and *line feed* (CHR\$(10)). Like the escape code, they have been given abbreviations which you'll find in many texts (including this one): <CR> and <LF>. The codes are simple, but their action is a little confusing (especially with BASIC). Carriage return is the easiest. Each time that the printer receives a CHR\$(13) it returns the print head to the left margin. It does not advance the paper (if DIP switch C-4 is off; see below).

Line feed is more complicated. Each time the printer receives a CHR\$(10) it both advances the paper one line and returns the print head to the left margin, ready to start a new line.

Now to add a little confusion—most (but not all) versions of BASIC add a line feed (CHR\$(10)) to every carriage return (CHR\$(13)) that they send. If your version of BASIC doesn't do this, then you should turn DIP switch C-4 on so that Radix will add the line feed for you. When you have DIP switch C-4 on the printer will do the same thing when it receives a carriage return as it does when it receives a line feed.

If you find that your printer double spaces when it should single space, then you probably need to turn DIP switch C-4 off.

### **Reverse line feeds**

Your Radix printer has a unique capability: it can move the paper up or down! Its unique tractor design allows the paper to be fed in either direction without jamming. This allows you to move around the page at will. You can use this feature to print several columns of text side by side, or print a graph and then move back up and insert descriptive legends. As you experiment you're bound to come up with more uses!

The simplest form of reverse paper feeding is a reverse line feed. The code is <ESC> <LF>, which causes the paper to move down (in effect, moving the printing up) one line. A "line" used in a reverse line feed is the same size as a line in a regular line feed (this is normally 1/6 inch). When you change the line spacing (which you'll read about next), you change it for both forward and reverse line feeds.

**Table 8-1**  
**Line feed commands**

Function	ASCII code	Control code
Return print head to left margin	<CR>	CHR\$(13)
Advance paper one line	<LF>	CHR\$(10)
Reverse paper one line		<ESC>CHR\$(10)

## Changing Line Spacing

When you turn Radix on the line spacing is set to 6 lines per inch (or 8 lines per inch if DIP switch A-5 is off). This is fine for most printing applications, but sometimes you may want something different. Radix makes it easy to set the line spacing to whatever value you want.

Try this program to see how easy it is to change the line spacing:

```

10 'Demo variable line spacing.
20 OPEN "LPT1:" AS #1 : WIDTH #1,255
30 FOR I = 1 TO 25
40 'Set line spacing.
50 PRINT #1,CHR$(27) "A" CHR$(I) ;
60 LPRINT "RADIX line spacing set to " I
70 NEXT I
80 LPRINT "Line spacing is set to 1/6 inch (normal)."
90 'Set line spacing to 1/6 inch (normal).
100 LPRINT CHR$(27) "2" ;
110 CLOSE #1

```

The printout is shown on the next page.

In this program, notice that we're sending codes to the printer a different way. In addition to the LPRINT statements for character strings, we've opened the printer as a random file in line 20. This method, which works with most versions of Microsoft BASIC, allows us to send codes that would otherwise be "problem codes" (such as CHR\$(13), which BASIC automatically follows with a CHR\$(10)). The codes are sent with the PRINT #1 statement in line 50. Unfortunately, this method doesn't work for all computers, but the appendix for your computer shows some other ways to send "problem codes."

Line 50 changes the line spacing. The command  $\langle \text{ESC} \rangle$  "A" CHR\$(n) changes the line spacing to  $n/72$  of an inch. The loop that is started in line 30 increases the value of  $n$  (the variable I in the program) each time it is executed. So the line spacing increases as the program continues. Finally, the  $\langle \text{ESC} \rangle$  "2" in line 100 resets the line spacing to 6 lines per inch. This is a shortcut that is the same as  $\langle \text{ESC} \rangle$  "A" CHR\$(12).

You may wonder why they picked  $1/72$  of an inch as the increment for the line spacing command. There's a good reason: the

```

RADIX line spacing set to 9
RADIX line spacing set to 10
RADIX line spacing set to 11
RADIX line spacing set to 12
RADIX line spacing set to 13
RADIX line spacing set to 14
RADIX line spacing set to 15
RADIX line spacing set to 16
RADIX line spacing set to 17
RADIX line spacing set to 18
RADIX line spacing set to 19
RADIX line spacing set to 20
RADIX line spacing set to 21
RADIX line spacing set to 22
RADIX line spacing set to 23
RADIX line spacing set to 24
RADIX line spacing set to 25

```

Line spacing is set to  $1/6$  inch (normal).

dots that the printer makes are  $1/72$  inch apart. So this means that you can vary the line spacing in increments as fine as one dot—unless you want finer spacing, like one half dot spacing.

The  $\langle \text{ESC} \rangle$  "3"  $\text{CHR}\$(n)$  command sets the line spacing in increments of 1/144 inch. Change line 50 in your program so it is like this:

```
50 PRINT #1,CHR$(27) "3" CHR$(I) ;
```

and run the program again. Now the results will look like this:

```

RADIX line spacing set to 16
RADIX line spacing set to 17
RADIX line spacing set to 18
RADIX line spacing set to 19
RADIX line spacing set to 20
RADIX line spacing set to 21
RADIX line spacing set to 22
RADIX line spacing set to 23
RADIX line spacing set to 24
RADIX line spacing set to 25
Line spacing is set to 1/6 inch (normal).

```

The program works just the same as before, but the line spacings are just half what they were. This is because  $\langle \text{ESC} \rangle$  "3"  $\text{CHR}\$(n)$  sets the line spacing to  $n/144$  inch.

Table 8-2 shows all the line spacing commands, including several "shortcut" commands for commonly used line spacings.

Let's take a look at the last two commands in the table, which give a one-time line feed (or reverse line feed) of  $n/144$  inch. The  $\langle \text{ESC} \rangle$  "J"  $\text{CHR}\$(n)$  command does not change the setting of the line spacing, but it does cause the printer to make one line feed of  $n/144$  inch. Try this program to see how it works:

```

10 'Demo one-time line feeds.
20 LPRINT "Line number 1."
30 LPRINT "Line number 2." ;
40 'One time line feed 100/144 inch.

```

**Table 8-2**  
**Line spacing commands**

Function	Control code
Set line spacing to $n/72$ inch	<ESC> "A" CHR\$(n)
Set line spacing to $n/144$ inch	<ESC> "3" CHR\$(n)
Set line spacing to 1/8 inch	<ESC> "0"
Set line spacing to 7/72 inch	<ESC> "1"
Set line spacing to 1/6 inch	<ESC> "2"
One-time line feed of $n/144$ inch	<ESC> "J" CHR\$(n)
One-time reverse line feed of $n/144$ inch	<ESC> "j" CHR\$(n)
Advance paper $n$ lines	<ESC> "a" CHR\$(n)

**Note:** If your computer does not support lowercase characters, use CHR\$(106) and CHR\$(97) for "j" and "a," respectively.

```
50 LPRINT CHR$(27) "J" CHR$(100) ;
60 LPRINT "Line number 3."
70 LPRINT "Line number 4."
```

Here is what Radix will produce:

```
Line number 1.
Line number 2.
```

```
Line number 3.
Line number 4.
```

The <ESC> "J" CHR\$(100) in line 50 changes the line spacing to 100/144 for one line only. The rest of the lines are printed with the normal line spacing. Notice that both line 30 and line 50 end with semicolons. This prevents the normal line feed from occurring.

The <ESC> "j" CHR\$(n) command works the same way except that the paper moves in the opposite direction. Try this simple change to your program and see what a difference it makes!

```
40 'One time reverse line feed 100/144 inch.
50 LPRINT CHR$(27) "j" CHR$(100) ;
```

```
Line number 3.  
Line number 4.
```

```
Line number 1.  
Line number 2.
```

The value of *n* in all four commands (<ESC> “A”, <ESC> “3”, <ESC> “J”, and <ESC> “j”) can range from 0 to 255. A value of 0 means that there is no line spacing. This allows you to print multiple lines in the same position on the page. This is useful when you want to overprint graphics and text.

### **Moving down the page without a carriage return**

So far, all the commands that move the paper also move the print head to the left margin. And normally this is what you want. Sometimes, though, you may wish to move down the page without moving the printhead back to the left margin. The <ESC> “a” CHR\$(*n*) command does just that. This command advances the paper *n* lines (using whatever the current line spacing is) without moving the printhead. Change lines 40 and 50 of your program so that they are like this:

```
40 'Advance paper 3 lines.  
50 LPRINT CHR$(27) "a" CHR$(3) ;
```

Now when you run the program the results will look like this:

```
Line number 1.  
Line number 2.
```

```
Line number 3.  
Line number 4.
```

The new line 50 moves the paper up 3 lines, but the printhead doesn't move. Therefore, line 60 prints its message starting in the column that the printhead was left in at the end of line 30.

## Forms Controls

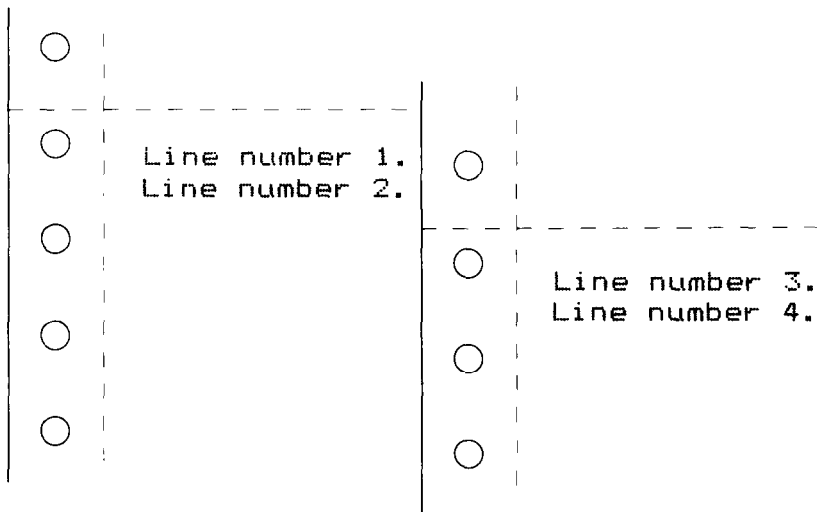
We have seen how to control the spacing between lines on a page. Radix also has commands that control the placement of printing on the page, and even adjust for different size pages.

### Form feed

The simplest forms control code is the *form feed*. Form feed (or <FF>) is CHR\$(12) and causes the printer to move the paper to the top of the next sheet. Try it by changing lines 40 and 50 to this:

```
40 'Form feed
50 LPRINT CHR$(12) ;
```

Before you run the program, turn your printer off and adjust the paper so that the top of the sheet is even with the top of the ribbon guide on the print head, then turn the printer back on. If you don't remember how to do this, review Chapter 1. When you run the program, the results will look like this:





The form feed (CHR\$(12)) in line 50 caused the printer to move to the top of a new page before printing the last two lines.

A note to TRS-80 users: CHR\$(12) is a problem code for the TRS-80. To send a form feed command to Radix you must add 128 to it making it CHR\$(140). Use CHR\$(140) where we use CHR\$(12) in these programs.

### Reverse form feed

Just as Radix can perform a reverse line feed, it can do a reverse form feed. This code moves the paper so that the print head is positioned at the top of the current page. This can be used, for example, to print text in a multi-column magazine format; print the first column, then reverse form feed back to the top of the page to start the second column. The code for reverse form feed is easy to remember: <ESC> <FF>.

**Table 8-3**  
**Form feed commands**

Function	ASCII code	Control code
Advance paper to top of next page	<FF>	CHR\$(12)
Reverse paper to top of current page		<ESC> CHR\$(12)

## Changing the Page Length

You may have some computer forms that you wish to use with Radix that are not 11 inches high. That's no problem, because you can tell Radix how high the forms are that you are using. There are two commands for doing this, shown in this table:

**Table 8-4**  
**Form length commands**

Function	Control code
Set the page length to n lines	<ESC> "C" CHR\$(n)
Set the page length to n inches	<ESC> "C" CHR\$(0) CHR\$(n)

Let's set up a 7 inch high form length, which is typical of many computer checks. The following program will do it.

```

10 'Demo variable form lengths.
20 LPRINT CHR$(27) "C" CHR$(0) CHR$(7) ; 'Form length 7
   inches.
30 LPRINT "Pay to the order of:"
40 LPRINT CHR$(12) ; 'Form feed.
50 LPRINT "Pay to the order of:"

```

This program should print "Pay to the order of:" twice, and they should be 7 inches apart. Line 20 sets the form length to 7 inches. After line 30 prints, line 40 sends a form feed to advance the paper to the top of the next form. Line 50 then prints its message.

After you have run this program, turn off the printer and adjust the top of form position. When you turn the printer back on the page length will be reset to its normal setting (usually 11 inches).

## Top and Bottom Margins

Many programs that use a printer don't keep track of where they are printing on the page. This causes a problem when you get to the bottom of a page because these programs just keep on printing, right over the perforation. This makes it very hard to read, especially if a line happens to fall right on the perforation. And if you separate the pages then you are really in trouble.

Of course Radix has a solution to this predicament. Radix can keep track of the position on the page, and advance the paper so that you won't print too near the perforation. There are two commands to do this. One controls the space at the top of the page and the other controls the space at the bottom of the page. The control codes are given in the following table.

**Table 8-5**  
**Top and bottom margin commands**

Function	Control code
Set top margin	<ESC> "R" CHR\$(n)
Set bottom margin	<ESC> "N" CHR\$(n)
Clear top and bottom margins	<ESC> "O"

In both cases the value of  $n$  tells Radix how many lines to skip, although there is a slight difference in the usage. When you set the top margin with  $\langle \text{ESC} \rangle$  "R" CHR\$( $n$ ), the value of  $n$  tells Radix what line to start printing on. When you set the bottom margin with  $\langle \text{ESC} \rangle$  "N" CHR\$( $n$ ), the value of  $n$  tells Radix how many blank lines should be left at the bottom of the page.

Let's try a simple application to see how these margins work. Enter this program, which will print 150 lines *without* top and bottom margins.

```
10 'Demo top and bottom margins.
20 LPRINT CHR$(12) ; 'Form feed.
30 FOR I = 1 TO 150
40 LPRINT "This is line" I
50 NEXT I
60 LPRINT CHR$(12) ; 'Form feed.
```

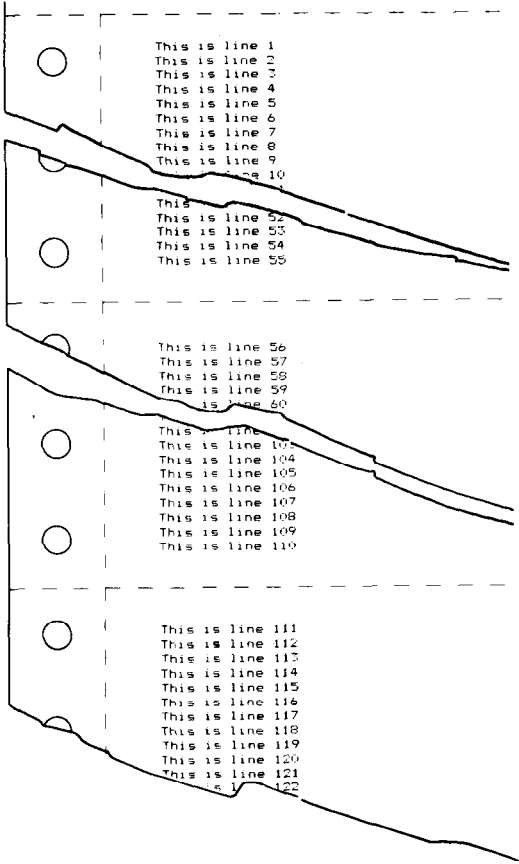
When you run this program it will print 150 lines right down the page and across the perforations. When it's done line 60 sends a form feed to advance the paper to the top of the next page. Look at the lines that have printed near the perforations. Separate the sheets and see if any of the lines have been torn in half. These are the problems that the top and bottom margins will solve.

Now add the following lines to your program. (Don't forget the semicolons or you won't get quite the same results that we did.)

```
11 'Leave 6 blank lines at bottom of page.
12 LPRINT CHR$(27) "N" CHR$(6) ;
13 'Start top of page at line 6.
14 LPRINT CHR$(27) "R" CHR$(6) ;
55 LPRINT CHR$(27) "O" ; 'Clear top & bottom margins.
```

Now when you run the program Radix will skip the first six lines and the last six lines on each page. Always send a form feed after setting the top margin, or it will not work on the first page printed. That's because the top margin only takes effect after a form feed.

Line 14 sets the top margin, line 12 sets the bottom margin, and line 55 clears both margins when we are done.



## Summary

### Control code

CHR\$(10)  
 <ESC> CHR\$(10)  
 CHR\$(13)  
 <ESC> "A" CHR\$(n)  
 <ESC> "3" CHR\$(n)  
 <ESC> "0"  
 <ESC> "1"  
 <ESC> "2"

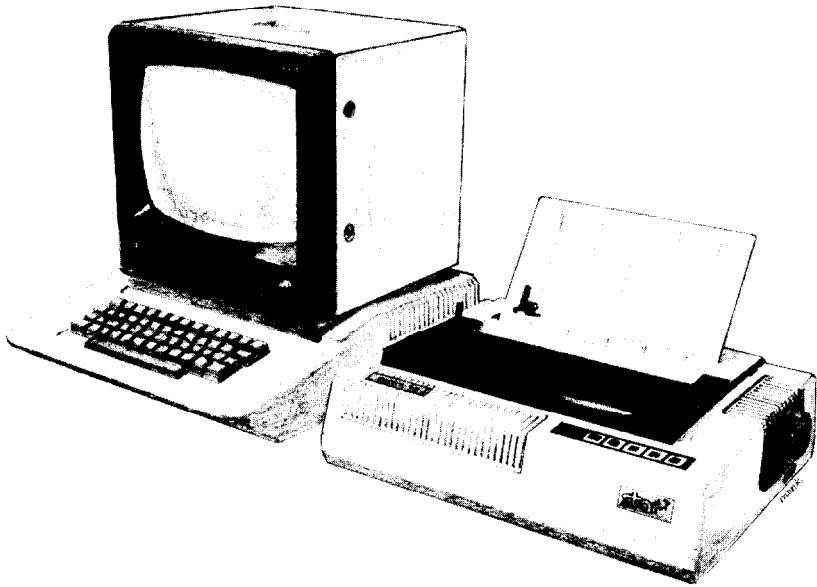
### Function

Line feed  
 Reverse line feed  
 Carriage return  
 Set line spacing to  $n/72$  inch  
 Set line spacing to  $n/144$  inch  
 Set line spacing to  $1/8$  inch  
 Set line spacing to  $7/72$  inch  
 Set line spacing to  $1/6$  inch

---

<ESC> "J" CHR\$(n)	One-time line feed of n/144 inch
<ESC> "j" CHR\$(n)	One-time reverse line feed of n/144 inch
<ESC> "a" CHR\$(n)	Advance the paper n lines
CHR\$(12)	Form feed
<ESC> CHR\$(12)	Reverse form feed
<ESC> "C" CHR\$(n)	Set page length to n lines
<ESC> "C" CHR\$(0) CHR\$(n)	Set page length to n inches
<ESC> "R" CHR\$(n)	Set top margin; start printing on line n
<ESC> "N" CHR\$(n)	Set bottom margin; leave n lines blank
<ESC> "O"	Clear top and bottom margins





## Chapter 9

# Formatting Your Output

You have probably used the tab and margin features on a typewriter. They make it easier to format the text on a page. Radix also has tabs and margins that you can set. But it goes beyond the capabilities of a typewriter because besides having tabs that go across the page, called *horizontal tabs*, Radix has *vertical tabs* that go down the page.

**In this chapter we will discover how to use:**

- **Horizontal tabs**
- **Vertical tabs**
- **Left and right margins**

## Using Horizontal Tabs

When you turn Radix on there are horizontal tabs set automatically every ten spaces. If you start counting at column 1 they are at columns 10, 20, 30, 40, etc. It's easy to use these tabs; you just send a CHR\$(9) to Radix and the print head will move to the next tab position. CHR\$(9) is the ASCII code <HT> for *horizontal tab*.

Try this one line program to demonstrate the use of the default horizontal tabs.

```
10 'Tabs demo.
20 LPRINT "one" CHR$(9) "two" CHR$(9) "three" CHR$(9)
   "four"
```

Here's what will print:

```
one           two           three          four
```

Even though the words are different lengths, they are spaced out evenly by the horizontal tabs.

CHR\$(9) is a problem with some computers. Some BASICs convert CHR\$(9) to a group of spaces that act like a sort of *pseudotab*. This is fine if the computer and the printer have the same tab settings, but it doesn't allow us to use our own tab settings on Radix. We can "outsmart" these computers by adding 128 to the ASCII value that we use. Instead of using CHR\$(9), use CHR\$(137) for a tab command. Even this trick won't work for Apple II computers, for they use CHR\$(9) for something else entirely. Apple users can get some help in Appendix C.

Now add the following line to your program to set different horizontal tabs:

```
15 LPRINT CHR$(27) "D" CHR$(8) CHR$(16) CHR$(24) CHR$(0)
```

<ESC> "D" is the command to begin setting horizontal tabs. It must be followed by characters representing the positions that you want the tabs set. In our program we are setting tabs in col-



umns 8, 16, and 24. The CHR\$(0) at the end ends the string of tabs. In fact, any character that is not greater than the previous one will stop setting tabs. This means that you must put all your tab values in order, from least to greatest, or they won't all get set. (It also means that a CHR\$(1) is just as good as a CHR\$(0) for ending a group of tabs; some computers have trouble sending CHR\$(0).)

When you run the program now it produces this:

one        two        three        four

The words are now closer together, but still evenly spaced. Turn your printer off and on again to reset the default tabs.

If you set tabs in one pitch, such as pica, and then change the pitch, say to elite, the tab settings will also change. If, for example, the tabs are set every eight spaces, when you change pitch they will still be set every eight spaces, but the spaces will be a different width.

### A one-shot tab command

Suppose you need to move to a position across the page, but you only need to do it once. It doesn't make much sense to set up a tab to use only one time. There must be an easier way—and of course there is.

The solution is called a *one-time tab* and is <ESC> "b" CHR\$(n). This command moves the print head n columns to the right. It has the same effect as sending n spaces to the printer.

**Table 9-1**  
**Horizontal tab commands**

Function	Control code
Advance to next tab position	CHR\$(9)
Set tabs at n1, n2, etc.	<ESC> "D" CHR\$(n1) CHR\$(n2)...CHR\$(0)
One-time tab of n spaces	<ESC> "b" CHR\$(n)

**Note:** If your computer does not support lowercase characters, use CHR\$(98) for "b."

## Setting Left and Right Margins

Radix's left and right margins work just like a typewriter—



When you want to reset the margins to the default values, you have two choices. You can either turn the printer off and back on, or you can set margin values equal to the default values. This means that you should set a left margin of 1 and a right margin of 80 on Radix-10 or 136 on Radix-15.

If you change the pitch of your printing after you set your margins, the margins will not change. They stay at the same place on the page. So if you set the margins to give you 65 columns of printing when you are using pica type, and then you change to elite type you will have room for more than 65 columns of elite printing between the margins.

## Using Vertical Tabs

Vertical tabs have the same kinds of uses that horizontal tabs do—they just work in the other direction. Horizontal tabs allow you to reach a specific column on the page no matter where you start from. Vertical tabs are the same. If you have a vertical tab set at line 20, a <VT> (or vertical tab) will move you to line 20 whether you start from line 5 or line 19.

The default vertical tab settings are every six lines. If you send a CHR\$(11), which is the ASCII code for <VT>, before we have set up tabs it will advance the paper to one of these preset tabs. Enter this program to see how this works.

```
10 'Demo vertical tabs.
20 LPRINT CHR$(11) "First tab."
30 LPRINT CHR$(11) "Second tab."
40 LPRINT CHR$(11) "Third tab."
50 LPRINT CHR$(11) "Fourth tab."
```

The CHR\$(11) in each line advances the paper to the next vertical tab. The lines should be spaced evenly, six lines apart.

Now let's set some vertical tabs of our own. Add these lines to the program:

```
12 LPRINT CHR$(27) "P" CHR$(10) ;
14 LPRINT CHR$(20) CHR$(40) CHR$(50) CHR$(0) ;
```

`<ESC> "P"` is the command to set vertical tabs. Like the horizontal tab setting command, tab positions must be defined in ascending order. Our example sets vertical tabs at lines 10, 20, 40 and 50. Then the `CHR$(11)` in each of the following lines advances the paper to the next vertical tab. The printout is shown below.

Add one more line to the program to demonstrate one more feature of vertical tabs.

```
60 LPRINT CHR$(11) "Fifth tab."
```

Now when you run the program the first page looks just like before, but line 60 sends one more `<VT>` than there are tabs. This doesn't confuse Radix—it advances the paper to the next tab position which happens to be the first tab position on the next page. That's nice, isn't it?

First tab.

Second tab.

Third tab.

Fourth tab.

**A one-shot vertical tab command**

There's a one-time vertical tab command that works just like the one-time horizontal tab command. It is  $\langle \text{ESC} \rangle$  "a" CHR\$(n), and it causes the paper to advance n lines. It doesn't change the settings of the vertical tabs.

**Table 9-3**  
**Vertical tab commands**

Function	Control code
Advance paper to next tab position	CHR\$(11)
Set vertical tabs at n1, n2, etc.	$\langle \text{ESC} \rangle$ "P" CHR\$(n1) CHR\$(n2)...CHR\$(0)
Advance paper n lines	$\langle \text{ESC} \rangle$ "a" CHR\$(n)

**Note:** If your computer does not support lowercase characters, use CHR\$(97) for "a."

**Summary**

Control code	Function
CHR\$(9)	Horizontal tab
$\langle \text{ESC} \rangle$ "D" n1 n2 n3 . . . CHR\$(0)	Set horizontal tabs
$\langle \text{ESC} \rangle$ "b" n	One-time horizontal tab of n spaces
$\langle \text{ESC} \rangle$ "M" n	Set left margin
$\langle \text{ESC} \rangle$ "Q" n	Set right margin
CHR\$(11)	Vertical tab
$\langle \text{ESC} \rangle$ "P" n1 n2 n3 . . . CHR\$(0)	Set vertical tabs
$\langle \text{ESC} \rangle$ "a" n	One-time vertical tab of n lines

