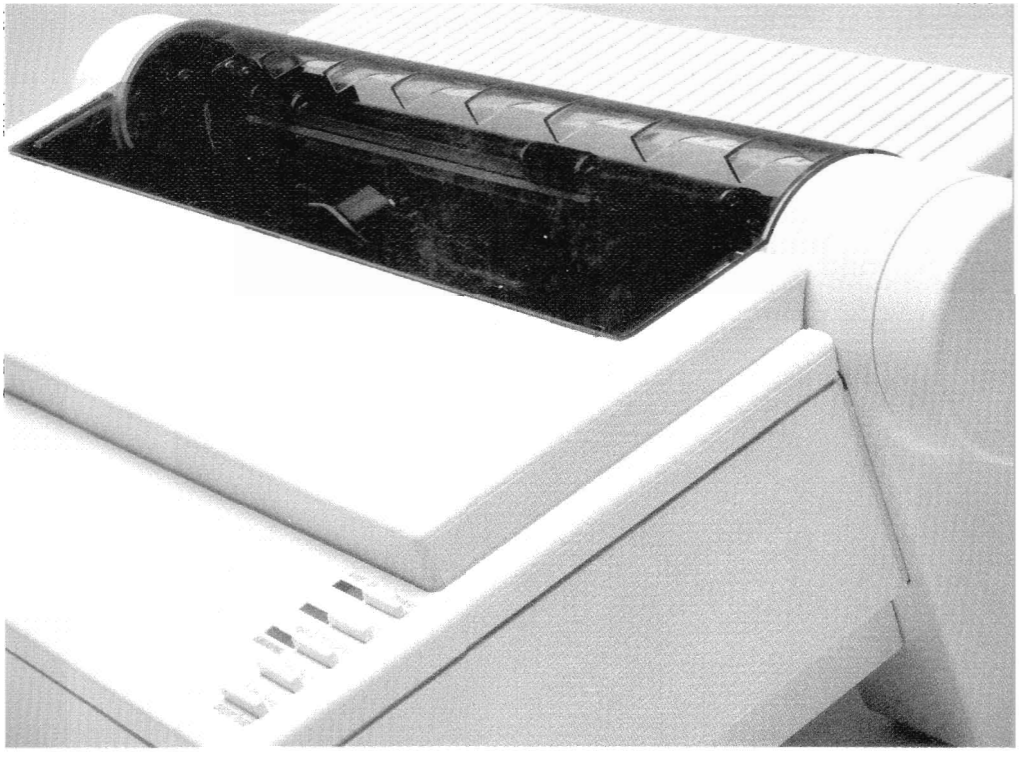
ImageWriter™ II
Technical Reference Manual



For all Macintosh™ and Apple® II Computers.

Limited Warranty on Media and Replacement

If you discover physical defects in the manuals distributed with an Apple product or in the media on which a software product is distributed, Apple will replace the media or manuals at no charge to you, provided you return the item to be replaced with proof of purchase to Apple or an authorized Apple dealer during the 90-day period after you purchased the software. In addition, Apple will replace damaged software media and manuals for as long as the software product is included in Apple's Media Exchange Program. While not an upgrade or update method, this program offers additional protection for up to two years or more from the date of your original purchase. See your authorized Apple dealer for program coverage and details. In some countries the replacement period may be different; check with your authorized Apple dealer.

ALL IMPLIED WARRANTIES ON THE MEDIA AND MANUALS, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

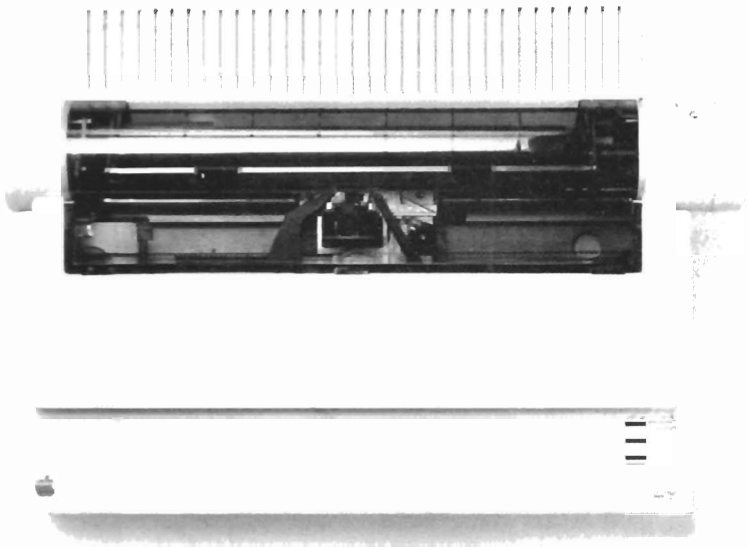
Even though Apple has tested the software and reviewed the documentation, **APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO SOFTWARE, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE IS SOLD "AS IS," AND YOU THE PURCHASER ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.**

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE OR ITS DOCUMENTATION, even if advised of the possibility of such damages. In particular, Apple shall have no liability for any programs or data stored in or used with Apple products, including the costs of recovering such programs or data.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Apple® ImageWriter™ II Technical Reference Manual



Addison-Wesley Publishing Company, Inc.
Reading, Massachusetts Menlo Park, California Don Mills, Ontario
Wokingham, England Amsterdam Sydney Singapore Tokyo
Madrid Bogotá Santiago San Juan

ISBN 0-201-17739-0

ABCDEFGHIJ-DO-89876
First printing, February 1986

Apple® ImageWriter™ II Technical Reference Manual

Acknowledgments

Manual Team

Writers: Paul Black, Dave Marson

Editor: Jody B. Larson

Project Supervisor: Rolly Reed

Art Supervisor: Nancy Hecht

Production Supervisor: Rani Cochran

Product Team

Product Management: Curtis Sasaki

Product Coordination: Ken Jochims

Engineering: Allan Erbes, Matthew Hernandez, Tak Mori, Lyle Morris, Eric Ness, Mark Pruitt, Roger Thomson

Testing: Brig Mills, Randy Thelen

Contents

Figures and Tables

About This Manual

Welcome to the Apple® ImageWriter™ II Technical Reference Manual. The information in this manual is provided for experienced computer users who would like to take advantage of some of the advanced features of the ImageWriter II. To use this manual, you should have some knowledge of a programming language, such as BASIC, Pascal, Logo, or assembly language, and should be familiar with the computer with which you intend to use your ImageWriter.

Important! You don't need to use this manual for running packaged programs for your Apple computer. This manual is for you if you are an experienced user who is running a program that can be configured to take advantage of the advanced ImageWriter II features, or if you are writing or modifying a program to use the ImageWriter II.

Instructions for connecting the ImageWriter II to your computer, for loading paper, changing ribbons, installing options, and other routine operating tasks are provided in the *ImageWriter II Owner's Manual*, which comes with your printer.

What This Manual Contains

The following is a brief outline of the contents of this manual. See the Table of Contents for a complete list of the subjects covered in each chapter. See Appendix A and the Quick Reference Card for summaries of all the commands, and the Index for a cross reference of topics discussed in this book.

- Chapter 1, **Introduction**, summarizes the ImageWriter II's features, provides an introduction to the use of the ImageWriter commands described in Chapters 3 through 8, explains how to use the examples found throughout this book, and gives programming tips for a variety of computer languages.

- Chapter 2, **DIP Switches**, describes the functions of the DIP Switches found under the front cover of the printer. These switches determine the configuration of certain printer features when the printer is turned on or reset.
- Chapter 3, **Software Switches**, describes commands that modify the DIP switch settings or perform functions assigned to DIP switches in earlier ImageWriter printers.
- Chapter 4, **Character Sets and Type Attributes**, describes command codes that select the character set used and that directly affect the appearance of the text, such as character spacing and underlining.
- Chapter 5, **Page Formatting**, describes command codes that directly affect the appearance of the page, such as the left margin setting, that control the movement of the print head, such as tabs, and that control the movement of paper through the printer, such as form feeds and line feeds. This chapter also describes commands that enable the automatic insertion of line feeds and carriage returns.
- Chapter 6, **Miscellaneous Commands**, describes commands that do not fit in the categories covered in other chapters, such as software reset and self ID.
- Chapter 7, **Custom Characters**, explains how to create your own custom characters, how to store them in the ImageWriter II's memory, and how to fetch them from memory and print them.
- Chapter 8, **Graphics and Color Printing**, explains how to create graphics patterns and control the placement of individual dots on the page; and describes the use of the color print codes to create color graphics or to print text in color.
- Chapter 9, **ImageWriter II Options**, describes variations in the use or function of printer commands caused by the installation of the SheetFeeder, 32K Memory, and AppleTalk options.
- Appendix A, **Command Summary**, provides a reference list of printer commands, default settings, DIP switch settings, and software switch settings.
- Appendix B, **ASCII, Binary, and Hexadecimal Codes**, lists ASCII characters, MouseText characters, and binary, decimal, and hexadecimal equivalent values for each ASCII code.
- Appendix C, **Character Specifications**, shows exactly how each character in each of the standard ImageWriter II fonts is printed.
- Appendix D, **Printer Specifications**, lists technical specifications for the ImageWriter II.
- Appendix E, **Interface Specifications**, provides technical information about the ImageWriter II's serial interface.

How to Use This Manual

Before reading about any of the control codes described in this manual, read the rest of this preface and all of Chapter 1 to learn about the conventions used in this manual, the nature of ImageWriter II commands, and the general principles involved in including ImageWriter control codes in your programs. Then look through the rest of the manual to get an idea of the features available, which ones are controlled by DIP switches, which by software commands, and which by both switches and commands. There is no need to read the description of a given command in detail until you are ready to use it.

If you want to use any of the software switch commands (Chapter 3), custom character commands (Chapter 7), or graphics commands (Chapter 8), read the introductory material in the appropriate chapter first before reading about the specific commands.

Once you are familiar with the purpose and use of a command, you need only refer to Appendix A or the Quick Reference Card to remind yourself of the exact command syntax when you want to use it.

This manual includes many example programs that show you how to use the various ImageWriter II commands in both Applesoft BASIC and Macintosh™ Pascal. Short examples are often presented as sample character strings. These strings are shown with a space between each two characters to improve legibility; do *not* include these spaces when you use the commands. For example, the command to set the left margin to column 10 is shown as:

```
ESCAPE L 0 1 0
```

Send this character string to the printer as:

```
[ESC]L010
```

Aids to Understanding

Look for these visual cues throughout the manual:

| ***By the Way:*** Notes like this contain sidelights or interesting pieces of information.

| Information set off like this contains important information that you should read before proceeding.

Applesoft BASIC | Information set off in this manner, with the name of an Apple computer, peripheral device, or programming language in the margin, applies specifically to that computer, device, or language.

▲Warning | Warnings like this direct your attention to something that could damage either software or hardware. They also are used to alert you to important safety precautions.

Marginal notes are used to provide references, definitions, and other helpful information.

Boldfaced terms are defined in the glossary.

A special typeface is used for characters that you type:

```
It looks like this.
```

Keys that you press look like this: ←, ♁, RETURN. When you see a hyphen joining two keys, it means to press them simultaneously. For instance, ♁-CONTROL-RESET means all three keys should be pressed at the same time.

In general, commands are sent to the printer with no spaces between characters in the command. Spaces between characters in sample commands in this manual are for legibility only; when the space character is included as part of a command, it is shown as SPACE.

Dollar signs: In this manual, hexadecimal numbers are preceded by a dollar sign ($), except in tables where hexadecimal numbers are clearly labeled; for example, the hexadecimal equivalent of decimal 16 is written as $10.

Where to Look for More Information

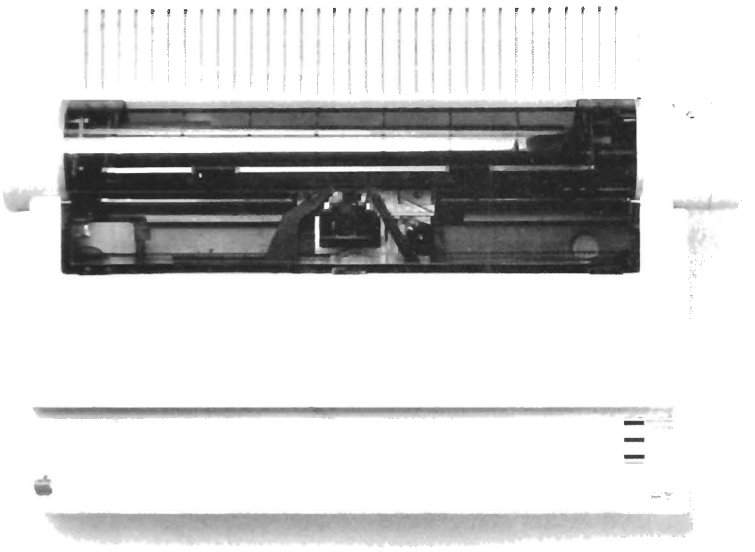
The following manuals have information about the Apple ImageWriter II printer, Apple computers, or programming languages that can help you in getting the most out of your printer.

- *ImageWriter II Owner's Manual*
- *ImageWriter II AppleTalk Option User's Manual*
- *ImageWriter II SheetFeeder Installation Manual*
- *Apple IIe Technical Reference Manual*
- *Apple IIc Technical Reference Manual*
- *Inside Macintosh*
- *ProDOS Technical Reference Manual*
- *BASIC Programming With ProDOS*
- *ProDOS Assembler Tools*

- *Applesoft BASIC Programmer's Reference Manual*
- *Macintosh Pascal Reference Manual*
- *Apple II Pascal Operating System Reference Manual*
- *Apple II Pascal Language Reference Manual*
- *Apple II Pascal 1.3 Manual*
- *Apple Logo Reference Manual*
- *Apple Logo II Reference Manual*

Chapter 1

Introduction



This chapter summarizes the features of your Apple® ImageWriter™ II, provides an introduction to the use of the ImageWriter commands described in Chapters 3 through 8, explains how to use the examples found throughout this book, and gives some programming tips for a variety of computer languages.

ImageWriter II Features

The dot matrix printing method gives you maximum flexibility to create written records in exactly the form you want. Besides printing color and black-and-white graphics, the ImageWriter II lets you print ordinary text with a wide range of typographical options (such as underlining, boldfacing, and several character spacings). It also provides a full set of horizontal tabbing controls, and lets you create sets of custom characters and integrate them with ordinary text.

Here's a summary of what you can tell your Apple ImageWriter II to do, using the DIP switches and programming procedures described in this manual:

- Print all the letters, numbers, and punctuation marks that you can type on your Apple computer keyboard or display on the video screen.
- Print special characters for several languages.
- Set the number of characters per inch (**character pitch**) to 9, 10, 12, 13.4, 15, or 17. Two additional options allow the character pitch to be determined by the width of each character (**proportional printing**). More than one pitch can be used in a single line.
- Set the spacing between lines (**line feed pitch**) in increments of 1/144 inch, including the standard six or eight lines per inch. Line spacing commands can be included within any line, and affect all following lines.
- Feed paper both up and down.
- Print subscripts and superscripts.
- Print in three text modes: near letter quality (**NLQ**), correspondence, and draft.
- Print 32 MouseText characters.
- Set page length to any length up to 69 inches in 1/144-inch increments; set the page to conform to standard page length; or allow the printout to be continuous, without page breaks.
- Print zeros either unslashed (0) or slashed (Ø).
- Print in boldface.
- Underline text.

- Print double-width characters.
- Change the location of the left-hand print margin. The location of the right-hand margin is usually determined by the program you use in your Apple computer.
- Program up to 32 horizontal tabs.
- Print symbols or characters you design yourself. You can add up to 175 8-dot-by-8-dot characters, or 95 characters that are 8 dots high by 16 dots wide. You can mix these special symbols freely with the standard alphabets and print them with all the formatting features listed above.
- Print drawings, graphs, diagrams, and similar line graphics, by instructing your Apple ImageWriter II to print individual dots in specific locations. The highest available resolution of 160 dots to the inch horizontally and 144 to the inch vertically produces sharp, unbroken lines; many other resolutions are available. You can print graphics patterns up to eight inches wide and any height.
- Print text or graphics in any combination of six colors, plus black. Many additional colors can be created by overprinting colors or printing patterns with dots of alternating colors. (You must install a color ribbon before the color print commands are activated.)

When the ImageWriter II is turned on it is set up to function with certain characteristics (the **default** settings). Some of these defaults can be altered by changing the setting of DIP switches 1-1 through 2-4, and some of them can be reset after the ImageWriter II is on by sending special commands to the printer. A complete description of the DIP switches and their use is given in Chapter 2. The commands that control the various characteristics of the ImageWriter II from your computer are described in Chapters 3 through 8. Defaults are listed in Appendix A.

Using ImageWriter II Commands

The ImageWriter II communicates with your computer through an asynchronous serial interface as described in Appendix E. The serial interface conforms to EIA Standards RS422 and RS423, and is electrically compatible with RS232 devices.

The ImageWriter II can receive 8-bit binary codes ranging in value from 0 to 255. Of these, 0–31 and 127 are non-printable characters; 32–126 represent characters printable by the ImageWriter II. Codes 128–255 are called **high ASCII** and can be used to print characters not defined in the

ASCII standard, such as the ImageWriter II's MouseText characters or characters you have designed. In the graphics modes, all 256 code combinations are used to specify graphics patterns.

What is ASCII? In order for a computer to communicate with a printer, both pieces of equipment must understand a common language. Computers communicate by means of numbers. A specific number sent to the printer can represent a command, a graphics pattern, or a text character.

The numbers representing a set of printable characters and some common commands have been standardized within the computer industry. This standard is called the American Standard Code for Information Interchange, or ASCII for short. In other words, ASCII is simply a standard set of characters represented by specified numbers.

Macintosh

Several special characters are available on the Macintosh™ computer by pressing the Option key and a character key simultaneously. These characters do not conform to the ASCII standard; do *not* use any of these special characters to send commands to the ImageWriter II.

Numbering Systems

This manual uses three of the numbering systems commonly used in the computer industry to represent the ASCII characters: binary, hexadecimal, and decimal.

The decimal (base 10) and hexadecimal (base 16) systems are used in most programming languages to represent the codes sent to the printer. In this manual, hexadecimal numbers are indicated by a dollar sign ($); for example, the hexadecimal equivalent of decimal 16 is written as $10.

The actual electrical signal sent to the printer through the serial interface consists of a series of high and low voltages representing a sequence of binary numbers (ones and zeros). The CHR$ command in BASIC and the CHR command in Pascal both use the decimal number system; assembly language programs use hexadecimal numbers.

The standard ASCII codes in the binary, decimal, and hexadecimal number systems are shown in Appendix B. The complete ImageWriter II character set is shown in Appendix C.

Control Codes

The non-printable ASCII characters with decimal values from 0 to 31 (see Appendix B) are referred to as **control codes**; several of the control codes are used to send commands to the ImageWriter II to control printer functions.

CTRL is used as an abbreviation for CONTROL.

Generating Control Codes: Many computers, including the Apple II family, have a [CONTROL] key; you can generate a control code on these computers by pressing [CONTROL] and a character key simultaneously. Pressing [CONTROL] has the effect of subtracting 64 (decimal) from ASCII keys with decimal values 64 through 95. For example, pressing [H] generates ASCII code 72, while pressing [CONTROL]-[H] generates ASCII code 8 (72–64).

As shown in Appendix B, the ASCII standard assigns a name to each control code; for example, CONTROL-H, ASCII code 8, is also called "BS," and is used by many printers and video display screens for the backspace command. CONTROL-[, ASCII code 27, is called "ESC"; on many computers (including the Apple II family), this code can be generated by pressing the Escape key ([ESC]).

ESC is an abbreviation for ESCAPE.

By the Way: Some computers do not have [CONTROL] or [ESC] keys; in this case you must use an ASCII character function (such as the CHR$ statement in BASIC or the CHR statement in Pascal) to send these codes to the printer.

Escape Sequences

The ImageWriter II accepts more commands than can be specified by the ASCII control codes alone. Most printer commands consist of an ESC control code (ASCII code 27) followed by one or more printable characters. These commands are called **escape sequences**.

When the printer receives the ESCAPE character, it interprets the character that follows it as part of a command, rather than as a printable character. For example, an ESC character followed by an exclamation point, ESC ! (27 33 or $1B $21), tells the ImageWriter II to begin boldface printing. The exclamation point is not printed because the ESC preceding it causes it to be interpreted as a printer command.

If the character following the ESC is part of a multi-character command, the printer recognizes this fact and continues to interpret ASCII codes as parts of the command until the command is complete. For example, to set the left margin to column 10, send the following sequence to the printer:

```
ESC L 0 1 0
```

The characters L, 0, 1, and 0 are not printed but are interpreted as parts of the command sequence.

> ***By the Way:*** If the first character after an ESC character is not a legitimate command identifier, both it and the ESC are ignored by the printer.

Using Packaged Programs

Many packaged programs have been designed to take full advantage of the ImageWriter II printer's advanced features. However, there are also many programs that have been designed to work with almost any printer; either the program provides a way for you to embed printer commands directly in your text or data file, or you must configure the program for the ImageWriter II.

If you are using a program that must be configured, then either you will be prompted with questions about your printer and its interface as you go through the configuration or setup section of the program, or you will have to modify a data file to reflect the attributes of the ImageWriter II. The documentation that came with the program should have a section that guides you through this process; refer to it for detailed instructions. You might be asked to identify the make and model of your printer interface card, the make and model of your printer, the number of characters it can print per line, and commands for certain printer functions.

When you are asked to specify printer commands, some programs allow you to enter the characters that make up the command by pressing the keys labeled with those characters on your computer keyboard. Other programs need the commands to be identified by their ASCII codes. If your program accepts ASCII codes, make sure you know which number system (such as decimal or hexadecimal) the program expects the codes to be in.

For example, assume you are prompted for the hexadecimal codes that turn on boldface printing. The first step is to look up the command in Chapter 4, where you find that it is an ESC character followed by the exclamation point (ESC !, or $1B $21) Enter 1B 21 in response to the prompt. (You may need to type some character—such as a dollar sign ($)—before each number to indicate that it is hexadecimal.) Refer to the documentation that came with the program to learn how it requires these command sequences to be entered.

Writing Programs

As you read through the other chapters in this book, you'll find many example programs that show you how to use the various ImageWriter II commands in both Applesoft BASIC and Macintosh Pascal. This section provides general instructions for writing programs that include printer commands, and gives some suggestions for using ImageWriter II commands with Logo, other BASICs, other Pascals, and assembly language.

Important!

Short examples in this manual are often presented as sample **character strings**. These strings are printed with a space between each two characters to improve legibility; do *not* include these spaces when you use the commands. For example, the command to set the left margin to column 10 is shown as:

```
ESC L 0 1 0
```

Send this character string to the printer as:

```
ESC L010
```

There are three steps to using a printer from within a program:

1. Open a communication channel to the printer.
2. Send the characters to the printer.
3. Close the communication channel.

Table 1-1 shows a flow chart of these steps and how they are implemented in Applesoft BASIC and Macintosh Pascal.

Table 1-1. Sending Characters to the Printer

Flow Chart	Applesoft BASIC	Macintosh Pascal
.	.	.
.	.	.
.	.	.
Open Printer Channel	`30 PRINT CHR$(4);"PR#1"`	`REWRITE(Prntr,'PRINTER:');`
Send Characters	`40 PRINT "Test"`	`WRITELN(Prntr,'Test');`
Close Printer Channel	`50 PRINT CHR$(4);"PR#0"`	`CLOSE(Prntr);`
.	.	.
.	.	.
.	.	.

Important! The example Applesoft BASIC programs in this manual are written to work with the disk operating system. If you have started your computer without a disk in the disk drive, these programs may not function correctly. Refer to the following manuals for more information on the PR# and PRINT commands: *Applesoft BASIC Programmer's Reference Manual*, and *BASIC Programming With ProDOS*.

The sample lines of Pascal code in Table 1-1 are extracted from a larger program and will not run without additional code. A more complete example is given in Figure 1-1. Refer to the following manuals for more information on Pascal and the REWRITE, WRITELN, and CLOSE procedures: *Macintosh Pascal Reference Manual*, *Apple II Pascal Operating System Reference Manual*, *Apple II Pascal Language Reference Manual*, *Apple II Pascal 1.3 Manual*.

The process of opening the communication channel to the printer is accomplished in different ways from one programming language to the next. In Applesoft BASIC, use the PR#n command (where n is the number of the slot your printer interface card is in). For example, if your computer is set up with the interface card in slot 1, then use PR#1. In Macintosh Pascal the same thing is done by the REWRITE command.

Once the communication channel to the printer is open, your program can send characters to the printer by using an output command. In Applesoft BASIC the output command is PRINT, while in Macintosh Pascal it's WRITE or WRITELN.

When you are finished using the printer, you must close the communication channel. In Applesoft BASIC, use the PR#n command again to close the channel, where this time n is the number (usually 0) that addresses your display device (video monitor or TV set). With Macintosh Pascal, use the CLOSE command to close the channel.

By the Way: Some programming languages have special commands for sending data to a printer that take care of opening and closing the communication channel for you. For example, in some versions of BASIC, you can use an LPRINT command in place of a PRINT command to send information directly to the printer.

To use a special feature of the ImageWriter II, send the command for that feature to the printer. Many of the characters that make up the printer commands are regular keyboard characters and can be typed right into the output commands. For example, to output the character string "L 0 1 0" using Applesoft BASIC, use the following statement:

```
10 PRINT "L010"
```

To do the same thing in Macintosh Pascal, use this statement:

```
WRITE(Prntr,'L010');
```

However, some of the characters used in printer commands cannot be entered directly from the keyboard. In order to send these characters to the printer, use an ASCII character function (found in most programming languages). In Applesoft BASIC the ASCII character function is CHR$(*d*) and in Macintosh Pascal it's CHR(*d*), where *d* is the ASCII code of the desired character in the decimal (base ten) numbering system. For example, to output the ESC character with Applesoft BASIC, use the following statement:

```
10 PRINT CHR$(27);
```

The same thing can be done in Macintosh Pascal by using this statement:

```
WRITE(Prntr,chr(27));
```

The following sections provide programming hints for Pascal and Applesoft BASIC, and provide sample programs in Microsoft® BASIC 2.0, Logo, and 6502 assembly language. If you are using a language other than Macintosh Pascal or Applesoft BASIC, read over the relevant section and compare the sample program in that section with the Applesoft BASIC and Macintosh Pascal samples shown in Figure 1-1. Then try it on your computer.

Whatever language you are using, the important things to study are how the language performs the three steps to using a printer (see Table 1-1), and how to output non-printable characters. Each of the sample programs does the same things: underlining, boldfacing, and double-width printing. (See Figure 1-2 for a sample printout.)

The Applesoft BASIC sample program in Figure 1-1 is written to work with either DOS or ProDOS®. Refer to the following manuals for more information on CHR$ (4) or Control-D: *DOS Programmer's Manual*, and *BASIC Programming With ProDOS*.

Figure 1-1. Applesoft BASIC and Macintosh Pascal Sample Programs

Applesoft BASIC

```
100 HOME
104 D$ = CHR$ (4)
110 PRINT "Test to Screen"
120 PRINT D$; "PR#1"
130 PRINT "Test to Printer"

140 PRINT CHR$(27);"X"
150 PRINT "Underlined"

160 PRINT CHR$(27);"Y"
170 PRINT "Underlined Ended"

180 PRINT CHR$(27);"!"
190 PRINT "This is in Boldface"

200 PRINT CHR$(27);CHR$ (34)
210 PRINT "Boldface Ended"

220 PRINT CHR$ (14)
230 PRINT "This is the Doublewide Mode"


240 PRINT CHR$ (15)
250 PRINT "This Ends Doublewide Mode"

260 PRINT D$; "PR# 0"
270 END
```

Macintosh Pascal

```
PROGRAM Printer_Features;

VAR Prntr:TEXT;

BEGIN

 WRITELN('Test to Screen');
 REWRITE(Prntr,'PRINTER:');
 WRITELN(Prntr,'Test to Printer');

 WRITELN(Prntr,chr(27),'X');
 WRITELN(Prntr,'Underlined');

 WRITELN(Prntr,chr(27),'Y');
 WRITELN(Prntr,'Underlined Ended');

 WRITELN(Prntr,chr(27),'!');
 WRITELN(Prntr,'This is in Boldface');

 WRITELN(Prntr,chr(27),chr(34));
 WRITELN(Prntr,'Boldface Ended');

 WRITELN(Prntr,chr(14));
 WRITELN(Prntr,'This is the Doublewide
 Mode');

 WRITELN(Prntr,chr(15));
 WRITELN(Prntr,'This Ends Doublewide Mode');

 CLOSE(Prntr)
END.
```

Figure 1-2. Sample Programs' Printer Output

```
Test to Printer

Underlined

Underlined Ended

This is in Boldface

Boldface Ended

This is the Doublewide Mode

This Ends Doublewide Mode
```

Applesoft BASIC

Although the CHR$ command in Applesoft BASIC accepts any number up to 255, the **most significant** (eighth) **bit** of each data **byte** is always sent to the printer as a 1; this feature can interfere with the use of the software switch commands (Chapter 3), with high-ASCII characters (such as MouseText characters and some custom characters), and with graphics data. You may be able to set the serial port on your computer to clear the most significant bit to zero; a program that does so is shown in Figure 3-2. You may also be able to load a short machine language routine into memory that can be used by your Applesoft BASIC program to send 8-bit data to the printer. To try this on an Apple II, Apple II Plus, Apple IIc, or Apple IIe, include the following lines at the beginning of your Applesoft BASIC program:

```
10 DATA 32,177,0,32,123,221,32,251,230,138,76,237,253
20 A = 768
30 FOR I = 1 TO 13
40 READ B
50 POKE A,B
60 A = A + 1
70 NEXT I
```

To use this routine to send data to the printer, use a statement of the form:

```
100 CALL 768 , d
```

Software switches are discussed in Chapter 3. Decimal values of ASCII characters are shown in Appendix B.

where *d* is any decimal number from 0 to 255, representing the ASCII character you want to send. Figure 1-3 illustrates enabling the "line feed when line is full" function with Applesoft BASIC using this machine language routine. In this example, the software switch data bytes are sent using CALL 768 statements so that the most significant bit of each data byte is not set to 1.

Figure 1-3. Applesoft BASIC Example: Sending 8-Bit Data

```
100 D$ = CHR$ (4)
110 PRINT D$; "PR#1"
120 REM  THE NEXT SEVEN LINES LOAD A MACHINE LANGUAGE
121 REM       PROGRAM INTO MEMORY
130 DATA 32,177,0,32,123,221,32,251,230,138,76,237,253
140 A = 768
150 FOR I = 1 TO 13
160 READ B
170 POKE A,B
180 A = A + 1
190 NEXT I
200 REM  THE NEXT LINE SENDS THE FIRST PART OF THE LF INSERT COMMAND
210 PRINT CHR$ (27);"D";
220 REM  THE NEXT LINE SENDS THE DATA PART OF THE COMMAND
230 CALL 768 , 32 : CALL 768 , 0
240 PRINT D$; "PR# 0"
250 END
```

By the Way: The DATA statement in the program in Figure 1-3 is equivalent to the following assembly language routine:

```
JSR $B1   ;Skip over the character following 768
JSR $DD7B ;Evaluate the expression following the
comma
JSR $E6FB ;Put high byte in A, low byte in X
TXA       ;Transfer low byte to A
JMP $FDED ;Print A and return from subroutine
```

Microsoft BASIC 2.0

This section shows how printer features can be selected by including the appropriate commands in your Microsoft BASIC 2.0 program. The example in Figure 1-4 selects the underline mode and then turns it off; selects

boldface and then turns it off; and selects double-width mode and then turns it off. The use of these three features is typical of how the other features of the ImageWriter can be selected.

Use the LPRINT command in Microsoft BASIC programs to send data to the printer.

Figure 1-4. Microsoft BASIC Sample Program

```
PRINT "Test to Screen"
LPRINT #1: "Test to Printer"

LPRINT #1:  CHR$(27); "X"              :REM CHR$(27) is ESCAPE
LPRINT #1: "Underlined"

LPRINT #1: CHR$(27); "Y"
LPRINT #1: "Underlined Ended"

LPRINT #1: CHR$(27); "!"
LPRINT #1: "This is in Boldface"

LPRINT #1: CHR$(27); CHR$ (34)         :REM CHR$(34) is "
LPRINT #1: "Boldface Ended"

LPRINT #1: CHR$ (14)                   :REM CHR$(14) is CONTROL-N
LPRINT #1: "This is the Doublewide Mode"

LPRINT #1: CHR$ (15)                   :REM CHR$(15) IS CONTROL-O
LPRINT #1: "This Ends Doublewide Mode"

END
```

When you choose Run from the Program menu to run this program, it displays `Test to Screen` on your monitor, then prints out seven lines to the printer, as shown in Figure 1-2.

Pascal

The Macintosh Pascal examples work without modification with Apple II Pascal, Instant Pascal®, and most other versions of Pascal. The examples assume that you have the Pascal Operating System running on your computer. If you don't already know how to use your Pascal Operating System, please read the manuals that came with it before trying these examples.

Refer to the following manuals for more information on Apple II Pascal: *Apple II Pascal Language Reference Manual* and *Apple II Pascal Operating System Reference Manual.* Refer to the following manual for more information on Apple II Pascal 1.3: *Apple II Pascal 1.3 Manual.*

Logo

Refer to the following manuals for more information on Logo primitives and procedures: For Apple Logo II see *Apple Logo II Reference Manual.* For 64K Apple Logo see *Apple Logo Reference Manual.*

This section provides some examples of using the ImageWriter II from Logo. It contains examples of using the underline, boldface, and double-width printing features. The use of these three features is typical of how the other features of the ImageWriter II can be selected.

The four procedures in Figure 1-5 demonstrate how to use the printer with Apple Logo II. The first procedure calls the three other procedures, which send the appropriate codes to the printer, and print a line on the printer. For 64K Apple Logo, see Figure 1-6.

Figure 1-5. Apple Logo II Sample Program

```
OPEN 1
SETWRITE 1
PR [Test to Printer]
PrintUnderlined [Underlined]
PR [Underlined Ended]
PrintBoldface [This is in Boldface]
PR [Boldface Ended]
PrintDoublewide [This is the Doublewide Mode]
PR [This Ends Headline Mode]
SETWRITE []
CLOSE 1
END

TO PrintUnderlined :Line
TYPE CHAR 27
TYPE "X
PR :Line
TYPE CHAR 27
Type "Y
END
```

```
TO PrintBoldface :Line
TYPE CHAR 27
TYPE "!
PR :Line
TYPE CHAR 27
TYPE CHAR 34
END

TO PrintDoublewide :Line
TYPE CHAR 14
PR :Line
TYPE CHAR 15
END
```

When you type **SHOWFEATURES** and press RETURN to execute this program, it displays **Test to Screen** on your monitor, then prints out four lines of text, as shown in Figure 1-2.

64K Apple Logo

You must make two changes in the Apple Logo II ShowFeatures procedure to use it with 64K Apple Logo. Change the OPEN 1 and SETWRITE 1 commands to .PRINTER 1 and change SETWRITE [] CLOSE 1 to .PRINTER 0, as shown in Figure 1-6. The other procedures will work as shown in Figure 1-5.

Figure 1-6. 64K Apple Logo Sample Procedure

```
.PRINTER 1
PR [Test to Printer]
PrintUnderlined [Underlined]
PR [Underlined Ended]
PrintBoldface [This is in Boldface]
PR [Boldface Ended]
PrintDoublewide [This is the Doublewide Mode]
PR [This Ends Doublewide Mode]
.PRINTER 0
END
```

Assembly Language

This section provides you with some examples of using your printer from 6502 Assembly Language. The sample program in Figure 1-7 contains examples of using the underline, boldface, and double-width printing features. The use of these three features is typical of how the other features of the ImageWriter can be selected.

Figure 1-7. ProDOS Assembler Sample Program

```
0300:           0300     1             ORG    $300          ;STARTING ADDRESS
0300:           000D     2 CRETURN     EQU    $0D           ;RETURN CHARACTER
0300:           000A     3 CLINEFEED   EQU    $0A           ;LINE FEED CHARACTER
0300:           001B     4 CESCAPE     EQU    $1B           ;ESCAPE CHARACTER
0300:           0022     5 CQUOTE      EQU    34            ;QUOTATION MARK CHARACTER
0300:           000D     6 PINIT       EQU    $0D           ;ENTRY POINT LOOKUP TABLE
0300:           000E     7 PREAD       EQU    $0E           ;      "
0300:           000F     8 PWRITE      EQU    $0F           ;      "
0300:           0010     9 PSTATUS     EQU    $10           ;      "
0300:                   10 ;
0300:4C 10 03           11 START      JMP    BEGIN
0303:                   12 ;
0303:                   13 ;Y=INDEX FOR ENTRY PT LOOKUP TABLE, SLOT #1 IS ASSUMED
0303:           0303    14 GOCARD     EQU    *             ;GO TO A PASCAL ENTRY PT
0303:BE 00 C1           15            LDX    $C100,Y       ;GET LOW BYTE OF ENTRY PT
0306:8E 0E 03           16            STX    VECTOR+1      ;SELF MODIFYING CODE
0309:A2 C1              17            LDX    #$C1          ;REQUIRED X = Cn
030B:A0 10              18            LDY    #$10          ;          Y = n0
030D:4C 00 C1           19 VECTOR     JMP    $C100         ;THIS ADDRESS GETS MODIFIED
0310:                   20 ;
0310:A0 0D              21 BEGIN      LDY    #PINIT        ;INITIALIZE CARD
0312:20 03 03           22            JSR    GOCARD        ;     "
0315:A2 00              23            LDX    #0            ;CHARACTER POINTER
0317:BD 2D 03           24 LOOP1      LDA    TEXT,X        ;GET CHARACTER
031A:F0 0F      032B    25            BEQ    DONE          ;IF 0 > NO MORE CHARACTERS
031C:8E 2C 03           26            STX    XSAVE         ;SAVE THE POINTER
031F:A0 0F              27            LDY    #PWRITE       ;OUTPUT CHAR
0321:20 03 03           28            JSR    GOCARD        ; TO CARD (PRINTER)
0324:AE 2C 03           29            LDX    XSAVE         ;RESTORE THE POINTER TO X
0327:E8                 30            INX                  ;POINT TO NEXT CHARACTER
0328:4C 17 03           31            JMP    LOOP1         ;GO DO NEXT CHARACTER
032B:60                 32 DONE       RTS                  ;RETURN
032C:                   33 ;
032C:00                 34 XSAVE      DFB    0             ;STORES X
032D:           032D    35 TEXT       EQU    *             ;HERE IS THE TEXT
032D:54 65 73 74        36            ASC    "Test to Printer"
033C:0D 0A              37            DFB    CRETURN,CLINEFEED
033E:1B                 38            DFB    CESCAPE
```

Figure 1-7. ProDOS Assembler Sample Program (Continued)

```
033F:58                 39              ASC     "X"
0340:0D 0A              40              DFB     CRETURN,CLINEFEED
0342:55 6E 64 65        41              ASC     "Underlined"
034C:0D 0A              42              DFB     CRETURN,CLINEFEED
034E:1B                 43              DFB     CESCAPE
034F:59                 44              ASC     "Y"
0350:0D 0A              45              DFB     CRETURN,CLINEFEED
0352:55 6E 64 65        46              ASC     "Underlined Ended"
0362:0D 0A              47              DFB     CRETURN,CLINEFEED
0364:1B                 48              DFB     CESCAPE
0365:21                 49              ASC     "!"
0366:0D 0A              50              DFB     CRETURN,CLINEFEED
0368:54 68 69 73        51              ASC     "This is in Boldface"
037B:0D 0A              52              DFB     CRETURN,CLINEFEED
037D:1B 22              53              DFB     CESCAPE,CQUOTE
037F:0D 0A              54              DFB     CRETURN,CLINEFEED
0381:42 6F 6C 64        55              ASC     "Boldface Ended"
038F:0D 0A              56              DFB     CRETURN,CLINEFEED
0391:0E                 57              DFB     14
0392:0D 0A              58              DFB     CRETURN,CLINEFEED
0394:54 68 69 73        59              ASC     "This is the Doublewide Mode"
03AD:0D 0A              60              DFB     CRETURN,CLINEFEED
03AF:0F                 61              DFB     15
03B0:0D 0A              62              DFB     CRETURN,CLINEFEED
03B2:54 68 69 73        63              ASC     "This Ends Doublewide Mode"
03C9:0D 0A              64              DFB     CRETURN,CLINEFEED
03CB:00                 65              DFB     0
```

Debugging Programs: Hexadecimal Data Dump

If your printer does not respond correctly to your program (or to an example in this manual), it may be because your computer is modifying the commands before they are sent to the printer. Applesoft BASIC, for example, automatically sets the most significant (eighth) bit of each data byte to 1; this feature can interfere with the use of the software switch commands (Chapter 3), with high-ASCII characters (such as MouseText characters and some custom characters), and with graphics data.

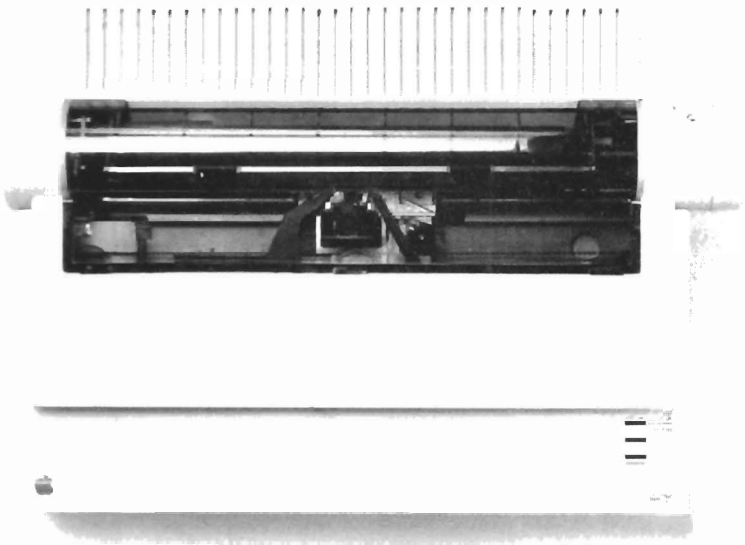
To determine the actual data being received by your printer, turn the printer off, press the SELECT button, and hold that button down while you turn the printer back on. Then run the program. The data received by the printer will be printed out as hexadecimal codes, each pair representing a data byte. For example, the software reset sequence ESC c prints out as 1B 63.

Check the hexadecimal codes against your program. Some codes (such as PR#1) are intended for the computer or the serial interface, and should not be passed on to the printer. If any of the codes intended for the printer are not received, or any of the codes received by the printer are incorrect, first check your program to make sure that you typed it in correctly. Then refer to your computer and serial card manuals to determine why the codes are being modified, and what you can do about it.

To reset your printer to run normally, turn it off and back on.

Sample Applesoft BASIC programs that can send a 0 for the eighth bit are shown in Figures 1-3 and 3-2.

Improper Line Feeds: When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the automatic line feed after carriage return function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.



This chapter describes how to take advantage of some of the capabilities of the Apple ImageWriter II by adjusting a set of switches located just under the front cover. With the information in this chapter you can change the way the printer is configured when it is turned on, before it receives any commands.

The first section explains what DIP switches are and why they are used. The second section describes how to set the DIP switches and summarizes their uses. The third section describes the function of each of the switches.

Power-On Configuration

The ImageWriter II printer is controlled by a **microprocessor** that receives instructions from read-only memory (ROM), from a set of miniature switches (called DIP switches) located inside the printer case, and from commands sent to the printer by your computer. The DIP switches determine several of the default configuration settings of the printer when power is turned on, including

DIP is an abbreviation for Dual In-Line Package.

- The national (alternate) character set to be used
- Whether the printer assumes an 11-inch or 12-inch paper length
- Whether the printer skips over the perforation between sheets of paper
- The character pitch to be used
- Whether a line feed (LF) is automatically inserted after every carriage return (CR)
- The communications rate (**baud**) to be used
- The communications protocol (hardware handshake or XON/XOFF) to be used
- Whether an installed option card has been enabled.

A complete list of power-on defaults is given in Appendix A.

There are 14 switches in the two DIP switch assemblies inside the ImageWriter II; you can set these switches to configure the printer's power-on defaults to suit your needs. Each switch setting is described in this chapter.

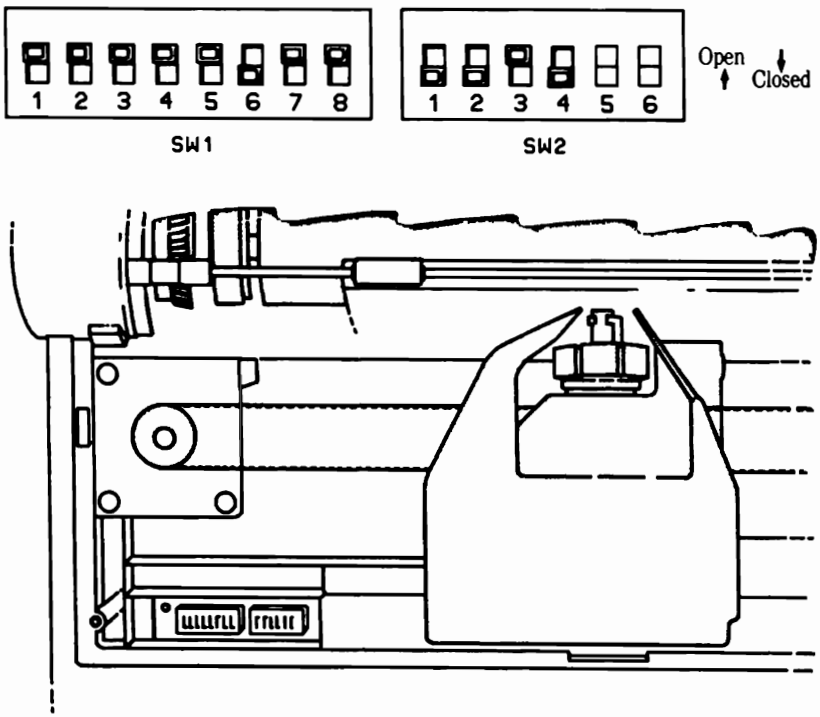
Setting DIP Switches

Many of the DIP switch settings can be overridden with software commands; see the descriptions of the DIP switch settings in this chapter for details.

Remove your printer's front cover as described in Chapter 2 of the *ImageWriter II Owner's Manual*. If the print head is at the far left, move it gently to the center. The two DIP switch assemblies are located in the bottom of the machine at the left front.

DIP switch assembly 1 (SW 1) is closest to the left side of the machine and contains eight switches, numbered 1 through 8. Switch assembly 2 (SW 2) is to the right of SW 1 and contains six switches, numbered 1 through 6; switches 5 and 6 of SW 2 are set by the factory and you should not change them. Individual switches are identified in this manual by the switch assembly name followed by the individual switch number—for example, switch 2-4 or SW 2-4 is switch number 4 on switch assembly 2. The DIP switch assemblies are illustrated in Figure 2-1.

Figure 2-1. Location of DIP Switches



▲Warning | Be sure to turn the power off before adjusting the switches.

To change a DIP switch setting, turn off the power (do *not* unplug the printer), then press the small switch handle to the opposite position with a pointed tool. A switch is open (off) when its handle is set nearest to the platen (towards the back of the printer). It is closed (on) when its handle is toward the front of the machine. Don't forget to turn the power back on when you are finished setting switches.

Important! | When you change a DIP switch setting, do not use a pencil or other object that may leave foreign matter that can get into the switch. A small, clean tool such as a small screwdriver or straightened paper clip works best.

The self-test is described in Chapter 2 of the *ImageWriter II Owner's Manual*. Options are described in Chapter 9 of this manual, and in Chapter 5 of the owner's manual.

You can run the ImageWriter II self-test to obtain a printout of the current switch positions. As set by the factory (for U.S. distribution), a self-test printout of DIPSW1(00000100) DIPSW2(1100) is standard with no option cards installed. A self-test printout of DIPSW1(00000100) DIPSW2(1101) is standard with an option card (32K Memory or AppleTalk™) installed and enabled.

| ***Factory Settings:*** The factory settings for DIP switches shown in this manual apply only to printers configured for sale in the USA. Check the self-test printout of your printer to determine the current setting of the DIP switches.

Table 2-1 summarizes the DIP switch settings. The boldfaced switch position indicates the factory setting for DIP switches 1-1 through 2-4. "Open" corresponds to "off" and a logic "0". "Closed" corresponds to "on" and a logic "1".

Each of these settings is discussed in detail in the following sections.

Table 2-1. DIP Switch Functions

SW1-1	SW1-2	SW1-3	Language
OPEN	**OPEN**	**OPEN**	**American**
Closed	Open	Open	Italian
Open	Closed	Open	Danish
Closed	Closed	Open	British
Open	Open	Closed	German
Closed	Open	Closed	Swedish
Open	Closed	Closed	French
Closed	Closed	Closed	Spanish

Table 2-1. DIP Switch Functions (Continued)

SW1-4		Form Length
OPEN		**11 Inches**
Closed		12 Inches
SW1-5		Perforation Skip
OPEN		**Perforation skip inactive**
Closed		Perforation skip active
SW1-6	SW1-7	Character Pitch
Open	Open	10 Characters per inch (Pica)
CLOSED	**OPEN**	**12 Characters per inch (Elite)**
Open	Closed	17 Characters per inch (Ultracondensed)
Closed	Closed	160 Dots per inch (Elite proportional)
SW1-8		LF on CR
OPEN		**No line feed is added after a carriage return**
Closed		A line feed is added after receiving a carriage return
SW2-1	SW2-2	Communication Rate
Open	Open	300 Baud
Closed	Open	1200 Baud
Open	Closed	2400 Baud
CLOSED	**CLOSED**	**9600 Baud**
SW2-3		Protocol
OPEN		**Hardware Handshake**
Closed		XON/XOFF
SW2-4		Option Card
OPEN		**Option Card Disabled**
Closed		Option Card Enabled
SW2-5	SW2-6	Hammer Timing
*	*	**Factory settings of hammer-fire timing**

* These two switches are factory-set to optimize bidirectional dot placement. Note how they are set and write the settings in the space above. **Do not change these settings.**

Descriptions of DIP Switch Functions

This section describes the purpose and use of each of the DIP switches. All of the settings of DIP switch assembly 1 can be overridden by software commands; the functions controlled by DIP switch assembly 2 can be changed only by turning off the printer and resetting the switch.

DIP Switch Assembly 1

DIP switch assembly 1 allows you to set the power-on defaults for a variety of functions that you will seldom want to change; however, each of these settings can be overridden by software commands. The following sections describe each of these functions and refer you to the appropriate section in this manual for more information on the software commands.

Alternate Language Characters

Your Apple ImageWriter II has eight different language fonts to help you in printing text in German, French, Italian, Swedish, Spanish, Danish, and British English, as well as American English. You can choose any one of these character groups to substitute for the ten American symbols shown in Figure 2-2.

Figure 2-2. Alternate Language Characters

American	#	$	@	[	\	]	^	`	{	|	}	~
British	£	$	@	[	\	]	^	`	{	|	}	~
German	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
French	£	$	à	°	ç	§	^	`	é	ù	è	¨
Swedish	#	$	@	Ä	Ö	Å	^	`	ä	ö	å	~
Italian	£	$	§	°	ç	é	^	ù	à	ò	è	ì
Spanish	£	$	§	¡	Ñ	¿	^	`	°	ñ	ç	~
Danish	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Decimal	35	36	64	91	92	93	94	96	123	124	125	126
Hexadecimal	$23	$24	$40	$5B	$5C	$5D	$5E	$60	$7B	$7C	$7D	$7E

| *By the Way:* To type a Spanish hyphen, press the single quote key.

The top line in Figure 2-2 shows the American characters. The other lines give the character groups that can be substituted for these characters by using the appropriate DIP switch settings. DIP switches 1-1, 1-2, and 1-3 specify the default character group. Table 2-1 shows how to set the three DIP switches to select each language character group.

Refer to Appendix C, Character Specifications, to see exactly how the foreign language characters are printed.

When you set DIP switches 1-1, 1-2, and 1-3 to the character group you normally use, that group is chosen every time you turn on your Apple ImageWriter II. You can choose only one language character group at a time. Once you have chosen a group, the ten special symbols always print the characters from that group until you choose a different group.

If you wish to switch to another language character group while printing text, you can send the printer command codes as described in Chapter 4.

Form Length

The setting of DIP switch 1-4 specifies the length of the paper (forms) you are using. The printer uses this setting to calculate the location of the end of the page. The 11-inch setting (SW 1-4 open) is for standard U.S. paper (see Table 2-1). When switch 1-4 is closed, a form length of 12 inches is specified.

If you want to switch to another form length while printing text, you can select any length up to approximately 69 inches in increments of 1/144 inch by using the printer command codes described in Chapter 5.

Perforation Skip

When DIP switch 1-5 is closed, the printer skips over the last half inch (3 lines at 6 lines per inch) of a page and the top half inch of the following page. This feature is used to prevent printing on the perforation of continuous (pin-feed) paper, and can also be used to provide top and bottom margins on pages inserted manually or by the SheetFeeder (see Chapter 9). When the perforation skip function is selected, the printer determines when it is one half inch above the end of the page by keeping track of the number of lines it has printed or scrolled past the top-of-form (TOF) position. The TOF is set at the position of the print line when the printer is turned on, and can be reset with a software command (see Chapter 5).

To assure that the perforation skip function performs correctly, set the form length to match the size of the paper you are using (see the preceding section on Form Length), and set the TOF at one half inch below the top

edge of the page. The easiest way to set the TOF to the correct position is to turn the printer off, roll the paper so that the top of the page is one half inch above the print head, and turn the printer back on.

The factory setting of switch 1-5 is open. When it is set to the open position the printer does not skip over the perforation (see Table 2-1).

Important! Most word processing programs keep track of the page length and perform the perforation skip function under software control. If you are using such a program, don't select the perforation skip function, as it will interfere with the correct operation of the word processing program.

The perforation skip function can also be selected or deselected through use of a printer command code as described in Chapter 5.

Character Pitch

The settings of DIP switches 1-6 and 1-7 specify the character pitch in effect when you turn on the printer. The factory settings are switch 1-6 closed and switch 1-7 open, so that the character pitch selected is 12 characters per inch (the same pitch as an elite typewriter). Refer to Table 2-1 for the other settings.

You can override these DIP switch settings by sending command codes to the printer (see Chapter 4).

Automatic Line Feed After Carriage Return

When DIP switch 1-8 is closed, the printer inserts a line feed after each carriage return character it receives. The factory setting is open: no line feed inserted (see Table 2-1).

Most word processing programs insert line feed characters after each carriage return character. If you close SW 1-8 and use the printer with such a program, it will cause double spacing.

You can override this switch setting by sending a command code to the printer (see Chapter 5).

DIP Switch Assembly 2

DIP switch assembly 2 allows you to set the power-on defaults for a variety of functions that you will rarely want to change. There are no software commands that override these settings; be sure these switches are set the way you want them before you turn the printer on.

Baud Setting

Your Apple ImageWriter II can receive data at 300, 1200, 2400, or 9600 baud. DIP switches 2-1 and 2-2 specify the rate in effect. Both switches are set on (closed) at the factory, for a setting of 9600 baud. See Table 2-1 for the other settings.

Communication Protocol

Hardware handshake and XON/XOFF communication protocols are described in Appendix E, Interface Specifications.

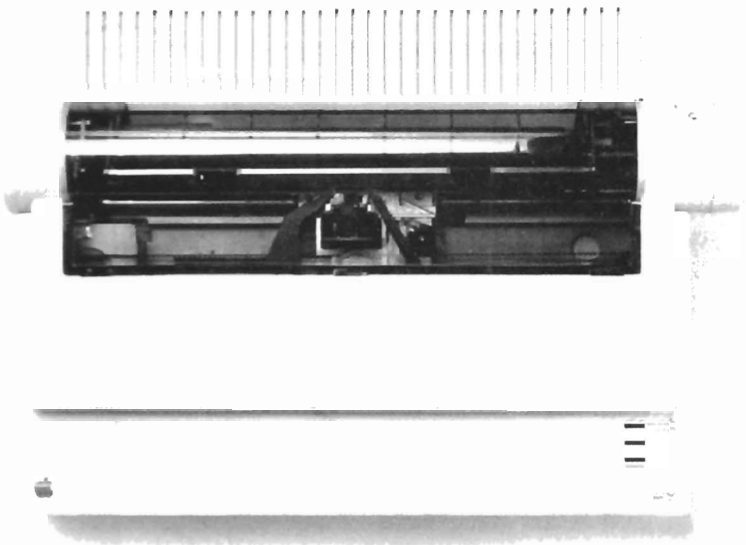
Your printer can receive data using two different types of communication protocol: XON/XOFF, or hardware handshake. DIP switch 2-3 specifies the protocol type when you turn on the printer; switch 2-3 is set open (off) at the factory, selecting hardware handshake as the data protocol (see Table 2-1).

Option Card Enabled

DIP switch 2-4 enables or disables an option card installed in the printer. The factory setting of this switch is open (option disabled), unless your printer comes equipped with the 32K memory or the AppleTalk option, in which case this switch is normally set closed (on). If you have either the memory or the AppleTalk option installed, this switch allows you to switch that option on or off. For example, if you have an AppleTalk option card installed and you wish to use the printer with its standard serial interface rather than the AppleTalk interface, set SW 2-4 to the open (disabled) position. To use (enable) the AppleTalk interface again, set the switch back to the closed position (see Table 2-1). Leave this switch open if you have no option card installed.

Hammer-Fire Timing

DIP switches 2-5 and 2-6 are factory-set to optimize **bidirectional** dot placement. Do not change the settings of these two switches, but write the current settings in the space provided in Table 2-1, in case they become altered accidentally.



Some Apple ImageWriter II commands can be combined in escape sequences that can set several functions at once. Because these escape sequences operate in a manner analogous to DIP switches, they are referred to as "software switches."

This chapter describes the use of the software switches, and briefly describes the function and use of each switch. Each switch can also be used independently as a printer command; these commands are included in Chapters 4 through 6, as shown in the following list.

The functions controlled by software switches include:

- Selecting a national character set (Chapter 4)
- Selecting and deselecting the printer (Chapter 6)
- Automatic line feed when the line is full (Chapter 5)
- Determining whether LF and FF cause printing (Chapter 6)
- Inserting a line feed after a carriage return (Chapter 5)
- Selecting whether zeros are slashed or unslashed (Chapter 4)
- Automatically skipping over perforations (Chapter 5)
- Determining whether the eighth data bit is included or ignored (Chapter 6).

How to Set Software Switches

Software switches, like DIP switches, affect functions of the printer that you seldom need to change. Some of the software switches control functions that were controlled by DIP switches on other Apple printers. Others allow you to modify the setting of some of the ImageWriter II DIP switches. The software-switch functions are summarized in Table 3-2.

There are two **registers** of eight bits each that make up the sixteen individual software switches. Each register is analagous to a DIP switch assembly; each contains eight bits that can be considered as eight switches (see Figure 3-1). The individual switches are identified by the register name (A or B) followed by the individual bit number—for example, software switch B-3 or SW B-3 is bit number 3 of register B.

Use the commands shown in Table 3-1 to set the software switches. Use ESC Z *a b* to set switches open (off) and ESC D *a b* to set switches closed (on), where *a* and *b* represent ASCII characters. Character *a* determines which switches (bits) are affected in register A, and *b* determines which switches are affected in register B.

To see exactly how MouseText characters are printed, refer to Appendix C, Character Specifications. A complete list of low and high ASCII characters is given in Appendix B.

The binary values of *a* and *b* are calculated by placing a one (1) in each bit position that corresponds to a switch you wish to affect; the remaining bit positions are filled with zeros (0). For example, to affect software switches SW A-1, SW A-3, and SW A-7, set the first, third, and seventh bits to 1 and the rest to 0. As illustrated in Figure 3-1, the result for register A is the binary number 01000101, which is equal to decimal 69 or hexadecimal $45, corresponding to the ASCII character E. Register B is 00000000, corresponding to the ASCII character CONTROL-@.

To open software switches SW A-1, SW A-3, and SW A-7 without affecting any other switches, then, send:

```
ESC Z E CTRL-@
```

To close these same switches, send:

```
ESC D E CTRL-@
```

Table 3-2 lists the software switches and their effects.

Figure 3-1. Setting Software Switches A-1, A-3, and A-7

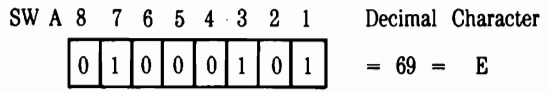


Table 3-1. Software Switch Commands

Characters	Decimal	Hexadecimal	Function
ESC Z *a b*	27 90 *da db*	1B 5A *ha hb*	Sets switches to open (off).
ESC D *a b*	27 68 *da db*	1B 44 *ha hb*	Sets switches to closed (on).

a: The ASCII character corresponding to the bit pattern for register A.
da: The decimal equivalent of *a;* from 0 to 255.
ha: The hexadecimal equivalent of *a;* from $00 to $FF.
b: The ASCII character corresponding to the bit pattern for register B.
db: The decimal equivalent of *b;* from 0 to 255.
hb: The hexadecimal equivalent of *b;* from $00 to $FF.

Table 3-2. Software Switches

SWA-1	SWA-2	SWA-3	Language
OPEN	OPEN	OPEN	American
Closed	Open	Open	Italian
Open	Closed	Open	Danish
Closed	Closed	Open	British
Open	Open	Closed	German
Closed	Open	Closed	Swedish
Open	Closed	Closed	French
Closed	Closed	Closed	Spanish

SW	Function	Z-Setting	D-Setting	Default
A-1	Language char set	SW 1-1 open	SW 1-1 closed	Per SW 1-1
A-2	Language char set	SW 1-2 open	SW 1-2 closed	Per SW 1-2
A-3	Language char set	SW 1-3 open	SW 1-3 closed	Per SW 1-3
A-4	Not used			
A-5	Soft select response	Enabled	Disabled	Disabled
A-6	LF when line full	No line feed	Add line feed	No line feed
A-7	Print commands	CR only	CR, LF, FF	CR, LF, FF
A-8	Auto LF after CR	CR only	CR plus LF	Per SW 1-8
B-1	Slash zero	Unslashed	Slashed	Unslashed
B-2	Not used			
B-3	Perforation skip	Skip	No skip	Per SW 1-5
B-4	Not used			
B-5	Not used			
B-6	Eighth data bit	Included	Ignored	Ignored
B-7	Not used			
B-8	Not used			

Applesoft BASIC

Although the CHR$ (n) command in Applesoft BASIC accepts values as high as 255 for n, the most significant bit is always sent as a 1. As a result, every time you send an ESC Z command to the printer with Applesoft BASIC, the automatic line feed after carriage return command is disabled (SW A-8 is opened), and every time you send an ESC D command to the printer with Applesoft BASIC, that function is enabled (SW A-8 is closed). Changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for more information on this function.

Applesoft BASIC

See the Applesoft BASIC section in Chapter 1 for a routine that you may be able to use to send 8-bit data to the printer. See the following note on the serial port for a way to set the most significant bit to zero. Alternatively, you can write your Applesoft BASIC program to reset SW A-8 as necessary. For example, if SW A-8 is closed and you use an ESC Z command to open SW A-6, you will open SW A-8 at the same time; follow the ESC Z command with the ESC D command in Table 5-15 to re-close SW A-8.

Serial Port

You can manipulate the Apple Super Serial Card or the Apple IIc serial port to gain more control over the eighth data bit. For example, as shown in Figure 3-2, you can set the serial port to use "space" parity, which means that every byte sent has the most significant bit cleared (0).

Figure 3-2 illustrates enabling the "line feed when line is full" function with Applesoft BASIC and the Apple Super Serial Card or the Apple IIc. In this example, the serial port is first set to space parity so that SW A-8 is not affected by the command. See the *Apple IIc Technical Reference Manual* or the *Apple IIe Technical Reference Manual* for more information on serial port commands.

Figure 3-2. Applesoft BASIC Example: Manipulating the Serial Port

```
100 D$ = CHR$ (4)
105 PRINT D$; "PR#1"
109 REM  THE NEXT LINE SENDS THE FIRST PART OF THE LF INSERT COMMAND
110 PRINT CHR$ (27);"D";
119 REM  THE NEXT LINE SETS 7-BIT DATA ON THE SERIAL PORT
120 PRINT CHR$ (9);"1D";
129 REM  THE NEXT LINE SETS SPACE PARITY SO THE MSB=0
130 PRINT CHR$ (9);"7P";
139 REM  THE NEXT LINE SENDS THE DATA PART OF THE COMMAND
140 PRINT CHR$ (32);CHR$ (0)
169 REM  THE NEXT LINE CANCELS PARITY
170 PRINT CHR$ (9);"0P";
175 PRINT D$; "PR# 0"
180 END
```

Each software switch can be set individually, and so can be used as an independent printer command. This section provides a brief description of the purpose and use of each of the software switches shown in Table 3-2. Each of these commands is also described in the appropriate chapter in this manual; for example, the Alternate Language Characters command is also described in Chapter 4, Character Sets and Type Attributes.

Alternate Language Characters

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

Software switches A-1, A-2, and A-3 correspond to DIP switches 1-1, 1-2, and 1-3 respectively (see Table 2-1). These switches allow you to choose any one of eight different language fonts to help you in printing text in German, French, Italian, Swedish, Spanish, Danish, or British English, as well as American English. These character groups are shown in Figure 4-2. To set this switch alone, use the commands in Table 4-3. For more information, see the Alternate Language Characters sections in Chapters 2 and 4.

Response to Software Select-Deselect

Software switch A-5 determines whether the printer responds to the software select and deselect commands. When SW A-5 is set open (off), the printer responds to the select and deselect commands; when the switch is closed (on), the printer does not respond. The default is for the printer not to respond. To set this switch alone, use the commands in Table 6-6. For more information on software select and deselect commands, see the Software Select-Deselect section in Chapter 6.

Automatic Line Feed When Line Is Full

The ImageWriter II executes an automatic carriage return when it receives enough data to fill an entire line. Software switch A-6 determines whether the printer inserts a line feed after the automatic carriage return. When the switch is set closed (on), a line feed is inserted after each automatic carriage return. The default is for no line feed to be inserted. To set this switch alone, use the commands in Table 5-16. This function is discussed in the Automatic Line Feed When Line Is Full section in Chapter 5.

Print Commands

Software switch A-7 determines which commands cause the data in the print buffer to be printed out. When this switch is open (off), only a carriage return causes printing. When the switch is closed (on), carriage returns (CR), line feeds (LF), and form feeds (FF) all cause the data in the print buffer to be printed. The default is for CR, LF, and FF all to cause printing. To set this switch alone, use the commands in Table 6-1.

Important! When you select CR as the only command that causes printing, the LF and FF commands function only when used immediately following a carriage return character. CR insertion (Chapter 5) has no effect when the LF and FF commands are disabled from causing printing.

Automatic Line Feed After Carriage Return

Software switch A-8 determines whether a line feed is automatically inserted after each carriage return. When this switch is open (off), no LF is inserted. The default is determined by DIP switch 1-8. To set this switch alone, use the commands in Table 5-15. For more information on this function, see the Automatic Line Feed After Carriage Return sections in Chapters 2 and 5.

Slashed and Unslashed Zeros

Software switch B-1 determines whether zeros are printed with a slash (Ø) or without (0). When this switch is open (off), zeros are unslashed; this is the default setting. To set this switch alone, use the commands in Table 4-12. For more information on this function, see the Slashed and Unslashed Zeros section in Chapter 4.

Perforation Skip

When software switch B-3 is open (off), the printer automatically skips over the last half inch (3 lines at 6 lines per inch) of a page and the top half inch of the following page. This feature is used to prevent printing on the perforation of continuous (pin-feed) paper, and can also be used to provide top and bottom margins on pages inserted manually or by the SheetFeeder (Chapter 9). The default is determined by the setting of DIP switch 1-5. To set this switch alone, use the commands in Table 5-12. For more information on this function, see the Perforation Skip section in Chapter 2.

Ignore or Include Eighth Data Bit

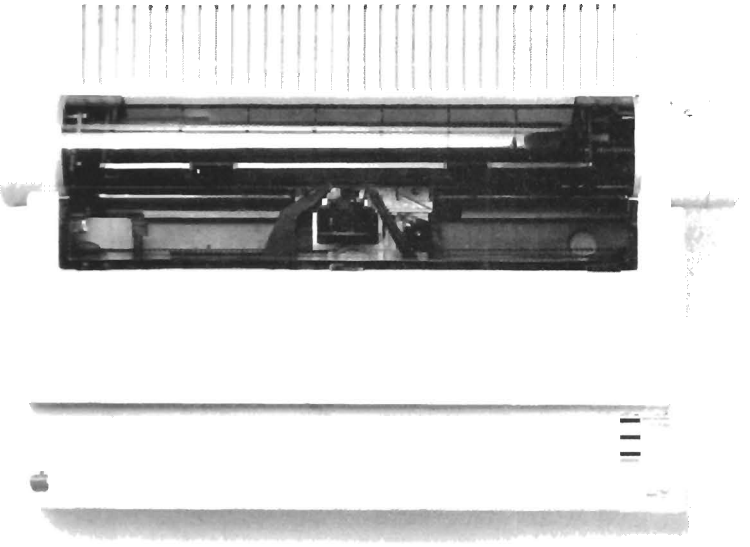
Software switch B-6 determines whether the ImageWriter II recognizes 8-bit data, or only the lower seven bits of each data byte. Data is sent over a serial interface in 8-bit data bytes; however, normal ASCII characters use only the lower seven bits, so usually the value of the most significant bit is ignored.

The ImageWriter II automatically changes to 8-bit mode when the escape-sequence command sent to it uses 8-bit data; for example, the software switch commands and the commands for down-loading a custom character set use 8-bit data. However, you might want to print 8-bit high-ASCII custom characters or MouseText characters on the ImageWriter II without using the ESC * (Chapter 7) or ESC & (Chapter 4) commands. Therefore, a soft switch is provided that sets the printer to recognize the eighth bit.

To set this switch alone, use the commands in Table 6-4.

Applesoft BASIC

Although the CHR$ (n) command in Applesoft BASIC accepts values as high as 255 for n, the most significant bit is always sent as a 1. Figure 3-2 illustrates using serial port commands to set the most significant bit to 0. Figure 1-3 shows a machine-language routine, executable from Applesoft BASIC, that can be used to send 8-bit data to the printer.



This chapter describes how to use ImageWriter II command codes to select the set of characters used and the appearance of the type. These commands control:

1. Character set selection
   - print-quality font
   - MouseText
   - alternate language characters
   - user-designed characters
2. Type attributes
   - character pitch
   - proportional character spacing
   - double-width printing
   - underlining
   - boldfacing
   - half-height printing
   - subscripts
   - superscripts
   - slashed and unslashed zeros

This chapter also provides programming examples in both Applesoft BASIC and Macintosh Pascal for most of the commands. If you need to use these commands with other languages or applications, see the Writing Programs section in Chapter 1.

Character Set Selection

The Apple ImageWriter II provides three different print-quality fonts, plus a variety of special characters, including MouseText, alternate language characters, and custom (user-defined) characters. This section describes the commands that select which font is used for printing, and that allow you to print the special character sets. See Chapter 7 for a complete explanation of how to design, load, and print custom characters.

Print-Quality Fonts

The ImageWriter II can print ASCII text using any of three different fonts: near letter quality (NLQ), correspondence, and draft. When you turn on the printer it is set to use the draft font. You can change to a different font either by pressing the Print Quality button on the control panel or by sending the command characters shown in Table 4-1 (the software commands override the setting of the control-panel button). The NLQ font prints at 45 characters per second (cps), the correspondence font prints at 180 cps, and the draft font prints at 250 cps.

Table 4-1. Font Selection Commands

Characters	Decimal	Hexadecimal	Function
ESC a 0	27 97 48	1B 61 30	Select correspondence font
ESC a 1	27 97 49	1B 61 31	Select draft font
ESC a 2	27 97 50	1B 61 32	Select NLQ font
ESC m	27 109	1B 6D	Same as ESC a 0*
ESC M	27 77	1B 4D	Same as ESC a 2*

* The ESC m and ESC M commands are included in the ImageWriter II to maintain compatibility with the Apple Scribe™ printer.

Default: The setting of the Print Quality switch on control panel.

The boldface, double-width, half-height, subscript, superscript, and proportional printing options are not available for the draft font. If you select any of these options while printing in the draft font, the printer changes to the correspondence font; when you deselect the option, the printer returns to draft font.

The example in Figures 4-1 and 4-2 selects the NLQ font, switches to correspondence font, and then back to NLQ font.

Figure 4-1. Applesoft BASIC Example of Font Selection

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
140 PRINT CHR$(27);"a2";
150 PRINT "Near Letter Quality is selected.";
160 PRINT CHR$(27);"a0";
170 PRINT "Here's some correspondence.";
180 PRINT CHR$(27);"a2";
190 PRINT "Now back to Letter Quality."
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-2. Macintosh Pascal Example of Font Selection

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'a2');
  WRITE(Prntr,'Near Letter Quality is selected.');
  WRITE(Prntr,chr(27),'a0');
  WRITE(Prntr,'Here's some correspondence.');
  WRITE(Prntr,chr(27),'a2');
  WRITELN(Prntr,'Now back to Letter Quality.');
  CLOSE(Prntr)
END.
```

MouseText

The binary, hexadecimal, and decimal equivalents of all the ASCII characters are shown in Appendix B.

MouseText is a set of 32 special characters used on Apple II computers. This set of characters is supported by the ImageWriter II to aid in printing hard copies of the Apple II computer's screen. There are two ways to access the MouseText characters contained in the ImageWriter II. The first and most direct method is to send the 8-bit (high ASCII) codes shown in Appendix B to the printer. In order for the ImageWriter to recognize the eighth data bit, you must use the ESC Z command in Table 4-2.

The second method is to use (temporarily) the standard (low ASCII) codes, 64 ($40) through 95 ($5F) for the MouseText characters by sending ESC & to the printer. To call a MouseText character when using the ESC & control code, subtract 128 decimal ($80 hexadecimal) from the ASCII value of the MouseText character and send the resulting standard ASCII character to the printer. The MouseText characters and their corresponding code values are shown in Appendix B.

When you wish to switch back to the standard ASCII character set, send ESC $ to the printer.

Table 4-2 lists the commands that allow you to use the MouseText characters.

Table 4-2. MouseText Character Commands

Characters	Decimal	Hexadecimal	Function
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit
ESC &	27 38	1B 26	Map MouseText to low ASCII ($40-$5F)
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC $	27 36	1B 24	Select standard ASCII characters (default)

SPACE: The space character

The example in Figures 4-3 and 4-4 demonstrates the use of MouseText characters by printing an Apple and the words "Apple Computer."

Applesoft BASIC

Applesoft BASIC always sets the eighth bit to 1 in every byte sent. If you use the ESC Z command in Table 4-2 to set the printer to include the eighth data bit, then every code sent to the printer using Applesoft BASIC will be read as high ASCII, and you will not be able to print ordinary characters until you return the printer to 7-bit mode. We recommend you use only the ESC & command to call MouseText characters in Applesoft BASIC programs.

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the automatic line feed after carriage return function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

Figure 4-3. Applesoft BASIC Example of MouseText Selection

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 PRINT CHR$ (27);"&";
141 PRINT "A";
142 PRINT CHR$ (27);"$";
150 PRINT "Apple Computer"
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-4. Macintosh Pascal Example of MouseText Selection

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'&');
  WRITE(Prntr,'A');
  WRITE(Prntr,chr(27),'$');
  WRITELN(Prntr,'Apple Computer');
  CLOSE(prntr)
END.
```

Alternate Language Characters

To see exactly how the foreign language characters are printed, refer to Appendix C, Character Specifications.

Your Apple ImageWriter II has eight different language fonts to aid in printing text in German, French, Italian, Swedish, Spanish, Danish, and British English, in addition to American English. Each of these fonts substitutes characters for these ten American symbols:

@ [\ ] ` { | } ~

Figure 2-2 shows the characters that can be substituted for these symbols. There are two ways to select the language font you wish to use: by setting DIP switches 1-1, 1-2, and 1-3 (as explained in Chapter 2), or by sending the command characters shown in Table 4-3. The DIP switch settings determine the language font in effect when the printer is turned on; when you use a software command, it overrides the DIP switch settings.

ASCII equivalents of decimal and hexadecimal numbers are shown in Appendix B.

You can choose only one language font at a time. Once you have chosen a language font, the ten special symbols always print the characters of that group until you choose a different group.

Table 4-3. Alternate (National) Character Group Commands

Characters	Decimal	Hexadecimal	Function
ESC Z CTRL-G CTRL-@	27 90 7 0	1B 5A 07 00	American
ESC Z CTRL-F CTRL-@ ESC D CTRL-A CTRL-@	27 90 6 0 27 68 1 0	1B 5A 06 00 1B 44 01 00	Italian
ESC Z CTRL-E CTRL-@ ESC D CTRL-B CTRL-@	27 90 5 0 27 68 2 0	1B 5A 05 00 1B 44 02 00	Danish
ESC Z CTRL-D CTRL-@ ESC D CTRL-C CTRL-@	27 90 4 0 27 68 3 0	1B 5A 04 00 1B 44 03 00	British
ESC Z CTRL-C CTRL-@ ESC D CTRL-D CTRL-@	27 90 3 0 27 68 4 0	1B 5A 03 00 1B 44 04 00	German
ESC Z CTRL-B CTRL-@ ESC D CTRL-E CTRL-@	27 90 2 0 27 68 5 0	1B 5A 02 00 1B 44 05 00	Swedish
ESC Z CTRL-A CTRL-@ ESC D CTRL-F CTRL-@	27 90 1 0 27 68 6 0	1B 5A 01 00 1B 44 06 00	French
ESC D CTRL-G CTRL-@	27 68 7 0	1B 44 07 00	Spanish

Default: The default is per DIP switches SW1-1, SW1-2, and SW1-3.

The example in Figures 4-5 and 4-6 calls the British language font, prints a British pound sign, then switches to the American language font. This example will work no matter which language font has been selected with the DIP switches.

Applesoft BASIC

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

Figure 4-5. Applesoft BASIC Example of Alternate (National) Character Group Selection

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 PRINT CHR$ (27);"Z";CHR$ (4);CHR$ (0);
131 PRINT CHR$ (27);"D";CHR$ (3);CHR$ (0);
150 PRINT "#"
200 PRINT CHR$ (27);"Z";CHR$ (7);CHR$ (0)
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-6. Macintosh Pascal Example of Alternate (National) Character Group Selection

```
PROGRAM Printer_Features;

VAR     Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'Z',chr(4),chr(0));
  WRITE(Prntr,chr(27),'D',chr(3),chr(0));
  WRITE(Prntr,chr(27),'Z',chr(7),chr(0));
  WRITELN(Prntr,'#');
  CLOSE(prntr)
END.
```

User-Designed Characters

The Apple ImageWriter II contains a special memory that is capable of holding up to 175 custom-designed characters. See Chapter 7 for a complete explanation of how to design custom characters and load them into memory. Once their specifications are stored in your ImageWriter's memory, you can fetch and print custom characters at any time. The codes shown in Table 4-4 switch the entire keyboard of printing characters between the normal set of ASCII characters and the custom character font.

Table 4-4. Custom Character Printing Commands

Characters	Decimal	Hexadecimal	Function
ESC '	27 39	1B 27	Switches to custom character font
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC *	27 42	1B 2A	Switches to custom character font (high ASCII values)
ESC $	27 36	1B 24	Switches back to normal font

Whenever you send one of the character-font control codes in Table 4-4 to the ImageWriter II, the printer changes fonts; it prints with that font until it receives one of the other codes or is reset. To print a single custom symbol, send the printer ESC ' (or ESC *), the keyboard character to which the custom symbol was assigned, and ESC $.

The ESC ' control code permits you to change to a font containing up to 95 custom characters of any width and fetch them with ordinary keyboard characters, and to call up to 80 custom characters assigned to high-ASCII values. The ESC * control code allows you to call up to 80 high-ASCII custom characters using only standard ASCII characters in the control code sequence. ESC ' and ESC * have the following characteristics:

- **ESC '** fetches custom characters stored in both 8-dot-maximum and 16-dot-maximum modes that were assigned to ordinary keyboard characters (that is, ASCII characters with decimal values 32 through 126—see Appendix B).

  ESC ' also fetches custom characters that were assigned to ASCII characters with decimal values 160 through 239. When you use ESC ' for this purpose, use the ESC Z command in Table 4-4 to set the printer to recognize the eighth data bit.

- **ESC *** lets you use standard ASCII characters (values 32 through 111 decimal) to fetch custom characters assigned to high-ASCII decimal values 160 through 239. To call a custom character when using the ESC * control code, subtract 128 decimal ($80 hexadecimal) from the ASCII value of the custom character and send the resulting standard ASCII character to the printer.

  For example, 193–128 = 65, which is the ASCII value for *A*; so to print a custom character that you loaded into memory at the high-ASCII value 193, send the following sequence of characters to the printer:

```
ESC * A
```

Applesoft BASIC

Applesoft BASIC always sets the eighth bit to 1 in every byte sent. If you use the ESC Z command in Table 4-4 to set the printer to include the eighth data bit, then every code sent to the printer using Applesoft BASIC will be read as high ASCII, and you will not be able to print low ASCII characters until you return the printer to 7-bit mode. We recommend you use only the ESC * command to call high-ASCII custom characters in Applesoft BASIC programs.

Applesoft BASIC

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the automatic line feed after carriage return function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

Type Attributes

In addition to selecting between print-quality character fonts and printing special characters, the ImageWriter allows you to vary the appearance of the type by changing the character width and spacing, by printing half-height characters, superscripts and subscripts, by boldfacing and underlining text, and by printing zeros with slashes through them. This section describes the printer commands you can use to control the appearance of the type.

Character Pitch

With the Apple ImageWriter II you can print characters in eight different widths (character pitches), from 9 characters per inch (cpi) to 17 cpi.

By printing any of these pitches in double-width mode, you can select from 16 character widths. See the Double-Width Printing section in this chapter for more information.

Two of the character pitch options print proportionally; that is, the space allotted to each character depends on the shape of the character. For instance, although *m* is wider than *i*, the ordinary character pitches allot the same number of dot spaces to an *i* as to an *m*. The proportional pitches eliminate the extra blank dot spaces around the *i* so it takes less space to print than the *m*. The proportional-width pitches are measured in dots per inch (dpi) rather than characters per inch.

Refer to Appendix C, Character Specifications, to see exactly how the characters are printed in both fixed-width and proportional fonts. See the Proportional Character Spacing section in this chapter for more information on printing with the proportional fonts.

The commands to select the eight character pitch options are listed in Table 4-5. The default character pitch is determined by the setting of DIP switches 1-6 and 1-7 (see Chapter 2).

Table 4-5. Character Pitch Commands

Characters	Decimal	Hexadecimal	Pitch	Characters/line
ESC n	27 110	1B 6E	9 cpi (Extended)	72
ESC N	27 78	1B 4E	10 cpi (Pica)	80
ESC E	27 69	1B 45	12 cpi (Elite)	96
ESC e	27 101	1B 65	13.4 cpi (Semicondensed)	107
ESC q	27 113	1B 71	15 cpi (Condensed)	120
ESC Q	27 81	1B 51	17 cpi (Ultracondensed)	136
ESC p	27 112	1B 70	144 dpi (Pica)	Proportional
ESC P	27 80	1B 50	160 dpi (Elite)	Proportional

cpi: characters per inch
dpi: dots per inch
Default: The default setting for character pitch is per DIP switches SW1-6 and SW1-7.

Any of these character pitches can be printed in boldface, underlined, or double-width styles.

The example in Figures 4-7 and 4-8 first prints with 17 cpi (ultracondensed), switches to 9 cpi (Extended), and then switches back to 17 cpi.

Figure 4-7. Applesoft BASIC Example of Character Pitch Selection

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
140 PRINT CHR$(27);"Q";
150 PRINT "It starts small, ";
160 PRINT CHR$(27);"n";
170 PRINT "gets BIG, ";
180 PRINT CHR$(27);"Q";
190 PRINT "and now it's small again."
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-8. Macintosh Pascal Example of Character Pitch Selection

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'Q');
  WRITE(Prntr,'It starts small, ');
  WRITE(Prntr,chr(27),'n');
  WRITE(Prntr,'gets BIG, ');
  WRITE(Prntr,chr(27),'Q');
  WRITELN(Prntr,'and now it''s small again.');
  CLOSE(Prntr)
END.
```

Proportional Character Spacing

The dot patterns and widths of proportional characters are shown in Appendix C

When your Apple ImageWriter II is printing in a proportional pitch (selected by the ESC P or ESC p commands), the spacing between characters is controlled by using the commands listed in Table 4-6. To set spacing between characters from zero to nine dots in both proportional pitches, use ESC s *n*, where *n* represents an ASCII character from 0 to 9 that specifies the number of dot spaces. Note that each proportional character includes one blank dot column, so that when the proportional character spacing command is set to 0, the actual separation between characters is one dot width.

To put extra spaces between individual characters, use ESC *m*, where *m* represents an ASCII numeral from 1 to 6 corresponding to the number of dot spaces to be inserted. For example, to print the word "ImageWriter" in elite proportional pitch, with one dot space between each character but two spaces between the "e" and the "W", you could send the following character string to the printer:

```
ESC P ESC s 1 I m a g e ESC 1 W r i t e r
```

If multiple ESC *m* commands are sent, the effect is cumulative.

Proportional character spacing is used primarily in programs that print justified text (text with both left and right margins straight, like text in a newspaper) by distributing the extra space in each line among the text characters. The example in Figures 4-9 and 4-10 demonstrates the use of proportional printing to print two lines that are left and right justified.

Table 4-6. Proportional Character Spacing Commands

Characters	Decimal	Hexadecimal	Function
ESC s *n*	27 115 *dn*	1B 73 *hn*	Set dot spacing to *n* for all characters (default *n* = 0)
ESC *m*	27 *dm*	1B *hm*	Insert *m* dot spaces

n: An ASCII character in the range 0 to 9, indicating the number of dot spaces between all characters.

dn: The decimal equivalent of *n* from 48 to 57.

hn: The hexadecimal equivalent of *n* from $30 to $39.

m: An ASCII character in the range 1 to 6, indicating the number of dot spaces to be inserted between two specific characters.

dm: The decimal equivalent of *m* from 49 to 54.

hm: The hexadecimal equivalent of *m* from $31 to $36.

Proportional printing is not available for the draft font. If you select a proportional pitch while printing in the draft font, the printer switches to the correspondence font; when you cancel the proportional pitch, the printer switches back to draft font.

Remember: These control codes work only in the two proportional character pitches. To use them with any other character pitch, you'll have to precede each character-spacing control code with an ESC P or ESC p code and follow it with the appropriate control code to return to the pitch you are using.

Figure 4-9. Applesoft BASIC Example of Proportional Character Spacing

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 PRINT CHR$ (27);"a2"; :REM Select NLQ font
140 PRINT CHR$ (27);"p"; :REM Proportional 144dpi
150 PRINT CHR$ (27);"s1"; :REM 1 dot spacing
160 PRINT "This line has 33 characters in it"
170 PRINT CHR$ (27);"s2"; :REM 2 dot spacing
180 PRINT "and this lin";
190 PRINT CHR$ (27);"s1"; :REM 1 dot spacing
200 PRINT "e ";
210 PRINT CHR$ (27);"2";  :REM Insert 2 spaces
220 PRINT "has 31 characters"
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-10. Macintosh Pascal Example of Proportional Character Spacing

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'a2');
  WRITE(Prntr,chr(27),'p');
  WRITE(Prntr,chr(27),'s1');
  WRITELN(Prntr,'This line has 33 characters in it');
  WRITE(Prntr,chr(27),'s2');
  WRITE(Prntr,'and this lin');
  WRITE(Prntr,chr(27),'s1');
  WRITE(Prntr,'e ');
  WRITE(Prntr,chr(27),'2');
  WRITELN(Prntr,'has 31 characters');
  CLOSE(Prntr)
END.
```

Double-Width Printing

The ImageWriter II creates double-width characters by printing two dots for every one normally printed. To start or end double-width printing, use the commands shown in Table 4-7.

Table 4-7. Double-Width Printing Commands

Characters	Decimal	Hexadecimal	Function
CTRL-N	14	0E	Start double width printing
CTRL-O	15	0F	Stop double width printing (default)

You can combine double-width printing with all of the character pitch options; the dot spacing remains the same, but you get half as many characters per inch. Thus you can select from eight double-width typefaces, ranging from 4.5 characters per inch (extended) to 8.5 characters per inch (ultracondensed). You can combine double-width printing with boldfacing to make excellent headlines.

Double-width printing also causes each dot in a graphics pattern to be printed twice; see Chapter 8 for information on printing graphics.

Once double-width printing is started, it remains in effect until it is canceled by either CTRL-O or by resetting the printer. If you start double-width printing while the draft font is in effect, the printer switches to the double-width correspondence font; it returns to draft font when you end double-width printing.

See Chapter 1 for example programs demonstrating how to use the double-width printing commands.

Underlining Text

The Apple ImageWriter II underlines characters by printing the bottom dots in each character matrix (see Appendix C for character specifications). Thus underlining does not require double striking of characters, nor does it affect printing speed. Use the commands in Table 4-8 to start and end underlining.

Table 4-8. Underline Text Commands

Characters	Decimal	Hexadecimal	Function
ESC X	27 88	1B 58	Start underline
ESC Y	27 89	1B 59	Stop underline (default)

When an ESC X command is in force, all text is underlined, including spaces and punctuation. You can combine underlining with boldface printing, with double-width printing, and with all character pitches. Once underlining is started, it remains in effect until it is canceled by either ESC Y or by resetting the printer.

See Chapter 1 for example programs demonstrating how to use the underlining commands.

Boldfacing Text

The Apple ImageWriter II uses a single pass to create boldface characters by printing each dot in the character twice with a small shift of position. The printer runs at half speed during boldface printing. Use the commands in Table 4-9 to start and end boldface printing.

Table 4-9. Boldface Text Commands

Characters	Decimal	Hexadecimal	Function
ESC !	27 33	1B 21	Start boldface
ESC "	27 34	1B 22	Stop boldface (default)

Boldface printing is available in NLQ and correspondence fonts for all character pitches, including double-width printing. Boldface printing is also available in all graphics modes (see Chapter 8). If you combine boldface printing with underlining, it increases the darkness of the underline as well as of the character.

Once boldfacing is started, it remains in effect until it is canceled by either ESC " or by resetting the printer. If you start boldface printing while the draft font is in effect, the printer switches to the boldface correspondence font; it returns to draft font when you end boldfacing.

See Chapter 1 for example programs demonstrating how to use the boldface printing commands.

Half-Height Text

The ImageWriter II creates half-height characters by printing correspondence font characters with a vertical dot density of 144 dots per inch (dpi), twice the normal density. The vertical resolution of 144 dpi is

achieved by printing one pass, executing a paper movement of 1/144 inch, and then overprinting. Half-height text can be used to increase the amount of data that can be printed on a single page; this is useful for long tables or large spreadsheet files, for example.

To start or end half-height printing, use the commands shown in Table 4-10.

Table 4-10. Half-Height Text Commands

Characters	Decimal	Hexadecimal	Function
ESC w	27 119	1B 77	Start half-height text
ESC W	27 87	1B 57	Stop half-height text (default)

When the printer is creating half-height text, print head wires 4-7 are the only ones used (see Appendix C for character specifications). If half-height text is underlined, wire 7 is used to print the underline (unless subscript or superscript is called at the same time, in which case wire 9 is used for underlining).

Once half-height printing is started, it remains in effect until it is canceled by either ESC W or by resetting the printer. If you start half-height printing while either the draft font or NLQ font is in effect, the printer switches to the half-height correspondence font; it returns to draft or NLQ font when you end half-height printing.

The example in Figures 4-11 and 4-12 first prints with full size text, switches to half-height text and then back to full size text.

Figure 4-11. Applesoft BASIC Example of Half-Height Text Selection

```
100 D$ = CHR$(4)
120 PRINT D$; "PR#1"
140 PRINT CHR$(27);"W";
150 PRINT "It starts normal size, ";
160 PRINT CHR$(27);"w";
170 PRINT "becomes small, ";
180 PRINT CHR$(27);"W";
190 PRINT "and now it's back to normal."
260 PRINT D$; "PR#0"
270 END
```

Figure 4-12. Macintosh Pascal Example of Half-Height Text Selection

```
PROGRAM Printer_Features;

VAR     Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITE(Prntr,chr(27),'W');
  WRITE(Prntr,'It starts normal size, ');
  WRITE(Prntr,chr(27),'w');
  WRITE(Prntr,'becomes small, ');
  WRITE(Prntr,chr(27),'W');
  WRITELN(Prntr,'and now it''s back to normal.');
  CLOSE(Prntr)
END.
```

Subscript and Superscript Text

The ImageWriter II creates subscript and superscript characters in a manner identical to half-height text, except that the bottom five print wires are used for subscripting, and the top five wires are used for superscripting. The paper is not rolled forward or backward to print subscripts or superscripts, since the characters are half-height and print on the same line as the normal characters.

The subscript and superscript commands are listed in Table 4-11.

Table 4-11. Subscript and Superscript Commands

Characters	Decimal	Hexadecimal	Function
ESC x	27 120	1B 78	Start superscript text
ESC y	27 121	1B 79	Start subscript text
ESC z	27 122	1B 7A	Stop either superscript or subscript (default)

Once subscript or superscript printing is started, it remains in effect until it is canceled by either ESC z or by resetting the printer. If you start subscript or superscript printing while either the draft or NLQ font is in effect, the printer switches to the subscript or superscript correspondence font; it

returns to draft or NLQ font when you end subscript or superscript printing. If either subscript or superscript characters are underlined, wire 9 is used to print the underline.

The example in Figures 4-13 and 4-14 first demonstrates the use of subscript and then the use of superscript. When you run this program, it prints the text shown in Figure 4-15.

Figure 4-13. Applesoft BASIC Example of Subscript/Superscript Selection

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 PRINT "This is a Subscript Sample:"
140 PRINT "A"; CHR$ (27);"y"; "t"; CHR$ (27);"z";
141 PRINT " = A"; CHR$ (27);"y"; "1"; CHR$ (27);"z";
142 PRINT " +  A"; CHR$ (27);"y"; "2"; CHR$ (27);"z"
150 PRINT "This is a Superscript Sample:"
160 PRINT "Area = PiR"; CHR$ (27);"x"; "2";
170 PRINT CHR$ (27);"z"
260 PRINT D$; "PR# 0"
270 END
```

Figure 4-14. Macintosh Pascal Example of Subscript/Superscript Selection

```
PROGRAM Printer_Features;

VAR     Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITELN(Prntr,'This is a Subscript Sample:');
  WRITE(Prntr,'A', chr(27),'y', 't', chr(27),'z');
  WRITE(Prntr,' = A', chr(27),'y', '1', chr(27),'z');
  WRITE(Prntr,' +  A', chr(27),'y', '2');
  WRITELN(Prntr,chr(27),'z');
  WRITELN(Prntr,'This is a Superscript Sample:');
  WRITE(Prntr,'Area = PiR', chr(27),'x', '2');
  WRITELN(Prntr,chr(27),'z');
  CLOSE(Prntr)
END.
```

Figure 4-15. Subscript and Superscript Printout Example

```
This is a Subscript Sample:
At = A1 + A2
This is a Superscript Sample:
Area = PiR²
```

Slashed and Unslashed Zeros

This command can be combined with several others by setting the "software switches"; see Chapter 3 for details.

Some people prefer to print the numeral zero with a slash through it (Ø)to distinguish it from the capital letter O. The Apple ImageWriter II can print zeros either way, but normally prints them unslashed. To switch between slashed and unslashed zeros, use the commands shown in Table 4-12.

Table 4-12. Slashed and Unslashed Zero Commands

Characters	Decimal	Hexadecimal	Function
ESC D CTRL-@ CTRL-A	27 68 0 1	1B 44 00 01	Zeros slashed
ESC Z CTRL-@ CTRL-A	27 90 0 1	1B 5A 00 01	Zeros unslashed (default)

Applesoft BASIC

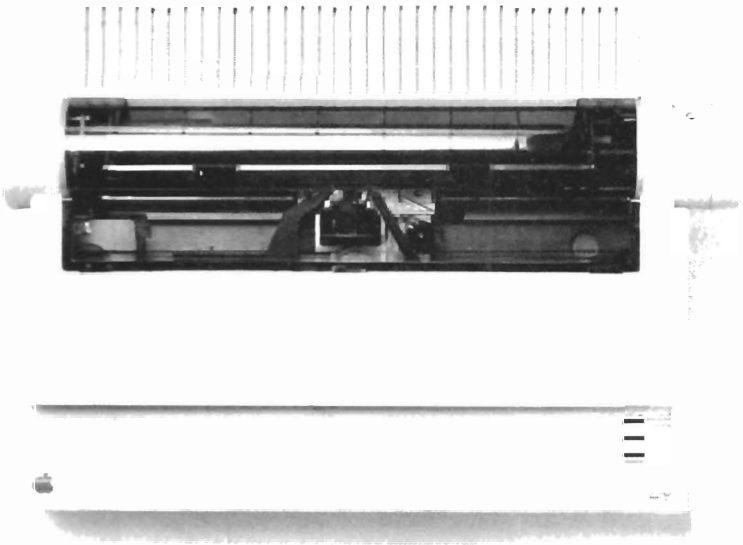
When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

The following line of Applesoft BASIC code illustrates setting the printer to slash zeros.

```
130 PRINT CHR$ (27);"D";CHR$ (0);CHR$ (1);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'D',chr(0),chr(1));
```



This chapter describes how use ImageWriter II command codes to control the placement of text on the page and the movement of paper through the printer. These commands control

1. Page formatting
   - left margin setting
   - page length
2. Print head motion
   - carriage returns
   - unidirectional printing
   - backspacing
   - tabbing, including setting and clearing tab stops
   - placement of the print head at any dot column
3. Paper motion
   - setting the top of form position
   - form feeds
   - line feeds, including spacing and direction
   - activating or deactivating the perforation skip function
   - activating or deactivating the paper-out sensor
4. Automatic carriage returns and line feeds
   - carriage return insertion before line feeds and form feeds
   - automatic line feed insertion after a carriage return
   - automatic line feed when the line is full

This chapter also provides programming examples in both Applesoft BASIC and Macintosh Pascal for most of the commands. If you need to use these commands with other languages or applications, see the Writing Programs section in Chapter 1.

Page Boundaries

Text and graphics are printed within the boundaries set by the left margin and page length commands, described in this section. The placement of text within those boundaries is controlled by the commands described in the Print Head Motion and Paper Motion sections.

Left Margin Setting

When you turn the power on, your Apple ImageWriter II is set to start each line of print as far left as the print head can travel. This is called position 0, and corresponds physically with the red line engraved at the left end of the roller shaft. You can change the left print margin at any time by sending the printer the left margin command shown in Table 5-1.

Table 5-1. Left Margin Command

Characters	Decimal	Hexadecimal	Function
ESC L *nnn*	27 76 *ddd*	1B 4C *hhh*	Set left margin at column *nnn*

n: An ASCII numeral character from 0 to 9. The range of *nnn* depends on the current character pitch as shown in Table 5-2. The default value is 000.

d: The decimal equivalent of *n*; from 48 to 57.

h: The hexadecimal equivalent of *n;* from $30 to $39.

The number *nnn* in this command is a three-digit decimal number represented in ASCII numerals; it is sent one digit at a time to the printer. For example, to set the left margin to column 23, send the following characters to the printer:

```
ESC L 0 2 3
```

While *nnn* must always be three digits long, leading zeros may be replaced by space characters. The range of *nnn* depends on the current character pitch as shown in Table 5-2.

Table 5-2. Left Margin Range vs. Character Pitch

Pitch	Characters/line	Range of *nnn*
9 cpi (Extended)	72	000 - 071
10 cpi (Pica)	80	000 - 079
12 cpi (Elite)	96	000 - 095
13.4 cpi (Semicondensed)	107	000 - 106
15 cpi (Condensed)	120	000 - 119
17 cpi (Ultracondensed)	136	000 - 135
144 dpi (Pica)	Proportional	000 - 071
160 dpi (Elite)	Proportional	000 - 079

Character pitch is discussed in Chapter 4.

The position of the margin is measured by counting characters, and so depends on the character pitch in force at the time the margin command is sent. For example, if 15 cpi printing is selected and the left margin is set at 15 characters, printing begins one inch to the right of position 0. If elite proportional pitch is selected, the margin setting is measured at 10 character positions to the inch; with pica proportional pitch, it is measured at 9 to the inch.

Important! The position of the margin depends on the character pitch in force at the time the margin command is sent. If you subsequently change the character pitch, the margin does *not* move to accommodate the change.

The following line of Applesoft BASIC code sets the left margin to the 36th character position (note that, since the first character position is in column 0, the 36th position is in column 35):

```
100 PRINT CHR$ (27);"L035"
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'L035');
```

The left margin remains at the selected setting until another ESC L *nnn* command is sent to the printer, or until the printer is reset. For example, to move the margin back to the far left end, use the following Applesoft BASIC command:

```
100 PRINT CHR$ (27);"L000"
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'L000');
```

Important! Margin positions start with 000, while tab positions start with 001. Thus, for example, a margin setting of 5 starts each line at the same position as a tab setting of 6.

Page Length

The Form Feed button on the Apple ImageWriter II control panel, or the form feed command (see the Paper Motion section in this chapter), advances the paper to the next top of form (TOF) position. The distance between TOFs is determined by DIP switch 1-4 (Chapter 2) or by the page length command shown in Table 5-3.

Table 5-3. Page Length Command

Characters	Decimal	Hexadecimal	Function
ESC H *nnnn*	27 72 *dddd*	1B 48 *hhhh*	Set page length to *nnnn*/144 inch

n: An ASCII numeral character from 0 to 9. The range of *nnnn* is from 0001 to 9999.

d: The decimal equivalent of *n;* each *d* can be from 48 to 57.

h: The hexadecimal equivalent of *n;* each *h* can be from $30 to $39.

Default: The default page length is per DIP switch 1-4.

The number *nnnn* in this command is a four-digit decimal number represented in ASCII numerals; it is sent one digit at a time to the printer. For example, to set the page length to ten inches (1440/144 inch), send the following characters to the printer:

```
ESC H 1 4 4 0
```

Although *nnnn* must always be four digits long, leading zeros may be replaced by space characters.

The following line of Applesoft BASIC code illustrates setting the page length to one inch (144/144 inch), as for printing mailing labels that measure one inch from the top of one to the top of the next:

```
130 PRINT CHR$ (27);"H0144";
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'H0144');
```

Print Head Motion

This section describes the commands that control the motion of the print head, and the horizontal positioning of text or graphics on the page.

Carriage Return

In manual typewriters, the paper moves horizontally as you type; the assembly that carries the paper is called the "carriage;" and to return to the left margin of the page you move the carriage all the way to the right. This action is called a "carriage return." The analogous feature on the ImageWriter II is also referred to as a carriage return, but the actual sequence of events is different.

Carriage return is abbreviated as CR.

When the printer receives data, it stores it in a random-access memory (RAM) called the print buffer; no printing actually occurs until a full line of text is stored in the print buffer or until a print command is received (CR is a print command; see the Print Commands section in Chapter 6). When the CR command is received, the current line in the print buffer is printed out, and the printer is set so that when the *next* line is full or print command is received, the text or graphics pattern starts at the left margin. The print head may physically print from left to right or from right to left (see the next section), but the line of text or graphics following a carriage return starts at the left margin, just as it would if typed on a typewriter.

The carriage return command is shown in Table 5-4.

Table 5-4. Carriage Return Command

Characters	Decimal	Hexadecimal	Function
CTRL-M	13	0D	Execute carriage return

A carriage return is always executed automatically when the printer receives enough text or graphics data to fill an entire line. A carriage return may also be inserted automatically before every line feed or form feed command (see the Carriage Return Insertion section in this chapter).

The carriage return does *not* necessarily include a line feed; that is, the next line of data overprints the current line unless a line feed command follows the carriage return. The ImageWriter II automatically inserts a line feed following a carriage return if set to do so with DIP switch 1-8 (Chapter 2) or by the automatic line-feed insertion command shown in Table 5-15.

The following line of Applesoft BASIC code sends a carriage return to the printer.

```
10 PRINT CHR$ (13);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(13));
```

Unidirectional Printing

When the Apple ImageWriter II receives a full line of text, or a command that causes printing (such as a carriage return), it prints out the data in the print buffer. When you turn the printer on, it is set to print lines both when the print head is travelling from left to right and when it is travelling from right to left. This feature, called bidirectional printing, increases the efficiency of printing and thus the printing speed.

In some cases, however, you may prefer to print in one direction only (unidirectional printing) because it improves the precision with which dots and characters line up to make vertical lines, text columns, and graphics patterns.

Table 5-5 shows the commands that allow you to control this function. Either command remains in force until canceled by the other or until the printer is reset.

Table 5-5. Unidirectional-Bidirectional Printing Commands

Characters	Decimal	Hexadecimal	Function
ESC >	27 62	1B 3E	Unidirectional printing
ESC <	27 60	1B 3C	Bidirectional printing (default)

By the Way: The vertical alignment of dots and characters when printed in bidirectional mode by the ImageWriter II is better than the alignment achieved by most other dot matrix printers in unidirectional mode. You will probably find that unidirectional printing is not needed to achieve excellent results.

Backspacing

The ImageWriter II can overprint one character with another by backspacing a single character at a time.

The backspace command is shown in Table 5-6.

Table 5-6. Backspacing Command

Characters	Decimal	Hexadecimal	Function
CTRL-H	8	08	Backspace

Use the backspace command to create new characters by overprinting one character with another; for example, to print the cents sign (¢), send the following character string to the printer:

```
c CTRL-H /
```

The backspace command must be followed by a printable character or another backspace command—it cannot be followed by any other control character.

ImageWriter

The original ImageWriter printer can not accept two backspace commands in sequence; to overprint a string of characters (for example, to print a line with slashes through all the characters) on the original ImageWriter, you must follow each character with CONTROL-H plus the symbol to be overprinted. Follow this procedure to make your program compatible with all ImageWriter printers; you can use multiple backspace commands for any program to be used only with ImageWriter II printers.

Tabbing

You can set up to 32 horizontal tab positions for your Apple Imagewriter; you can set an entire line of tab positions at one time, and add individual tabs at any time.

Each tab character (TAB or CONTROL-I) sent to the printer advances the print head to the next tab position. The printer ignores tab commands if no tab positions have been set or if the print head is at or beyond the last tab position. There is a control code that clears individual tab positions, and one that clears all tab positions at once. Turning off or resetting the printer also clears all tab positions.

The tabbing control codes are shown in Table 5-7

Table 5-7. Tabbing Commands

Characters	Decimal	Hexadecimal	Function
ESC (*aaa,bbb,...,nnn.*	27 40 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 28 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Sets tabs at positions *aaa,bbb,...,nnn*
ESC u *nnn*	27 117 *dndndn*	1B 75 *hnhnhn*	Adds one tab stop at column *nnn*
CTRL-I	9	09	Moves print head to next tab
ESC) *aaa,bbb,...,nnn.*	27 41 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 29 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Clears tabs at positions *aaa,bbb,...,nnn*
ESC Ø	27 48	1B 30	Clears all tabs

a,b,...,n: ASCII characters in the range 0 to 9. The maximum for each three-digit number depends on the character pitch as shown in Table 5-2.

da,db,...,dn: The decimal equivalents of *a,b,...,n;* from 48 to 57.

ha,hb,...,hn: The hexadecimal equivalents of *a,b,...,n;* from $30 to $39.

Default: When the printer is reset, all tabs are cleared.

Setting a Line of Tab Stops

To set a line of up to 32 tab positions, send an ESC character, a left parenthesis, a sequence of three-digit numbers separated by commas, and a period. Send the numerals to the printer in ascending numerical order, using ASCII characters or an ASCII character function (such as CHR$ in BASIC or CHR in Pascal). The numbers specify the columns at which tabs are to be set, starting from the left margin. The left margin has the column position number 1.

Important!

The ESC (control sequence clears all previously-set tab stops. To add a single tab stop, use the ESC u control sequence shown in Table 5-7.

To set tabs at column positions 5, 23, and 67, for example, use the following command sequence:

```
ESC ( 0 0 5 , 0 2 3 , 0 6 7 .
```

While *aaa, bbb, ..., nnn* must always be three digits long, leading zeros may be replaced by space characters.

The character pitch determines the maximum tab setting as shown in Table 5-8.

Table 5-8. Maximum Tab Setting vs. Character Pitch

Character Pitch	Character Columns/Inch	Range of *aaa,bbb,...,nnn*
Extended	9	001 - 072
Pica	10	001 - 080
Elite	12	001 - 096
Semicondensed	13.4	001 - 107
Condensed	15	001 - 120
Ultracondensed	17	001 - 136
Pica proportional	9	001 - 072
Elite proportional	10	001 - 080

Important!

Tab positions are based on the character pitch and left margin position in force at the time you enter the tabs. If you later change the character pitch or the left margin, the tabs remain in the same absolute locations and no longer correspond to current character column positions.

Margin position column numbers start with 000, while tab position column numbers start with 001. Thus, for example, a margin setting of 5 starts each line at the same position as a tab setting of 6.

Character pitches are discussed in Chapter 4; the left margin command is discussed in this chapter.

Adding a Single Tab Stop

The ESC (control sequence can be used only to set an entire line of tabs; it clears all previously-set tabs. To add a single tab to a line of tabs already set, use the ESC u control sequence shown in Table 5-7. The original tab settings are retained and the new one added. Only one tab can be added at a time. For example, to add the tab at column position 25 to those already set, use the following command sequence:

```
ESC u 0 2 5
```

You can replace leading zeros with space characters.

See the previous section for restrictions on setting tabs.

Using Tabs

To cause the next character or graphics line printed to start at the next tab position, send a CONTROL-I to the printer. This character is sometimes called the ASCII HT ("horizontal tab") command, and is the code generated when you press TAB on your computer. Several CONTROL-I characters can be sent in a row to cause the next character to print several tab positions from the last character printed.

Important! If you have an Apple Super Serial Card or are using an Apple IIc, then the CONTROL-I command is normally interpreted as a command character for the serial port, and is not passed on to the printer. You can change the command character from CONTROL-I to another control character by sending CONTROL-I CONTROL-*X* where *X* is any valid control character (see Appendix B); or you can disable the serial port from receiving any command characters by sending CONTROL-I Z CONTROL-M. To pass a single CONTROL-I to the ImageWriter II through a Super Serial Card, you can send two CONTROL-I's in a row (this does not work for the Apple IIc). See the *Apple IIe Technical Reference Manual* or the *Apple IIc Technical Reference Manual* for more information.

By the Way: In many text editing and word processing programs, when you press TAB the program inserts space characters into the file instead of the CONTROL-I character. If you are using such a program, you may not be able to enter CONTROL-I directly into your text file.

Clearing Tab Stops

You can clear one or more individual tabs with the ESC) control sequence shown in Table 5-7. To clear a set of tab positions, send an ESC character, a right parenthesis, a sequence of three-digit numbers separated by commas,

and a period. Send the numerals to the printer in ascending numerical order using ASCII characters or an ASCII character function (such as CHR$ in BASIC or CHR in Pascal). The numbers specify the columns at which tabs are to be cleared, starting from the left margin. The left margin has the column position number 1. For example, to clear tabs at column positions 5, 23, and 67, use the following command sequence:

```
ESC ) 0 0 5 , 0 2 3 , 0 6 7 .
```

You can replace leading zeros with space characters. If you attempt to clear a tab that is not set, you won't affect the rest of the command; bad syntax will cause the whole command to be ignored, however.

Important!

Tab positions are based on the character pitch and left margin position in force at the time you enter the tabs. If you later change the character pitch or the left margin, the tabs remain in the same absolute locations and no longer correspond to current character column positions. Be sure you reset the character pitch and left margin to their original settings before attempting to clear individual tab stops.

To clear all tabs at once, use the ESC Ø control code shown in Table 5-7.

Exact Print Head Placement

Text or Graphics printing can be started at any dot position on the page. To start printing with the print head a specified number of dot columns from the left margin, use the commands in Table 8-5. The Exact Print Head Placement section in Chapter 8 describes this command in detail and provides an example of its use.

Paper Motion

This section describes the commands that control the movement of paper through the printer, including form feeds, line feeds, and automatic skipping over perforations; commands to set the top of form position, line feed spacing, and line feed direction; and commands to activate and deactivate the paper-out sensor.

Top of Form Set

The perforation skip function is discussed in Chapter 2.

When you turn on your ImageWriter II, the top of form (TOF) position is set at the location of the current print line. The TOF determines the point to which the paper scrolls when a form feed or perforation skip takes place.

You can reset the TOF to the current print-line position at any time by using the command shown in Table 5-9.

Table 5-9. Top of Form Set Command

Characters	Decimal	Hexadecimal	Function
ESC v	27 118	1B 76	Sets TOF to current position

Important! The ESC v command is ignored when the SheetFeeder option is installed. The SheetFeeder option is discussed in Chapter 9.

The following line of Applesoft BASIC code sets the top of form to the current print-line position:

```
130 PRINT CHR$ (27);"v";
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'v');
```

Form Feed

The form feed (FF) command has the same effect as pressing the Form Feed button on the control panel of your ImageWriter II: it causes the printer to advance the paper to the next TOF position. The form feed command is shown in Table 5-10. The TOF is discussed in the previous section.

Table 5-10. Form Feed Command

Characters	Decimal	Hexadecimal	Function
CTRL-L	12	0C	Feeds paper to the next TOF

When you turn on the ImageWriter II, it is set so that the form feed, line feed, and carriage return commands all cause the current line in the print buffer to be printed, and so that a carriage return is inserted before any line feed or form feed command. Use the command in Table 6-1 to cause the carriage return to be the only command that causes printing. Use the commands in Table 5-14 to determine whether a carriage return is automatically inserted before a form feed.

The following line of Applesoft BASIC code sends a form feed command to the printer:

```
130 PRINT CHR$ (12);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(12));
```

Line Feeds

The line feed (LF) command causes the paper to scroll by one line. The multiple line feed command causes the paper to scroll by up to 15 lines. The line feed spacing and line feed direction commands affect the action of both line feed commands. All of these commands are shown in Table 5-11.

Table 5-11. Line Feed Commands

Characters	Decimal	Hexadecimal	Function
CTRL-J	10	0A	Feeds paper one line (LF)
CTRL-_ *n*	31 *dn*	1F *hn*	Feeds 1 to 15 lines of blank paper (*n* = 1 to 9, :, ;, <, =, >, or ?)
ESC A	27 65	1B 41	6 lines per inch (default)
ESC B	27 66	1B 42	8 lines per inch
ESC T *mm*	27 84 *dmdm*	1B 54 *hmhm*	Distance between lines to be *mm*/144 inch (*mm* = 01 to 99)
ESC f	27 102	1B 66	Forward line feeding (default)
ESC r	27 114	1B 72	Reverse line feeding

n: An ASCII character representing the number of lines to feed, as follows:
Number of Lines: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ASCII Character: 1 2 3 4 5 6 7 8 9 : ; < = > ?
See Appendix B for decimal and hexadecimal equivalents.

dn: The decimal equivalent of *n;* from 49 to 63.

hn: The hexadecimal equivalent of *n;* from $31 to $3F.

m: An ASCII character from 0 to 9. The range of *mm* is from 01 to 99.

dm: The decimal equivalent of *m;* from 48 to 57.

hm: The hexadecimal equivalent of *m;* from $30 to $39.

Note that for multiple line feeds of 10 to 15 lines, you use symbols rather than numbers for *n*.

When you turn on the ImageWriter II, it is set so that the form feed, line feed, and carriage return commands all cause the current line in the print buffer to be printed, and so that a carriage return is inserted before any line feed or form feed command. Use the command in Table 6-1 to cause the carriage return to be the only command that causes printing. Use the commands in Table 5-14 to determine whether a carriage return is automatically inserted before a line feed command.

The following line of Applesoft BASIC code illustrates a multiple line feed of twelve lines.

```
130 PRINT CHR$ (31);"<";
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(31),'<');
```

Line Feed Spacing

When the printer is turned on, it is set to a vertical line spacing of six lines to the inch. You can change at any time to eight lines per inch, or to any line spacing from 1/144 inch to 99/144 inch in increments of 1/144 inch.

The three line feed spacing commands are shown in Table 5-11. These three commands affect all line feeds executed *after* the line in which they are included; that is (for forward line feeding), they affect the space immediately below the line containing the command. The line feed spacing commands remain in effect until you set a new line feed pitch, or until the printer is reset.

The vertical distance between dots is 1/72 of an inch, so the ESC T *mm* command permits line feeding to be as little as one-half of a dot dimension. Line feeds of 1/144 inch are used to create half-height and NLQ fonts.

You can use ESC T 18 and ESC T 24 as alternate ways to set line feeds of eight and six to the inch respectively. To print graphics, use ESC T 16, which leaves no gaps between vertically aligned dots, so you can draw continuous vertical lines or graphics patterns.

The following lines of Applesoft BASIC code illustrate two ways to set the printer to use a line pitch of 8 lines per inch.

```
130 PRINT CHR$ (27);"B";
```

```
130 PRINT CHR$ (27);"T18";
```

Use either of the following lines of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'B');
```

```
WRITE(Prntr,chr(27),'T18');
```

Line Feed Direction

Your ImageWriter II can can feed paper forward or backward. When you turn the power on, forward line feeding is automatically selected. Use the commands shown in Table 5-11 to change the line feed direction. The printer continues to feed in the direction last selected until it receives the opposite command or the printer is reset.

The performance of reverse line feeding is not guaranteed once the bottom edge of a form has left the forms tractor, or once a cut sheet has advanced past the pressure roller at the bottom of the platen.

When using the ImageWriter II SheetFeeder, reverse paper motion is limited to one half inch, and does not scroll beyond the top-of-form position. However, the printer keeps track of the total reverse line-feed distance sent to it, and no paper motion occurs until the net motion is positive.

The following line of Applesoft BASIC code sets the printer to reverse line-feed mode:

```
130 PRINT CHR$ (27);"r";
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'r');
```

Perforation Skip

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

DIP switch SW 1-5 determines whether the printer automatically skips over perforations when using continuous (fanfold) paper. This switch can be overridden with the commands in Table 5-12.

For more information on this function, see the Perforation Skip section in Chapter 2.

Table 5-12. Perforation Skip Commands

Characters	Decimal	Hexadecimal	Function
ESC D CTRL-@ CTRL-D	27 68 0 4	1B 44 00 04	Perforation skip disabled
ESC Z CTRL-@ CTRL-D	27 90 0 4	1B 5A 00 04	Perforation skip enabled

Default: The default is determined by DIP switch SW 1-5.

Applesoft BASIC

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in this chapter for information on resetting this function.

The following line of Applesoft BASIC code illustrates setting the printer to skip over perforations.

```
130 PRINT CHR$ (27);"Z";CHR$ (0);CHR$ (4);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'Z',chr(0),chr(4));
```

Paper-Out Sensor

The paper-out sensor detects when the printer is about to run out of paper, and monitors paper flow when the SheetFeeder is installed. If the paper-out sensor is enabled and no SheetFeeder is installed, then when approximately 7/144 inch of paper remains, the Error Light comes on and the printer is deselected. If a SheetFeeder is installed and the paper-out sensor detects the end of the page, the current page is automatically ejected.

Use the commands in Table 5-13 to enable and disable the paper-out sensor.

Table 5-13. Paper-Out Sensor Commands

Characters	Decimal	Hexadecimal	Function
ESC O	27 79	1B 4F	Paper-out sensor off
ESC o	27 111	1B 6F	Paper-out sensor on (default)

When a SheetFeeder is installed, the ESC O command disables the automatic eject feature, so that a page is ejected only when the programmed page length is exceeded, or when you press the Form Feed button or send a form feed command to the printer. The ESC o command reenables the automatic eject feature, so that the page is automatically ejected when the end of the page is sensed. (The new page is not fed into the printer until the printer receives a print command.)

The following line of Applesoft BASIC code illustrates switching off the paper-out sensor.

```
130 PRINT CHR$ (27);"O";
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'O');
```

Automatic Carriage Returns and Line Feeds

When printing text and most graphics patterns, each carriage return is normally followed by a line feed, so that subsequent lines do not overprint previous text or graphics. Printer commands are provided that let you specify whether a carriage return is inserted before every line feed and form feed character, and whether a line feed is automatically executed after every carriage return command. A carriage return is automatically executed whenever a full line's worth of data is sent to the printer; a command is provided that lets you specify whether a line feed is also automatically executed when a line is full. The commands that control automatic carriage returns and line feeds are described in this section.

Carriage Return Insertion

When you turn on the ImageWriter II, it's set to print out the data in the print buffer and to insert a carriage return (CR) before every line feed (LF) or form feed (FF) character it receives.

To control whether a CR is inserted before each LF or FF, use the commands in Table 5-14. To control whether each LF or FF character causes printing, use the commands shown in Table 6-1.

Table 5-14. Carriage Return Insertion Commands

Characters	Decimal	Hexadecimal	Function
ESC l 1	27 108 49	1B 6C 31	No carriage return insertion
ESC l Ø	27 108 48	1B 6C 30	Insert CR before LF and FF (default)

Important! When you select CR as the only command that causes printing, the LF and FF commands function only when used immediately following a carriage return character. CR insertion has no effect when the LF and FF commands are disabled from causing printing.

The CR command sets the printer to start the next line at the left margin. For example, when CR insertion is in effect and a set of characters containing a line feed (LF) is sent to the printer, the following sequence occurs:

1. The characters in the print buffer, up to the LF, are printed out.
2. The paper scrolls by one line, with the amount and direction of the line feed controlled by the options in Table 5-11.
3. The printer is set to start the next line at the left margin (see the Carriage Return section in this chapter).
4. The rest of the characters are loaded into the print buffer, to be printed out the next time a CR, LF, or FF is received.

Suppose the following line is sent to the printer while CR insertion is in effect:

```
----------------------(LF)----------------------(CR)
```

The printed result is as follows:

```
----------------------
----------------------
```

If you disable CR insertion, the LF does not cause a carriage return; in this case, when the printer is sent the same line, the LF causes the paper to scroll, but does not return the print head to the left margin. The result is as follows:

```
----------------------
                       ----------------------
```

The programming code samples in Figures 5-1 and 5-2 illustrate this example.

Figure 5-1. Applesoft BASIC Example of Carriage Return Insertion Commands

```
104 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
129 REM  THE NEXT LINE SETS SOFTWARE SWITCH SO LF CAUSES PRINTING
130 PRINT CHR$ (27);"D";CHR$ (64);CHR$ (0);
135 PRINT CHR$ (27);"10";
140 PRINT "----------------------";CHR$ (10);"----------------------"
145 PRINT CHR$ (10)
146 PRINT CHR$ (10)
150 PRINT CHR$ (27);"11";
160 PRINT "----------------------";CHR$ (10);"----------------------"
260 PRINT D$; "PR# 0"
270 END
```

Figure 5-2. Macintosh Pascal Example of Carriage Return Insertion Commands

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
    {The next line sets software switch so LF causes printing}
  WRITE(Prntr,chr(27),'D',chr(64),chr(0));
  WRITE(Prntr,chr(27),'10');
  WRITELN(Prntr,'----------------------',chr(10),'----------------------');
  WRITELN(Prntr,''); WRITELN(Prntr,'');
  WRITE(Prntr,chr(27),'11');
  WRITELN(Prntr,'----------------------',chr(10),'----------------------');
  CLOSE(prntr)
END.
```

Automatic Line Feed After Carriage Return

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

Many standard programs automatically add a line feed after every carriage return sent to the printer; others offer this feature as an option. If the program you are using sends only a carriage return character to start a new line, you can cause the printer to add the line feed by itself. If the program you are using already adds a line feed, inserting another line feed produces double-spaced printing.

There are two ways to control this feature: by setting DIP switch 1-8 (Chapter 2) or by sending the commands shown in Table 5-15. The DIP switch settings determine whether line feed insertion is in effect when the printer is turned on; the software commands override the DIP switch setting.

Table 5-15. Automatic Line Feed After Carriage Return Commands

Characters	Decimal	Hexadecimal	Function
ESC D {@} CTRL-@	27 68 128 0	1B 44 80 00	Adds automatic line feed after CR
ESC Z {@} CTRL-@	27 90 128 0	1B 5A 80 00	No line feed added after CR

{@}: A high-ASCII CONTROL-@ (decimal 128, hex $80)

Default: The default is per DIP switch 1-8.

Once one of these commands has been sent to the printer, it remains in effect until the other command is sent or the printer is reset.

Applesoft BASIC

Any time you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. Use the commands in Table 5-15 to reset the automatic line feed function. See also Figures 1-3 and 3-2 for possible ways to use ESC Z and ESC D commands without resetting this function. (If you are uncertain as to the default setting of this function, check the setting of DIP switch 1-8 as described in Chapter 2.)

The following line of Applesoft BASIC code sets the printer for automatic line feed insertion:

```
130 PRINT CHR$ (27);"D";CHR$ (128);CHR$ (0);
```

The following line of Applesoft BASIC code cancels automatic line feed insertion:

```
130 PRINT CHR$ (27);"Z";CHR$ (128);CHR$ (0);
```

The following line of Macintosh Pascal code sets the printer for automatic line feed insertion:

```
WRITE(Prntr,chr(27),'D',chr(128),chr(0));
```

Automatic Line Feed When Line Is Full

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

When the ImageWriter II receives enough text or graphics to fill an entire line, it automatically prints the line and executes a carriage return. When you turn the printer on, it is not set to insert a line feed character after an automatic carriage return. In this case, if you send a long string of characters to the printer without any line feeds, each line of text overprints the previous line. Use the commands in Table 5-16 to cause the printer to insert a line feed after an automatic carriage return when the line is full.

Table 5-16. Automatic Line Feed When Line Is Full Commands

Characters	Decimal	Hexadecimal	Function
ESC D SPACE CTRL-@	27 68 32 0	1B 44 20 00	Adds line feed when line is full
ESC Z SPACE CTRL-@	27 90 32 0	1B 5A 20 00	No line feed added when line is full (default)

SPACE: The space character

Applesoft BASIC

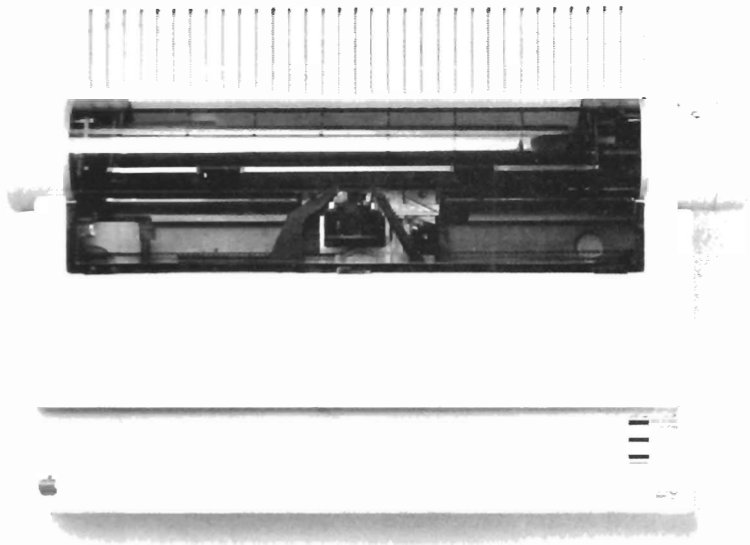
When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; see the Automatic Line Feed After Carriage Return section in this chapter for more information.

The following line of Applesoft BASIC code illustrates setting the printer to add a line feed when a full line of data has been received.

```
130 PRINT CHR$ (27);"D";CHR$ (32);CHR$ (0);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'D',chr(32),chr(0));
```



This chapter describes the use of several commands not discussed elsewhere. The functions controlled by these commands include:

- Determining whether LF and FF cause printing
- Character repetition
- Erasing the current line from the print buffer
- Selecting and deselecting the printer
- Ignoring or including the eighth data bit
- Software reset
- Software select and deselect
- Printer self-identification.

This chapter also provides programming examples in both Applesoft BASIC and Macintosh Pascal for most of the commands. If you need to use these commands with other languages or applications, see the examples in Chapter 1.

Print Commands

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

When the printer receives data, it stores it in a random-access memory (RAM), called the print buffer; no printing actually occurs until a full line of text is stored in the print buffer or a print command is received. The commands in Table 6-1 determine which commands cause the data in the print buffer to be printed when the command is executed. The default is for carriage returns (CR), line feeds (LF), and form feeds (FF) all to cause printing.

Table 6-1. Print Commands

Characters	Decimal	Hexadecimal	Function
ESC Z @ CTRL-@	27 90 64 0	1B 5A 40 00	Only CR causes printing
ESC D @ CTRL-@	27 68 64 0	1B 44 40 00	CR, LF, FF cause printing (default)

Important! When you select CR as the only command that causes printing, the LF and FF commands function only when used immediately following a carriage return character. CR insertion has no effect when the LF and FF commands are disabled from causing printing. See Chapter 5 for more information on CR insertion.

Applesoft BASIC

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

The following line of Applesoft BASIC code illustrates setting the printer so that only a carriage return causes printing.

```
130 PRINT CHR$ (27);"Z";CHR$ (64);CHR$ (0);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'Z',chr(64),chr(0));
```

Character Repetition

The character repetition command causes the Apple ImageWriter II to print the same character up to 999 times. This command can be used for printing a row of hyphens, for example.

The character repetition command is shown in Table 6-2.

Table 6-2. Character Repetition Command

Characters	Decimal	Hexadecimal	Function
ESC R *nnn c*	27 82 *ddd dc*	1B 52 *hhh hc*	Repeat character *c nnn* times

n: An ASCII numeral character from 0 to 9. The range of *nnn* is from 001 to 999.
d: The decimal equivalent of *n;* from 48 to 57.
h: The hexadecimal equivalent of *n;* from $30 to $39.
c: Any printable ASCII character that you wish to repeat *nnn* times.
dc: The decimal equivalent of *c.*
hc: The hexadecimal equivalent of *c.*

The number *nnn* in this command is a three-digit decimal number represented in ASCII numerals; it is sent one digit at a time to the printer. For example, to repeat the asterisk character 23 times, send the following characters to the printer:

```
ESC R 0 2 3 *
```

While *nnn* must always be three digits long, leading zeros may be replaced by space characters.

If *nnn* is so large that the repeated characters run beyond the end of the line, then the excess characters print over other characters on the same line unless the "automatic line feed when line is full" function is enabled (Chapter 5).

The example in Figures 6-1 and 6-2 prints a line consisting of 80 underline characters.

Figure 6-1. Applesoft BASIC Example of the Character Repetition Command

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 PRINT CHR$ (27);"R080_"
260 PRINT D$; "PR# 0"
270 END
```

Figure 6-2. Macintosh Pascal Example of the Character Repetition Command

```
PROGRAM Printer_Features;

VAR     Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  WRITELN(Prntr,chr(27);'R080_');
  CLOSE(Prntr)
END.
```

Cancel Current Line

As the Apple ImageWriter II receives data, it stores it in the print buffer until it can be printed. No text is printed until a print command (such as a carriage return) is received or the buffer contains a full line of text. If any line of text contains the "cancel current line" command shown in Table 6-3, that line is cancelled (that is, not printed). If there is more data in the print buffer following the cancelled line, it is not affected by this command.

Table 6-3. Cancel Current Line Command

Characters	Decimal	Hexadecimal	Function
CTRL-X	24	18	Erase current line from print buffer

CONTROL-X cancels all text and unexecuted commands in the next line of text to be printed (the "current line") in the print buffer. Since no command is read by the ImageWriter II until it is in the next line of text to be printed, the cancel current line command is not executed until the printer gets to the line that contains it. If a long queue of commands or data is being sent to the printer, it could take several minutes for the printer to receive and respond to the CONTROL-X. Only the line containing the CONTROL-X is cancelled; all lines before and after the current line are printed.

If you abort a program, or it crashes, the print buffer can be left with less than one line of text. This text is not printed until a print command (such as a carriage return) or enough additional text to complete the line is received, or until you press the SELECT button on the front panel. You can use the cancel current line command to purge the print buffer of this text without printing it, so you can resume your program or print something else. (If you were printing graphics when the program terminated, send 1280 null characters (CONTROL-@) before the CONTROL-X to assure that the printer does not interpret the command as graphics data.)

Ignore or Include Eighth Data Bit

When you turn on the ImageWriter II, it is set to ignore the eighth bit (the most significant bit) of each data byte. Although data is sent over the serial interface in 8-bit data bytes, normal ASCII characters use only the lower seven bits of each byte.

These commands can be combined with several others by setting the "software switches"; see Chapter 3 for details.

The ImageWriter II automatically changes to 8-bit mode for graphics data or when the escape-sequence command sent to it uses 8-bit data; for example, the soft switch commands and the commands for down loading a custom character set use 8-bit data. However, you might want to print 8-bit high-ASCII custom characters or MouseText characters on the ImageWriter II without using the ESC * (Chapter 7) or ESC & (Chapter 4) commands. Use the command in Table 6-4 to set the printer to recognize the eighth data bit.

Table 6-4. Data-Byte Length Commands

Characters	Decimal	Hexadecimal	Function
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit

SPACE: The space character

Applesoft BASIC

Applesoft BASIC always sets the eighth bit to 1 in every byte sent. If you use the ESC Z command in Table 6-4 to set the printer to include the eighth data bit, then every code sent to the printer using Applesoft BASIC will be read as high ASCII, and you will not be able to print ordinary characters until you return the printer to 7-bit mode. We recommend you use the ESC & command discussed in Chapter 4 to print MouseText characters and the ESC * command discussed in Chapter 7 to print custom characters. You do not have to enable 8-bit data to use the ESC & and ESC * commands. See also the Applesoft BASIC section in Chapter 1 for a routine you may be able to use to send 8-bit data to your printer.

When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

The following line of Applesoft BASIC code illustrates setting the printer to recognize the eighth data bit.

```
130 PRINT CHR$ (27);"Z";CHR$ (0);CHR$ (32);
```

Use the following line of code to achieve the same result with Macintosh Pascal:

```
WRITE(Prntr,chr(27),'Z',chr(0),chr(32));
```

Software Reset

When you turn the printer off and turn it back on, all of the software switches and commands are reset to their default conditions. However, when you turn the printer off, any data in the print buffer is lost along with any custom characters you have down-loaded. To reset the printer without turning it off, send the command in Table 6-5.

Table 6-5. Software Reset Command

Characters	Decimal	Hexadecimal	Function
ESC c	27 99	1B 63	Reset defaults

When you issue an ESC c control code, the printer prints all data in the print buffer entered prior to the ESC c, and resets all software-controllable operating instructions as shown in Appendix A, Table A-2.

Important!

The reset procedure may require up to three seconds to execute, as it involves moving the print head all the way to the left margin. Data sent immediately after an ESC c command will be lost if your program does not allow sufficient time for the reset to complete execution.

Since no command is read by the ImageWriter II until it is in the next line of text to be printed, the reset command is not executed until the printer gets to the line that contains it. If a long queue of commands or data is being sent to the printer, it could take several minutes for the printer to receive and respond to the reset command.

Software Select-Deselect

The Apple ImageWriter II can be selected or deselected with the front-panel SELECT button, or with the software command shown in Table 6-6. See the *ImageWriter II Owner's Manual* for more information on the SELECT button.

Whether the printer responds to the software select and deselect commands is determined by software switch A-5 (Chapter 3). The default is for the printer not to respond. To set this switch alone, use the commands in Table 6-6.

Table 6-6. Software-Select Response Commands

Characters	Decimal	Hexadecimal	Function
CTRL-Q	17	11	Select printer (default)
CTRL-S	19	13	Deselect printer
ESC D CTRL-P CTRL-@	27 68 16 0	1B 44 10 00	Software select disabled (default)
ESC Z CTRL-P CTRL-@	27 90 16 0	1B 5A 10 00	Software select enabled

Important! When you deselect the printer, either with the SELECT button or the command in Table 6-6, the DTR signal from the printer goes false. As long as this signal is false, your Apple computer will not send data to the printer, so you will be unable to send the CTRL-S command to the printer once it is deselected. These commands are provided for use with systems that can be set to ignore the DTR signal. For more information on the DTR signal, see Appendix E, Interface Specifications.

Applesoft BASIC When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

Self ID

The ImageWriter II can send a string of ASCII characters to your computer to indicate:

- An ImageWriter printer
- The carriage width
- Whether a color ribbon is in place
- Whether the SheetFeeder option is installed.

To cause the identification (ID) string to be sent, use the command in Table 6-7.

Table 6-7. Self ID Command

Characters	Decimal	Hexadecimal	Function
ESC ?	27 63	1B 3F	Send ID string

Communication protocols are discussed in Appendix E, Interface Specifications

The ID string sent by your ImageWriter II printer in response to this command consists of ASCII codes, sent at the current communication rate set by DIP switches 2-1 and 2-2, with the most significant (eighth) bit set to 0 ("Space parity"), one **start bit**, and one **stop bit**. The ASCII string is as follows:

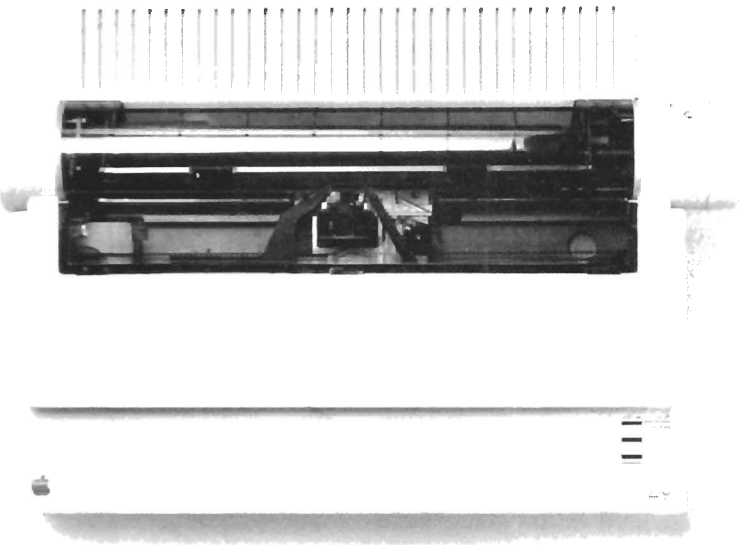
I W 1 0 C F

Where IW indicates an ImageWriter printer, 10 indicates a 10-inch carriage width, C is sent if a color ribbon is in place, and F is sent if the SheetFeeder is installed.

Important!

Since no command is read by the ImageWriter II until it is in the next line of text to be printed, the self ID command is not executed until the printer gets to the line that contains it. If a long queue of commands or data is being sent to the printer, it could take several minutes for the printer to receive and respond to the ESC ? command. If your program requires an immediate response to the self ID query, be sure to use the command only when the printer is idle.

For maximum control of timing and communications protocols, we recommend that you use the self ID command with assembly-language programs only.



In addition to the standard ASCII characters and MouseText characters shown in Appendix C, the Apple ImageWriter II contains a special memory that is capable of holding up to 175 custom-designed characters. Control codes allow you to switch the entire Apple character set from normal characters to new ones you have designed, and back again. You can easily mix custom symbols with ordinary text. Your custom-designed characters act just like the others: they expand and contract with changes in character pitch; you can underline them, print them in boldface, and expand them to double-width.

> *By the Way:* Unlike some other printers, loading custom characters in the ImageWriter II has no effect on the size of the print buffer available for printing.

Custom characters are available only in correspondence font. If you call custom characters while either the draft font or NLQ font is in effect, the printer switches to the correspondence font; it returns to draft or NLQ font when you return to printing standard characters.

This chapter explains how to create your own custom characters, how to load them into your Apple ImageWriter's memory, and how to fetch them from memory and print them. Each of these steps is illustrated with a specific example.

Creating Custom Characters

The ImageWriter II print head contains a vertical array of nine dot-strikers (called wires), spaced 1/72 of an inch apart. Normal characters are printed by striking up to seven eight-dot columns in a row, with the eighth dot column left blank to separate the characters. The fixed-width alphabets (pica, extended, condensed, and so on) print all characters eight dot-columns wide (including a one-dot space); the proportional alphabets vary the number of dot columns per character to fit the width of each letter or symbol. The difference between one character pitch and another is not in the number of dot columns per character, but in their horizontal spacing. Thus an ultracondensed *A* is printed with the same dot pattern as a pica *A*, but with 136 dots to the horizontal inch instead of 80.

To see how the standard fixed-width and proportional-width characters are designed, see Appendix C.

Custom characters can be up to 16 dots wide; you specify the width of each individual character, so you can make all your custom characters the same width, or can make their widths proportional. You can assign a custom character to any key that normally prints something, including uppercase and lowercase characters, punctuation, special symbols and the space bar.

The dot-striking wires in the print head are numbered from 1 at the top to 9 at the bottom. When designing a custom character you can use either the top eight wires (1 through 8) or the bottom eight wires (2 through 9). The normal alphabet uses wires 1 through 7 for capital letters and 3 through 7 for lowercase, with 8 and 9 being reserved for descenders, such as the tails on lowercase *y*, *g*, and *p*. Wire 9 is used for underlining as well. Thus the base line for normal printing is wire 7.

An example illustrating the process of designing and down loading a custom character is given in the last section of this chapter.

We suggest that you design each new custom character on square-ruled graph paper, which makes it easier for you to check its proportions and its vertical placement with respect to other characters. Plotting the dot pattern on graph paper also makes it easier to translate each column of dots into a data byte, as discussed below.

The data-byte calculation for custom characters is identical to that for graphics data as described in Chapter 8.

To send the character to the printer, you must translate each vertical column of eight dots into a binary number in which each bit set to 1 corresponds to a dot. The least significant bit (the last binary digit) of this number corresponds to the top of the character (wire 1 or 2, depending on whether the upper or lower eight wires are being used), and the most significant bit (the first digit) to the bottom. When you design the character, remember to include one or more columns of all zeros at its right edge to provide spacing between it and the character to follow. Figure 7-1 illustrates the correlation between dot positions and bit values.

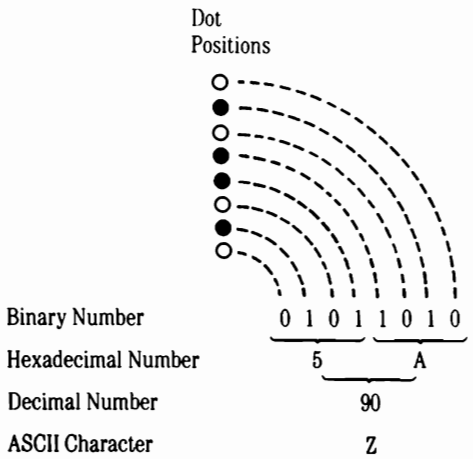
Figure 7-1. Custom-Character Data Byte

Bit	Dot Position	Binary Value	Decimal Value	Hexadecimal Value
0	●	00000001	1	$01
1	●	00000010	2	$02
2	●	00000100	4	$04
3	●	00001000	8	$08
4	●	00010000	16	$10
5	●	00100000	32	$20
6	●	01000000	64	$40
7	●	10000000	128	$80

ASCII equivalents of decimal and hexadecimal numbers are shown in Appendix B.

For each dot position that you want to print, use the value for that dot position shown in Figure 7-1. The sum of all the bit values for that dot pattern is the value of the data byte. For example, to print the column of dots shown in Figure 7-2, you would send a data byte with decimal value 90 (hexadecimal $5A).

Figure 7-2. Calculating a Custom-Character Data Byte



Bit	Dot Pattern	Binary Value	Decimal Value	Hexadecimal Value
0		00000000		
1	●	00000010	2	$02
2		00000000		
3	●	00001000	8	$08
4	●	00010000	16	$10
5		00000000		
6	●	01000000	64	$40
7		00000000		
Byte Value		**01011010**	**90**	**$5A**

Loading Custom Characters

When you have designed one or more new custom characters, you can load them into the memory of your Apple ImageWriter II using the codes in Table 7-1.

By the Way: Unlike some other printers, loading custom characters in the ImageWriter II has no effect on the size of the print buffer available for printing.

Table 7-1. Custom-Character Data-Loading Commands

Characters	Decimal	Hexadecimal	Function
ESC –	27 45	1B 2D	Maximum width is 8 dots (default)
ESC +	27 43	1B 2B	Maximum width is 16 dots
ESC I	27 73	1B 49	Starts loading new character(s)
CTRL-D	4	04	Ends loading of new character(s)

To load one or more new custom characters into memory, send one of the following sequences to the printer (each of these items is discussed in detail below):

ESC - ESC I *KEY1 WIDTH1 DATA1 ... KEYn WIDTHn DATAn...* **CTRL-D**

ESC + ESC I *KEY1 WIDTH1 DATA1 ... KEYn WIDTHn DATAn...* **CTRL-D**

▲Warning

When you send the ESC – or ESC + control codes to the ImageWriter II, any previously-loaded custom characters are erased from memory.

The custom characters you load into memory are saved only as long as the printer is on; when you turn the power off, the characters are lost. When you design custom characters, you should save them in a file on disk that can be loaded each time you turn the printer on.

The custom characters are *not* erased when you send a software reset command to the printer (Chapter 6).

The characters in the sequence used to load custom characters into memory have the following significance:

1. **ESC – or ESC +** Select a maximum width of 8 or 16 dots by sending the printer ESC – or ESC +, respectively. If you choose a maximum width of 8 dots, the memory can hold a maximum of 175 characters; if you choose a maximum width of 16 dots, the memory can hold a maximum of 95 characters. The maximum-width selection remains until it is superseded by the other control code. When you turn the printer on, it is set to 8-dot maximum mode.
2. **ESC I** Send the printer an ESC I to tell it you are starting a group of new character specifications.

3. For each new character, send the following information to the printer:

   a. *KEY* The regular character that is to be assigned a new symbol. This can be any character with ASCII value from 32 through 126, that is, any uppercase or lowercase letter, any numeral, or any punctuation mark, including the space character. It may not be RETURN, ESCAPE, TAB, or any other nonprinting character. If you select a maximum width of eight dots in step 1 above, you can also use any value from 160 to 239 (that is, you can assign it to a high-ASCII character), for a total of 175 custom characters. To use a high-ASCII character, you must use the Pascal CHR statement or its equivalent.

   b. *WIDTH* A width code from Table 7-2. If the new character is seven dots wide (including spacing) and you want it printed using wires 1 through 8, for example, send G; if it is 13 dots wide and uses wires 2 through 9, send m; and so on.

ASCII equivalents of decimal and hexadecimal numbers are shown in Appendix B.

   c. *DATA* As many 8-bit binary numbers (in the form of ASCII characters, Pascal CHR statements, or the equivalent) as were specified by the width code just sent. Each binary number specifies a one-column 8-bit dot pattern, from the left to the right. Bit 0 of each number corresponds to the top wire, and bit 7 corresponds to the bottom, as shown in Figure 7-1.

4. **CTRL-D** After repeating step 3 for each custom character, send the printer a CONTROL-D character (ASCII 4) to signal that the loading sequence is complete.

▲Warning

Be careful when sending this sequence to the printer. If a single character is omitted or wrong, the whole result can become garbled. We recommend you put your entire set of custom character sequences in a file that you can edit if you find mistakes.

Applesoft BASIC

The ESC - and ESC + commands automatically switch the ImageWriter II into 8-bit mode (see the Ignore or Include Eighth Data Bit section in Chapter 6), but Applesoft BASIC always sets the eighth bit to 1 in every byte sent. You can send custom characters to the printer using Applesoft BASIC if every data byte has a value of 128 or higher. Alternatively, you may be able to set your serial port to clear the eighth data bit to zero, as illustrated in Figure 3-2, or call a machine-language routine from your BASIC program that sends 8-bit data to the printer, as illustrated in Figure 1-3.

Table 7-2. Custom-Character Width Codes

Width Dots	Width Code Top 8 Wires	Decimal	Hexadecimal
1	A	65	41
2	B	66	42
3	C	67	43
4	D	68	44
5	E	69	45
6	F	70	46
7	G	71	47
8	H	72	48
9	I	73	49
10	J	74	4A
11	K	75	4B
12	L	76	4C
13	M	77	4D
14	N	78	4E
15	O	79	4F
16	P	80	50
Dots	**Bottom 8 Wires**		
1	a	97	61
2	b	98	62
3	c	99	63
4	d	100	64
5	e	101	65
6	f	102	66
7	g	103	67
8	h	104	68
9	i	105	69
10	j	106	6A
11	k	107	6B
12	l	108	6C
13	m	109	6D
14	n	110	6E
15	o	111	6F
16	p	112	70

For example, suppose you want to load a special character consisting of three columns of the pattern shown in Figure 7-2, followed by two blank columns for spacing; you want to assign it to the [A] key, and to print it using the top eight print wires. Since this character is five dot columns wide, you can use the 8-dot maximum-width command, ESC –. The width code for this character is *E* (Table 7-2). The binary number representing

this pattern, 01011010, corresponds to the ASCII character *Z*, while the ASCII character corresponding to a blank column is CTRL-@. To load this special character into memory, send the following character string to the printer:

```
ESC – ESC I A E Z Z Z CTRL-@ CTRL-@ CTRL-D
```

Printing Custom Characters

Once their specifications are stored in your Apple ImageWriter's memory, you can fetch and print custom characters at any time. The codes shown in Table 7-3 switch the entire keyboard of printing characters from one font to another.

Table 7-3. Custom-Character Printing Commands

Characters	Decimal	Hexadecimal	Function
ESC '	27 39	1B 27	Switches to custom character font
ESC *	27 42	1B 2A	Switches to custom character font (high ASCII values)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC $	27 36	1B 24	Switches back to standard font

Whenever one of these font commands appears in text being printed, the printer changes fonts and continues to print with that font until it receives one of the other codes or the printer is reset. To print a single custom symbol, send the printer ESC ' (or ESC *), the keyboard character to which the custom symbol was assigned, and ESC $.

The ESC ' control code permits you to change to a font containing up to 95 custom characters of any width and fetch them with ordinary keyboard characters, and to call up to 80 custom characters assigned to high-ASCII values. The ESC * control code allows you to call up to 80 high-ASCII custom characters using only standard ASCII characters in the control code sequence. ESC ' and ESC * have the following characteristics:

- **ESC '** fetches custom characters that were assigned to ordinary keyboard characters (that is, ASCII characters with decimal values 32 through 126—see Appendix B).

  ESC ' also fetches custom characters that were assigned to ASCII characters with decimal values 160 through 239. When you use ESC ' for this purpose, use the ESC Z command in Table 7-3 to set the printer to recognize the eighth data bit.

- **ESC *** lets you use standard ASCII characters (values 32 through 111 decimal) to fetch custom characters assigned to high-ASCII decimal values 160 through 239. To call a custom character when using the ESC * control code, subtract 128 decimal ($80 hexadecimal) from the ASCII value of the custom character and send the resulting standard ASCII character to the printer.

ASCII equivalents of decimal and hexadecimal numbers are shown in Appendix B.

For example, 193–128 = 65, which is the ASCII value for *A*; so to print a custom character that you loaded into memory at the high-ASCII value 193, send the following sequence of characters to the printer:

```
ESC * A
```

Applesoft BASIC

Applesoft BASIC always sets the eighth bit to 1 in every byte sent. If you use the ESC Z command in Table 7-3 to set the printer to include the eighth data bit, then every code sent to the printer using Applesoft BASIC will be read as high ASCII, and you will not be able to print low ASCII characters until you return the printer to 7-bit mode. We recommend you use only the ESC * command to call high- ASCII custom characters in Applesoft BASIC programs.

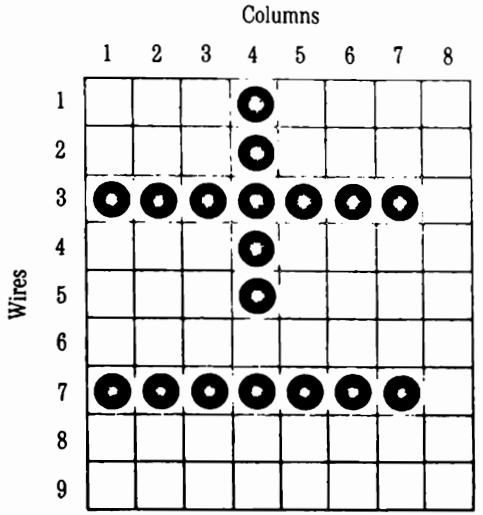
When you use an ESC Z command or an ESC D command with Applesoft BASIC, you may inadvertently change the setting of the "automatic line feed after carriage return" function; changing this function can cause too many or too few line feeds, depending on what other options are in effect. See the Automatic Line Feed After Carriage Return section in Chapter 5 for information on resetting this function.

A Sample Custom Character

Suppose you are using your Apple Imagewriter to print technical specifications in which the plus-or-minus sign (±) is used extensively. You could print this symbol by printing plus-backspace-underline, but the result would look awkward. A better solution is to use a custom character.

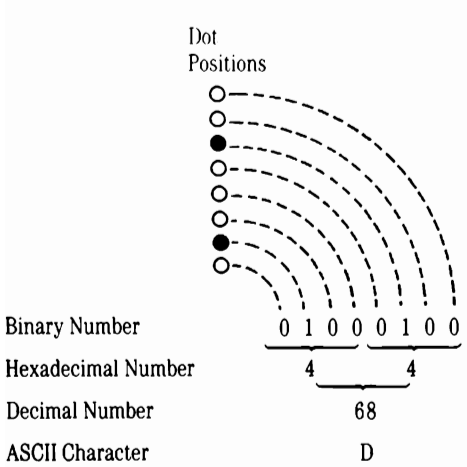
The first step is to design the new symbol on square-ruled paper, numbering the columns, as shown in Figure 7-3.

Figure 7-3. Designing a Custom Character

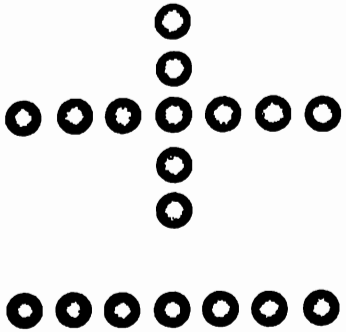


The next step is to translate the columns into binary numbers, and then into ASCII characters. The new symbol uses wires 1 through 8, so the least significant (or last) digit of each binary number corresponds to wire 1 and the most significant to wire 8. Figure 7-4 shows the calculation of a data byte, the binary number corresponding to each column in the custom character, and the equivalent ASCII characters.

Figure 7-4. Sample Custom-Character Data Calculation



Wires	Bits	Decimal Values
1	0	1
2	1	2
3	2	4
4	3	8
5	4	16
6	5	32
7	6	64
8	7	128



Data-Byte Values:

Decimal	68	68	68	95	68	68	68	0
Hexadecimal	$44	$44	$44	$5F	$44	$44	$44	$00
ASCII Character	D	D	D	_	D	D	D	CTRL-@

Since the new symbol is eight dots wide, you can enter it in the eight-dot maximum storage mode. Let's assume that you want to fetch it from the custom character font by using the & (ampersand) character. The resulting loading sequence is:

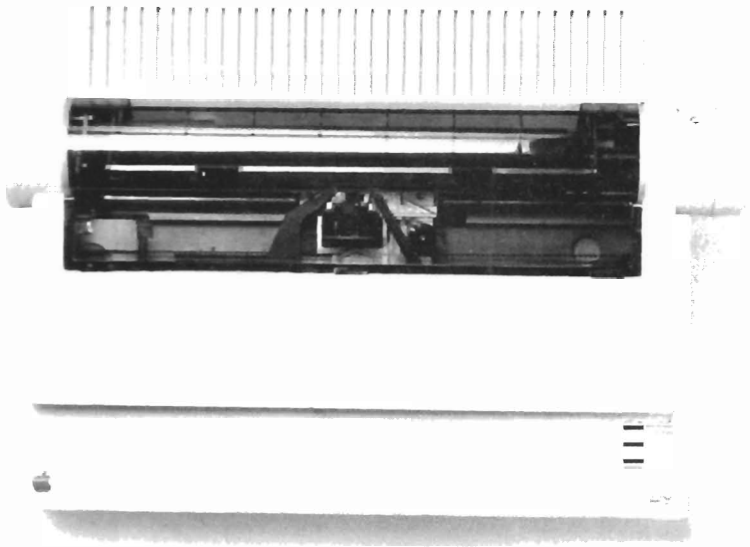
```
ESC - ESC I & H D D D _ D D D CTRL-@ CTRL-D
```

The character *H* is the width code. It indicates that the top eight wires print the symbol and that it is eight dots wide (see Table 7-2).

After you load the new character by means of the sequence above, you can print the character at any time by sending the following sequence to the printer:

```
ESC ' & ESC $
```

These five characters change the printer to the custom font, print the symbol assigned to the [&] key, and change the printer back to the standard font.



The Apple ImageWriter II can print graphics patterns in eight densities, ranging from 72 dots per inch (dpi) vertical by 72 dpi horizontal up to 72 dpi vertical by 160 dpi horizontal. A vertical resolution of 144 dpi is also possible with two-pass printing.

The Imagewriter II prints graphics by printing lines of vertical dot patterns, each eight dots high. Several features are supported by this printing method:

- You can quickly skip over blank spaces to start printing at any dot position on the page.
- You can individually control every dot on the page.
- You can create lines and horizontal stripes by repeating a single vertical dot pattern for up to an entire line.
- Each dot pattern can be printed in any of seven colors (plus white spaces); by combining colors you can create many more colors.

Bit Image Printing

This section describes how to create and print graphics patterns on your ImageWriter II.

The ImageWriter II print head contains a vertical array of nine dot-strikers (called wires), spaced 1/72 of an inch apart. Graphics patterns are printed by the upper eight wires, so for each horizontal print head position, a pattern of up to eight vertically-aligned dots can be printed. To print a graphics pattern, you must specify a byte of data for each 8-dot vertical pattern. A single line of graphics may require up to 1,280 bytes for its definition (8 inches x 160 dots/inch = 1280 columns). Each eight-dot high line of graphics is sent to the printer as a continuous stream of data, preceded by a command that puts the printer in graphics mode and specifies the number of data bytes for the line.

To help clarify the procedures for printing graphics with the ImageWriter II, this section is divided into several subsections, as follows:

- **Graphics Commands** describes the commands you can use to put the printer in graphics mode and specify the number of data bytes in the line.
- **Graphics Data** explains how to convert a graphics pattern into data bytes.

- **Dot Spacing** describes how to control the vertical and horizontal spacing of dots in a graphics printout, and explains how to create graphics with twice the normal vertical resolution (144 dots to the inch instead of 72 dots to the inch).
- **Generating a Graphics Pattern** shows you how to create a repeating graphics pattern without having to explicitly list every graphics byte in your program.
- **Dot-Column Repeating Graphics** describes a graphics command that repeats a single eight-dot pattern for as many columns as you specify.
- **Exact Print Head Placement** describes how to start graphics or text printing at any dot position on the page.

Graphics Commands

Graphics data is sent to the printer as a string of data bytes, each byte representing one column of up to eight dots. Up to one line's worth of data may be sent at one time; each data string must be preceded with one of the commands shown in Table 8-1.

Table 8-1. Graphics Data Prefixes

Characters	Decimal	Hexadecimal	Function
ESC G *nnnn*	27 71 *dddd*	1B 47 *hhhh*	Prints 8-dot high patterns corresponding to the following *nnnn* data bytes
ESC S *nnnn*	27 83 *dddd*	1B 53 *hhhh*	Same as ESC G
ESC g *nnn*	27 103 *ddd*	1B 67 *hhh*	Prints 8-dot high patterns corresponding to the following *nnn* x 8 data bytes

ESC S and ESC G are identical and may be used interchangeably.
n: An ASCII numeral character from 0 to 9. The maximum *nnnn* is shown in Table 8-2; the maximum *nnn* is shown in Table 8-3.
d: The decimal equivalent of *n;* each *d* can be from 48 to 57.
h: The hexadecimal equivalent of *n;* each *h* can be from $30 to $39.
Default: When the printer is reset, graphics printing is turned off.

The number *nnnn* in the ESC G and ESC S commands specifies the number of bytes of graphics data that follow the command. The number *nnn* in the ESC g command specifies the number of eight-byte groups of graphics data that follow the command; that is, multiply *nnn* times eight to calculate the

number of bytes of graphics data following the ESC g command. Each data byte defines a vertical column of eight dots printed by the top eight wires in the print head, as described in the Graphics Data section, below.

The number *nnnn* is a four-digit decimal number represented in ASCII numerals; *nnn* is a three-digit decimal number represented in ASCII numerals. Each command is sent one character at a time to the printer.

For example, to set the printer to graphics mode and send 232 graphics bytes, send the ASCII codes for one of the following character strings to the printer:

```
ESC G 0 2 3 2
```

```
ESC g 0 2 9
```

While *nnnn* must always be four digits long and *nnn* must always be three digits long, leading zeros may be replaced by space characters.

Up to one full line's worth of data bytes, each byte specifying an 8-dot vertical pattern, can be sent at one time. After the specified number of graphics data bytes have been received, the printer automatically returns to text mode until the next graphics command is received.

The ESC g command operates somewhat faster than the ESC G (or ESC S) command; if the number of graphics data bytes you are sending to the computer is evenly divisible by eight, use ESC g. For example, ESC G 0080 and ESC g 010 are equivalent.

The maximum number of dots that can be printed on a line depends on the horizontal resolution, which is determined by the character pitch in effect (Chapter 4). Table 8-2 specifies the maximum value for *nnnn* (maximum number of dots per line) and Table 8-3 shows the maximum value for *nnn* for each of the character pitch modes.

Table 8-2. Horizontal Resolution vs. Maximum Numbers of Dots per Line

Pitch Command	**Horizontal Density Dots per Inch**	**Dots per Line Maximum *nnnn***
ESC n (Extended)	72	0576
ESC N (Pica)	80	0640
ESC E (Elite)	96	0768
ESC e (Semicondensed)	107	0856
ESC q (Condensed)	120	0960
ESC Q (Ultracondensed)	136	1088
ESC p (Pica)	144	1152
ESC P (Elite)	160	1280

Table 8-3. Horizontal Resolution vs. Range for *nnn*

Pitch Command	Horizontal Density Dots per Inch	Range of *nnn*	Dots per Line
ESC n (Extended)	72	001 - 072	0008 - 0576
ESC N (Pica)	80	001 - 080	0008 - 0640
ESC E (Elite)	96	001 - 096	0008 - 0768
ESC e (Semicondensed)	107	001 - 107	0008 - 0856
ESC q (Condensed)	120	001 - 120	0008 - 0960
ESC Q (Ultracondensed)	136	001 - 136	0008 - 1048
ESC p (Pica)	144	001 - 144	0008 - 1152
ESC P (Elite)	160	001 - 160	0008 - 1280

By the Way: You can use double-width and boldface printing modes (Chapter 4) with any of the graphics modes. In double-width mode, the ImageWriter II prints two identical columns of dots for every byte of data (and therefore can accept only half the maximum number of data bytes per line shown in Tables 8-2 and 8-3). In boldface mode the ImageWriter II prints each dot in the graphics pattern twice with a small shift of position. The printer runs at half speed during boldface printing.

To use these printing modes with graphics, you must send the double-width or boldface command *before* entering graphics mode, or the command is interpreted as graphics data.

Graphics Data

The data-byte calculation for graphics data is identical to that for custom characters as described in Chapter 7.

Each data byte defines a vertical column of eight dots printed by the top eight wires in the print head. A dot is printed for each bit set to 1 in the data byte. Bit 7 causes the dot at the bottom of the column to be printed and bit 0 causes the dot at the top to be printed. Figure 8-1 illustrates the correlation between dot positions and bit values.

Figure 8-1. Graphics Data Byte

Bit Numbers	Dot Positions	Binary Values	Decimal Values	Hexadecimal Values
0	●	00000001	1	$01
1	●	00000010	2	$02
2	●	00000100	4	$04
3	●	00001000	8	$08
4	●	00010000	16	$10
5	●	00100000	32	$20
6	●	01000000	64	$40
7	●	10000000	128	$80

For each dot position that you want to print, use the value for that dot position shown in Figure 8-1. The sum of all the bit values for that dot pattern is the value of the data byte. For example, to print the column of dots shown in Figure 8-2, you would send a data byte with decimal value 90 (hexadecimal $5A). Note that the byte value is the equivalent of a binary number for which there is a 1 corresponding to each dot that is printed, and a 0 for each dot that is not printed.

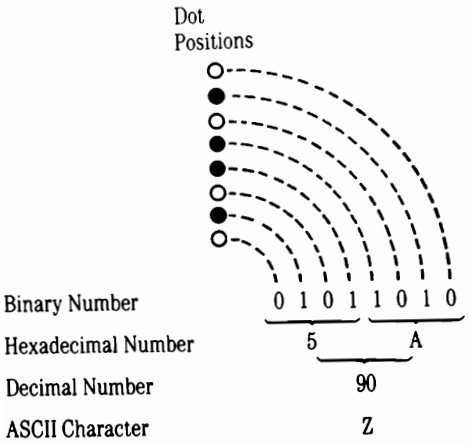
ASCII equivalents of decimal and hexadecimal numbers are shown in Appendix B.

Send the data to the printer as ASCII characters corresponding to the data byte values, or use the decimal values directly in the Pascal CHR statement or its equivalent. Examples of programs that send graphics data to the printer are shown in Figures 8-5, 8-7, and 8-9.

Applesoft BASIC

The ESC G, ESC S, and ESC g commands automatically switch the ImageWriter II into 8-bit mode (see the Ignore or Include Eighth Data Bit section in Chapter 6), but Applesoft BASIC always sets the eighth bit to 1 in every byte sent. If you can set your serial port to send a zero for the eighth data bit ("space parity"), you can still send a graphics pattern to the printer by using dot columns seven dots high (that is, no data byte can have a value higher than $7F). Remember to set the line feed to 7/72 inch to eliminate open spaces between subsequent lines. Figure 3-2 gives an example of an Applesoft BASIC program that sets the Apple Super Serial Card or the Apple IIc serial port to set the eighth data bit to zero.

Figure 8-2. Calculating a Graphics Data Byte



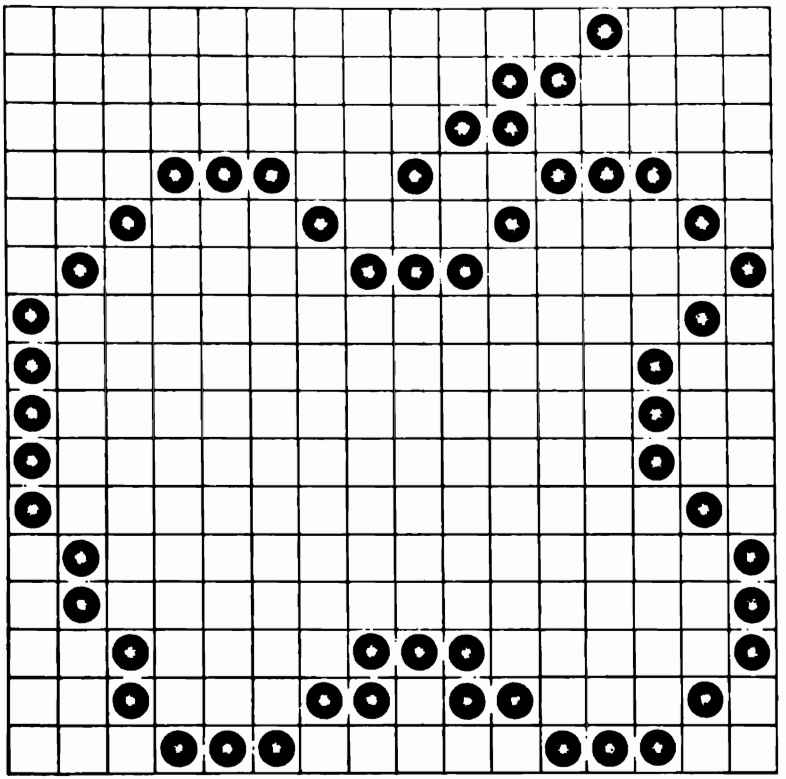
Bit Numbers	Dot Positions	Binary Values	Decimal Values	Hexadecimal Values
0		00000000	0	$00
1	●	00000010	2	$02
2		00000000	0	$00
3	●	00001000	8	$08
4	●	00010000	16	$10
5		00000000	0	$00
6	●	01000000	64	$40
7		00000000	0	$00
Byte Value Sum	**=**	**01011010**	**90**	**$5A**

The ASCII equivalent of decimal 90 is Z, so to print the column of dots shown in Figure 8-2, send the following character string to the printer:

```
ESC G 0 0 0 1 Z
```

Graphics dot patterns print from left to right, starting at the left margin of the page. The example in Figure 8-5 prints the pattern shown in Figure 8-3. Note that this pattern is more than eight dots high, and so requires two lines of data.

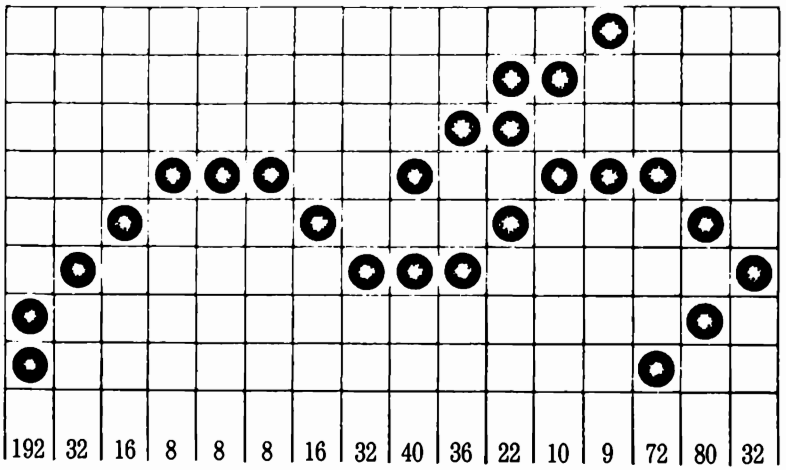
Figure 8-3. Graphics Example



To print this graphics pattern, first divide it into columns eight dots high, as shown in Figure 8-4, then determine the data-byte value of each column, as shown in the figure. It is helpful to plot the dots on graph paper if you are designing a graphics pattern by hand.

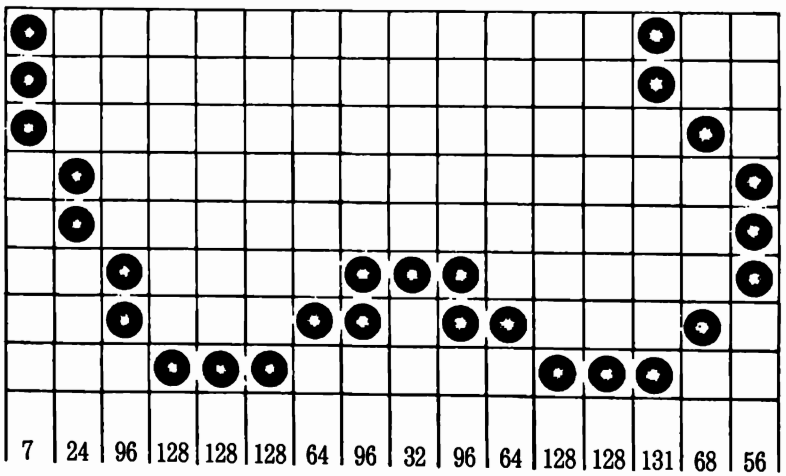
Figure 8-4. Graphics Example Converted to Data Bytes

Bit	Decimal Value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128



Data-Byte Value:	192	32	16	8	8	8	16	32	40	36	22	10	9	72	80	32

Bit	Decimal Value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128



Data-Byte Value:	7	24	96	128	128	128	64	96	32	96	64	128	128	131	68	56

The example in Figure 8-5 selects 72 dots per inch (dpi) horizontal resolution and 16/144-inch line feed (for 72 dpi vertical resolution), then prints the graphics pattern illustrated in Figure 8-3.

Figure 8-5. Macintosh Pascal Example of Printing Graphics

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  {Set character and line pitch for 72 per inch}
  WRITE (Prntr, CHR(27), 'n', CHR(27), 'T16');
  {Set Graphics Mode and width}
  WRITE (Prntr, CHR(27), 'G0016';
  {send data bytes for top section}
  WRITE (Prntr, CHR(192), CHR(32), CHR(16), CHR(8));
  WRITE (Prntr, CHR(8), CHR(8), CHR(16), CHR(32));
  WRITE (Prntr, CHR(40), CHR(36), CHR(22), CHR(10));
  WRITELN (Prntr, CHR(9), CHR(72), CHR(80), CHR(32));
  {Set Graphics Mode and width}
  WRITE (Prntr, CHR(27), 'G0016';
  {send data bytes for bottom section}
  WRITE (Prntr, CHR(7), CHR(24), CHR(96), CHR(128));
  WRITE (Prntr, CHR(128), CHR(128), CHR(64), CHR(96));
  WRITE (Prntr, CHR(32), CHR(96), CHR(64), CHR(128));
  WRITELN (Prntr, CHR(128), CHR(131), CHR(68), CHR(56));
  CLOSE(Prntr)
END.
```

Dot Spacing

The vertical spacing of the eight striker wires in the print head is 1/72 of an inch, so a vertical line feed of 8/72 inch (16/144 inch) maintains the dot spacing between subsequent lines. The "Extended" (9 characters per inch) horizontal pitch produces 72 dots per inch, so a line feed of 16/144 together with Extended character pitch results in a uniform matrix of dot positions, 72 per inch in each direction. The dot size is such that horizontal and vertical lines appear connected for this matrix.

To produce nine characters per inch horizontally and a line feed pitch of 16/144 of an inch, send the following characters to the printer:

```
ESC n ESC T 1 6
```

Because the minimum line feed is 1/144 inch, you can create graphics patterns with twice the density of the print head wire spacing. To generate double-density graphics, divide the pattern into vertical columns of 16 dots, then calculate the data byte value of every other dot in each column. Print a line including every even-numbered dot (starting with 0), then execute a 1/144-inch line feed, and print the odd-numbered dots. For example, to print the column of dots shown in Figure 8-6, first print a data byte with the decimal value 204, then execute a carriage return and 1/144-inch line feed, and print a data byte with the value 50, as shown in Figure 8-7.

Figure 8-6. Calculating High-Resolution Data Bytes

Bit Numbers	**Dot Positions**	**Decimal Values Even Bits**	**Decimal Values Odd Bits**
0			
1	●		
2			
3	●		2
4	●	4	
5			
6	●	8	
7			
8			
9	●		16
10			
11	●		32
12	●	64	
13			
14	●	128	
15			
Byte Value		**204**	**50**

Figure 8-7. Macintosh Pascal Example of Printing High-Density Data Bytes

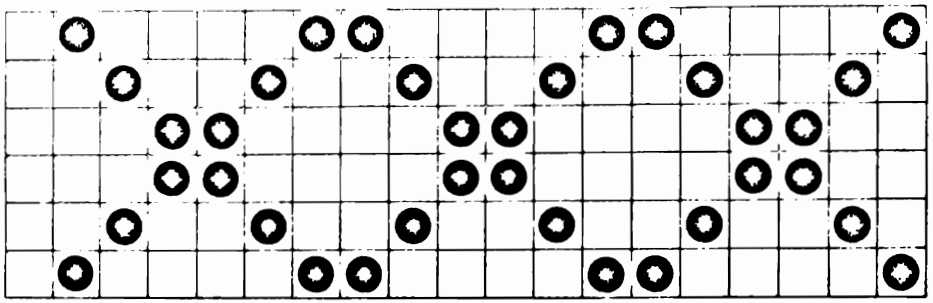
```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  {Set character and line pitch for 144 per inch}
  WRITE (Prntr, CHR(27), 'p', CHR(27), 'T01');
  {Set Graphics Mode and width}
  WRITE (Prntr, CHR(27), 'G0001';
  {send data byte for even bits}
  WRITELN (Prntr, CHR(204);
  {Set Graphics Mode and width}
  WRITE (Prntr, CHR(27), 'G0001';
  {send data byte for odd bits}
  WRITELN (Prntr, CHR(50);
  CLOSE(Prntr)
END.
```

Generating a Graphics Pattern

You can generate patterns by combining graphics commands with mathematical expressions and programming loops. For example, suppose you are using your Apple Imagewriter II to print custom business forms in which you want to crosshatch certain areas. The Pascal program in Figure 8-9 crosshatches blocks of a specified size in a specified location on the page.

The crosshatch pattern is generated from eight-column diamonds, as shown in Figure 8-8.

Figure 8-8. Crosshatch Graphics Pattern



The pattern shown in Figure 8-8 is generated by the following string of data-byte values:

129 66 36 24 24 36 66 129 129 66 ...

These values are calculated from the dot pattern as follows (see Figure 8-1):

$129 = 1 + 128 = 2^0 + 2^7$

$66 = 2 + 64 = 2^1 + 2^6$

$36 = 4 + 32 = 2^2 + 2^5$

$24 = 8 + 16 = 2^3 + 2^4$

$24 = 16 + 8 = 2^4 + 2^3$

$36 = 32 + 4 = 2^5 + 2^2$

$66 = 64 + 2 = 2^6 + 2^1$

$129 = 128 + 1 = 2^7 + 2^0$

$129 = 1 + 128 = 2^0 + 2^7$

$66 = 2 + 64 = 2^1 + 2^6$

•

•

•

You can generate this pattern by setting up a programming loop in which the data byte is calculated from the following formula, with the variable L varying from 0 to 7:

(Data byte value) = $2^L + 2^{7-L}$

The program in Figure 8-9 uses this formula to generate the crosshatching, and controls the size and location of the crosshatching with the following three parameters:

- LEFTEDGE: the character position of the left edge of the crosshatched block
- WIDTH: the width, in ninths of an inch, of the block
- HEIGHT: the height, in ninths of an inch, of the block

Figure 8-9. Macintosh Pascal Example: Generating a Graphics Pattern

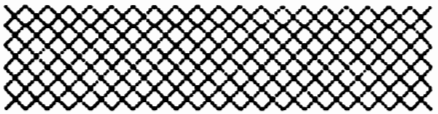
```
PROGRAM GRAPHICS;

VAR
 LEFTEDGE,WIDTH,HEIGHT : INTEGER;
 PRINTER : TEXT;

FUNCTION POWER(I : INTEGER) : INTEGER;
  {RETURN A POWER OF 2}
   VAR
    J,VALUE : INTEGER;
  BEGIN
    VALUE := 1;
    IF I=0 THEN
     POWER:=1
    ELSE
     BEGIN
      FOR J := 1 TO I DO
       BEGIN
        VALUE := 2*VALUE
       END;
      POWER := VALUE;
     END;
   END;

PROCEDURE HATCH;
 VAR
  N, M, L : INTEGER;
BEGIN
{Set left margin}
 WRITE(PRINTER, CHR(27), 'L', LEFTEDGE:3);
  {Set character and line pitch to 72 per inch}
 WRITE(PRINTER, CHR(27), 'n', CHR(27), 'T16');
  FOR N := 1 TO HEIGHT DO
   BEGIN
   {Set graphics mode and width}
    WRITE (PRINTER, CHR(27), 'G', (WIDTH * 8):4);
    FOR M := 1 TO WIDTH DO
     BEGIN
     {8 bytes to produce diamond pattern}
      FOR L := 0 TO 7 DO
       BEGIN
        WRITE(PRINTER, CHR(POWER(L)+POWER(7-L)));
       END;
     END;
    {Type each line when complete}
    WRITELN(PRINTER);
   END;
  END;
```

Figure 8-9. Macintosh Pascal Example: Generating a Graphics Pattern (Continued)

```
BEGIN
  REWRITE(PRINTER,'PRINTER:');
  WIDTH:=5;
  HEIGHT:=5;
  LEFTEDGE:=10;
  HATCH;
 END.
```

When you run the program in Figure 8-9, it prints the pattern shown in Figure 8-10.

Figure 8-10. Sample Graphics Printout



Dot-Column Repeating Graphics

The control code in Table 8-4 causes your Apple Imagewriter II to repeat a single vertical dot pattern for up to one line of graphics (1280 columns at 160 columns/inch). The result is a set of horizontal lines whose thickness and spacing depend on the dot pattern you specify.

Table 8-4. Dot-Column Repeating Graphics Commands

Characters	Decimal	Hexadecimal	Function
ESC V *nnnn c*	27 86 *dddd dc*	1B 56 *hhhh hc*	Prints nnnn repetitions of the dot column specified by *c*

n: An ASCII numeral character from 0 to 9. The maximum *nnnn* is shown in Table 8-2. *nnnn* must always have four digits; leading zeros can be replaced with spaces.

c: An ASCII character representing a data byte that specifies an eight-dot graphics pattern.

d: The decimal equivalent of *n*; each *d* can be from 48 to 57.

dc: The decimal equivalent of *c*.

h: The hexadecimal equivalent of *n;* each *h* can be from $30 to $39.

hc: The hexadecimal equivalent of *c*.

The character *c* is the ASCII equivalent of an 8-bit binary number specifying where dots appear in the column. The relationship between dot position and data value is shown in Figure 8-1; an example of a data byte is shown in Figure 8-2.

The dot column repeat command causes the data byte that follows to be printed as bit-image graphics. The number *nnnn* specifies the number of times this pattern is repeated.

For example, to print the pattern shown in Figure 8-11, send the following character string to the computer:

```
ESC V 0 0 0 5 K
```

The ASCII character K corresponds to decimal 75 or binary 01001011.

Figure 8-11. Dot-Column Repeating Graphics Example

Bit Numbers	Decimal Values	Dot Positions	Graphics Pattern
0	1	●	● ● ● ● ●
1	2	●	● ● ● ● ●
2			
3	8	●	● ● ● ● ●
4			
5			
6	64	●	● ● ● ● ●
7			
Data Byte Value	**75**		

The dot-column repeating graphics command is useful for generating horizontal rules for tables and forms. The length of the rule is controlled by the number of dots (*nnnn*).

Here are some sample values for the data byte *c*:

	ASCII	Dec	Hex	Binary MSB LSB
One hairline rule:	@	64	40	01000000
Two hairline rules:	H	72	48	01001000
Three hairline rules:	I	73	49	01001001
One heavy rule:	0	48	30	00110000
Two heavy rules:	f	102	66	01100110
One extra-heavy rule:	8	56	38	00111000

Exact Print Head Placement

Graphics or text printing can be started at any dot position on the page. To move the print head horizontally a specified number of dot columns, use the commands in Table 8-5.

Table 8-5. Print Head Placement Command

Characters	Decimal	Hexadecimal	Function
ESC F *nnnn*	27 70 *dddd*	1B 46 *hhhh*	Place print head *nnnn* dot columns from the left margin

n: An ASCII numeral character from 0 to 9. The range of *nnnn* depends on the character pitch; the maximum is shown in Table 8-2.

d: The decimal equivalent of *n;* each *d* can be from 48 to 57.

h: The hexadecimal equivalent of *n;* each *h* can be from $30 to $39.

The number *nnnn* in this command is a four-digit decimal number represented in ASCII numerals; it is sent one digit at a time to the printer. For example, to start printing at the twenty seventh dot column, send the following characters to the printer:

```
ESC F 0 0 2 7
```

While *nnnn* must always be four digits long, leading zeros may be replaced by space characters.

The left margin command is discussed in Chapter 5; character pitch is controlled by either a DIP switch, as discussed in Chapter 2, or by a control code, discussed in Chapter 4.

Printing begins *nnnn* dot columns from the left margin. The actual placement thus depends on the position of the left margin as set by the ESC L command and on the character pitch in effect. The first possible print position (left-most dot column) is *nnnn*=0000.

Follow each ESC F command with at least one text character, a graphics control code (such as ESC G or ESC V), or another ESC F control code.

Important! To use more than one ESC F command on a single line, send them to the printer in ascending numerical order of *nnnn* (that is, from left to right).

The example in Figures 8-12 and 8-13 selects 10 cpi character spacing, prints the words "Apple ImageWriter", then prints the words "Is Grrrreat" on the next line. The letters on the second line are positioned exactly half way between those on the first line. The results of this example are shown in Figure 8-14.

Figure 8-12. Applesoft BASIC Example of Print Head Placement

```
100 D$ = CHR$ (4)
120 PRINT D$; "PR#1"
130 REM  SELECT 10 CPI CHARACTER PITCH
140 PRINT CHR$(27);"N";
150 PRINT "Apple ImageWriter"
160 PRINT CHR$(27);"F0018";
170 PRINT "Is";
180 PRINT CHR$(27);"F0068";
190 PRINT "Grrrreat"
260 PRINT D$; "PR# 0"
270 END
```

Figure 8-13. Macintosh Pascal Example of Print Head Placement

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
   {select 10 cpi character pitch}
  WRITE(Prntr,chr(27),'N');
  WRITELN(Prntr,'Apple ImageWriter');
  WRITE(Prntr,chr(27),'F0018');
  WRITE(Prntr,'Is');
  WRITE(Prntr,chr(27),'F0068');
  WRITELN(Prntr,'Grrrreat');
  CLOSE(Prntr)
END.
```

Note in Figure 8-14 that the letters in the second line of text start at positions midway between the horizontal positions of the letters in the first line; the ESC F command has been used to reposition the starting print positions of the words "Is" and "Grrrreat."

Figure 8-14. Print Head Placement Printout Example

```
Apple ImageWriter
   is     Grrrreat
```

Color Printing

The Apple ImageWriter II can print text and graphics in color; the color option is activated automatically when you install a color ribbon. The color ribbon contains four bands of color: yellow, cyan (greenish-blue), magenta (purplish-red), and black. In addition, the ImageWriter II automatically prints orange, green, and purple by overprinting one color with another, as shown in Table 8-6.

To select the color to be printed, send one of the codes in Table 8-6 to the printer.

Table 8-6. Color Selection Commands

Characters	Decimal	Hexadecimal	Function
ESC K 0	27 75 48	1B 4B 30	Black (Default)
ESC K 1	27 75 49	1B 4B 31	Yellow
ESC K 2	27 75 50	1B 4B 32	Magenta
ESC K 3	27 75 51	1B 4B 33	Cyan
ESC K 4	27 75 52	1B 4B 34	Orange (Yellow & Magenta)
ESC K 5	27 75 53	1B 4B 35	Green (Yellow & Cyan)
ESC K 6	27 75 54	1B 4B 36	Purple (Magenta & Cyan)

> ***By the Way:*** Many printer manufacturers refer to the orange color created by overprinting of yellow and magenta as "red," and to the purple color created from magenta and cyan as "blue." Thus the orange, green, and purple discussed here correspond to the "red, green, and blue" referred to in other printer manuals.

Color mixing is done by overprinting one color on another. For example, to obtain a yellow-green color, print yellow, then overprint with green. To avoid contamination of the yellow band when printing combined colors, print yellow first followed by the other color or colors.

A large variety of colors and intensities can be achieved by using dot patterns to print blocks of colors. For example, a field of light blue can be created by printing patterns of four dots, alternating white (no dot) and cyan, as shown on the next page.

cyan white cyan white
white cyan white cyan
cyan white cyan white
white cyan white cyan

The most efficient and fastest way to print color patterns is to print an entire line of graphics in one color, followed by a carriage return with no line feed, then an entire line in the next color, and so on until all colors are printed. See the Bit Image Printing section in this chapter for information on printing graphics.

The example in Figure 8-15 prints a color rainbow by printing each of the three primary colors, with the color bands overlapping to print the other three colors. Figure 8-16 shows the sequence of color dots used to create this pattern.

Figure 8-15. Macintosh Pascal Example of Color Graphics Commands

```
PROGRAM Printer_Features;

VAR    Prntr:TEXT;

BEGIN
  REWRITE(Prntr,'PRINTER:');
  {Set character pitch for 72 per inch}
  WRITE (Prntr, CHR(27), 'n');
  {No linefeed after carriage return}
  WRITE (Prntr, CHR(27), 'Z', CHR(128), CHR(0));
  {Set line pitch to overlap last four dots}
  WRITE (Prntr, CHR(27), 'T8');
  {print yellow, then carriage return}
  WRITE (Prntr, CHR(27), 'K1');
  WRITE (Prntr, CHR(27), 'V0500', CHR(63));
  WRITE (Prntr, CHR(13));
  {print cyan, then carriage return and line feed}
  WRITE (Prntr, CHR(27), 'K3');
  WRITE (Prntr, CHR(27), 'V0500', CHR(3));
  WRITE (Prntr, CHR(13)), CHR(10));
  {print magenta, then carriage return}
  WRITE (Prntr, CHR(27), 'K2');
  WRITE (Prntr, CHR(27), 'V0500', CHR(63));
  WRITE (Prntr, CHR(13));
  {print cyan, then carriage return and
   two line feeds}
  WRITE (Prntr, CHR(27), 'K3');
  WRITELN (Prntr, CHR(27), 'V0500', CHR(240));
  WRITE (Prntr, CHR(13)), CHR(10)), CHR(10));
  CLOSE(Prntr)
END.
```

Figure 8-16. Color Graphics Example

First Pass: Print Yellow

Bit	Decimal Values	Dot Positions		Color
0	1	●		Yellow
1	2	●		Yellow
2	4	●		Yellow
3	8	●		Yellow
4	16	●		Yellow
5	32	●		Yellow
6	64			
7	128			

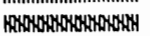
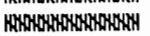
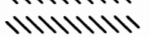
Data-Byte Value: 63

Second Pass: Print Cyan

Bit	Decimal Values	Dot Positions		Color
0	1	●		Green
1	2	●		Green
2	4			Yellow
3	8			Yellow
4	16			Yellow
5	32			Yellow
6	64			
7	128			

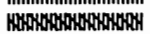
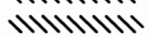
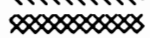
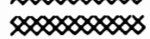
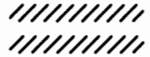
Data-Byte Value: 3

Third Pass: Line Feed, Print Magenta

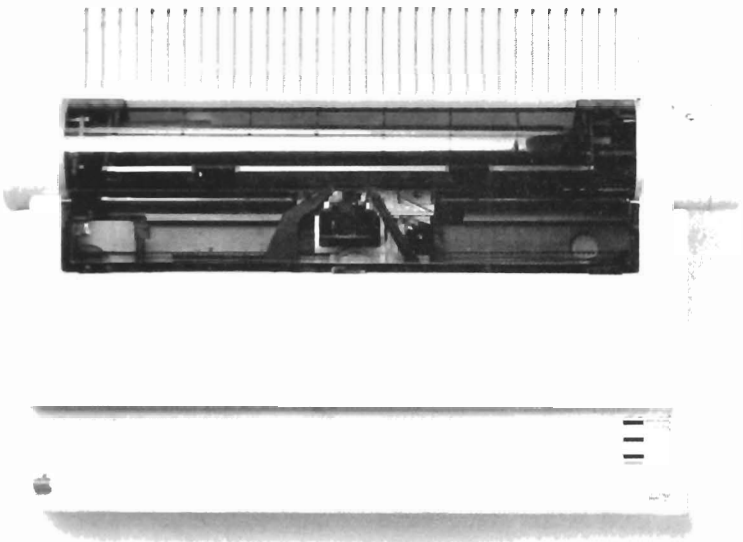
Bit	Decimal Values	Dot Positions		Color
.				Green
.				Green
.				Yellow
.				Yellow
0	1	●		Orange
1	2	●		Orange
2	4	●		Magenta
3	8	●		Magenta
4	16	●		Magenta
5	32	●		Magenta
6	64			
7	128			

Data-Byte Value: 63

Fourth Pass: Print Cyan

Bit	Decimal Values	Dot Positions		Color
.				Green
.				Green
.				Yellow
.				Yellow
0	1			Orange
1	2			Orange
2	4			Magenta
3	8			Magenta
4	16	●		Purple
5	32	●		Purple
6	64	●		Cyan
7	128	●		Cyan

Data-Byte Value: 240



Three options are available for the Apple ImageWriter II: the SheetFeeder, the 32K Memory option, and the AppleTalk option. The SheetFeeder allows you to print on letterhead or standard typing paper on the ImageWriter II without having to stop between pages to load a new sheet of paper. The 32K Memory option allows you to feed an entire document of 16 to 20 pages into the print buffer, then continue to use your computer while the document is printing. The AppleTalk option allows one ImageWriter to be shared by as many as 31 computers on an AppleTalk network. This chapter describes the way the installation of each of these options affects the use of the commands described in Chapters 3 through 8. See the manual that came with the option for information on installing and using the option.

Note: You cannot install both the 32K memory option and the AppleTalk option in the same ImageWriter II.

SheetFeeder

The Apple ImageWriter II SheetFeeder automatically feeds individual sheets of letterhead and other standard paper into the printer. All ImageWriter II commands and functions operate exactly as they do with continuous (fanfold) paper, except as noted in this section.

The SheetFeeder uses the paper-out sensor to monitor the flow of paper. The paper-out sensor has three distinct functions for the SheetFeeder:

1. It determines whether there is currently any paper in the printer. (If there is paper in the printer when it is turned on, the page is automatically ejected.) After a page has been ejected, a new sheet of paper is not fed in until the printer is ready to start printing; that is, until after a print command (Chapter 6) is received or there is a full line of text in the print buffer.
2. It determines where the top of the page is when a new sheet is fed into the printer.
3. It determines when the bottom of the page is reached. Paper is ejected when the print head position is approximately 7/144 inch from the bottom of the page.

▲Warning If you have left paper in the printer for some time (even an hour or so), it acquires a curl. In this case, the paper may not eject correctly and the SheetFeeder may jam. Do not leave any paper in the SheetFeeder when you turn it off; if anyone *has* left paper in the SheetFeeder, remove it before turning on the printer.

If the paper-out sensor is deactivated (Chapter 5), only the third function is affected. In this case, paper is ejected automatically only when the page length is exceeded as determined by DIP switch 1-4 (Chapter 2) or by a software command (Chapter 5); you can eject paper at any time by using the form feed command (Chapter 5) or the FORM FEED button on the printer.

If the perforation-skip function is enabled (via DIP switch 1-5, Chapter 2, or via software command, Chapter 5), then printing begins 1/2 inch below the top of the page (three lines at 6 lines per inch) and ends 1/2 inch above the bottom of the page. If the perforation-skip function is disabled, then the paper is fed to the top line on the page. Disable the perforation-skip function to use the ImageWriter II with programs that set the top margin of the page by skipping lines before printing. The accuracy of the page feed is plus or minus a half line, so it is possible that the first line of print will not fall entirely on the page unless at least one line feed is executed before printing begins.

If reverse line feeding (Chapter 5) is in effect while you are using the SheetFeeder, the paper is not scrolled more than 1/2 inch backwards. The printer keeps track of the total number of reverse and forward line feeds sent to it, however, and does not start to scroll the page forward until the net line feed motion is forward. In no case will the ImageWriter II scroll paper past the top of the form with the SheetFeeder installed; again, the printer keeps track of the line feeds sent to it, and does not respond to forward line feed commands until the net motion is forward.

The set top of form command (Chapter 5) is ignored when the SheetFeeder is in place; the SheetFeeder uses the paper-out sensor to determine the actual top of the page.

See the *ImageWriter II SheetFeeder Installation Manual* for more information on installing and using the SheetFeeder.

32K Memory

The 32K Memory option replaces the ImageWriter II's 2K print buffer with a 32K print buffer. This memory is large enough to hold an entire document of 16 to 20 pages. As a result, when you send a document to the printer, your computer sends the data as fast as the interface allows (see Appendix E); once the document is in the printer's memory, you can go back to work with your computer while the ImageWriter prints the document.

By the Way: Some computers or programs do not send data to the printer all at once, or at the maximum transmission rate, so your computer may not be immediately freed for other use even if the 32K Memory option is installed and enabled.

All of the ImageWriter commands and DIP switches operate with the 32K Memory option installed exactly as they do without it. The 32K memory is only active when DIP switch 2-4 is closed (on); you can use this DIP switch to enable or disable the memory option at any time once it has been installed.

When you turn on your ImageWriter II, it automatically checks the RAM circuit chips to make sure they are good. When only the 2K memory is enabled, this check is so fast that the printer appears to come on immediately; however, when the 32K memory option is enabled, there is a delay of about two seconds before the printer is ready to be used. If the memory is bad, the error light flashes on and off, and the printer can not be used. To determine if an error condition is caused by the 32K memory option, turn off (do *not* unplug) the printer, reset DIP switch SW 2-4 to open (off), and turn the printer back on. If the printer comes up normally, then there is something wrong with your 32K memory option—check with your authorized Apple service representative.

To find out if the 32K memory has been installed, run the ImageWriter II self test as described in your *ImageWriter II Owner's Manual*. The printout at the top of the test specifies "RAM = 32KB" if the 32K memory is installed and enabled.

Note that since no command is read by the ImageWriter II until the command is in the next line of text to be printed, there is no command that can be used to cancel printing when a long file has been sent to the 32K memory. The only way to cancel printing in this case is to turn off the printer.

▲Warning Turning off the ImageWriter II to cancel printing erases any custom characters from memory and resets all default values. (Defaults are listed in Appendix A.)

AppleTalk

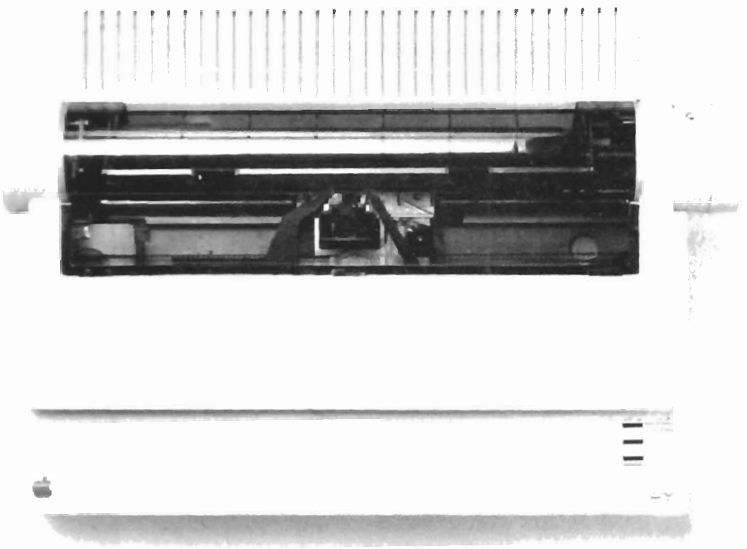
The AppleTalk™ option allows one ImageWriter II to be shared by as many as 31 computers on an AppleTalk network. The *ImageWriter II AppleTalk Option User's Manual* describes how to install the AppleTalk card in your ImageWriter II, how to connect your printer to the AppleTalk network, and how to name your printer for use on the AppleTalk network.

The only printer command you cannot use when you are using an ImageWriter II on an AppleTalk network is the self ID command (Chapter 6). Also, do not change the setting of the communication protocol DIP switch (SW 2-3, Chapter 2), as all communication is handled by AppleTalk. All other ImageWriter commands and DIP switches operate exactly as they do without AppleTalk.

To connect your ImageWriter II directly to a computer once the AppleTalk option has been installed, turn off (do *not* unplug) the printer, disconnect the AppleTalk system connector from the serial interface socket at the back of the printer, and reset DIP switch 2-4 to open (off). The AppleTalk option is then disabled, and you can connect the printer to a computer as described in the *ImageWriter II Owner's Manual* that came with your printer.

To return to the AppleTalk network, turn off (do *not* unplug) the printer, disconnect the printer from the computer, reset DIP switch 2-4 to closed, and reconnect the AppleTalk system connector. It is not necessary to rename the printer each time it is disconnected and then reconnected to AppleTalk in this fashion, but you should use the Chooser Desk Accessory to reselect the printer for use on AppleTalk. See the *ImageWriter II AppleTalk Option User's Manual* for more information on using AppleTalk with the ImageWriter.

Appendix A Command Summary



This appendix is a complete collection of all Apple ImageWriter II control codes and DIP switch settings.

Printer Defaults

The following are the settings for the various functions of the Apple ImageWriter in effect each time the printer is turned on.

Table A-1. Hard-Reset Defaults

Function	Default
Color Option	On if color ribbon installed, otherwise off
Communications rate	Per DIP switches 2-1, 2-2
Communications protocol	Per DIP switch 2-3
Option card enabled	Per DIP switch 2-4
Font	Draft
Character pitch	Per DIP switches 1-6 and 1-7
Proportional character spacing	0
Slashed-unslashed zeros	Unslashed
Select MouseText characters	Standard ASCII
Alternate language characters	Per DIP switches 1-1, 1-2, and 1-3
Underline text	Off
Boldface text	Off
Double-width printing	Off
Half-height text	Off
Subscript text	Off
Superscript text	Off
Left margin	0
Top of form	Current position of print head
Page length	Per DIP switch 1-4
Paper-out sensor	On
Line feed spacing	Six lines to the inch
Line feed direction	Forward
Carriage return insertion	On
Line feed after carriage return	Per DIP switch 1-8
Line feed when line is full	Off
Software select response	Disabled
Commands that cause printing	CR, LF, FF
Perforation skip	Per DIP switch 1-5
Include or ignore 8th bit	Ignore
Print head motion	Bidirectional
Horizontal tabs	Cleared

Table A-1. Hard-Reset Defaults (Continued)

Function	Default
Graphics printing	Off
Ribbon color	Black
Custom characters	Cleared

The following are the settings for the various software-controllable functions of the Apple ImageWriter in effect after the ESC c (software reset) command is sent to the printer:

Table A-2. Soft-Reset Defaults

Function	Default
Font	Draft
Character pitch	Per DIP switches 1-6 and 1-7
Proportional character spacing	0
Slashed-unslashed zeros	Unslashed
Select MouseText characters	Standard ASCII
Alternate language characters	Per DIP switches 1-1, 1-2, and 1-3
Underline text	Off
Boldface text	Off
Double-width printing	Off
Half-height text	Off
Subscript text	Off
Superscript text	Off
Left margin	0
Top of form	Unchanged
Page length	Per DIP switch 1-4
Paper-out sensor	On
Line feed spacing	Six lines to the inch
Line feed direction	Forward
Carriage return insertion	On
Line feed after carriage return	Per DIP switch 1-8
Line feed when line is full	Off
Software select response	Disabled
Commands that cause printing	CR, LF, FF
Perforation skip	Per DIP switch 1-5
Include or ignore 8th bit	Ignore
Print head motion	Bidirectional
Horizontal tabs	Cleared
Graphics printing	Off
Ribbon color	Black
Custom characters	Saved

All of the DIP switch settings for SW 1 can be overridden with software commands; there are no software commands corresponding to SW 2.

Table A-3. DIP Switch Settings

SW 1-1	SW 1-2	SW 1-3	Language
Open	Open	Open	American
Closed	Open	Open	Italian
Open	Closed	Open	Danish
Closed	Closed	Open	British
Open	Open	Closed	German
Closed	Open	Closed	Swedish
Open	Closed	Closed	French
Closed	Closed	Closed	Spanish

SW 1-4	Form Length
Open	11 Inches
Closed	12 Inches

SW 1-5	Perforation Skip
Open	Perforation skip inactive
Closed	Perforation skip active

SW 1-6	SW 1-7	Character Pitch
Open	Open	10 Characters per inch (Pica)
Closed	Open	12 Characters per inch (Elite)
Open	Closed	17 Characters per inch (Ultracondensed)
Closed	Closed	160 Dots per inch (Elite proportional)

SW 1-8	LF on CR
Open	No line feed is added after a carriage return
Closed	A line feed is added after receiving a carriage return

Table A-3. DIP Switch Settings (Continued)

SW 2-1	SW 2-2	Communication Rate
Open	Open	300 Baud
Closed	Open	1200 Baud
Open	Closed	2400 Baud
Closed	Closed	9600 Baud

SW 2-3	Protocol
Open	Hardware Handshake
Closed	XON/XOFF

SW 2-4	Option Card
Open	Option Disabled
Closed	Option Card Enabled

SW 2-5	SW 2-6	Hammer-Fire Timing
*	*	Factory setting of hammer-fire timing

* These two switches are factory-set to optimize bi-directional dot placement. Note how they are set and write the settings in the space above. **Do not change these settings.**

Software Switches

These commands are used to set several printer functions simultaneously. The use of the software switches is described in detail in Chapter 5.

Table A-4. Software Switch Commands

Characters	Decimal	Hexadecimal	Function
ESC Z *a b*	27 90 *da db*	1B 5A *ha hb*	Sets switches corresponding to the bit pattern of *a* and *b* to open (off)
ESC D *a b*	27 68 *da db*	1B 44 *ha hb*	Sets switches corresponding to the bit pattern of *a* and *b* to closed (on)

Table A-5. Software Switch Settings

SWA-1	SWA-2	SWA-3	Language
Open	Open	Open	American
Closed	Open	Open	Italian
Open	Closed	Open	Danish
Closed	Closed	Open	British
Open	Open	Closed	German
Closed	Open	Closed	Swedish
Open	Closed	Closed	French
Closed	Closed	Closed	Spanish

Table A-5. Software Switch Settings (Continued)

SW	Function	Z-Setting	D-Setting	Default
A-1	Language char set	SW 1-1 open	SW 1-1 closed	Per SW 1-1
A-2	Language char set	SW 1-2 open	SW 1-2 closed	Per SW 1-2
A-3	Language char set	SW 1-3 open	SW 1-3 closed	Per SW 1-3
A-4	Not used			
A-5	Soft select response	Enabled	Disabled	Disabled
A-6	LF when line full	No line feed	Add line feed	No line feed
A-7	Print commands	CR only	CR, LF, FF	CR, LF, FF
A-8	Auto LF after CR	CR only	CR plus LF	Per SW 1-8
B-1	Slash zero	Unslashed	Slashed	Unslashed
B-2	Not used			
B-3	Perforation skip	Skip	No skip	Per SW 1-5
B-4	Not used			
B-5	Not used			
B-6	Data byte length	8 Bits	7 Bits	7 Bits
B-7	Not used			
B-8	Not used			

Control Codes

The following tables list all of the control codes that can be used with the ImageWriter II printer.

Table A-6. Print Quality Control Codes

These commands select the print quality font to be used. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC a 0	27 97 48	1B 61 30	Select correspondence font
ESC a 1	27 97 49	1B 61 31	Select draft font (default)
ESC a 2	27 97 50	1B 61 32	Select NLQ font
ESC m	27 109	1B 6D	Same as the ESC a 0 command
ESC M	27 77	1B 4D	Same as the ESC a 2 command

Table A-7. MouseText Control Codes

These commands allow you to print MouseText characters. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC &	27 38	1B 26	Map MouseText to low ASCII ($40—$5F)
ESC $	27 36	1B 24	Select standard ASCII characters (default)
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit

Table A-8. Alternate Language Characters Control Codes

These commands select the language-character set to be used. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC Z CTRL-G CTRL-@	27 90 7 0	1B 5A 07 00	American
ESC Z CTRL-F CTRL-@ ESC D CTRL-A CTRL-@	27 90 6 0 27 68 1 0	1B 5A 06 00 1B 44 01 00	Italian
ESC Z CTRL-E CTRL-@ ESC D CTRL-B CTRL-@	27 90 5 0 27 68 2 0	1B 5A 05 00 1B 44 02 00	Danish
ESC Z CTRL-D CTRL-@ ESC D CTRL-C CTRL-@	27 90 4 0 27 68 3 0	1B 5A 04 00 1B 44 03 00	British
ESC Z CTRL-C CTRL-@ ESC D CTRL-D CTRL-@	27 90 3 0 27 68 4 0	1B 5A 03 00 1B 44 04 00	German
ESC Z CTRL-B CTRL-@ ESC D CTRL-E CTRL-@	27 90 2 0 27 68 5 0	1B 5A 02 00 1B 44 05 00	Swedish
ESC Z CTRL-A CTRL-@ ESC D CTRL-F CTRL-@	27 90 1 0 27 68 6 0	1B 5A 01 00 1B 44 06 00	French
ESC D CTRL-G CTRL-@	27 68 7 0	1B 44 07 00	Spanish

Table A-9. User-Designed Characters Control Codes

These commands control the loading of user-designed characters into the memory of the ImageWriter II, and print the characters. See Chapter 7.

Characters	Decimal	Hexadecimal	Function
ESC -	27 45	1B 2D	Maximum width of custom characters is 8 dots (default)
ESC +	27 43	1B 2B	Maximum width of custom characters is 16 dots
ESC I	27 73	1B 49	Starts loading new character(s)
CTRL-D	4	04	Ends loading of new character(s)
ESC '	27 39	1B 27	Switches to custom character font
ESC *	27 42	1B 2A	Switches to custom character font (high ASCII values)
ESC $	27 36	1B 24	Switches back to normal font

Table A-10. Character Pitch Control Codes

These commands control the horizontal size of characters. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC n	27 110	1B 6E	9 cpi (Extended)
ESC N	27 78	1B 4E	10 cpi (Pica)
ESC E	27 69	1B 45	12 cpi (Elite)
ESC e	27 101	1B 65	13.4 cpi (Semicondensed)
ESC q	27 113	1B 71	15 cpi (Condensed)
ESC Q	27 81	1B 51	17 cpi (Ultracondensed)
ESC p	27 112	1B 70	144 dpi (Pica)
ESC P	27 80	1B 50	160 dpi (Elite)
CTRL-N	14	0E	Start double width printing
CTRL-O	15	0F	Stop double width printing (default)

cpi: characters per inch
dpi: dots per inch

Table A-11. Proportional Character Spacing Control Codes

These commands control the spacing between proportionally-printed characters. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC s *n*	27 115 *dn*	1B 73 *hn*	Set dot spacing to *n* for all characters (*n* = 0 to 9)
ESC *n*	27 *d*	1B *h*	Insert *n* dot spaces (*n* = 1 to 6)

Table A-12. Character Attributes Control Codes

These commands affect the appearance of characters. See Chapter 4.

Characters	Decimal	Hexadecimal	Function
ESC X	27 88	1B 58	Start underline
ESC Y	27 89	1B 59	Stop underline (default)
ESC !	27 33	1B 21	Start boldface
ESC "	27 34	1B 22	Stop boldface (default)
CTRL-N	14	0E	Start double width printing
CTRL-O	15	0F	Stop double width printing (default)
ESC w	27 119	1B 77	Start half-height text
ESC W	27 87	1B 57	Stop half-height text (default)
ESC x	27 120	1B 78	Start superscript text
ESC y	27 121	1B 79	Start subscript text
ESC z	27 122	1B 7A	Stop either superscript or subscript (default)
ESC D CTRL-@ CTRL-A	27 68 0 1	1B 44 00 01	Zeros slashed
ESC Z CTRL-@ CTRL-A	27 90 0 1	1B 5A 00 01	Zeros unslashed (default)

Table A-13. Page Formatting Control Codes

These commands control the overall layout of the page. See Chapter 5.

Characters	Decimal	Hexadecimal	Function
ESC L *nnn*	27 76 *ddd*	1B 4C *hhh*	Set left margin at column *nnn* (default is *nnn* = 000)
ESC H *nnnn*	27 72 *dddd*	1B 48 *hhhh*	Set page length to *nnnn*/144 inch (*nnnn* = 0001 to 9999)

Table A-14. Print Head Motion Control Codes

These commands control the horizontal placement of text on the page. See Chapter 5.

Characters	Decimal	Hexadecimal	Function
CTRL-M	13	0D	Carriage Return
ESC >	27 62	1B 3E	Unidirectional printing
ESC <	27 60	1B 3C	Bidirectional printing (default)
CTRL-H	8	08	Back space
ESC (*aaa,bbb,...,nnn.*	27 40 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 28 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Sets horizontal tabs at positions *aaa,bbb,...,nnn*
ESC u *nnn*	27 117 *dndndn*	1B 75 *hnhnhn*	Adds one tab stop at column *nnn*
CTRL-I	9	09	Moves print head to next tab
ESC) *aaa,bbb,...,nnn.*	27 41 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 29 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Clears tabs at positions *aaa,bbb,...,nnn*
ESC 0	27 48	1B 30	Clears all tabs
ESC F *nnnn*	27 70 *dddd*	1B 46 *hhhh*	Place print head *nnnn* dot columns from the left margin

Table A-15. Paper Motion Control Codes

These commands control the motion of paper through the printer. See Chapter 5.

Characters	Decimal	Hexadecimal	Function
ESC v	27 118	1B 76	Sets TOF to current position
CTRL-L	12	0C	Feeds paper to the next TOF
CTRL-J	10	0A	Feeds paper one line (LF)
CTRL-_ *n*	31 *d*	1F *h*	Feeds 1 to 15 lines of blank paper (*n* = 1 to 9, :, ;, <, =, >, or ?)
ESC A	27 65	1B 41	6 lines per inch (default)
ESC B	27 66	1B 42	8 lines per inch
ESC T *nn*	27 84 *dd*	1B 54 *hh*	Distance between lines to be *nn*/144 inch (*nn* = 01 to 99)
ESC f	27 102	1B 66	Forward line feeding (default)
ESC r	27 114	1B 72	Reverse line feeding
ESC D CTRL-@ CTRL-D	27 68 0 4	1B 44 00 04	Perforation skip disabled
ESC Z CTRL-@ CTRL-D	27 90 0 4	1B 5A 00 04	Perforation skip enabled
ESC O	27 79	1B 4F	Paper-out sensor off
ESC o	27 111	1B 6F	Paper-out sensor on (default)

Table A-16. Automatic Carriage Returns and Line Feeds Control Codes

These commands enable or disable automatic insertion of carriage returns and line feeds. See Chapter 5.

Characters	Decimal	Hexadecimal	Function
ESC ! 1	27 108 49	1B 6C 31	No carriage return insertion before LF and FF
ESC ! 0	27 108 48	1B 6C 30	Insert CR before LF and FF (default)
ESC D {@} CTRL-@	27 68 128 0	1B 44 80 00	Adds automatic line feed after CR
ESC Z {@} CTRL-@	27 90 128 0	1B 5A 80 00	No line feed added after CR
ESC D SPACE CTRL-@	27 68 32 0	1B 44 20 00	Adds line feed when line is full
ESC Z SPACE CTRL-@	27 90 32 0	1B 5A 20 00	No line feed added when line is full (default)

{@}: A high-ASCII CONTROL-@ (decimal 128, hex $80)

Table A-17. Graphics Control Codes

These commands control the printing of graphics patterns. See Chapter 8.

Characters	Decimal	Hexadecimal	Function
ESC G *nnnn*	27 71 *dddd*	1B 47 *hhhh*	Prints line of graphics corresponding to the following *nnnn* data bytes
ESC S *nnnn*	27 83 *dddd*	1B 53 *hhhh*	Same as ESC G
ESC g *nnn*	27 103 *ddd*	1B 67 *hhh*	Prints line of graphics corresponding to the following *nnn* x 8 data bytes.
ESC V *nnnn c*	27 86 *dddd dc*	1B 56 *hhhh hc*	Prints nnnn repetitions of the dot column specified by *c*
ESC F *nnnn*	27 70 *dddd*	1B 46 *hhhh*	Place print head *nnnn* dot columns from the left margin
CTRL-N	14	0E	Start double width printing
CTRL-O	15	0F	Stop double width printing (default)

Table A-18. Color Printing Control Codes

These commands control the selection of a color when a color ribbon is installed. The ImageWriter II prints in Orange, Green, and Purple by printing first with one color, and then overprinting with another. See Chapter 8.

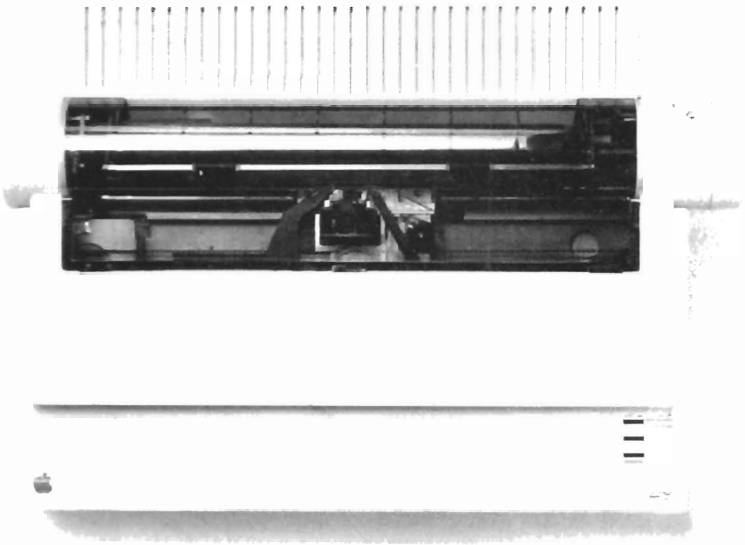
Characters	Decimal	Hexadecimal	Print Color
ESC K 0	27 75 48	1B 4B 30	Black (Default)
ESC K 1	27 75 49	1B 4B 31	Yellow
ESC K 2	27 75 50	1B 4B 32	Magenta
ESC K 3	27 75 51	1B 4B 33	Cyan
ESC K 4	27 75 52	1B 4B 34	Orange (Yellow & Magenta)
ESC K 5	27 75 53	1B 4B 35	Green (Yellow & Cyan)
ESC K 6	27 75 54	1B 4B 36	Purple (Magenta & Cyan)

Table A-19. Miscellaneous Commands

These commands control assorted printer functions not covered in the other categories in this appendix. See Chapter 6.

Characters	Decimal	Hexadecimal	Function
ESC Z @ CTRL-@	27 90 64 0	1B 5A 40 00	Only CR causes printing
ESC D @ CTRL-@	27 68 64 0	1B 44 40 00	CR, LF, FF cause printing (default)
ESC R *nnn c*	27 82 *ddd dc*	1B 52 *hhh hc*	Repeat character *c nnn* times (*nnn* = 001 to 999)
CTRL-X	24	18	Erase current line from print buffer
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit
ESC c	27 99	1B 63	Reset defaults
CTRL-Q	17	11	Select printer (default)
CTRL-S	19	13	Deselect printer
ESC D CTRL-P CTRL-@	27 68 16 0	1B 44 10 00	Software select disabled (default)
ESC Z CTRL-P CTRL-@	27 90 16 0	1B 5A 10 00	Software select enabled
ESC ?	27 63	1B 3F	Send ID string

Appendix B ASCII, Binary, and Hexadecimal Codes



There are 256 possible eight-bit binary numbers, from 00000000 to 11111111. Of these, the first 128 (from 00000000 to 01111111) have been assigned to a standard set of characters and commands used in data processing and communication. These assignments form the ASCII character set, where ASCII stands for the American Standard Code for Information Interchange.

The remaining 128 binary numbers, those for which the most significant bit (first digit) is 1 instead of 0, are not assigned in the ASCII standard. Because they have higher numerical values than the first 128 characters, they are often referred to as high-ASCII characters. In the ImageWriter II, the high-ASCII numbers are assigned to either the MouseText characters (see Appendix C) or to characters which you have designed yourself (see Chapter 8).

The following chart lists the 128 standard ASCII character assignments plus the high-ASCII equivalent for each number. Each number is given in binary, decimal, and hexadecimal forms. The first 27 characters can be sent from the keyboard by pressing CONTROL simultaneously with the desired character key; these are referred to as control characters.

Table B-1. ASCII, Binary, and Hexadecimal Codes

		Low ASCII			High ASCII		
	ASCII	**Dec**	**Hex**	**Binary**	**Dec**	**Hex**	**Binary**
CTRL-@	NUL	0	00	00000000	128	80	10000000
CTRL-A	SOH	1	01	00000001	129	81	10000001
CTRL-B	STX	2	02	00000010	130	82	10000010
CTRL-C	ETX	3	03	00000011	131	83	10000011
CTRL-D	EOT	4	04	00000100	132	84	10000100
CTRL-E	ENQ	5	05	00000101	133	85	10000101
CTRL-F	ACK	6	06	00000110	134	86	10000110
CTRL-G	BEL	7	07	00000111	135	87	10000111
CTRL-H	BS	8	08	00001000	136	88	10001000
CTRL-I	HT	9	09	00001001	137	89	10001001
CTRL-J	LF	10	0A	00001010	138	8A	10001010
CTRL-K	VT	11	0B	00001011	139	8B	10001011
CTRL-L	FF	12	0C	00001100	140	8C	10001100
CTRL-M	CR	13	0D	00001101	141	8D	10001101
CTRL-N	SO	14	0E	00001110	142	8E	10001110
CTRL-O	SI	15	0F	00001111	143	8F	10001111
CTRL-P	DLE	16	10	00010000	144	90	10010000
CTRL-Q	DC1	17	11	00010001	145	91	10010001

Table B-1. ASCII, Binary, and Hexadecimal Codes (Continued)

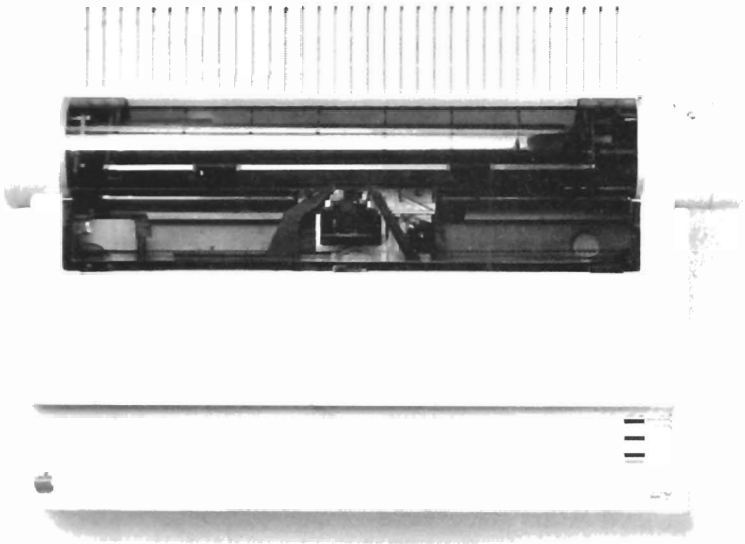
	Low ASCII				High ASCII		
	ASCII	**Dec**	**Hex**	**Binary**	**Dec**	**Hex**	**Binary**
CTRL-R	DC2	18	12	00010010	146	92	10010010
CTRL-S	DC3	19	13	00010011	147	93	10010011
CTRL-T	DC4	20	14	00010100	148	94	10010100
CTRL-U	NAK	21	15	00010101	149	95	10010101
CTRL-V	SYN	22	16	00010110	150	96	10010110
CTRL-W	ETB	23	17	00010111	151	97	10010111
CTRL-X	CAN	24	18	00011000	152	98	10011000
CTRL-Y	EM	25	19	00011001	153	99	10011001
CTRL-Z	SUB	26	1A	00011010	154	9A	10011010
CTRL-[	ESC	27	1B	00011011	155	9B	10011011
CTRL-\	FS	28	1C	00011100	156	9C	10011100
CTRL-]	GS	29	1D	00011101	157	9D	10011101
CTRL-^	RS	30	1E	00011110	158	9E	10011110
CTRL-_	US	31	1F	00011111	159	9F	10011111
SPACE		32	20	00100000	160	A0	10100000
	!	33	21	00100001	161	A1	10100001
	"	34	22	00100010	162	A2	10100010
	#	35	23	00100011	163	A3	10100011
	$	36	24	00100100	164	A4	10100100
	%	37	25	00100101	165	A5	10100101
	&	38	26	00100110	166	A6	10100110
	'	39	27	00100111	167	A7	10100111
	(	40	28	00101000	168	A8	10101000
	)	41	29	00101001	169	A9	10101001
	*	42	2A	00101010	170	AA	10101010
	+	43	2B	00101011	171	AB	10101011
	,	44	2C	00101100	172	AC	10101100
	-	45	2D	00101101	173	AD	10101101
	.	46	2E	00101110	174	AE	10101110
	/	47	2F	00101111	175	AF	10101111
	0	48	30	00110000	176	B0	10110000
	1	49	31	00110001	177	B1	10110001
	2	50	32	00110010	178	B2	10110010
	3	51	33	00110011	179	B3	10110011
	4	52	34	00110100	180	B4	10110100
	5	53	35	00110101	181	B5	10110101
	6	54	36	00110110	182	B6	10110110
	7	55	37	00110111	183	B7	10110111
	8	56	38	00111000	184	B8	10111000
	9	57	39	00111001	185	B9	10111001

Table B-1. ASCII, Binary, and Hexadecimal Codes (Continued)

Low ASCII					High ASCII		
ASCII	**Dec**	**Hex**	**Binary**	**Mouse Text**	**Dec**	**Hex**	**Binary**
:	58	3A	00111010		186	BA	10111010
;	59	3B	00111011		187	BB	10111011
<	60	3C	00111100		188	BC	10111100
=	61	3D	00111101		189	BD	10111101
>	62	3E	00111110		190	BE	10111110
?	63	3F	00111111		191	BF	10111111
@	64	40	01000000		192	C0	11000000
A	65	41	01000001		193	C1	11000001
B	66	42	01000010		194	C2	11000010
C	67	43	01000011		195	C3	11000011
D	68	44	01000100		196	C4	11000100
E	69	45	01000101		197	C5	11000101
F	70	46	01000110		198	C6	11000110
G	71	47	01000111		199	C7	11000111
H	72	48	01001000		200	C8	11001000
I	73	49	01001001		201	C9	11001001
J	74	4A	01001010		202	CA	11001010
K	75	4B	01001011		203	CB	11001011
L	76	4C	01001100		204	CC	11001100
M	77	4D	01001101		205	CD	11001101
N	78	4E	01001110		206	CE	11001110
O	79	4F	01001111		207	CF	11001111
P	80	50	01010000		208	D0	11010000
Q	81	51	01010001		209	D1	11010001
R	82	52	01010010		210	D2	11010010
S	83	53	01010011		211	D3	11010011
T	84	54	01010100		212	D4	11010100
U	85	55	01010101		213	D5	11010101
V	86	56	01010110		214	D6	11010110
W	87	57	01010111		215	D7	11010111
X	88	58	01011000		216	D8	11011000
Y	89	59	01011001		217	D9	11011001
Z	90	5A	01011010		218	DA	11011010
[	91	5B	01011011		219	DB	11011011
\	92	5C	01011100		220	DC	11011100
]	93	5D	01011101		221	DD	11011101
^	94	5E	01011110		222	DE	11011110
_	95	5F	01011111		223	DF	11011110
`	96	60	01100000		224	E0	11100000
a	97	61	01100001		225	E1	11100001
b	98	62	01100010		226	E2	11100010

	Low ASCII			High ASCII		
ASCII	**Dec**	**Hex**	**Binary**	**Dec**	**Hex**	**Binary**
c	99	63	01100011	227	E3	11100011
d	100	64	01100100	228	E4	11100100
e	101	65	01100101	229	E5	11100101
f	102	66	01100110	230	E6	11100110
g	103	67	01100111	231	E7	11100111
h	104	68	01101000	232	E8	11101000
i	105	69	01101001	233	E9	11101001
j	106	6A	01101010	234	EA	11101010
k	107	6B	01101011	235	EB	11101011
l	108	6C	01101100	236	EC	11101100
m	109	6D	01101101	237	ED	11101101
n	110	6E	01101110	238	EE	11101110
o	111	6F	01101111	239	EF	11101111
p	112	70	01110000	240	F0	11110000
q	113	71	01110001	241	F1	11110001
r	114	72	01110010	242	F2	11110010
s	115	73	01110011	243	F3	11110011
t	116	74	01110100	244	F4	11110100
u	117	75	01110101	245	F5	11110101
v	118	76	01110110	246	F6	11110110
w	119	77	01110111	247	F7	11110111
x	120	78	01111000	248	F8	11111000
y	121	79	01111001	249	F9	11111001
z	122	7A	01111010	250	FA	11111010
{	123	7B	01111011	251	FB	11111011
|	124	7C	01111100	252	FC	11111100
}	125	7D	01111101	253	FD	11111101
~	126	7E	01111110	254	FE	11111110
DEL	127	7F	01111111	255	FF	11111111

Appendix C Character Specifications



This appendix shows the dot patterns used to print all of the characters that are stored in the ImageWriter's permanent memory. All ImageWriter II fonts are shown, including:

- Draft
- Correspondence fixed width
- Correspondence proportional
- NLQ fixed width
- NLQ proportional

The ASCII decimal and hexadecimal codes for each character are shown at the character's upper left corner.

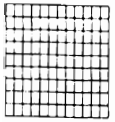
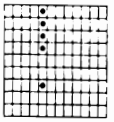
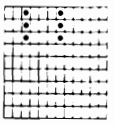
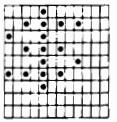
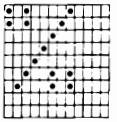
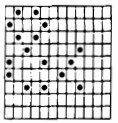
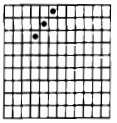
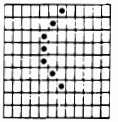
Draft-Quality Characters

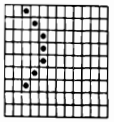
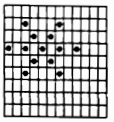
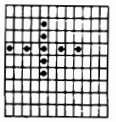
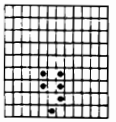
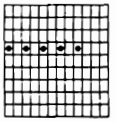
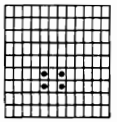
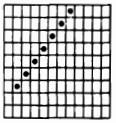
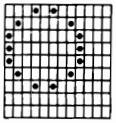
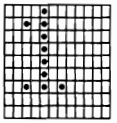
All draft-quality characters consist of twelve vertical columns, each nine dots high. The right-most column never contains dots, and the bottom two dots in each column are used only by lowercase characters with descenders, and by some punctuation. No horizontal row contains dots in two adjacent columns.

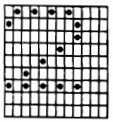
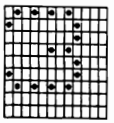
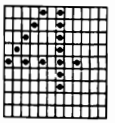
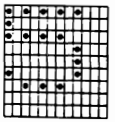
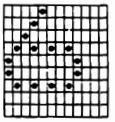
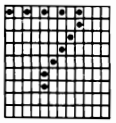
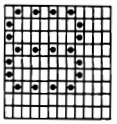
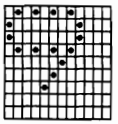
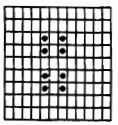
The decimal and hexadecimal codes for each character are shown at its upper left. Draft-quality characters include:

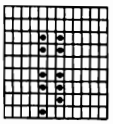
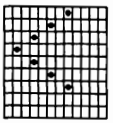
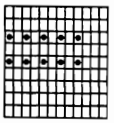
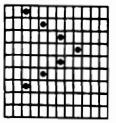
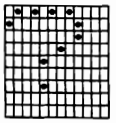
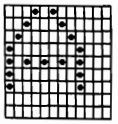
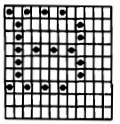
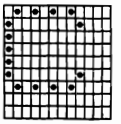
- Fixed-width standard ASCII
- Fixed-width MouseText
- Fixed-width alternate language

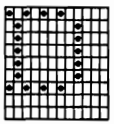
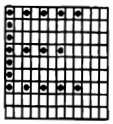
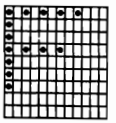
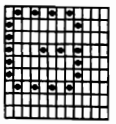
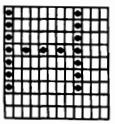
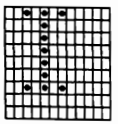
Each of these character sets is shown in this section.

32/$20	33/$21	34/$22	35/$23	36/$24	37/$25	38/$26	39/$27	40/$28
			See Table C-3					

41/$29	42/$2A	43/$2B	44/$2C	45/$2D	46/$2E	47/$2F	48/$30	49/$31
								

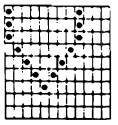
50/$32	51/$33	52/$34	53/$35	54/$36	55/$37	56/$38	57/$39	58/$3A
								

59/$3B	60/$3C	61/$3D	62/$3E	63/$3F	64/$40	65/$41	66/$42	67/$43
					See Table C-3			

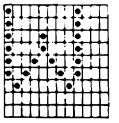
68/$44	69/$45	70/$46	71/$47	72/$48	73/$49	74/$4A	75/$4B	76/$4C
								

77/$4D	78/$4E	79/$4F	80/$50	81/$51	82/$52	83/$53	84/$54	85/$55

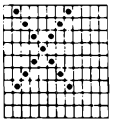
86/$56



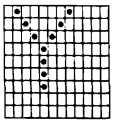
87/$57



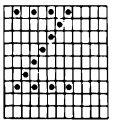
88/$58



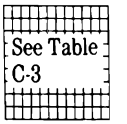
89/$59



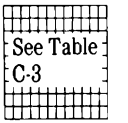
90/$5A



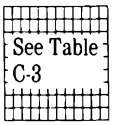
91/$5B



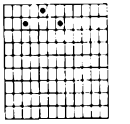
92/$5C



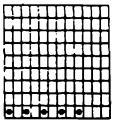
93/$5D



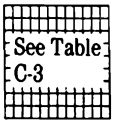
94/$5E



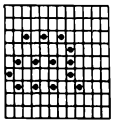
95/$5F



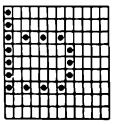
96/$60



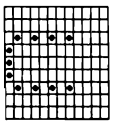
97/$61



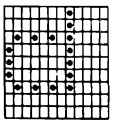
98/$62



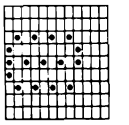
99/$63



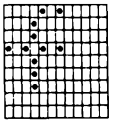
100/$64



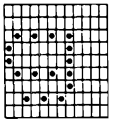
101/$65



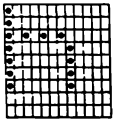
102/$66



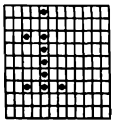
103/$67



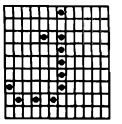
104/$68



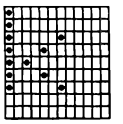
105/$69



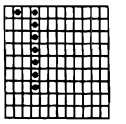
106/$6A



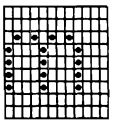
107/$6B



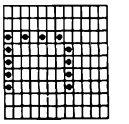
108/$6C



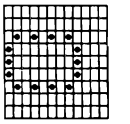
109/$6D



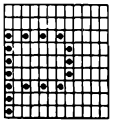
110/$6E



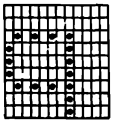
111/$6F



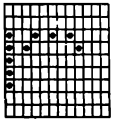
112/$70



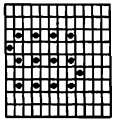
113/$71



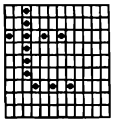
114/$72



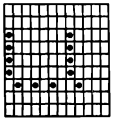
115/$73



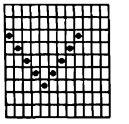
116/$74



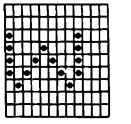
117/$75



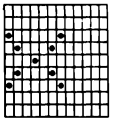
118/$76



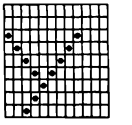
119/$77



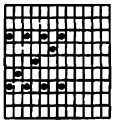
120/$78



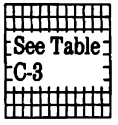
121/$79



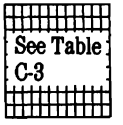
122/$7A



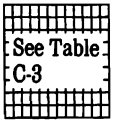
123/$7B



124/$7C

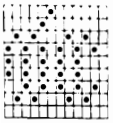


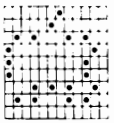
125/$7D

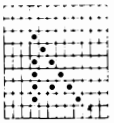


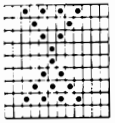
126/$7E

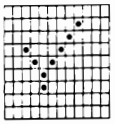
Table C-2. Draft-Quality Mousetext Characters

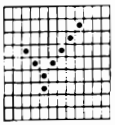
192/$C0 

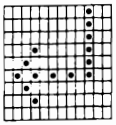
193/$C1 

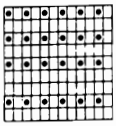
194/$C2 

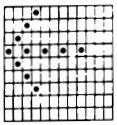
195/$C3 

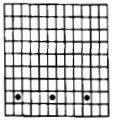
196/$C4 

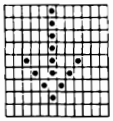
197/$C5 

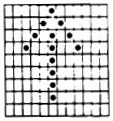
198/$C6 

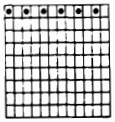
199/$C7 

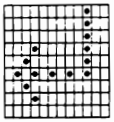
200/$C8 

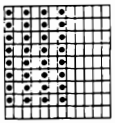
201/$C9 

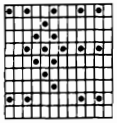
202/$CA 

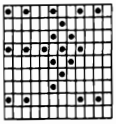
203/$CB 

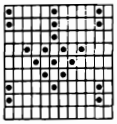
204/$CC 

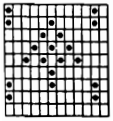
205/$CD 

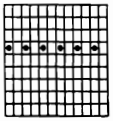
206/$CE 

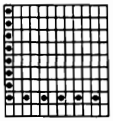
207/$CF 

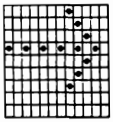
208/$D0 

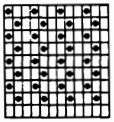
209/$D1 

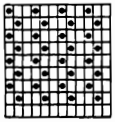
210/$D2 

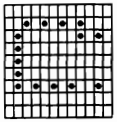
211/$D3 

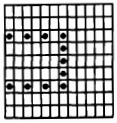
212/$D4 

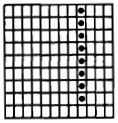
213/$D5 

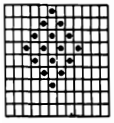
214/$D6 

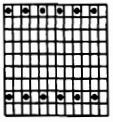
215/$D7 

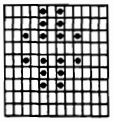
216/$D8 

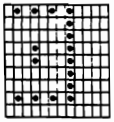
217/$D9 

218/$DA 

219/$DB 

220/$DC 

221/$DD 

222/$DE 

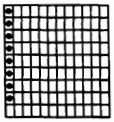
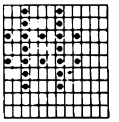
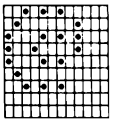
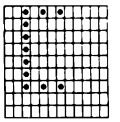
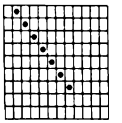
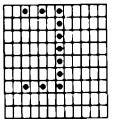
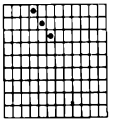
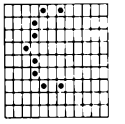
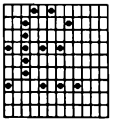
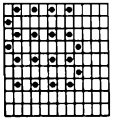
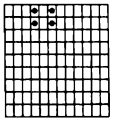
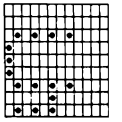
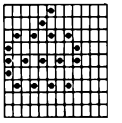
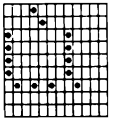
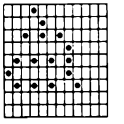
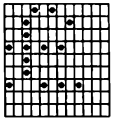
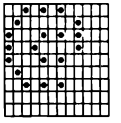
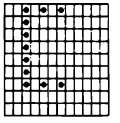
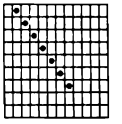
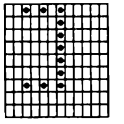
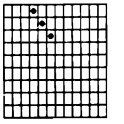
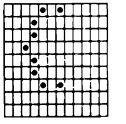
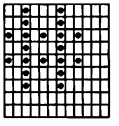
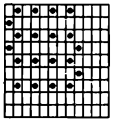
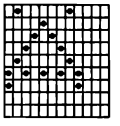
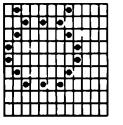
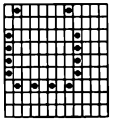
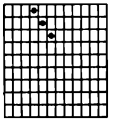
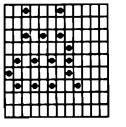
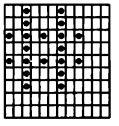
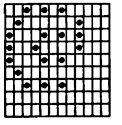
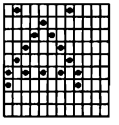
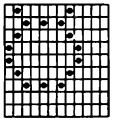
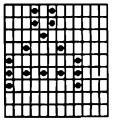
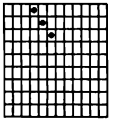
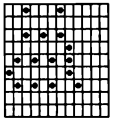
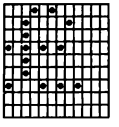
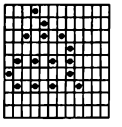
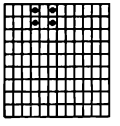
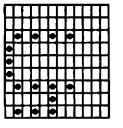
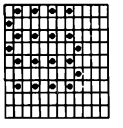
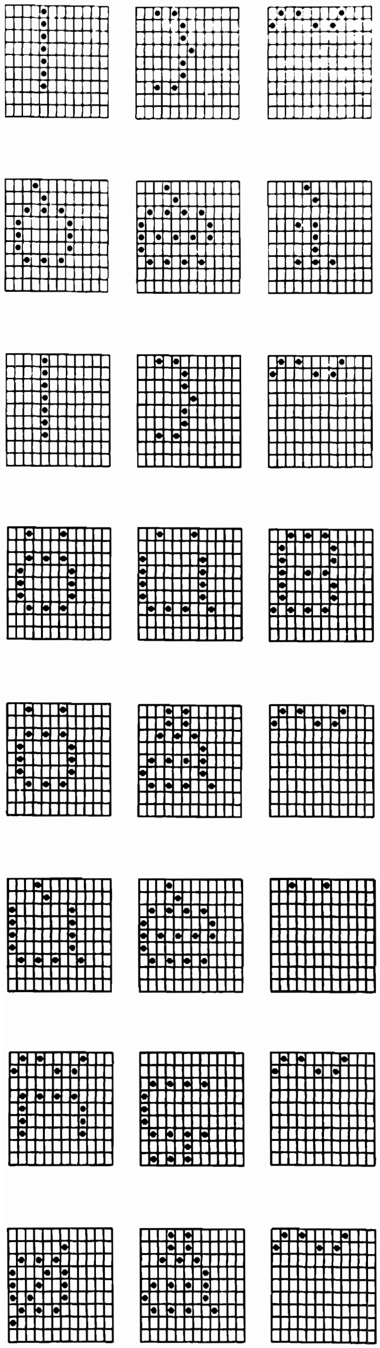
223/$DF 

Table C-3. Draft-Quality Alternate-Language Characters

	35/$23	64/$40	91/$5B	92/$5C	93/$5D	96/$60	123/$7B
ENGLISH (USA)							
ITALIAN							
ENGLISH (UK)							
GERMAN							
SWEDISH							
FRENCH							
SPANISH							
DANISH							

124/$7C **125/$7D** **126/$7E**



Correspondence-Quality Characters

Correspondence-quality characters include:

- Fixed-width standard ASCII
- Fixed-width MouseText
- Fixed-width alternate language
- Proportional standard ASCII
- Proportional alternate language

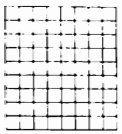
Each of these character sets is shown in this section.

Correspondence-Quality Fixed-Width Characters

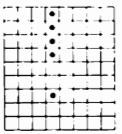
All correspondence-quality fixed-width characters consist of eight vertical columns, each nine dots high. The right-most column never contains dots, and the bottom two dots in each column are used only by lowercase characters with descenders, and by some punctuation.

The decimal and hexadecimal codes for each character are shown at its upper left.

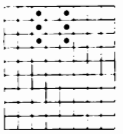
32/$20



33/$21



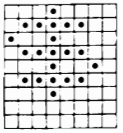
34/$22



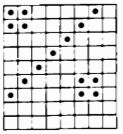
35/$23

See Table C-6

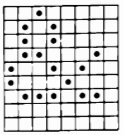
36/$24



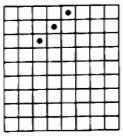
37/$25



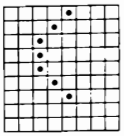
38/$26



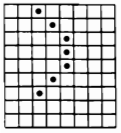
39/$27



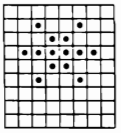
40/$28



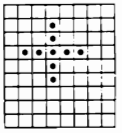
41/$29



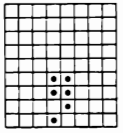
42/$2A



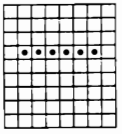
43/$2B



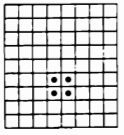
44/$2C



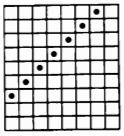
45/$2D



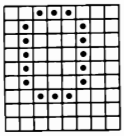
46/$2E



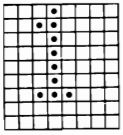
47/$2F



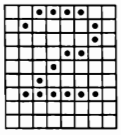
48/$30



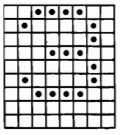
49/$31



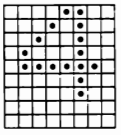
50/$32



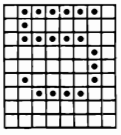
51/$33



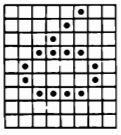
52/$34



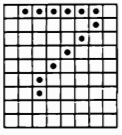
53/$35



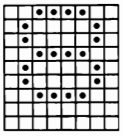
54/$36



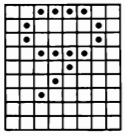
55/$37



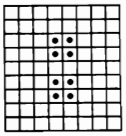
56/$38



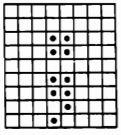
57/$39



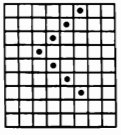
58/$3A



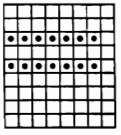
59/$3B



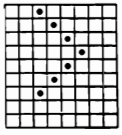
60/$3C



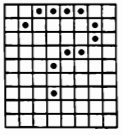
61/$3D



62/$3E



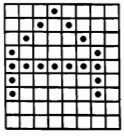
63/$3F



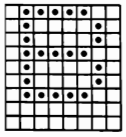
64/$40

See Table C-6

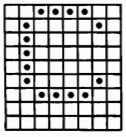
65/$41



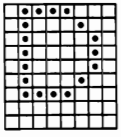
66/$42



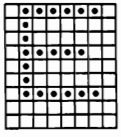
67/$43



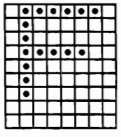
68/$44



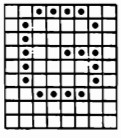
69/$45



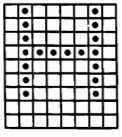
70/$46



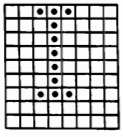
71/$47



72/$48



73/$49



74/$4A

75/$4B

76/$4C

77/$4D

78/$4E

79/$4F

80/$50

81/$51

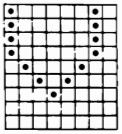
82/$52

83/$53

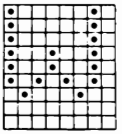
84/$54

85/$55

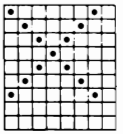
86/$56



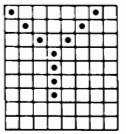
87/$57



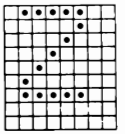
88/$58



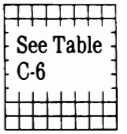
89/$59



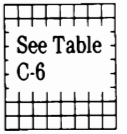
90/$5A



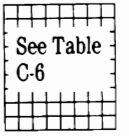
91/$5B



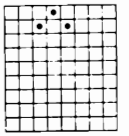
92/$5C



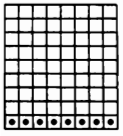
93/$5D



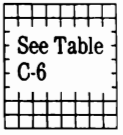
94/$5E



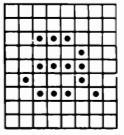
95/$5F



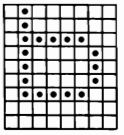
96/$60



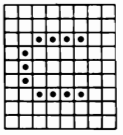
97/$61



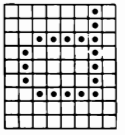
98/$62



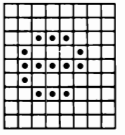
99/$63



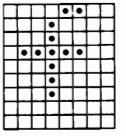
100/$64



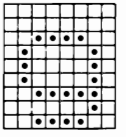
101/$65



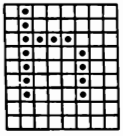
102/$66



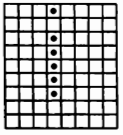
103/$67



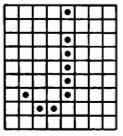
104/$68



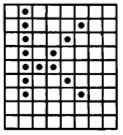
105/$69



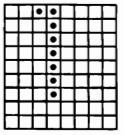
106/$6A



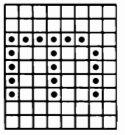
107/$6B



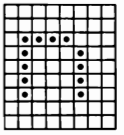
108/$6C



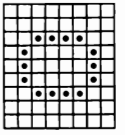
109/$6D



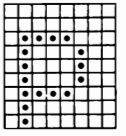
110/$6E



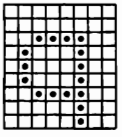
111/$6F



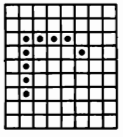
112/$70



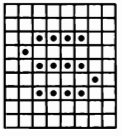
113/$71



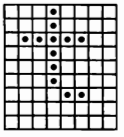
114/$72



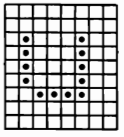
115/$73



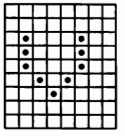
116/$74



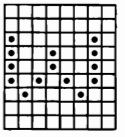
117/$75



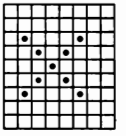
118/$76



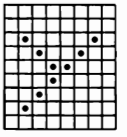
119/$77



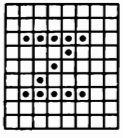
120/$78



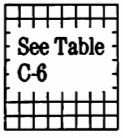
121/$79



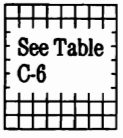
122/$7A



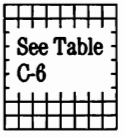
123/$7B



124/$7C



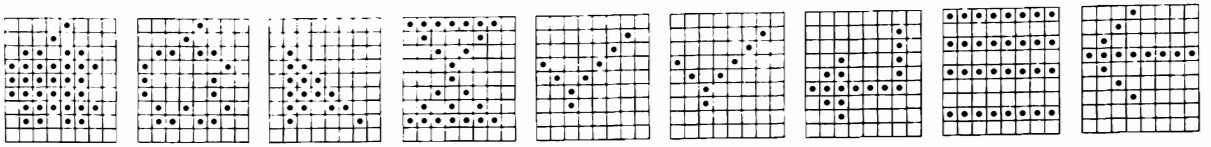
125/$7D



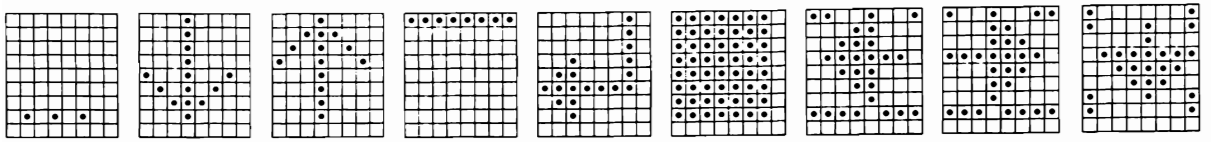
126/$7E

Table C-5. Correspondence-Quality Fixed-Width Mousetext Characters

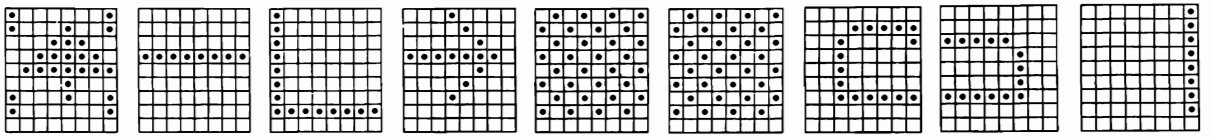
192/$C0 193/$C1 194/$C2 195/$C3 196/$C4 197/$C5 198/$C6 199/$C7 200/$C8



201/$C9 202/$CA 203/$CB 204/$CC 205/$CD 206/$CE 207/$CF 208/$D0 209/$D1



210/$D2 211/$D3 212/$D4 213/$D5 214/$D6 215/$D7 216/$D8 217/$D9 218/$DA



219/$DB 220/$DC 221/$DD 222/$DE 223/$DF

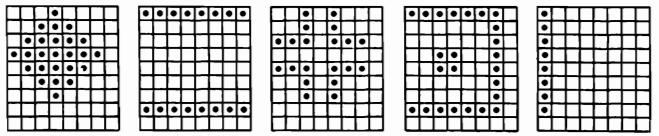
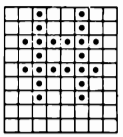
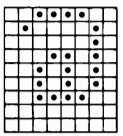
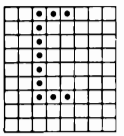
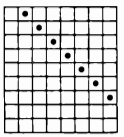
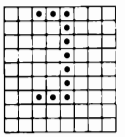
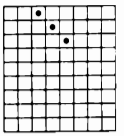
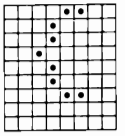
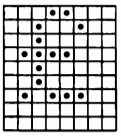
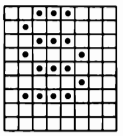
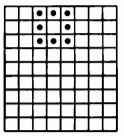
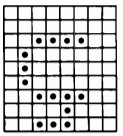
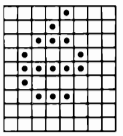
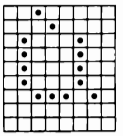
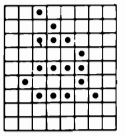
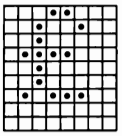
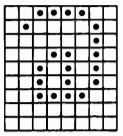
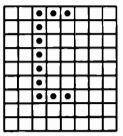
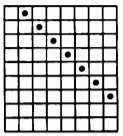
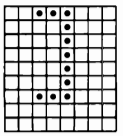
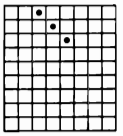
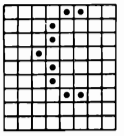
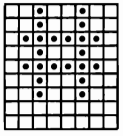
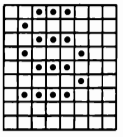
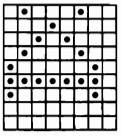
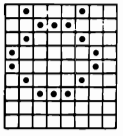
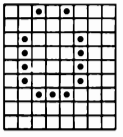
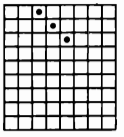
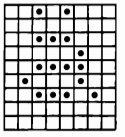
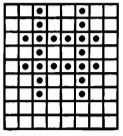
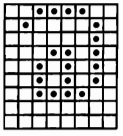
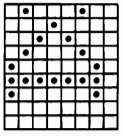
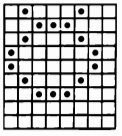
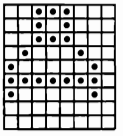
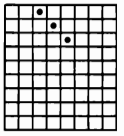
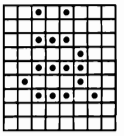
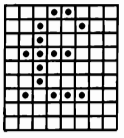
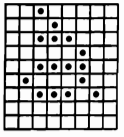
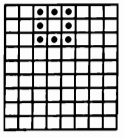
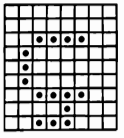
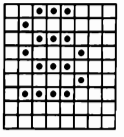
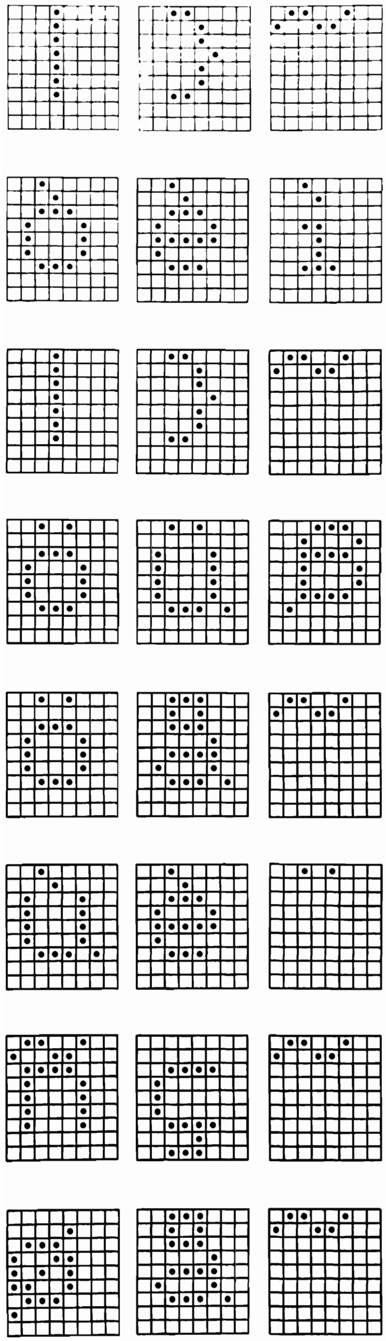


Table C-6. Correspondence-Quality Fixed-Width Alternate-Language Characters

	35/$23	64/$40	91/$5B	92/$5C	93/$5D	96/$60	123/$7B
ENGLISH (USA)							
ITALIAN							
ENGLISH (UK)							
GERMAN							
SWEDISH							
FRENCH							
SPANISH							
DANISH							

124/$7C 125/$7D 126/$7E

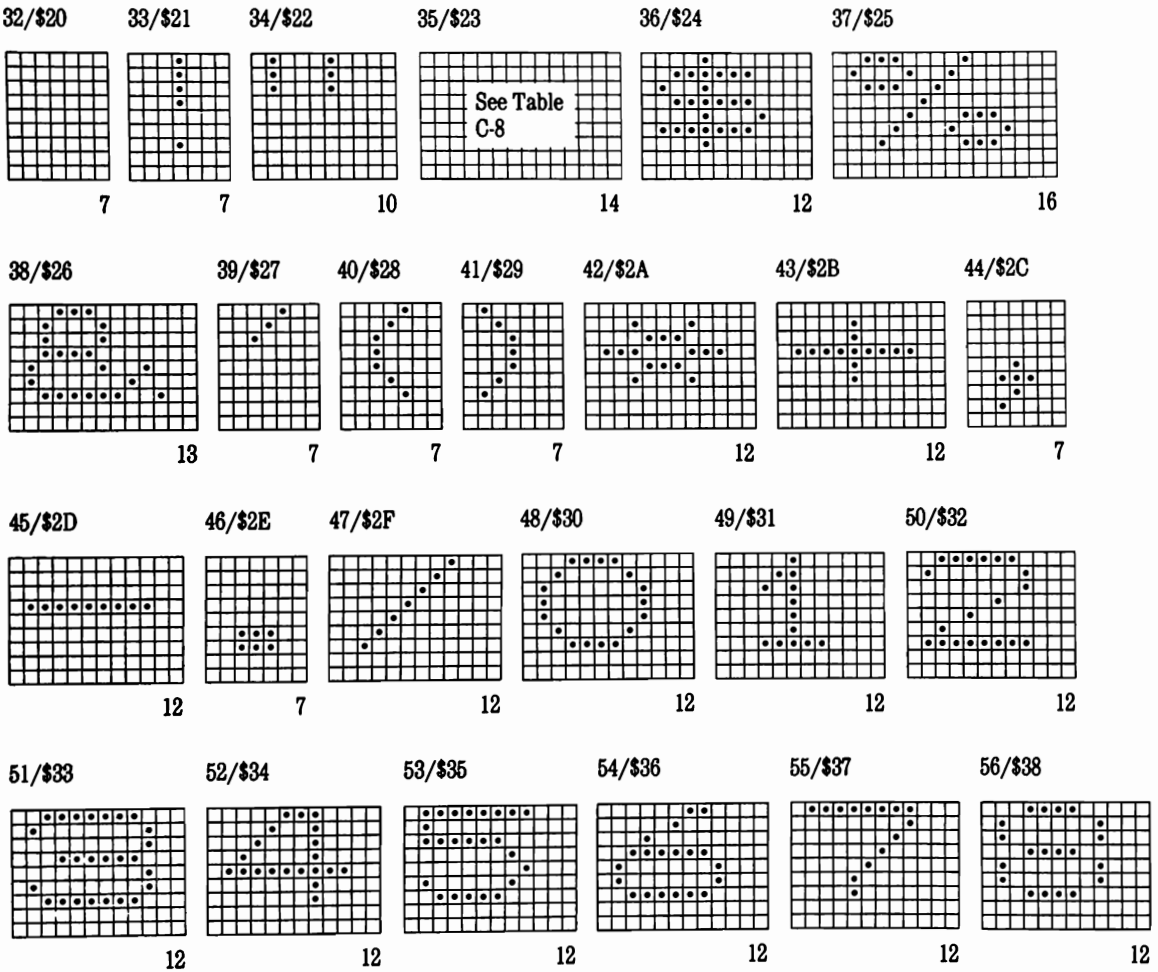


Correspondence-Quality Proportional Characters

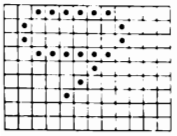
All correspondence-quality proportional characters consist of several vertical columns, each nine dots high. The right-most column never contains dots, and the bottom two dots in each column are used only by lowercase characters with descenders, and by some punctuation. The number of columns varies from character to character, as shown below.

The decimal and hexadecimal codes for each character are shown at its upper left. The number of columns for each character is shown at its lower right.

Table C-7. Correspondence-Quality Proportional Standard-ASCII Characters

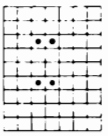


57/$39



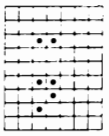
12

58/$3A



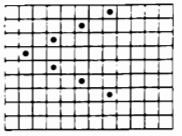
7

59/$3B



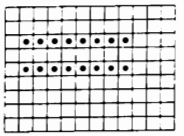
7

60/$3C



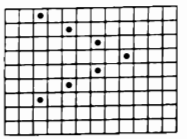
12

61/$3D



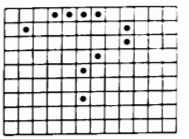
12

62/$3E



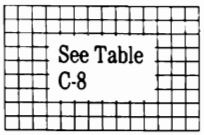
12

63/$3F



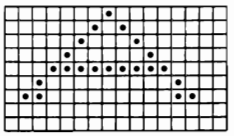
12

64/$40



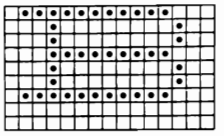
14

65/$41



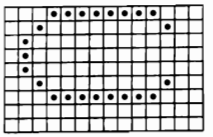
16

66/$42



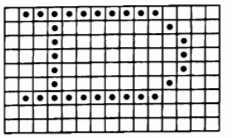
15

67/$43



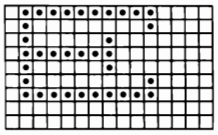
14

68/$44



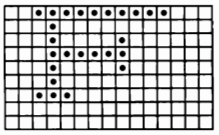
15

69/$45



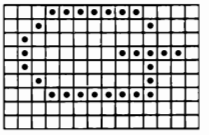
15

70/$46



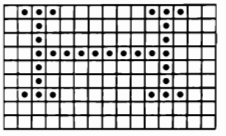
15

71/$47



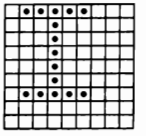
15

72/$48



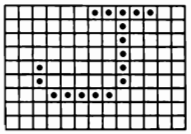
16

73/$49



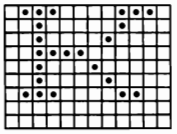
9

74/$4A



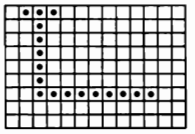
13

75/$4B



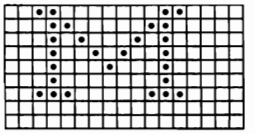
12

76/$4C



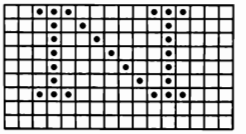
13

77/$4D



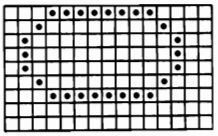
17

78/$4E



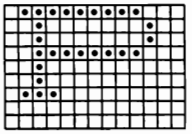
16

79/$4F



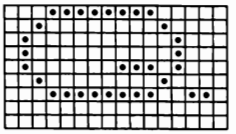
15

80/$50



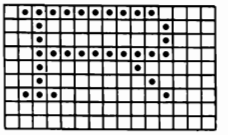
13

81/$51



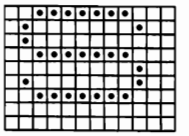
16

82/$52



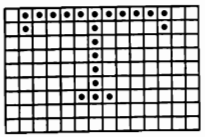
15

83/$53



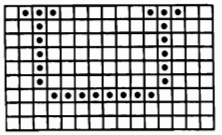
12

84/$54



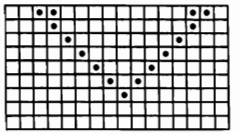
14

85/$55



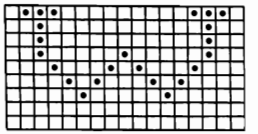
15

86/$56



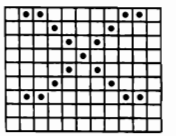
16

87/$57



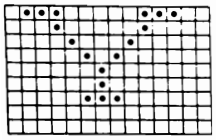
17

88/$58



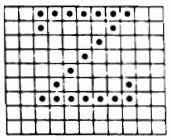
11

89/$59



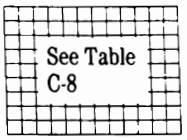
14

90/$5A



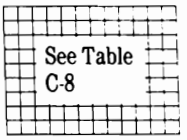
11

91/$5B



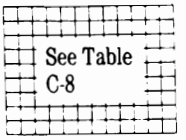
12

92/$5C



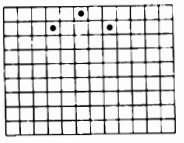
12

93/$5D



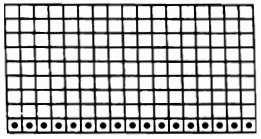
12

94/$5E



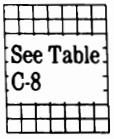
12

95/$5F



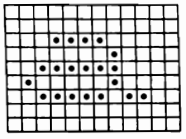
17

96/$60



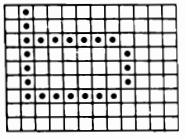
7

97/$61



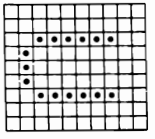
12

98/$62



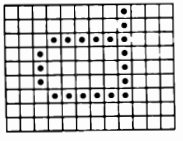
12

99/$63



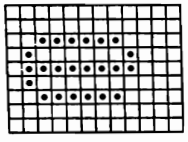
10

100/$64



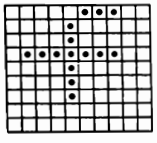
12

101/$65



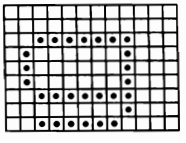
12

102/$66



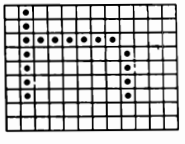
10

103/$67



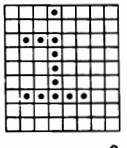
12

104/$68



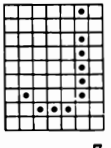
12

105/$69



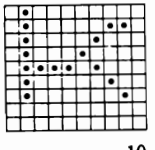
8

106/$6A



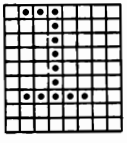
7

107/$6B



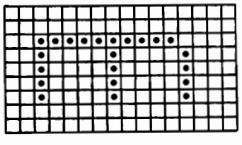
10

108/$6C



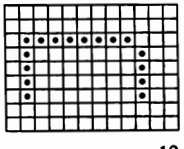
8

109/$6D



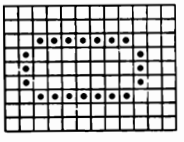
16

110/$6E



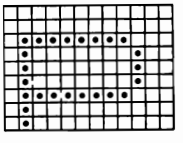
12

111/$6F



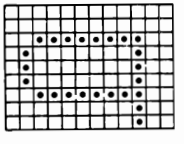
12

112/$70



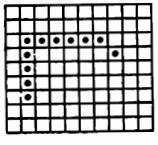
12

113/$71



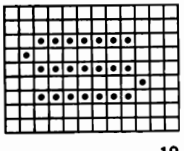
12

114/$72



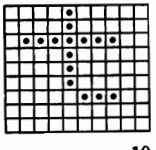
10

115/$73



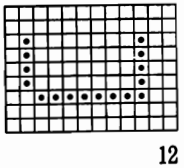
12

116/$74



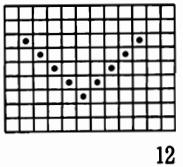
10

117/$75



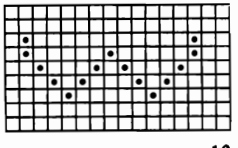
12

118/$76



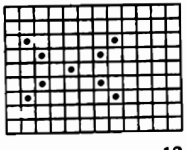
12

119/$77



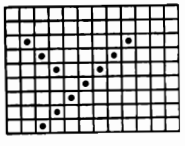
16

120/$78



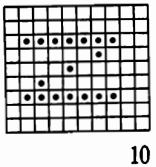
12

121/$79



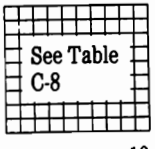
12

122/$7A



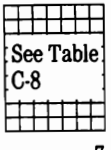
10

123/$7B



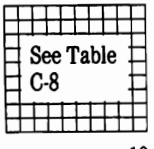
10

124/$7C



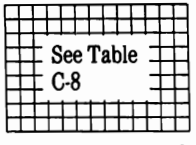
7

125/$7D



10

126/$7E



13

Table C-8. Correspondence-Quality Proportional Alternate-Language Characters

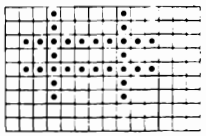
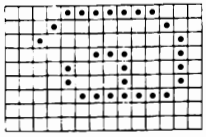
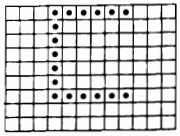
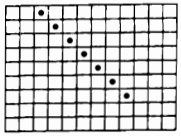
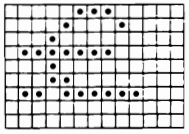
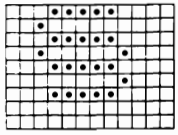
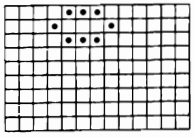
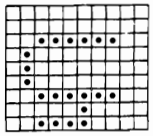
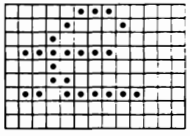
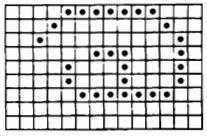
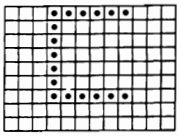
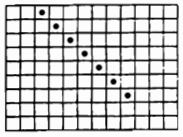
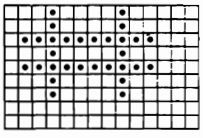
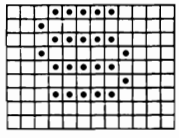
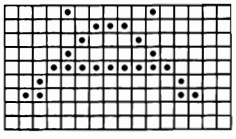
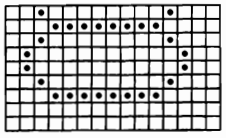
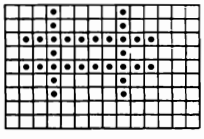
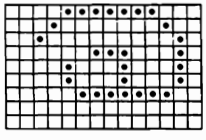
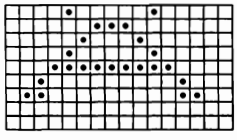
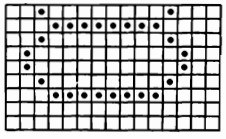
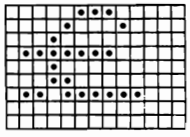
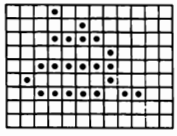
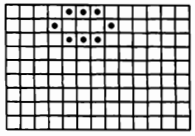
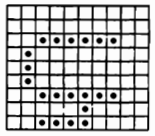
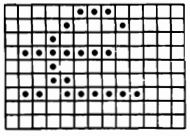
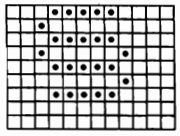
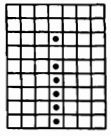
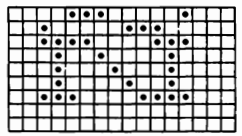
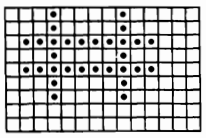
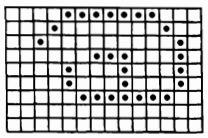
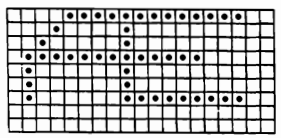
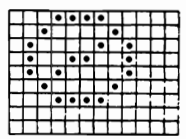
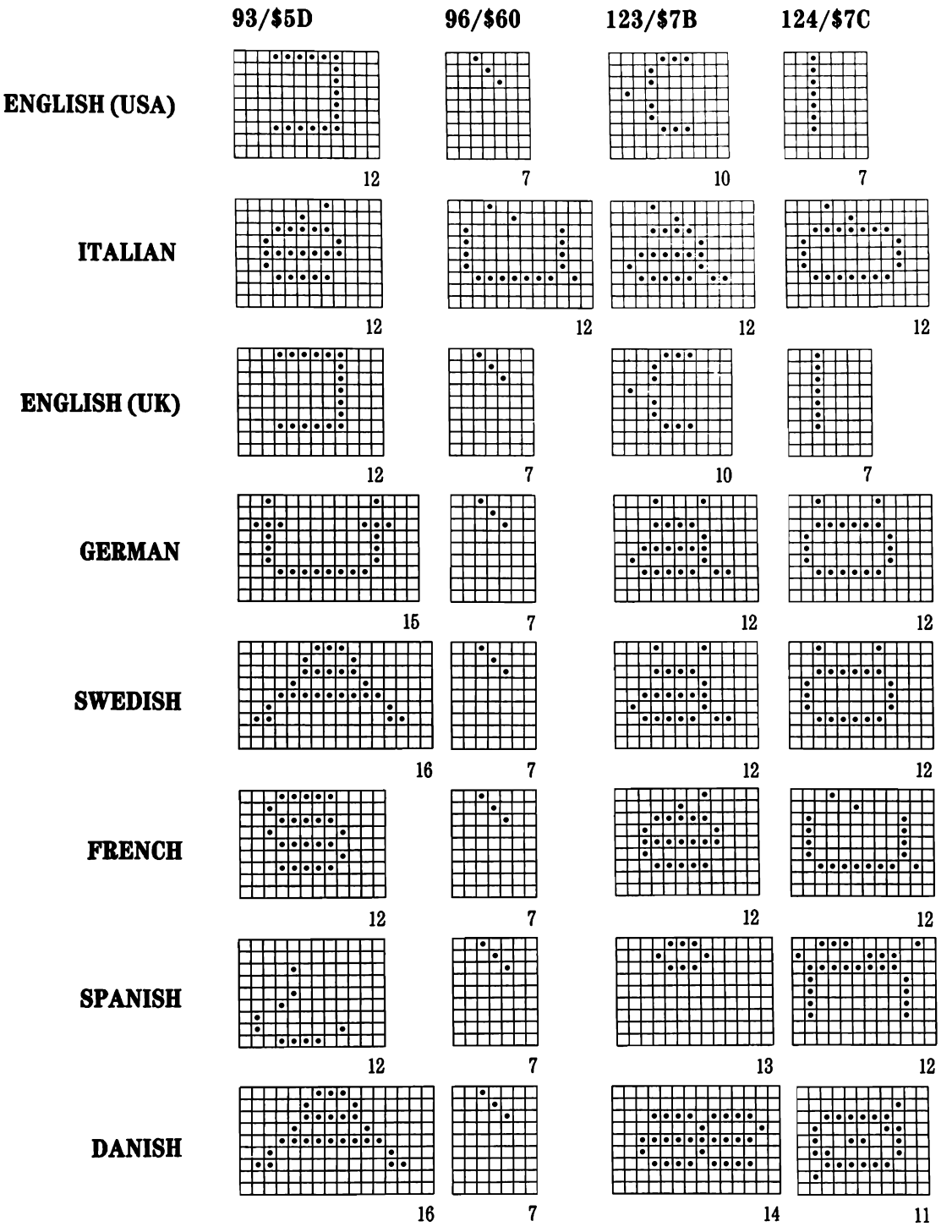
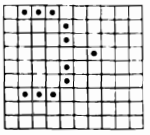
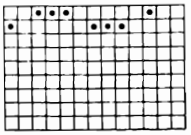
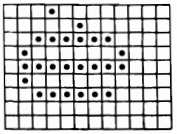
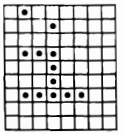
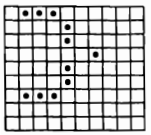
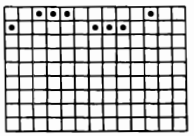
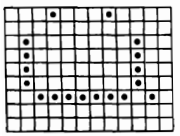
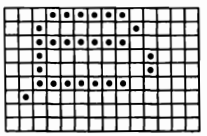
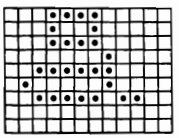
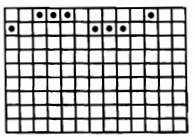
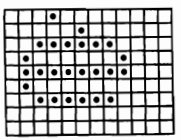
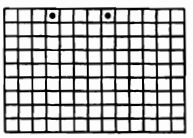
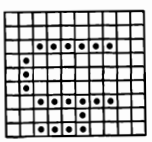
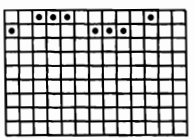
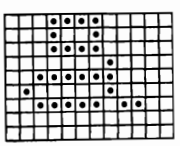
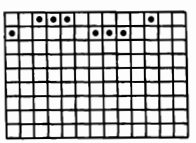
	35/$23	64/$40	91/$5B	92/$5C
ENGLISH (USA)	 14	 14	 12	 12
ITALIAN	 13	 12	 13	 10
ENGLISH (UK)	 13	 14	 12	 12
GERMAN	 14	 12	 16	 15
SWEDISH	 14	 14	 16	 15
FRENCH	 13	 12	 13	 10
SPANISH	 13	 12	 7	 16
DANISH	 14	 14	 19	 12

Table C-8. Correspondence-Quality Proportional Alternate-Language Characters (Continued)



125/$7D	126/$7E
 10	 13
 12	 8
 10	 13
 12	 14
 12	 13
 12	 13
 10	 13
 12	 13

Near-Letter-Quality Characters

Near-letter-quality characters include:

- Fixed-width standard ASCII
- Fixed-width MouseText
- Fixed-width alternate language
- Proportional standard ASCII
- Proportional alternate language

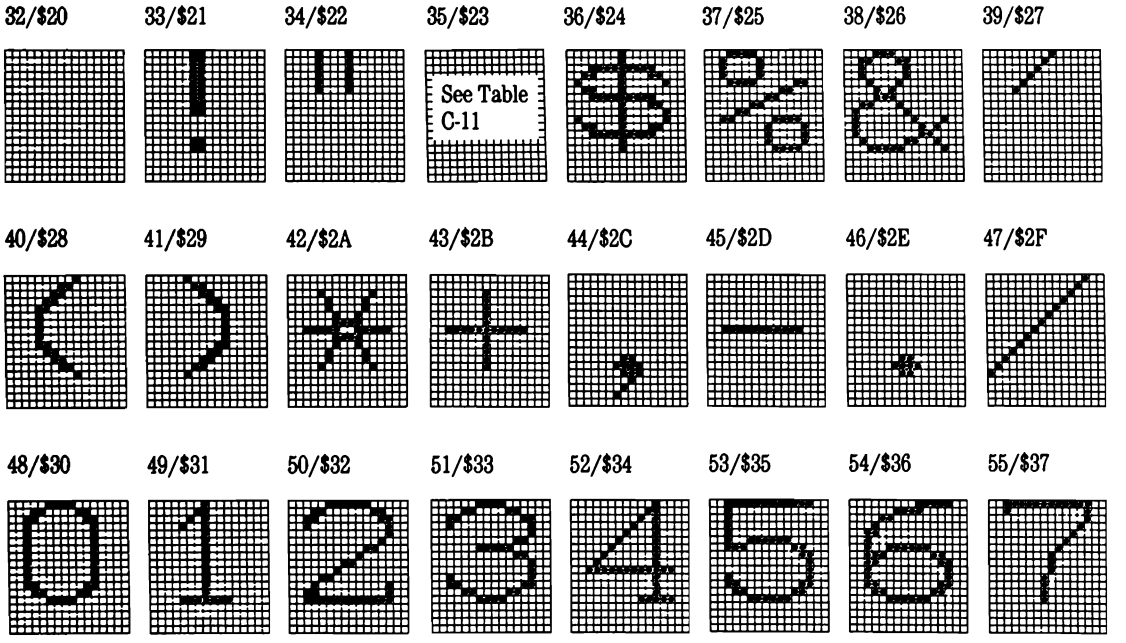
Each of these character sets is shown in this section.

Near-Letter-Quality Fixed-Width Characters

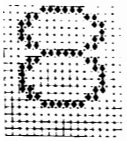
All NLQ fixed-width characters consist of sixteen vertical columns, each eighteen dots high. The right-most column never contains dots, and the bottom four dots in each column are used only by lowercase characters with descenders, and by some punctuation.

The decimal and hexadecimal codes for each character are shown at its upper left.

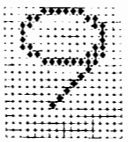
Table C-9. NLQ Fixed-Width Standard-ASCII Characters



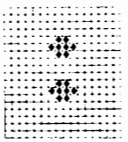
56/$38



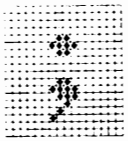
57/$39



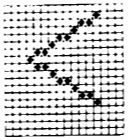
58/$3A



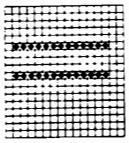
59/$3B



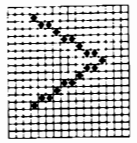
60/$3C



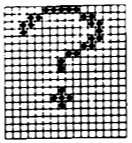
61/$3D



62/$3E



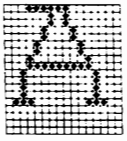
63/$3F



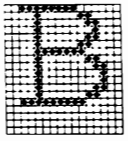
64/$40

See Table C-11

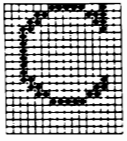
65/$41



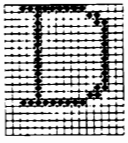
66/$42



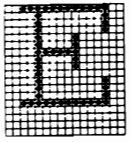
67/$43



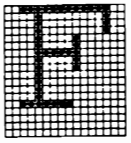
68/$44



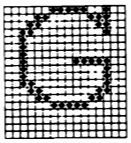
69/$45



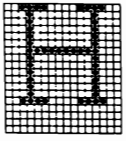
70/$46



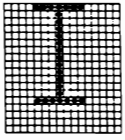
71/$47



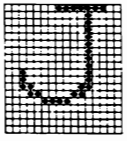
72/$48



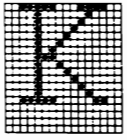
73/$49



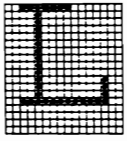
74/$4A



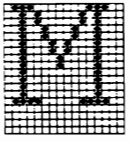
75/$4B



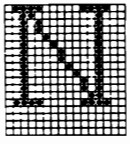
76/$4C



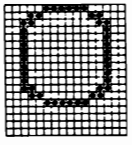
77/$4D



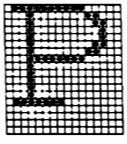
78/$4E



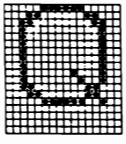
79/$4F



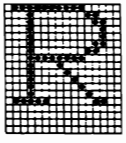
80/$50



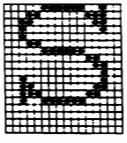
81/$51



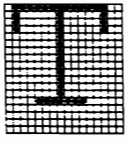
82/$52



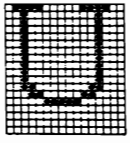
83/$53



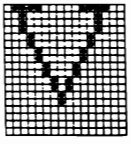
84/$54



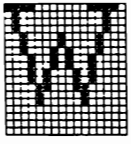
85/$55



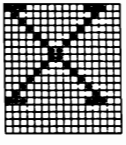
86/$56



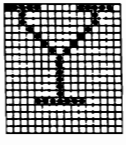
87/$57



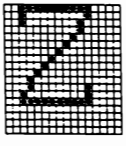
88/$58



89/$59



90/$5A



91/$5B

See Table C-11

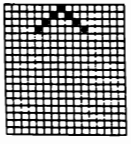
92/$5C

See Table C-11

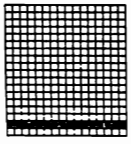
93/$5D

See Table C-11

94/$5E



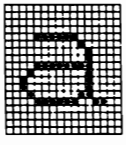
95/$5F



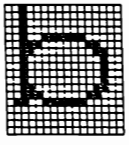
96/$60

See Table C-11

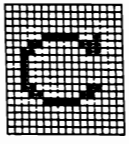
97/$61



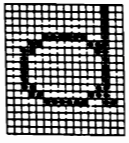
98/$62



99/$63



100/$64

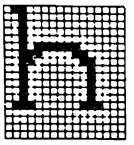


101/$65

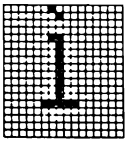
102/$66

103/$67

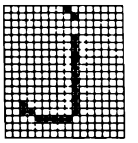
104/$68



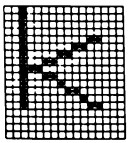
105/$69



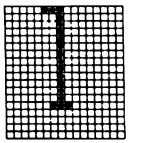
106/$6A



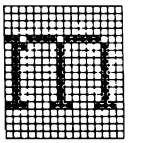
107/$6B



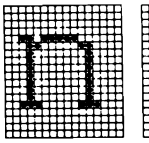
108/$6C



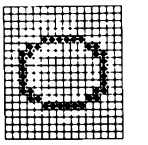
109/$6D



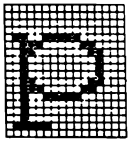
110/$6E



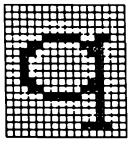
111/$6F



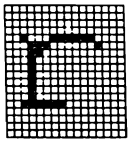
112/$70



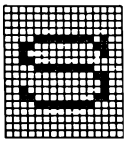
113/$71



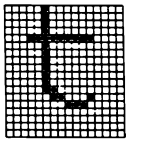
114/$72



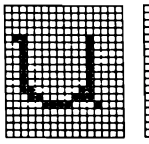
115/$73



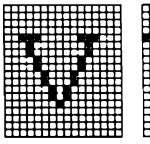
116/$74



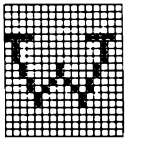
117/$75



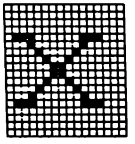
118/$76



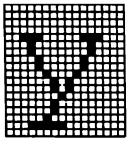
119/$77



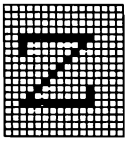
120/$78



121/$79



122/$7A



123/$7B

See Table C-11

124/$7C

See Table C-11

125/$7D

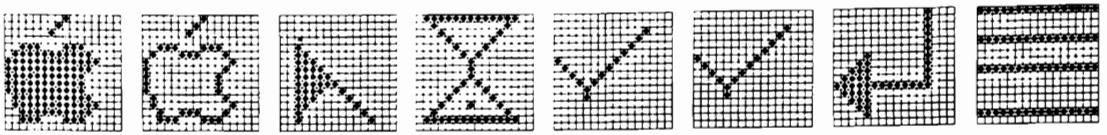
See Table C-11

126/$7E

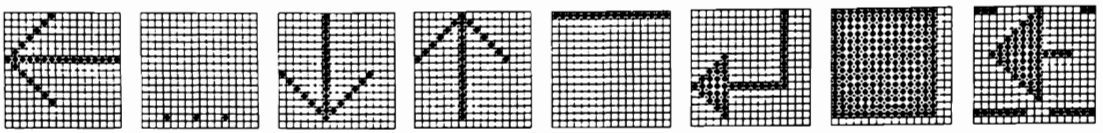
See Table C-11

Table C-10. NLQ Fixed-Width MouseText Characters

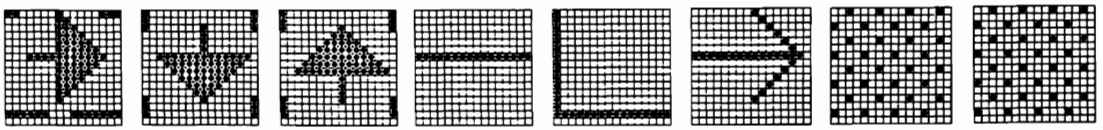
192/$C0 193/$C1 194/$C2 195/$C3 196/$C4 197/$C5 198/$C6 199/$C7



200/$C8 201/$C9 202/$CA 203/$CB 204/$CC 205/$CD 206/$CE 207/$CF



208/$D0 209/$D1 210/$D2 211/$D3 212/$D4 213/$D5 214/$D6 215/$D7



216/$D8 217/$D9 218/$DA 219/$DB 220/$DC 221/$DD 222/$DE 223/$DF

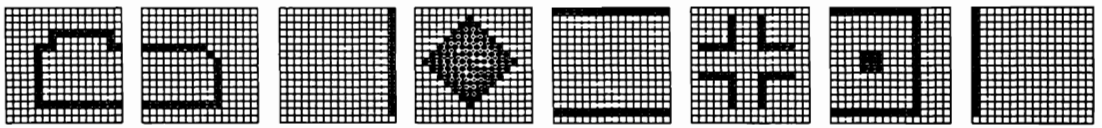
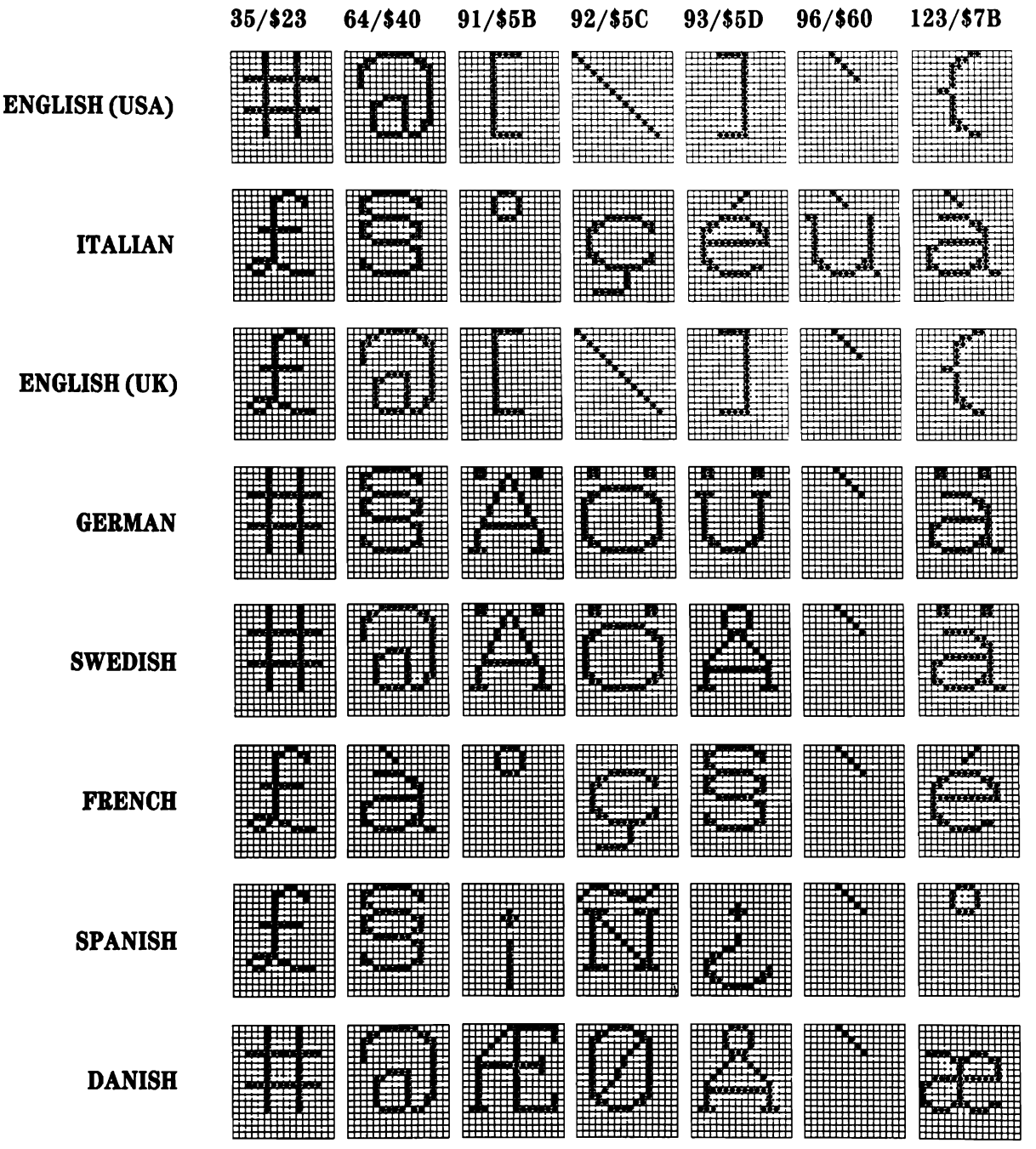
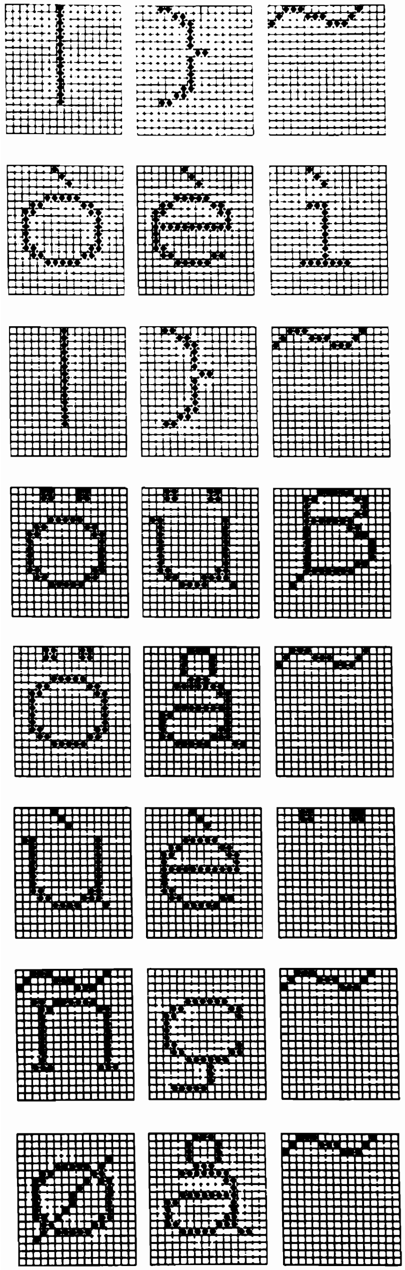


Table C-11. NLQ Fixed-Width Alternate-Language Characters



124/$7C 125/$7D 126/$7E



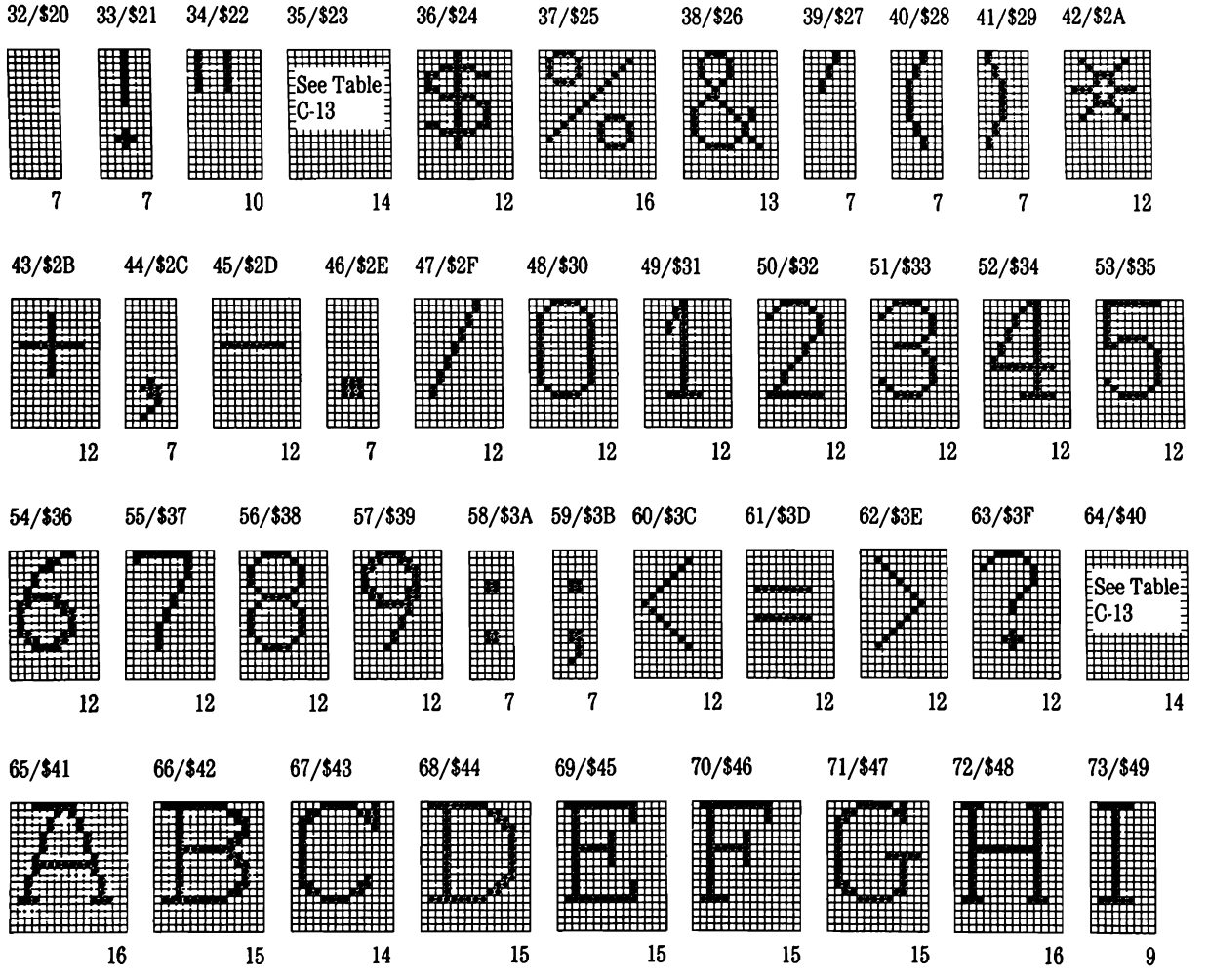
Near-Letter-Quality Characters

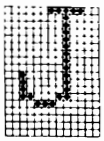
Near-Letter-Quality Proportional Characters

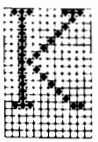
All NLQ proportional characters consist of several vertical columns, each eighteen dots high. The right-most column never contains dots, and the bottom four dots in each column are used only by lowercase characters with descenders, and by some punctuation. The number of columns varies from character to character, as shown below.

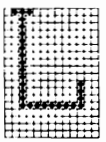
The decimal and hexadecimal codes for each character are shown at its upper left. The number of columns for each character is shown at its lower right.

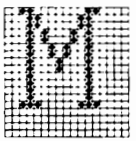
Table C-12. NLQ Proportional Standard-ASCII Characters

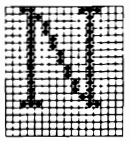


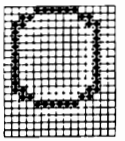
74/$4A

13

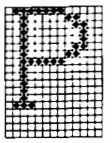
75/$4B

12

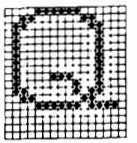
76/$4C

13

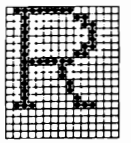
77/$4D

17

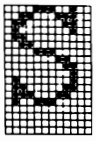
78/$4E

16

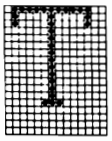
79/$4F

15

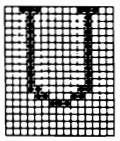
80/$50

13

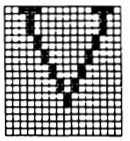
81/$51

16

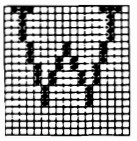
82/$52

15

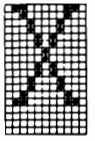
83/$53

12

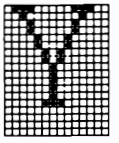
84/$54

14

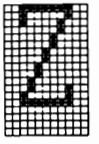
85/$55

15

86/$56

16

87/$57

17

88/$58

11

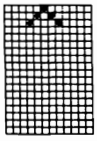
89/$59

14

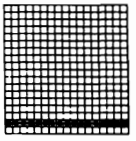
90/$5A

11

91/$5B
See Table C-13
12

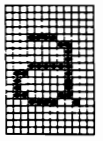
92/$5C
See Table C-13
12

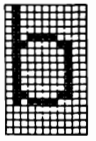
93/$5D
See Table C-13
12

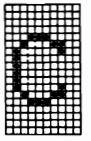
94/$5E

12

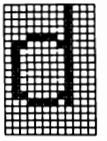
95/$5F

17

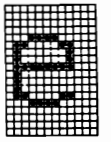
96/$60
See Table C-13
7

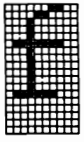
97/$61

12

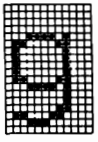
98/$62

12

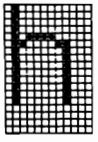
99/$63

10

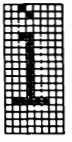
100/$64

12

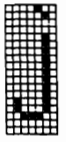
101/$65

12

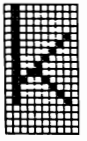
102/$66

10

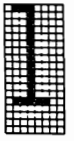
103/$67

12

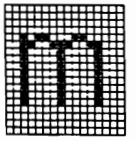
104/$68

12

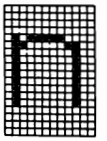
105/$69

8

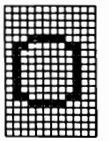
106/$6A

7

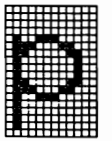
107/$6B

10

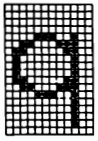
108/$6C

8

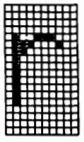
109/$6D

16

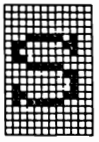
110/$6E

12

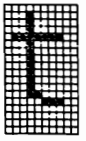
111/$6F

12

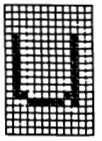
112/$70

12

113/$71

12

114/$72

10

115/$73

12

116/$74

10

117/$75

12

118/$76

12

119/$77

16

120/$78

12

121/$79

12

122/$7A

10

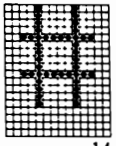
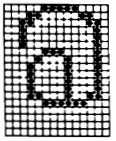
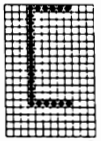
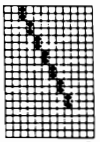
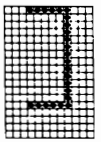
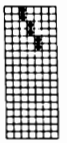
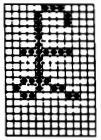
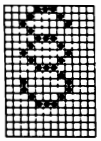
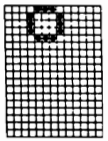
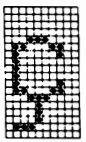
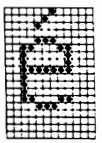
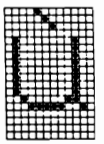
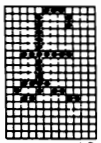
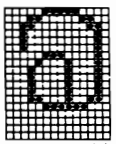
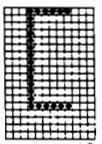
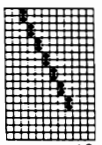
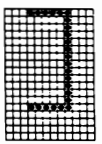
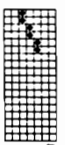
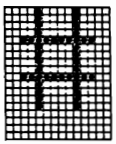
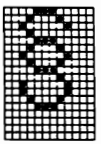
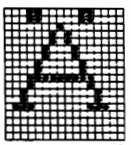
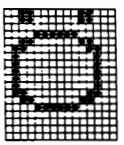
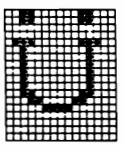
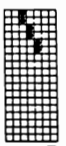
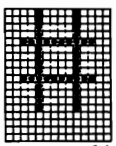
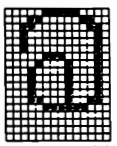
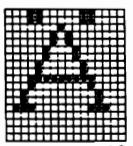
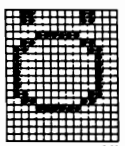
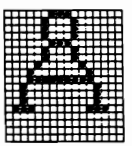
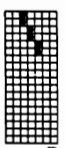
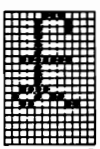
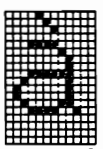
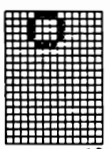
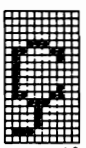
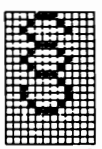
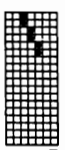
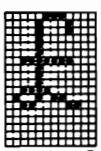
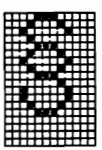
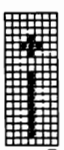
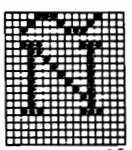
123/$7B
See Table C-13
10

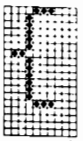
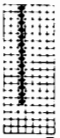
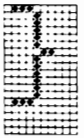
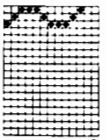
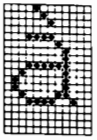
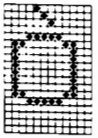
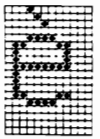
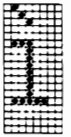
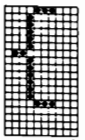
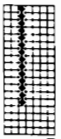
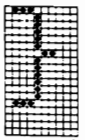
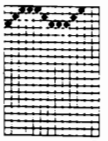
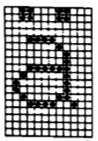
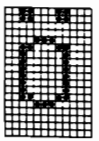
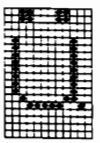
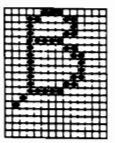
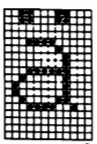
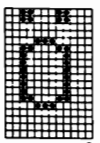
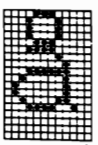
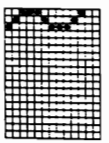
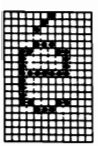
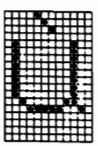
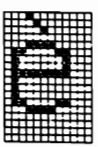
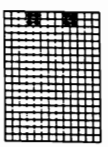
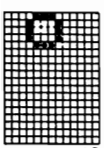
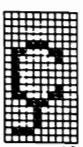
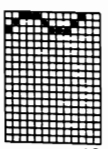
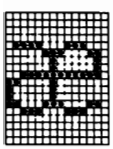
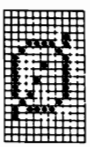
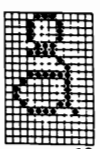
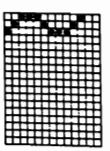
124/$7C
See Table C-13
7

125/$7D
See Table C-13
10

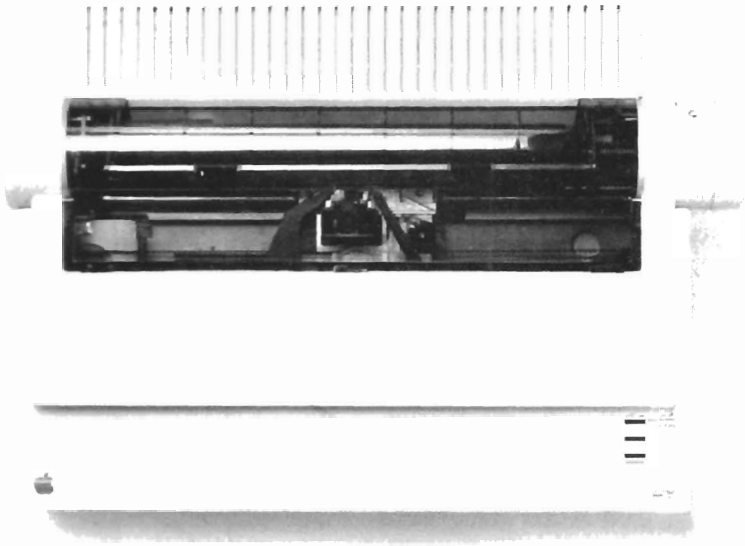
126/$7E
See Table C-13
13

Table C-13. NLQ Proportional Alternate-Language Characters

	35/$23	64/$40	91/$5B	92/$5C	93/$5D	96/$60
ENGLISH (USA)	 14	 14	 12	 12	 12	 7
ITALIAN	 13	 12	 13	 10	 12	 13
ENGLISH (UK)	 13	 14	 12	 12	 12	 7
GERMAN	 14	 12	 16	 15	 15	 7
SWEDISH	 14	 14	 16	 15	 16	 7
FRENCH	 13	 12	 13	 10	 12	 7
SPANISH	 13	 12	 7	 16	12	7
DANISH	14	14	19	12	16	7

123/$7B	124/$7C	125/$7D	126/$7E
 10	 7	 10	 13
 12	 12	 12	 8
 10	 7	 10	 13
 12	 12	 12	 14
 12	 12	 12	 13
 12	 12	 12	 13
 13	 12	 10	 13
 14	 11	 12	 13

Appendix D Printer Specifications



Below are the specifications for the Apple ImageWriter II printer.

Print Method:	Dot Matrix, logic seek (line by line)
Printing Speed:	Draft 250 cps Correspondence 180 cps NLQ 45 cps
Character Format:	Draft characters: 12 dots wide by 7 dots high, using every other dot on a horizontal line (1/2 dot method) Correspondence characters: 7 dots wide by 7 dots high Correspondence proportional characters: Up to 16 dots wide by 7 dots high NLQ characters: 16 dots wide by 14 dots high NLQ proportional characters: Up to 16 dots wide by 14 dots high Custom (down loaded) characters: Up to 16 dots wide by 8 dots high
Standard Characters:	96 ASCII (alphanumeric and symbols) 28 European language characters 32 MouseText characters
Vertical Dot Spacing:	1/72 of an inch
Printed Line Length:	8 inches maximum

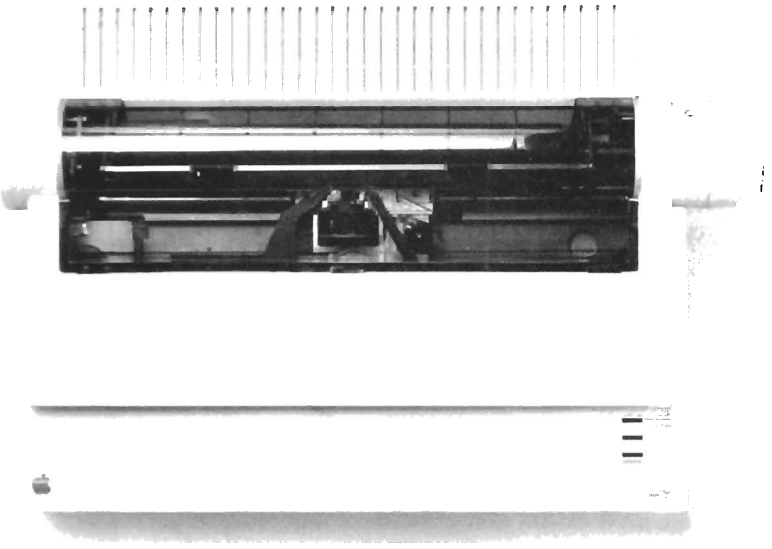
Horizontal Pitches:

Characters per Inch	Characters per Line	Dots per Inch (Approx.)
17	136	136
15	120	120
13.4	107	107
12	96	96
10	80	80
9	72	72
Proportional-elite	variable	160
Proportional-pica	variable	144

Paper Feed Direction:	Forward and Reverse
Line Spacing:	1/144 to 99/144 inch, selectable in increments of 1/144 inch
Line Feed Speed:	Max. 24 lps @ 6 lpi
Paper Width:	3.5 to 9.5 inches pin to pin (pin feed), 10 inches maximum

Paper Thickness:	0.05–0.28 millimeter (0.002–0.011 inch) Original + 3 copies maximum		
Paper Feed Method:	Selectable, friction or friction/pin-feed		
Paper Types:	Single Sheets Pin-feed paper (hole centers 3.5–9.5 inches)		
Ribbon:	Cassette containing black inked fabric ribbon: 13 mm wide by 13000 mm long, continuous Optional Four-color ribbon: 21 mm wide by 18000 mm long, continuous		
Power Options:	American Universal	120 volts AC ± 10%, 60 hertz 100 volts AC ± 10%, 50/60 hertz 120 volts AC ± 10%, 50/60 hertz 140 volts AC ± 10%, 50/60 hertz 200 volts AC ± 10%, 50/60 hertz 220 volts AC ± 10%, 50/60 hertz 240 volts AC ± 10%, 50/60 hertz	
Power Consumption:	Operating: 180 watts maximum Standby: 20 watts maximum		
Data Interface:	8-bit serial		
Weight:	11.36 kilograms (25 pounds)		
Dimensions:	Width 431.8 17.0	Depth 304.8 12.0	Height 127.0 millimeters 5.0 inches
Ambient Temperature:			
Operating:	10 to 40 degrees Celsius (50 to 104 degrees F.)		
Storage:	–40 to +47 degrees Celsius (–40 to +116 degrees F.)		
Humidity:			
Operating:	20% to 95% relative humidity, noncondensing		
Storage:	10% to 95% relative humidity, noncondensing		

Appendix E Interface Specifications



This appendix describes the connections and signals that interface the ImageWriter II to computers and the AppleTalk network.

Table E-1. Interface Specifications Summary

Data Input Form:	8-bit serial: 1 start bit, 8 data bits, and 1 stop bit (no parity bit); asynchronous
Data Output Form:	8-bit serial: 1 start bit, 8 data bits, and 1 stop bit; space parity (8th bit set to 0); asynchronous
Transmission Speed:	300, 1200, 2400, or 9600 baud set by the DIP switches 2-1 and 2-2
Input Buffer Size:	2K bytes; 32K bytes with 32K Memory Option
Printer Connector:	Mini-circular 8-pin socket
Cable Connector:	Mini-circular 8-pin plug

Figure E-1. Mini-Circular Jack

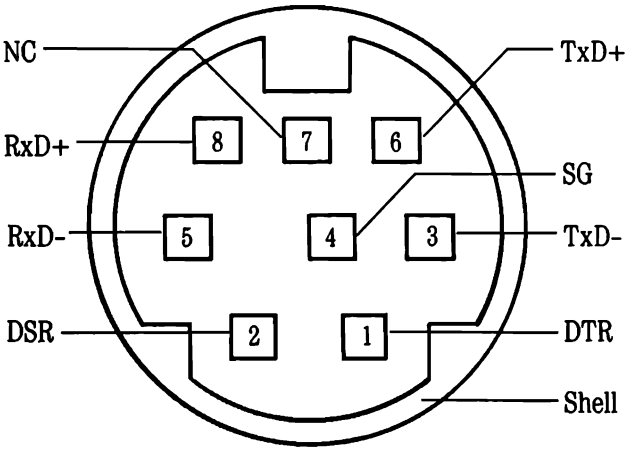


Table E-2. Mini-Circular Jack Pin Assignments

Pin Number	Symbol	Description	Direction
1	DTR	Data Terminal Ready	Output
2	DSR	Data Set Ready	Input
3	TxD-	Transmitted Data	Output
4	SG	Signal Ground	
5	RxD-	Received Data	Input
6	TxD+	Balanced Transmit	Output
7	NC	No Connection	
8	RxD+	Balanced Receive	Input
Shield	PG	Protective Ground	

Table E-3. Mini-Circular to RS232 Connections

Mini-Circular Pin Number	RS232 Symbol	Signal Description	DB25 Pin Number
1	DSR	Data Set Ready	6**
1	DCD	Data Carrier Detect	8**
2	DTR	Data Terminal Ready	20
3	RxD	Received Data	3
4*	GND	Ground	7
5	TxD	Transmitted Data	2
6	NC	No Connection	
7	NC	No Connection	
8*	GND	Ground	
Shield	Shield		

* Connect together at Mini-Circular Connector
** Connect together at DB25

Signal Descriptions

All signals comply with the Electronics Industry Association (EIA) RS422 standard. To implement an RS423 interface, ground the RxD+ line and do not use the TxD+ line. To implement an RS232 interface, use the connections shown in Table E-3.

Data Terminal Ready: Output signal from the printer. If the printer is set to Hardware Handshake protocol, this line is true (logic 0) when the printer is on and able to receive data; false (logic 1) when the printer is unable to receive data (deselected or busy).

If the printer is set to XON/XOFF protocol, the DTR line goes true (0) when the power is turned on, and remains true even if the printer is deselected or busy.

Data Set Ready: Input signal to the printer. The DSR signal is used by Apple technicians for diagnostic tests. During normal operation, the ImageWriter II ignores this signal.

Transmitted Data: Output signal from the printer. The TxD- line and TxD+ line together comprise a differential signal used to transmit data from the printer to the computer. This line is used to transmit printer self ID information, Appletalk status information, and XON/XOFF characters. When no data is being transmitted, the TxD- line is held in the logic 1 state.

The TxD- line is of the proper polarity to serve as the single transmitted data line for an RS423 or an RS232 interface.

Signal Ground: This ground provides a reference voltage level for the interface signals.

Received Data: Input signal to the printer. The RxD- line and RxD+ line together comprise a differential signal used by the printer to receive data. This line receives all the command, text, and graphics data sent to the printer by the computer. When no data is being received, the RxD- line is held in the logic 1 state.

The RxD- line is of the proper polarity to serve as the single received data line for an RS423 or an RS232 interface. The RxD+ line must be grounded if RS423- or RS232-compatible communication is desired.

Balanced Transmit: Output signal from the printer. The TxD- line and TxD+ line together comprise a differential signal used to transmit data from the printer to the computer.

If RS423- or RS232-compatible communication is desired, the TxD+ line is not used and may be allowed to float without termination.

Balanced Receive: Input signal to the printer. The RxD- line and RxD+ line together comprise a differential signal used by the printer to receive data.

If RS423- or RS232-compatible communication is desired, the RxD+ line must be grounded.

Shield: A low-impedance ground used for circuit protection.

Figure E-2. Input Circuit

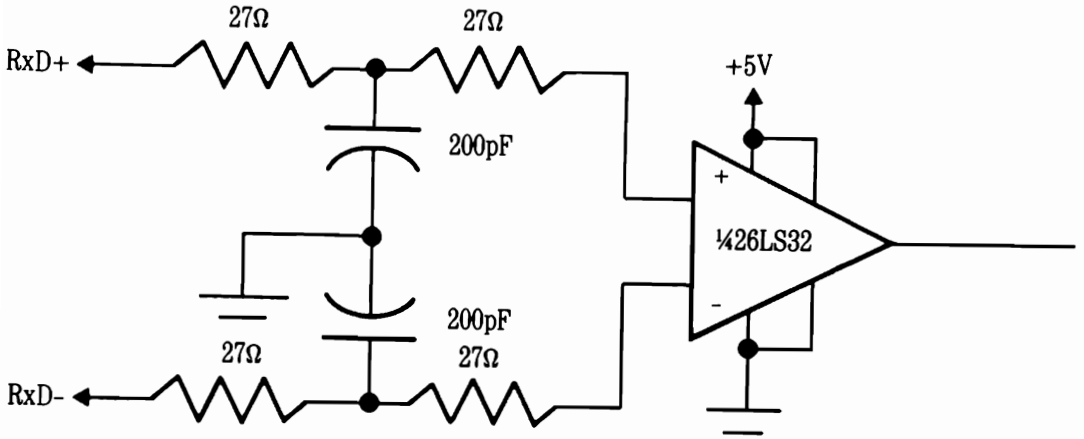
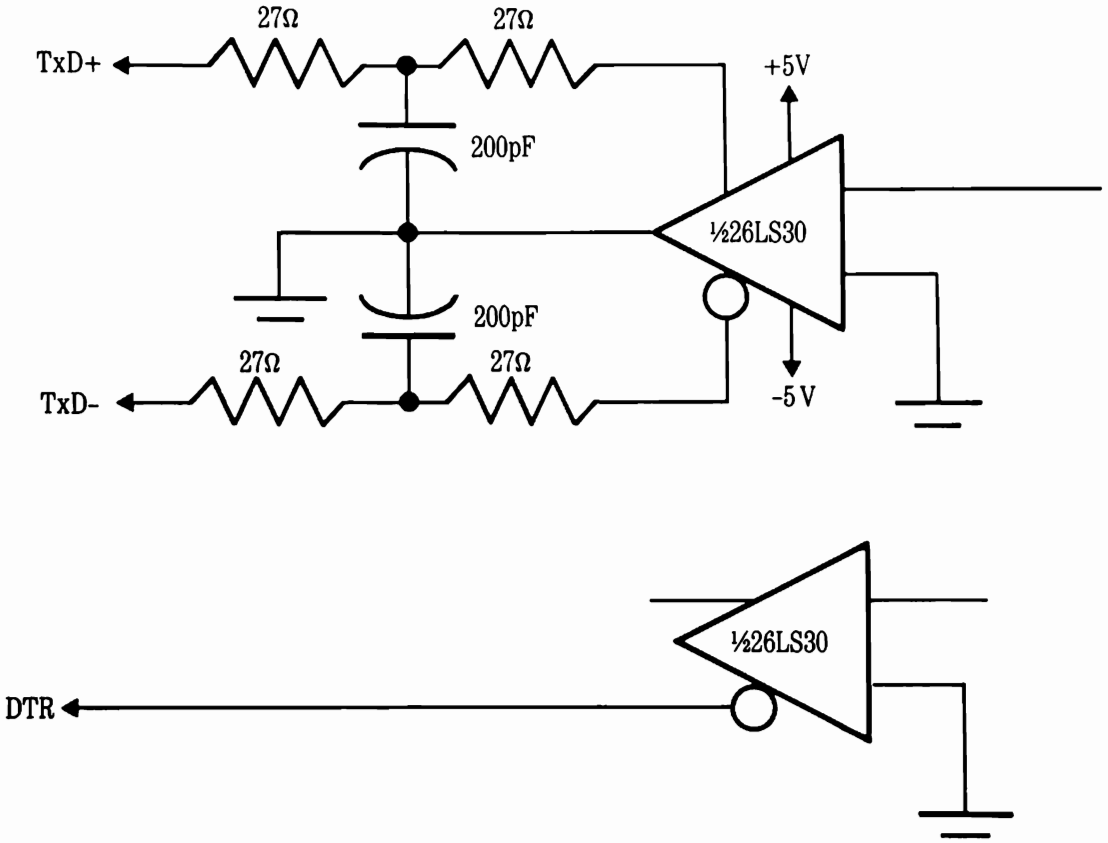


Figure E-3. Output Circuits



Communication Protocol

The ImageWriter II printer is capable of two types of data protocol: hardware handshake and XON/XOFF. The hardware handshake protocol uses the DTR signal to indicate the printer state, while the XON/XOFF protocol uses ASCII codes CONTROL-Q (XON) and CONTROL-S (XOFF) for the same purpose. These protocols are described in the following two sections.

Hardware Handshake

Whenever the input buffer has room left for less than 30 characters, the printer sends a busy signal by setting the DTR line false. The computer must stop transmission within the next 27 characters; if it does not, the excess data is lost. The DTR line is set true again when there is room for at least 100 characters in the input buffer.

The DTR line is also set false when the printer is deselected, either by pressing the front-panel switch or by sending it a CONTROL-S (DC3) character. The DTR line is set true whenever the printer is selected either by the front-panel switch or by sending it a CONTROL-Q (DC1) character. When the printer is turned on, it comes up in the selected state.

Important! The ImageWriter II responds to the software select and deselect signals (DC1 and DC3) only when software switch SWA-5 is open. Software switches are described in Chapter 5.

XON/XOFF

Whenever the input buffer has room left for less than 266 characters, the printer sends a CONTROL-S (XOFF) character. When the computer receives the XOFF character, it must stop transmission within the next 256 characters; if it does not, the excess data is lost. An XOFF character is also sent when the printer is deselected, either by the front-panel switch or by sending it a DC3 character (unless an XOFF has already been sent as described above).

The printer sends a CONTROL-Q (XON) character when there is room for at least 337 characters in the input buffer (and an XOFF character was sent earlier), when the printer is turned on, and when the printer is selected.

By the Way: You may have noticed that ASCII code CONTROL-Q is sometimes referred to as DC1, and sometimes as XON, while CONTROL-S is called both DC3 and XOFF. The names given to these two ASCII codes depend on whether they are sent by the computer to select or deselect the printer (DC1 and DC3), or by the printer to control data transmission from the computer (XON and XOFF); the actual ASCII codes sent are the same in both cases.

Applesoft BASIC: A programming language that can be used with Apple II Plus, IIe, and IIc computers. BASIC stands for *Beginner's All-Purpose Symbolic Instruction Code.*

ASCII: Acronym for *American Standard Code for Information Interchange,* a standard that assigns a unique binary number to each text character and control code. ASCII codes are listed in Appendix B. See also **high ASCII character, low ASCII character**.

baud: A unit of signalling speed equal to the number of discrete signal events per second; usually equal to bits per second.

bidirectional printing: A mode of printer operation in which characters are printed both when the print head is moving from left to right and when it is moving from right to left. Bidirectional printing is the default mode for the ImageWriter II. Compare with **unidirectional printing**.

binary: The base-two number system, in which all numbers are represented by a string of zeros and ones. The binary number 01 has the value (in the decimal system) of 1, binary 10 equals decimal 2, binary 100 equals decimal 4, binary 1000 equals decimal 8, and so on. An 8-bit binary number consisting of all ones has the decimal value 255.

bit: A single binary digit, consisting of either a zero or a one.

boot: The process by which a computer automatically loads a program into memory when the computer is turned on or reset. The computer is said to *pull itself up by its own bootstraps*. See also **boot disk, reset**.

boot disk: A disk containing the programs and data that the computer needs to get started. The computer automatically reads the boot disk when power is turned on or the computer is reset. See also **boot, reset.**

buffer: A memory area that holds information until it can be processed. See also **print buffer**.

byte: A binary number of fixed length. In the Apple ImageWriter II, bytes are eight bits long. Every character processed by the printer can be expressed as one byte, using the ASCII code.

carriage return: The printer function that prints the current line and sets the printer so that the next line will begin at the left margin. See also **current line** and **line feed**.

character: An ASCII code representing any letter, number, punctuation mark, or control code that can be acted upon by the printer. There are 256 possible characters, corresponding to the range of 8-bit binary numbers (from 0 to 255). The letter, number, punctuation mark, other symbol, or control code represented by an ASCII code.

character pitch: The number of characters per inch printed along a horizontal line.

character set: The entire set of characters that can be printed by a device such as the Apple ImageWriter II. See Appendix C for the ImageWriter II character set.

character string: Two or more characters read or sent in sequence; for example, *Z 3 CONTROL-4 5 A 2* is a character string.

clear: To erase information or commands from memory, as when the Apple ImageWriter II clears its horizontal tab stops.

command: An instruction that can be sent to a device such as a computer or printer to cause it to perform a specific function. The Apple ImageWriter II's commands are summarized in Appendix A.

compile: To convert a program (source file) written in a high-level programming language (such as BASIC) into a file of commands in a lower-level language (such as machine language) for later execution.

control character: A nonprinting character that has an ASCII code between $00 and $1F hexadecimal (0 and 31 decimal). Control characters can be generated at the keyboard by holding down CONTROL while typing a printing key, as shown in Appendix B. Control characters are often used to send control codes.

control code: A control character. One or more characters used to send a command to the printer (as opposed to text or graphics data, which is printed). See also **escape sequence**.

control sequence: A control code consisting of more than one character.

current line: The data in the print buffer that is currently being processed by the printer; the control codes in this line are read and acted on as appropriate. The current line is not printed until a print command is received or a full line's worth of text is in the print buffer. Command codes in the line following the current line in the print buffer are not read by the printer until the current line has been printed or cancelled.

default: A value, action, or setting that is assumed or set in the absence of explicit instructions otherwise. Defaults are used by the printer when it is first turned on or reset, before it receives any control codes. The Apple ImageWriter II defaults are listed in Appendix A.

deselect: A command to a device such as a printer to place it into a condition in which it will not receive data. A deselect command has an effect opposite to that of a **select** command.

device: A hardware component of a computer system, such as a video monitor, a disk drive, or the Apple ImageWriter II.

DIP switch: A small switch that affects a printer function and can be operated manually. There are 14 switches in two DIP switch assemblies inside the case of the Apple ImageWriter II, described in Chapter 2. DIP stands for *dual in-line package*.

dot column: A horizontal position that can be taken by the print head. The number of dot columns on a page depends on the horizontal resolution set by the character pitch command; see Table 8-2. Also refers to a column of dots, as in a graphics pattern or custom character specification.

dot matrix: The method by which the Apple ImageWriter II prints. Each character or graphics design is formed by a pattern of dots, printed by nine strikers (or wires) in the print head.

dot space: The horizontal distance between dot centers. This distance depends on the character pitch in effect (see Table 8-2).

editor: A program that helps the user create and change text files by providing commands to manipulate text. Compare with **word processor**.

end-of-line character: Any character that tells the printer that the preceding text constitutes a full line and may now be printed. Same as **Print Command**.

ESCAPE character: ASCII character 27 (decimal). You can send an ESCAPE character with the ESC key on the Apple II.

escape code: An ESCAPE character, or an escape sequence.

escape sequence A sequence of characters, beginning with ESCAPE, that constitutes a complete command to the printer.

execute: To perform the actions specified by a program command or sequence of commands.

fanfold paper: Paper supplied for computer printers such as the Apple ImageWriter II. The paper is continuous, with individual sheets separated by perforations, and folded along the perforations so that it lies in a stack. The left and right edges have regularly spaced holes (pin holes) that the printer uses to advance the paper. Also known as *pin-feed paper, continuous-form paper,* or *computer paper.*

file: In a computer, any named, ordered collection of data. Text files contain text, data files contain data, and code files contain programming code.

font: A complete set of type in one size and style of characters.

format: The general shape and appearance of the printer's output, including page size, character width and spacing, line spacing, and other factors.

function: In a programming language, an instruction that converts data from one form to another. The CHR or CHR$ function, for example, converts an ASCII code number into its corresponding character. As applied to the ImageWriter II, a specific ability or task, such as boldfacing text or recognition of the eighth data bit.

graphics: Designs and pictures as opposed to written text, when referring to the capabilities of a device such as a printer.

hardware: In computer technology, the physical machinery, as opposed to software, the program instructions.

hex, hexadecimal: The base-16 number system. (Ordinary decimal numbers are base 10.) Hexadecimal numerals consist of 0, the numerals 1-9, and the capital letters A-F. Appendix B lists hexadecimal equivalents of decimal numbers 0 to 255.

high ASCII character: A character with decimal equivalent between 128 and 255, inclusive. Called *high* ASCII because the high bit (leftmost binary digit) is set to 1 rather than 0.

input buffer: See **print buffer**.

interface: The equipment that accepts electrical signals from one part of a computer system and renders them into a form that can be used by another part. The act of converting signals from one form to another and passing them between two pieces of equipment.

leading zero: A zero occurring at the beginning of a number, as when a number must be sent to the printer as a certain number of digits; for example, twenty can be written as a three digit number with one leading zero: 020.

least significant bit: The binary digit in a number or data byte that, if changed, has the smallest effect on the value of the number; usually written at the right end of the number. For example, the number four is written in binary as 100, while five is written as 101; the right-most 0 or 1 is the least significant bit. Compare with **most significant bit**.

line feed: An ASCII character (hex $0A) that instructs a device such as a printer to feed one line of paper. Also refers to a vertical motion of the platen, moving the paper up or down one print line.

line feed pitch: The number of lines printed per vertical inch.

low ASCII character: A character with decimal equivalent between 0 and 127, inclusive. Called *low* ASCII because the high bit (leftmost binary digit) is set to 0 rather than 1. The low ASCII characters comprise the standard ASCII character set.

mark parity: The most significant bit of every data byte is set to 1.

memory: Any part of a computer system that stores data.

microprocessor: A circuit component made from a chip of silicon (or other semiconductor) that performs a complete set of basic computing functions.

most significant bit: The binary digit in a number or data byte that, if changed, has the greatest effect on the value of the number; usually written at the left end of the number. For example, the number five is written in binary as 101, while one is written as 001; the left-most 0 or 1 is the most significant bit. Compare with **least significant bit**.

NLQ: Near Letter Quality; one of the character fonts available on the ImageWriter II; see Appendix C.

operating system: A computer program that allows the user to run other computer programs, obtain information about disks, files, and devices connected to the computer, and control some computer functions. Normally the operating system is the first program loaded into a computer before any other programs are run.

override: To modify or cancel a default instruction or previously-sent command.

parity: Having the quality that the number of like bits (ones or zeros) is even (*even parity*) or odd (*odd parity*). A *parity bit* may be added to each byte to assure that the proper parity condition is met, and to allow for error checking. For example, if the most significant bit is the parity bit and even parity is set, then the number four is transmitted as 10000100 while five is transmitted as 00000101. If the number 10000101 is received and even parity is set, then an error is indicated, since there is an odd number of ones (or zeros) in this number. See also **mark parity** and **space parity**.

Pascal: A high-level programming language with statements that resemble English sentences. Named after the philosopher and mathematician Blaise Pascal.

pin-feed paper: Multisheet paper that has sprocket holes along both edges, for printers such as the Apple ImageWriter II. Same as **fanfold paper**.

platen: The rubber roller in a printer or typewriter that supports the paper and provides a backing for the printing action.

print buffer: Memory in the printer that stores text until it can be printed, as when the data transmission rate exceeds the printing speed. The last line remaining in the print buffer is not printed until a print command is received or a full line of text has been received. The print buffer is sometimes called the **input buffer**. See also **current line** and **print command**.

print command: A command that causes printing, such as a carriage return. Data in the print buffer is not printed until the buffer contains a full line of text or a print command is received.

print head: In the Apple ImageWriter II, the part that moves horizontally along the platen and performs the actual printing.

proportional printing: Printing in which the space taken by each letter depends on the width of the letter. In non-proportional printing, the letter I takes as much horizontal space on a line as the letter M. In proportional printing, the I takes less space than the M.

register: A location in memory where a small amount of information, such as a single byte, is stored temporarily under program control.

reset: To restore all the default settings for a device with one action or command. The ImageWriter II can be reset by turning it off and back on (a *hard reset*) or by sending it an ESCAPE c command (*software reset* or *soft reset*). The ImageWriter II defaults are shown in Appendix A.

select: A command to a device such as a printer to place it into a condition to receive data.

software: Programs and program instructions; as opposed to hardware, the machinery that runs software.

space parity: The most significant bit of every data byte is set to 0.

stack: In a computer, a memory that is used for temporary storage of operating data during operation of a program.

standard ASCII character: Same as **low ASCII character**.

standard instruction: An instruction automatically present when no superseding instruction has been received. Same as **default**.

start bit: A bit sent at the beginning of a data byte indicating that the following string of bits represents a character.

stop bit: A bit sent at the end of a data byte indicating the end of a character.

striker wire: A pin in the print head that strikes the ribbon to place a dot on the paper. The ImageWriter II print head contains a vertical column of nine striker wires that are used to make columns of dots when printing characters and graphics.

string: See **character string**.

TAB: An ASCII character that commands a device such as a printer to start printing at a preset horizontal location (called a tab stop). Corresponds to tab on a typewriter.

TOF: Top of form; the vertical position recognized by the printer as the top of a page. The position to which the printer scrolls paper when it receives a form feed command.

tractor: In the Apple ImageWriter II, either of the toothed wheels at the back that engage pin-feed paper and pull it through the machine.

type: Printed or typewritten characters.

unidirectional printing: A mode of printer operation in which characters are printed only when the print head is moving from left to right. Compare with **bidirectional printing**.

wire: Same as **striker wire**.

word processor: A program that helps the user create, change, and format documents by providing commands to manipulate text, assign character attributes such as underlining, and format pages by setting margins, line spacing, and so forth. Compare with **editor.**

Cast of Characters

A

B

C

Print Quality

Command	Decimal	Hexadecimal	Function
ESC a 0	27 97 48	1B 61 30	Select correspondence font
ESC a 1	27 97 49	1B 61 31	Select draft font (default)
ESC a 2	27 97 50	1B 61 32	Select NLQ font
ESC m	27 109	1B 6D	Same as ESC a 0 command
ESC M	27 77	1B 4D	Same as ESC a 2 command

MouseText

Command	Decimal	Hexadecimal	Function
ESC &	27 38	1B 26	Map MouseText to low ASCII ($40–$5F)
ESC $	27 36	1B 24	Select standard ASCII characters (default)
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit

Alternate Language Characters

Command	Decimal	Hexadecimal	Language
ESC Z CTRL-G CTRL-@	27 90 7 0	1B 5A 07 00	American
ESC Z CTRL-F CTRL-@	27 90 6 0	1B 5A 06 00	Italian
ESC D CTRL-A CTRL-@	27 68 1 0	1B 44 01 00	
ESC Z CTRL-E CTRL-@	27 90 5 0	1B 5A 05 00	Danish
ESC D CTRL-B CTRL-@	27 68 2 0	1B 44 02 00	
ESC Z CTRL-D CTRL-@	27 90 4 0	1B 5A 04 00	British
ESC D CTRL-C CTRL-@	27 68 3 0	1B 44 03 00	
ESC Z CTRL-C CTRL-@	27 90 3 0	1B 5A 03 00	German
ESC D CTRL-D CTRL-@	27 68 4 0	1B 44 04 00	
ESC Z CTRL-B CTRL-@	27 90 2 0	1B 5A 02 00	Swedish
ESC D CTRL-E CTRL-@	27 68 5 0	1B 44 05 00	
ESC Z CTRL-A CTRL-@	27 90 1 0	1B 5A 01 00	French
ESC D CTRL-F CTRL-@	27 68 6 0	1B 44 06 00	
ESC D CTRL-G CTRL-@	27 68 7 0	1B 44 07 00	Spanish

User-Designed Characters

Command	Decimal	Hexadecimal	Function
ESC –	27 45	1B 2D	Maximum width of custom characters 8 dots (default)
ESC +	27 43	1B 2B	Maximum width of custom characters 16 dots
ESC I	27 73	1B 49	Start loading new character(s)
CTRL-D	4	04	End loading new character(s)
ESC '	27 39	1B 27	Switch to custom-character font
ESC *	27 42	1B 2A	Switch to custom-character font (high-ASCII values)
ESC $	27 36	1B 24	Switch back to normal font

Apple® ImageWriter™ II
Quick Reference Card

DIP Switch Settings

The factory setting for each switch is in **boldface**. All of the DIP switch settings for SW1 can be overridden with software commands; there are no software commands corresponding to SW2.

Language	SW1-1	SW1-2	SW1-3
American	**Open**	**Open**	**Open**
Italian	Closed	Open	Open
Danish	Open	Closed	Open
British	Closed	Closed	Open
German	Open	Open	Closed
Swedish	Closed	Open	Closed
French	Open	Closed	Closed
Spanish	Closed	Closed	Closed

Form Length	SW1-4
11 Inches	**Open**
12 Inches	Closed

Perforation Skip	SW1-5
Inactive	**Open**
Active	Closed

Characters per Inch and Dots per Inch	SW1-6	SW1-7
10 cpi (pica)	Open	Open
12 cpi (elite)	**Closed**	**Open**
17 cpi (ultracondensed)	Open	Closed
160 dpi (elite proportional)	Closed	Closed

Line Feed on Carriage Return	SW1-8
No LF after CR	**Open**
LF after CR	Closed

Baud Rate	SW2-1	SW2-2
300	Open	Open
1200	Closed	Open
2400	Open	Closed
9600	**Closed**	**Closed**

Protocol	SW2-3
Hardware Handshake	**Open**
XON/XOFF	Closed

Option Card	SW2-4
Disabled	**Open**
Enabled	Closed

Hammer Timing	SW2-5*	SW2-6*
Factory setting of hammer-fire timing.		

*These two switches are factory-set to optimize bidirectional dot placement. Note how they are set and write the settings in the space above. **Do not change these settings.**

Software Switch Commands

Command	Decimal	Hexadecimal	Function
ESC Z *a b*	27 90 *da db*	1B 5A *ha hb*	Set switches corresponding to bit patterns of *a* and *b* to Open (Off).
ESC D *a b*	27 68 *da db*	1B 44 *ha hb*	Set switches corresponding to bit patterns of *a* and *b* to Closed (On).

SW	Function	Z-Setting	D-Setting	Default
A-1	Language character set	SW1-1 Open	SW1-1 Closed	Per SW1-1
A-2	Language character set	SW1-2 Open	SW1-2 Closed	Per SW1-2
A-3	Language character set	SW1-3 Open	SW1-3 Closed	Per SW1-3
A-4	Not used			
A-5	Soft-select response	Enabled	Disabled	Disabled
A-6	LF when line is full	No LF	Add LF	No LF
A-7	Print commands	CR only	CR, LF, FF	CR, LF, FF
A-8	Automatic LF after CR	CR only	CR plus LF	Per SW1-8
B-1	Slash zero	Don't slash	Slash	Don't slash
B-2	Not used			
B-3	Perforation skip	Skip	Don't skip	Per SW1-5
B-4	Not used			
B-5	Not used			
B-6	8th data bit	Include	Ignore	Ignore
B-7	Not used			
B-8	Not used			

Alternate Language Fonts

American	#	$	@	[	\	]	^	`	{	|	}	~
British	£	$	@	[	\	]	^	`	{	|	}	~
German	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
French	£	$	à	°	ç	§	^	`	é	ù	è	¨
Swedish	#	$	@	Ä	Ö	Å	^	`	ä	ö	å	~
Italian	£	$	§	°	ç	é	^	ù	à	ò	è	ì
Spanish	£	$	§	¡	Ñ	¿	^	`	°	ñ	ç	~
Danish	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Decimal	35	36	64	91	92	93	94	96	123	124	125	126
Hexadecimal	$23	$24	$40	$5B	$5C	$5D	$5E	$60	$7B	$7C	$7D	$7E

To type a Spanish hyphen, press the single quotation mark key.

Character Pitch

Command	Decimal	Hexadecimal	Characters per Inch and Dots per Inch
ESC n	27 110	1B 6E	9 cpi (extended)
ESC N	27 78	1B 4E	10 cpi (pica)
ESC E	27 69	1B 45	12 cpi (elite)
ESC e	27 101	1B 65	13.4 cpi (semicondensed)
ESC q	27 113	1B 71	15 cpi (condensed)
ESC Q	27 81	1B 51	17 cpi (ultracondensed)
ESC p	27 112	1B 70	144 dpi (pica proportional)
ESC P	27 80	1B 50	160 dpi (elite proportional)
CTRL-N	14	0E	Start double-width printing
CTRL-O	15	0F	Stop double-width printing (default)

Proportional Character Spacing

Command	Decimal	Hexadecimal	Function
ESC s *n*	27 115 *dn*	1B 73 *hn*	Set dot spacing to *n* for all characters (*n* = 0 to 9)
ESC *n*	27 *d*	1B *h*	Insert *n* dot spaces (*n* = 1 to 6)

Text Attributes

Command	Decimal	Hexadecimal	Function
ESC X	27 88	1B 58	Start underline
ESC Y	27 89	1B 59	Stop underline (default)
ESC !	27 33	1B 21	Start boldface
ESC "	27 34	1B 22	Stop boldface (default)
CTRL-N	14	0E	Start double-width printing
CTRL-O	15	0F	Stop double-width printing (default)
ESC w	27 119	1B 77	Start half-height text
ESC W	27 87	1B 57	Stop half-height text (default)
ESC x	27 120	1B 78	Start superscript text
ESC y	27 121	1B 79	Start subscript text
ESC z	27 122	1B 7A	Stop either superscript or subscript text (default)
ESC D CTRL-@ CTRL-A	27 68 0 1	1B 44 00 01	Slash zeros
ESC Z CTRL-@ CTRL-A	27 90 0 1	1B 5A 00 01	Don't slash zeros (default)

Page Formatting

Command	Decimal	Hexadecimal	Function
ESC L *nnn*	27 76 *ddd*	1B 4C *hhh*	Set left margin at column *nnn* (default is *nnn* = 000)
ESC H *nnnn*	27 72 *dddd*	1B 48 *hhhh*	Set page length to *nnnn*/144 inch (*nnnn* = 0001 to 9999)

Print Head Motion

Command	Decimal	Hexadecimal	Function
CTRL-M	13	0D	Carriage return
ESC >	27 62	1B 3E	Unidirectional printing
ESC <	27 60	1B 3C	Bidirectional printing (default)
CTRL-H	8	08	Backspace one character
ESC (*aaa,bbb,...,nnn.*	27 40 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 28 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Set horizontal tabs at positions *aaa,bbb,...,nnn*
ESC u *nnn*	27 117 *dndndn*	1B 75 *hnhnhn*	Add one tab stop at column *nnn*
CTRL-I	9	09	Move print head to next tab
ESC) *aaa,bbb,...,nnn.*	27 41 *dadada* 44 *dbdbdb* 44 ... 44 *dndndn* 46	1B 29 *hahaha* 2C *hbhbhb* 2C ... 2C *hnhnhn* 2E	Clear tabs at positions *aaa,bbb,...,nnn*
ESC 0	27 48	1B 30	Clear all tabs
ESC F *nnnn*	27 70 *dddd*	1B 46 *hhhh*	Place print head *nnnn* dot columns from left margin

Paper Motion

Command	Decimal	Hexadecimal	Function
CTRL-J	10	0A	Feed paper one line (LF)
CTRL-_ *n*	31 *d*	1F *h*	Feed 1 to 15 lines of blank paper (*n* = 1 to 9, :, ;, <, =, >, or ?)
ESC A	27 65	1B 41	6 lines per inch (default)
ESC B	27 66	1B 42	8 lines per inch
ESC T *nn*	27 84 *dd*	1B 54 *hh*	Distance between lines to be *nn*/144 inch (*nn* = 01 to 99)
ESC f	27 102	1B 66	Forward line feeding (default)
ESC r	27 114	1B 72	Reverse line feeding
CTRL-L	12	0C	Feed paper to next TOF
ESC v	27 118	1B 76	Set TOF to current position
ESC D CTRL-@ CTRL-D	27 68 0 4	1B 44 00 04	Disable perforation skip
ESC Z CTRL-@ CTRL-D	27 90 0 4	1B 5A 00 04	Enable perforation skip
ESC O	27 79	1B 4F	Paper-out sensor off
ESC o	27 111	1B 6F	Paper-out sensor on (default)

Automatic Carriage Returns and Line Feeds

Command	Decimal	Hexadecimal	Function
ESC I 1	27 108 49	1B 6C 31	No CR inserted before LF and FF
ESC I 0	27 108 48	1B 6C 30	Insert CR before LF and FF (default)
ESC D {@} CTRL-@	27 68 128 0	1B 44 80 00	Add automatic LF after CR ({@} is a high-ASCII CTRL-@)
ESC Z {@} CTRL-@	27 90 128 0	1B 5A 80 00	No LF after CR ({@} is a high-ASCII CTRL-@)
ESC D SPACE CTRL-@	27 68 32 0	1B 44 20 00	Add LF when line is full
ESC Z SPACE CTRL-@	27 90 32 0	1B 5A 20 00	No LF added when line is full (default)

Graphics

Command	Decimal	Hexadecimal	Function
ESC G *nnnn*	27 71 *dddd*	1B 47 *hhhh*	Print line of graphics corresponding to following *nnnn* data bytes
ESC S *nnnn*	27 83 *dddd*	1B 53 *hhhh*	Same as ESC G
ESC g *nnn*	27 103 *ddd*	1B 67 *hhh*	Print line of graphics corresponding to following *nnn* × 8 data bytes
ESC V *nnnn c*	27 86 *dddd dc*	1B 56 *hhhh hc*	Print *nnnn* repetitions of dot column specified by *c*
ESC F *nnnn*	27 70 *dddd*	1B 46 *hhhh*	Place print head *nnnn* dot columns from left margin
CTRL-N	14	0E	Start double-width printing
CTRL-O	15	0F	Stop double-width printing

Color Printing

Command	Decimal	Hexadecimal	Function
ESC K 0	27 75 48	1B 4B 30	Black (default)
ESC K 1	27 75 49	1B 4B 31	Yellow
ESC K 2	27 75 50	1B 4B 32	Magenta
ESC K 3	27 75 51	1B 4B 33	Cyan
ESC K 4	27 75 52	1B 4B 34	Orange (Yellow and Magenta)
ESC K 5	27 75 53	1B 4B 35	Green (Yellow and Cyan)
ESC K 6	27 75 54	1B 4B 36	Purple (Magenta and Cyan)

Miscellaneous

Command	Decimal	Hexadecimal	Function
ESC Z @ CTRL-@	27 90 64 0	1B 5A 40 00	Only CR causes printing
ESC D @ CTRL-@	27 68 64 0	1B 44 40 00	CR, LF, FF cause printing (default)
ESC R *nnn c*	27 82 *ddd dc*	1B 52 *hhh hc*	Repeat character *c nnn* times (*nnn* = 001 to 999)
CTRL-X	24	18	Erase current line from print buffer
ESC D CTRL-@ SPACE	27 68 0 32	1B 44 00 20	Ignore 8th data bit (default)
ESC Z CTRL-@ SPACE	27 90 0 32	1B 5A 00 20	Include 8th data bit
ESC c	27 99	1B 63	Reset defaults
ESC D CTRL-P CTRL-@	27 68 16 0	1B 44 10 00	Software-select disabled (default)
ESC Z CTRL-P CTRL-@	27 90 16 0	1B 5A 10 00	Software-select enabled
CTRL-Q	17	11	Select printer (default)
CTRL-S	19	13	Deselect printer
ESC ?	27 63	1B 3F	Send ID string

Tell Apple About Your ImageWriter II Reference Manual

- Would you like to tell Apple what you think about this manual? After you have had an opportunity to use this manual we would like to hear from you. You can help us to improve our manuals by responding to the questionnaire below and marking the appropriate boxes on the card at the right with a #2 pencil. If you have more than one response to a question, mark all the boxes that apply. Please detach the card and mail it to Apple.

- Please contact your authorized Apple dealer when you have questions about your Apple products. Dealers are trained by Apple Computer and are given the resources to handle service and support for all Apple products. If you need the name of an authorized Apple dealer in your area, call toll-free: 800-538-9696.

1. What is your overall rating of this manual? (**1**=low . . . **5**=high)
2. How would you rate the physical appearance of this manual? (**1**=low . . . **5**=high)
3. Where did you buy this manual? (**1**=dealer, **2**=bookstore, **3**=other)
4. Was it easy to find a copy of this manual? (**1**=yes, **2**=no)
5. Which Apple computer are you using? (**1**=Apple II family, **2**=Macintosh family)
6. How much experience have you had with computers? (**1**=no experience . . . **5**=lots of experience)
7. If you program, why do you program? (**1**=self, **2**=employer, **3**=sale, **4**=other)
8. How much of this manual have you read? (**1**=entire manual, **2**=specific chapters, **3**=reference only)
9. How easy was the manual to read and understand? (**1**=difficult . . . **5**=very easy)
10. How would you rate the organization of this manual? (**1**=poor . . . **5**=excellent)
11. How easy was it to find the information you needed? (**1**=difficult . . . **5**=very easy)
12. To what degree did the technical material in the manual meet your expectations? (**1**=low . . . **5**=high)
13. What did you like best about this manual?

14. What did you like least about this manual?

15. Please describe any errors or inconsistencies you may have encountered in this manual. (Page numbers would be helpful.)

16. What suggestions do you have for improving this manual?

ImageWriter II Technical Reference Manual

> $19.95 FPT
USA

The Official Publication from Apple Computer, Inc.

Written and produced by the people at Apple Computer, this manual is the definitive guide to the advanced features of the ImageWriter™ II printer. If you are an experienced computer user and you want to write your own program to use the ImageWriter II, or if you are running a program that can be configured for a specific printer, this manual is for you.

Included in this fact-filled book are the following:

- General principles for using printer control codes in your programs, and programming tips for BASIC, Pascal, Logo, and assembly language.
- Descriptions of the 12 hardware and 10 software switches that control printer functions such as communication rate, foreign-language characters, and the default character pitch.
- A discussion of more than 80 printer commands that select character sets; control character features, such as underlining and boldfacing; determine format, such as character pitch, left margin setting, and tabs; and set other functions, such as direction of paper motion and software resets.
- Explanations of how to design and load into the printer's memory a set of custom characters that you can then integrate with other text.
- A description of how to program graphic designs and send graphics and color printing commands to your ImageWriter II.
- Appendixes that include a summary of all the ImageWriter II commands, an ASCII table, a complete set of dot patterns for ImageWriter II characters, printer specifications, and a description of the ImageWriter II serial interface.

For serious computer users and system managers, the *ImageWriter II Technical Reference Manual* is the essential and comprehensive reference to the most advanced features of this state-of-the-art printer.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
408 996-1010
TLX 171-576

030-2003-A

Printed in U.S.A.

Addison-Wesley Publishing Company, Inc.

ISBN 0-201-17739-0