# PaperClip

PROFESSIONAL WORD PROCESSOR ®

# PaperClip

## Professional Word Processor

**ATTENTION**
This program no longer
requires a key for the
joystick port.

Thank you for purchasing PaperClip.

We believe that PaperClip is one of the best micro-computer wordprocessors available. PaperClip is easy to learn and use, yet powerful. We think you will find it to be an exceptional value.

BATTERIES INCLUDED
30 Mural Street
Richmond Hill, Ontario
Canada, L4B 1B5
(416) 881-9941

**PaperClip written by Steven Douglas**
**PaperClip User's Manual by Steven Douglas and Keith Hope**
**Produced by Batteries Included**

Fourth Printing March, 1986

# PAPERCLIP USER'S MANUAL

## LIMITATIONS OF WARRANTY AND LIABILITY

Batteries Included, or any dealer or distributor distributing this product, makes NO WARRANTY, EXPRESS OR IMPLIED, with respect to this manual, the related floppy disks and any other related items, their quality, performance, merchantability, or fitness for any particular use. It is solely the purchaser's responsibility to determine their suitability for any particular purpose.

Batteries Included will in no event be held liable for direct, indirect or incidental damages resulting from any defect or omission in this manual, the floppy disks, or other related items and processes, including but not limited to any interuption of services, loss of business or anticipatory profit, or other consequential damages.

THIS STATEMENT OF LIMITED LIABILITY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Batteries Included neither assumes nor authorizes any other person to assume for it any other warranty or liability in connection with the sale of its products.

## PRODUCT IMPROVEMENTS

Batteries Included reserves the right to make corrections or improvements to this manual and to the related floppy disks at any time without notice and with no responsibility to provide these changes to purchasers of earlier versions of its products.

## NOTICE OF REGISTERED TRADEMARK

Reference is made in several places in this manual to Commodore, CBM, PET, and Commodore 64 which are registered trademarks of Commodore Business Machines Inc.

# The PaperClip Word Processor

PaperClip is designed for ease and simplicity in use, yet is powerful enough to tackle almost any wordprocessing job. Many advanced features not found with micro-computer word processors are incorporated in PaperClip. The result is a document which reads better, being created in less time.

Just look at some of the things PaperClip can do for you...

- Create and edit text quickly.
- Store text on cassette or disk.
- Correct spelling mistakes easily with just a few keystrokes.
- Effortlessly move text, anywhere, with a touch of the keys.
- Re-arrange words, phrases, sentences, paragraphs and columns.
- Transfer text, even between documents.
- Produce personalized form letters with speed and accuracy.
- Review the disk directory without erasing text.
- Column commands allow simple editing of charts and tables.
- Horizontal scrolling - create wide documents with ease.
- Advanced sorting capabilities. Put everything in order.
- Edit and use data created by Database Management, Electronic Spreadsheet and telecomunications programs.
- Make complicated financial typing easy with numeric tabs.
- Arithmetic. Add and subtract, place totals anywhere in text.
- Printer support system - use all your printer's features... Built in commands for underlining, boldface, italics, super/subscript, change pitch, line spacing...

# Table of Contents

## Getting Started

## Tutorial

## Simple Editing

## Loading and Saving Text

## Disk Directories

## Moving Text

## Searching And Modifying Text

## Miscellaneous Commands

## Formatting Printed Output

## Linked Files

## Embedded Format Characters

## Outputting The Text

## Form Letters

## Advanced Editing Features

## Column Manipulation

## Advanced Column Functions

## Appendices

# Getting Started

## What You Need To Use PaperClip

PaperClip is designed for the Commodore computer and is compatable with most 80 column display adapters, disk and printer interfaces and can be used with both color and black and white displays.

Standard PaperClip works with the following Commodore computers:
PET or CBM (minimum 32K, Basic4.0)
CBM 8096
SuperPET SP9000

PaperClip Expanded can be used with the CBM 8096 and SuperPET computers to allow editing of large documents.

PaperClip 64/128 works with the Commodore 64 and 128 computers.

PaperClip can use any Commodore disk drive, including models 1541, 1571, 1572, 2031, 4040, 8050, 8250, 9060, 9090. The Commodore cassette unit may also be used, although some functions are not available. PaperClip will work with any properly interfaced printer, Commodore manufacture or other, but it may be necessary to create a printer file using the printer manual and our simple printer setup program. We will come back to setting up your printer later.

Before using PaperClip for the first time it is essential that you read APPENDIX D on pages 90 & 91 in this manual. Please read this section now before proceeding.

## How To Use This Manual

This book is designed to be both a self instructional guide and a reference manual. PLEASE READ IT. A few moments now can save a lot of frustration later.

On the next page you will find a list of special keys used throughout this manual. Because of differences in the operation of PaperClip on various computers, we have adopted a set of standard names for certain actions and keyboard keys. Take a moment to familiarize yourself with these.

The first section is a tutorial on simple editing. Please read it. PaperClip is easy to use once you have the basics. The following sections will show you how to use all the features PaperClip has to offer.

The appendices contain information which will be useful to all PaperClip owners. In particular, have a look at the following:

| | |
|---|---|
| Appendix A | - PaperClip Commands |
| Appendix B | - PaperClip Directives |
| Appendix C | - PaperClip Error Messages |
| Appendix F | - Utility programs for general houskeeping |
| Appendix G | - Making copies of a diskette with a single drive. |
| Appendix K | - Solving problems |
| Appendix Q | - Common Disk Errors |
| Appendix R | - The PaperClip Diskette |
| Appendix S | - Diskette and Cassette Care |
| Appendix U | - Notes for Commodore 128 Owners |

## Special Terminology

Press: This means strike and release the single key which is specified immediately after the colon. For example:

Press: **N** [strike the **N** key on your keyboard]

Press: **RETURN** [strike the key labeled **RETURN**]

All keys pressed should be unshifted. If a key is to be shifted it will be specified.

Press: **(shift) C** [Hold down the shift key and press the **C** key.]

Type: This means type the characters following the colon. For example:

Type: **The lazy dog lay down.**

This specifies type the full sentence **The lazy dog lay down.** on your keyboard, including all capitals and punctuation.

## Special Keys

**CTRL** The **CTRL** key is on the left hand side of the keyboard.
On PET and CBM computers the **CTRL** key is labeled **OFF/RVS**.

**ESCAPE** The **ESCAPE** key is the ← key at the top left corner of the keyboard.
On CBM computers it is labeled **ESC**.

✓ The ✓ key is the £ key near the top right corner of the keyboard.
On PET and CBM computers it is the \ (backslash) key.

**TAB** The **TAB** key is the **RUN/STOP** key on the left side of the keyboard.
On CBM computers it is the **TAB** key.

**INSERT** The **INSERT** key is the C= (Commodore logo) key at the bottom left corner of the keyboard.
On PET and CBM computers it is the **(shift) OFF/RVS** key.

## Setting The Display Color

To set the Display Color on the Commodore64 tap the key until the desired color appears.

**(shift) f2** — text color
**(shift) f4** — background color
**(shift) f6** — border color

## Loading PaperClip Into The Commodore64

1) Place a *copy* of the PaperClip diskette into the disk drive. If you have a dual disk drive place the diskette in drive Ø.

2) Make sure the PaperClip key is inserted in *Control Port 1* on the right side of the computer. (For PaperClip128 use Port 2.)

3) Type: **load "Ø :*",8**
For PaperClip128 Type: **load "Ø : paperclip128?",8**
Press: **RETURN**

```
SEARCHING FOR Ø:*
LOADING
READY.
```

4) Type: **run**
Press: RETURN

After a few moments the program will sign on with the copyright notices. If PaperClip does not sign on then reset the computer, make sure the key is firmly in place and try again. If the program does not load after two or three tries then review the appendix D Unpacking PaperClip.

# Lesson 1

## The Cursor

When PaperClip starts, the screen will be clear except for the top two lines and a small rapidly blinking box in the upper left corner. This small box is called the cursor.

PaperClip uses the cursor to indicate where the next key you type on the keyboard will show up. Try this (don't worry about any typing errors).

Type: **We are going to learn how to use the cursor.**

Notice that as you type, the cursor moves along the line just ahead of your typing. The cursor always indicates where the next key will appear.

Press: **CLR/HOME** [the key marked with **CLR** and **HOME**]

The cursor will return to the upper left hand corner. This is known as the "home" position. The **CLR/HOME** key will always move the cursor to the home position. Press the **SPACE** key a few times. Unlike a typewriter, the **SPACE** key *erases* text as it moves the cursor along. We can move the cursor without erasing text by using the **CRSR** keys.

Press: **CRSR⇒** [the key with the left and right arrows]

Try pressing it several more times. Now the cursor is moving over the text without altering it.

Type: **This is a new start**

The new text has replaced the old text. Now try this:

Press: **(shift) ⇐CRSR**

Keep pressing this key and move the cursor to the first **T** in **This is a new start**.

Press: **CRSR⇓** [the key with the up and down arrows]

This time the cursor moved down one line.

If you press (shift) **CRSR⇑** , the cursor will move up. Move the cursor about the screen to get used to its motion. If any of the cursor keys are held down, the cursor will move rapidly.

Remember that the cursor can be moved in all four directions.

| | |
|---|---|
| Right: | **CRSR⇒** |
| Left: | **(shift) ⇐CRSR** |
| Down: | **CRSR⇓** |
| Up: | **(shift) CRSR⇑** |

## The Tab and Status lines

The top line of the screen is called the tab line. Usually it indicates where tab stops have been set. Sometimes it is used to display a question when PaperClip needs more information about a command. The second line is called the Status line. It tells you what PaperClip is going to do next. Notice the right end of the Status line. The numbers there will always indicate where the cursor is. **C** stands for column and **L** is the line number.

# Lesson 2

## The HOME Key

The HOME key has two funtions. The first time you press it, the cursor will move to the upper left corner of the screen. If you press it a second time, the cursor will move to column one, line one. So, if you want to move back to the beginning of text just press the CLR/HOME key twice.

Press: CLR/HOME [cursor will jump to the upper left corner]
Press: CLR/HOME [cursor moves to column 1, line 1]

## The CTRL key

Press: CTRL

The tab line will read **Control option?** The main cursor will stop blinking and a second, non-blinking cursor will appear on the tab line. This means that PaperClip is waiting for you to enter a command. Most functions in PaperClip are initialed by pressing CTRL, then pressing a key to indicate what command you want to use. If you accidently press CTRL, or you do not want to finish a command, simply press CTRL a second time to cancel. Try the command below.

Press: **E**
Press: RETURN

The tab line should display the question **Are you sure?** Below, on the Status line, the message **Erase Lines** indicates that PaperClip is about to erase all text below the cursor including the line the cursor is on.

Press: **Y**
Press: RETURN

If you did not want to erase the text, pressing the CTRL key or answering **N** (for "No") would leave the text intact.

# Lesson 3

## Inserting And Deleting

Type: **There is a dog.**

Move the cursor to the **a** using the cursor keys.

Press: INST/DEL

The entire line moved one space to the left. Try it again. The letter just to the left of the cursor disappears each time you press INST/DEL.

Press: **(shift)** INST/DEL

Each time you press **(shift)** INST/DEL, the line will move to the right one space. This can be used whenever you wish to insert some space in the middle of a sentence.

## Insert Mode

Put the **s** and the space back in the sentence. Move the cursor over the **d** in **dog**.

Press: INSERT [the commodore logo key — bottom left of keyboard]

The status line should display the message ***INS***. This shows that PaperClip is in *insert mode.*

Type: **brown**
Press: SPACE

Now the sentence should read: **There is a brown dog.** While insert mode is active, PaperClip makes room for new text as it is typed in.

Press: INSERT

Insert mode is now cancelled. Press the INSERT key a few times while watching the status line. The INSERT key works like a an on/off switch. Make sure that the ***INS*** message is not displayed before continuing.

# Lesson 4

## Loading A Document

Press: CTRL
Press: L

The Status Line displays the message **Load File**. The tab line will change to **File name?**.

Type: **sample document**

The load command tells PaperClip which document we want to fetch. PaperClip is specific about document names: the words **Sample** and **sample** are not the same to PaperClip — one starts with an upper-case **S** and the other starts with a lower-case **s**.

Press: RETURN

The screen will clear and the disk drive will become active. The document called **sample document** stored on the PaperClip disk will be retrieved and loaded into PaperClip. If the file does not load and an error message appears then go back to the start of this Lesson, check that the PaperClip diskette is in the disk drive and make sure that the name **sample document** is correctly entered. Once the file has loaded, inspect the text.

Notice that at the end of a display line, words may be broken up. This is done automatically. When you are creating a document and come to the edge of the screen the cursor will automatically drop to the next line. Don't worry about this — when PaperClip prints the document everything will look fine and words will not be broken up. Notice the ← (left arrow) symbol located at the end of each paragraph. When entering text with PaperClip, do not press RETURN at the end of each line, just keep on typing and let the words wrap around. Press the RETURN key only at the end of a paragraph. When you do, PaperClip will place the ← symbol there to indicate the end of the paragraph. Each line in a document, have either some text or a ← symbol on it. There should never be a line in the middle of your text with absolutely nothing on it.

# Exercise

Press: CLR/HOME

Use the CRSR keys and position the cursor to the second **d** in **movedd**.

Press: INST/DEL

Notice that the first **d** is deleted and the rest of the paragraph is moved back one space. We have just made our first correction.

Move the cursor over the letter **u** in the word **spellung** in the second sentence.

Type: **i**

Corrections can be typed directly over mistakes.

Position the cursor over the **b** in **neverbe** in the next sentence.

Press: **(shift)** INST/DEL

Notice that each time you press INST/DEL a space is inserted under the cursor and the whole paragraph moves over. The words **never** and **be** are now separated.

Move the cursor to the space just before the word **Space** in line two.

Press: INSERT

Remember that pressing the INSERT key turns on the insert mode.

Type: **This is a sample with the insert function on.**

As we type in the new line, text is automatically shifted as each new character is typed. Notice that all insertion and deletion only affects text up to the ← symbol which signifies the end of the paragraph.

Press: INSERT

Pressing the INSERT key causes the insert mode to turn off.

# Lesson 5

## Saving A Document.

Any document can be saved for future reference or update. We will save the sample document with the corrections that have been made during the exercise.

Press: CTRL (Remember you must have the copy of PaperClip in the drive at this
Press: S point because you cannot save on the original.)

The tab line will change to **File name?** and the non-blinking cursor will be waiting for a response. **Save File** will be displayed in the Status Line.

Type: **test one**

We are giving this document a name. A name must be assigned to the document when it is to be saved. This name will be used later to retrieve the document. Make sure all letters are entered in lower case. If a mistake has been made on entry of the name use the **CRSR** and **INST/DEL** keys to make corrections.

Press: **RETURN**

The tab line will prompt **Drive number?**

Press: Ø [the digit zero]

This indicates that we want to save the document on drive zero.

Press: **RETURN**

The disk drive becomes active. Notice that the line counter increments as the file is saved. When the document is saved the disk drive light goes out and the cursor will start blinking.

# Lesson 6

## Scrolling

Press: **CLR/HOME**
Press: **CLR/HOME**

Pressing **CLR/HOME** tells PaperClip to move the cursor to the home position (top left hand corner) on the screen. PaperClip64 has space for hundreds of lines of text, but the screen can only display twenty three lines at any one time. Move the cursor down to line twenty three. Press **CRSR⇓** a few times — each time we do this a line of the text disappears off the top of the screen and a new one appears at the bottom. This is called *scrolling.*

Press: **CLR/HOME**

Looking at the status line we can see that the cursor is not on line one.

Press: **CLR/HOME**

The cursor now moves to line one, column one and the text which scrolled off the screen has reappeared.

Press: CTRL
Press: +

Notice that each time this key is pressed all text moves down one line and a <-, the RETURN symbol, is placed on that line.

Press: CTRL

Any time **CONTROL OPTION?** appears pressing CTRL will cancel it.

# Lesson 7

## Setting Left And Right Margins

Press: ✓

Remember that the ✓ key on the Commodore64 is the £ key. Pressing this key produces a ✓ character on the screen. The ✓ is used to give PaperClip special information about how the printed text is to be shaped. This information is used only when PaperClip is printing a document.

Type: **lm2:rm38** [don't forget the colon]

This sets the left and right margins — the left margin is set at 2 and the right margin is set at 38.

Press: RETURN

The top line of the screen should look like this:

✓ lm2:rm38 <-

# Lesson 8

## Outputting Text To The Screen

Press: CTRL
Press: **V**

The control option that we requested is **V** for video. The status line message is:

**Video Output**.

The tab line reads: **Fill variable blocks? N**

Press: RETURN

The tab line reads: **Page? 1**

Press: RETURN

The tab line reads: **Global? N**

Press: RETURN

The screen will clear momentarily and then formatted text will start scrolling up the screen. At the end of text the scrolling stops and a thin line indicates the end of the page. Notice that words at the edge of the screen are not broken up. When text is output to video or to the printer (as we shall see later) the text is formatted. Pressing any key will cancel the video output mode.
Now repeat the process but this time when the screen starts to scroll, stop the scrolling by tapping a key. Let the screen display a few lines before you press a key to stop the scrolling. If the line indicating the end of text appears then tap a key and try again.

Now, with PaperClip in video output mode, try this:

Press: **CRSR⇒**

Notice that each time this key is pressed the text on the screen moves to the left. This allows you to view the formatted text even if margins chosen are greater than the screen width (4Ø characters). Holding down the **⇐CRSR⇒** key will cause the horizontal scrolling to repeat.

Press: **(shift) ⇐CRSR**

This causes the text to scroll to the right.

Press: **CTRL**

This will abort the video output mode.

While in the video output mode almost any key can be used to start and stop the screen scroll. The two exceptions are the **CTRL** key (this key will abort the video output mode) and the **⇐CRSR⇒** key which will stop the scrolling downward and allow horizontal scrolling.

# Lesson 9

## Formatting The Text.

Thus far we have seen a bit of the formatting capabilities (setting left and right margins) and have discussed the ✓ character which is put on the screen by pressing the **£** key.

Press: **CTRL**
Press: **V**
Press: **RETURN**
Press: **RETURN**
Press: **RETURN**

Let the screen scroll through. The left margin is still set at two and the right margin is set at thirty-eight.

## Justification

Looking at the formatted text you can see that the left margin is even but the right margin is ragged. Justification means that we want the right hand margin to be even as well.

Press: CLR/HOME
Press: CLR/HOME

Press: CTRL
Press: +
Press: CTRL

This should open up a line at the top of the screen.

Press: ✓
Type: **ju1** [the digit 1, not the letter I or l]
Press: RETURN

The first line in text should look like this:

✓ju1←

Press: CTRL
Press: **V**
Press: RETURN
Press: RETURN
Press: RETURN

Let the screen scroll through. Notice that now the text is lined up evenly at both margins.

Press: CLR/HOME
Press: CLR/HOME

## Centering

Move the cursor so that it is in column 1 between the first two paragraphs.

Press: CTRL
Press: +
Press: +
Press: +
Press: +
Press: CTRL

There should be four blank lines between the paragraphs.

Press: ✓
Type: **cn1** [the digit 1]
Press: RETURN
Type: **Paragraph Two**
Press: RETURN
Press: ✓
Type: **cnØ** [the digit Ø]
Press: RETURN

The three lines should read:

**✓cn1←** [the digit 1]
**Paragraph two←**
**✓cnØ←** [the digit Ø]

Press: **CTRL**
Press: **V**
Press: **RETURN**
Press: **RETURN**
Press: **RETURN**

Notice that the phrase **Paragraph Two** is centered between the two paragraphs.

Press: **CLR/HOME**
Press: **CLR/HOME**

Now we will save the modified text.

Press: **CTRL**
Press: **S**
Type: **new test**
Press: **RETURN**
Press: **Ø** [the number zero]
Press: **RETURN**

We have saved our document on the diskette with the file name **new test**.

# Lesson 10

## Viewing The Disk Directory

Press: **CTRL**
Press: **Ø** [the number zero]

The Status Line displays **Disk Directory**. The screen will clear and the bottom line of the screen will display:

**Disk Drive #Ø:paperclip**

This is the diskette *header*. It tells us that drive zero has a diskette called **paperclip**.

Hold down the **CRSR⇓** key for a few seconds.

Information on a diskette is stored as files. The different files stored on the diskette are displayed. Beside each directory entry is the file type. For example, each document is a separate file. **prg** is used for program files and your documents. The numbers on the right indicate how many blocks each file has used. Each diskette, when empty, has 664 blocks available to store data. Hold down the space key until the screen stops scrolling. The last line will tell you how many blocks are free to be used.

The cursor will be flashing on line 1 and the Status Line will display ***DRC***. This means that PaperClip is in directory mode.

Notice that the last two entries in the diskette directory are **test one** and **new test**, the two documents you saved earlier.

Press: CLR/HOME
Press: CLR/HOME

This sequence takes us out of the directory mode and back into the text mode. Viewing the disk directory does not erase the text currently in the computer.

# Lesson 11

## Outputting Text To The Printer

Before PaperClip can print you must choose an appropriate "printer file". Turn to the appendix: Choosing A Printer File, right now and select such a file. Once you have a printer file which works with your printer, put your copy of the PaperClip diskette in the drive, and continue with the lesson below.

Press: CTRL
Press: **L**
Type: **new test**
Press: RETURN
Press: CTRL
Press: **(shift) O** [the letter O]
Press: RETURN

Your printer should start printing the document "new test." If you have a problem, see the appendices Choosing A Printer File, and Solving Problems.

Press: CLR/HOME

Notice that the printed text does not go across the full page. Remember that in "new test" the left margin was set at 2 and the right at 38. Most printers can print at least 8Ø characters across the page.

Position the cursor under the ✓ of ✓cnØ<

Press: ✓
Type: **lm1Ø:rm7Ø**
Press: RETURN

The line should read:

**✓lm1Ø:rm7Ø<**

Press: CTRL
Press: **(shift) O**
Press: RETURN

Notice that the second paragraph took on the new margins. Margins, just like any of the other formatting commands, remain in effect until they are changed.

Press: CLR/HOME

As the output goes to the printer it also prints on the screen. Pressing any key will return to main text mode.

# Simple Editing

## Using TAB Stops

Normally, a thin horizontal line is displayed across the entire top of the screen. However, when a tab is set, a small block will appear over the column where the tab stop has been set. To set a tab, move the cursor to the column on the screen where you wish to set a tab, then:

Press: **(shift)** CLR/HOME

This command will either set or clear a tab stop, depending upon whether a tab stop was previously set at that column. If one was set, then it will be cleared. If there wasn't one set, then one will be set. To move the cursor to a tab stop,

Press: TAB [remember, the TAB key is the RUN/STOP key on the Commodore64]

If none have been set the message **No tab stops set.** will be printed on the tab line when the tab key is struck. This is just one of many messages which will appear from time to time to let you know what is going on. Most functions (discussed later) have prompts which display on the tab line. When a message is printed, the tab line is temporarily removed. The tab line will appear when the next function is performed, such as pressing CTRL, etc. To clear all tab stops:

Press: CTRL
Press: **(shift)** CLR/HOME

## Finding The End Of Text

Press: **(shift)** RUN/STOP

This command automatically moves the cursor to the last character on the last line of text. If the cursor was already past the last character in text then the cursor will not be moved.

## The RETURN Key

As mentioned before, the RETURN key should only be entered at the end of a paragraph. When you do, several things occur. First, any text to the right of the cursor will be erased (except during Insert Mode as explained later). A ← character will be placed at the current cursor position to mark the end of the paragraph. Any other ← symbols on the line will be erased. The cursor will then move to the beginning of the next line.

Press: **(shift)** RETURN

This moves the cursor to column 1 on the next line *without* altering text.

## All Caps Mode

Press: ↑ [the up arrow key, not the CRSR key]

One of the problems with the **Shift Lock** key is that numbers cannot easily be mixed with text. When it is pressed, *all* keys are shifted which means that letters come out in capitals but instead of numbers, other symbols are produced. For this reason, PaperClip offers an All Caps Mode, which is activated or deactivated by pressing the ↑ key. While it is active all letters will be entered in upper case while numbers and punctuation are not affected.

When All Caps is active the letters **LOK** will be displayed on the Status Line. All Caps is turned off by pressing the ↑ key a second time.

## Insert Mode

Press: **INSERT** [the **C=** key in the bottom left of the keyboard

While Insert Mode is active, ***INS*** will be displayed on the Status Line, and text will automatically be inserted as you type. This is particularly useful when you wish to insert a word, sentence or more into a paragraph and you don't know how many characters you are going to insert. Just activate Insert Mode and start typing.

Insert Mode can also be used to split up paragraphs. If when you are editing a paragraph, you decide that the paragraph is too long and should be split into two. Striking the **RETURN** key while Insert mode is active will automatically split the paragraph at the current cursor position. The normal action of the **RETURN** key will still occur, but any characters on the line after the cursor will be moved to the beginning of the next line. To see this, move the cursor to the middle of a paragraph, activate Insert Mode, and press the **RETURN** key.

## CTRL Functions

Paperclip offers a large number of advanced editing features. There are not nearly enough keys on the keyboard to accomodate each of these functions separately so these functions are called up by pressing the **CTRL** key, then the key corresponding to the desired function.

# Loading And Saving Text

## Saving Text On Disk

There are basically two different ways to save and load text. Each has advantages and will be described in turn.

To save the document in memory onto a diskette the command is:

Press: **CTRL**
Press: **S**

When you strike the **CTRL** key, the question **Control option?** will appear on the top line, where the tab line is usually displayed. Also, the cursor in text will stop blinking. Another cursor will appear after the question on the top line. All prompts and questions will act like this with a second cursor appearing on the top line.

You have now started the procedure to save the document in memory onto diskette. However, the actual saving has not started yet. Since documents are stored on diskettes as 'files', you must supply a name for the file before it can be stored. The question **File name?** should now be on the top line. Type in the name you wish to give to the particular document in memory and press the **RETURN** key. The **INSERT**, **DELETE**, **⇐CRSR**, **CRSR⇒** and **HOME** keys still work so you can edit the file name.

Once the file name has been entered, one more question will be asked. That is the drive number on which the file is to be saved. Commodore dual drives each have two slots or "drives". They are referred to as "Drive Ø" and "Drive 1".

Press: **Ø** [you must enter Ø if you have a single slot drive]

If you don't have a diskette in the disk drive, put one in. If you have a Commodore 2Ø31 or 1541 disk drive press **Ø** to indicate the single drive.

Press: **RETURN**

The red light on the drive should light up as the file is saved. While the file is being stored on the diskette, the line number on the screen will change to show the progress of text being saved.

If you enter the save function accidently, press the **CTRL** key to return to edit mode. This is true of all **CTRL** functions. Most of the functions have prompts since they require more information before they can be executed. Striking the **CTRL** key in response to any prompt will abort the command.

During a save operation, the actual text in memory is not the only information which is written out. The tab line is saved as well as the width of the text lines. The reason for saving the text width will be explained in the section on Horizontal Scrolling.

## Saving A Sequential File (disk or tape)

Press: CTRL
Press: **Z**
Type: **document name**
Press: **Y**
Press: RETURN
Press: **∅** (or **1**)
Press: RETURN

You may wonder why there should be two ways of saving a document onto diskette. The reason is that the format of the file generated by CTRL **Z** is standard Pet ASCII. The type of file generated by CTRL **S** has the characters stored in screen format. All of the functions described later assume that the file was saved using CTRL **S**. But there will be times when you want to use CTRL **Z**. This will be true if you want the file to be read by another program such as a database, mail list, etc. Standard Pet ASCII is a more universal format for communication between programs.

There are several errors that could occur during the saving of a file onto diskette. The first such error is not really a mistake. It occurs when there is already a file on the diskette with the name you specified. If so, the save function will not be aborted. Instead, you will be asked **Replace existing file?**. If you respond by pressing: **Y**, and pressing: RETURN to signify **Yes** then the old file will be erased from the diskette and the document in memory will be saved in its place.

If you do not wish to replace the existing file, either press: CTRL or press: **N** and press: RETURN. Remember that CTRL is a universal abort key and can be entered in response to any prompt to abort whatever function is in progress.

The **file exists** error is the most common error and usually occurs when a document has been loaded into text memory and modified. When the updated document is saved with the same file name the **file exists** error will occur. The question is merely a precaution in case the wrong file name was given.

If, at any time during the save operation, the disk drive does not respond to instructions from PaperClip, either the message **Error: Device not present.** or **Error: Timeout on IEEE.** will be printed on the tab line. Both messages mean basically the same thing. Usually it means that the disk drive is not connected to the computer. Check to make sure that the disk drive is turned on. If that is not the problem, it is possible that the drive has a different device number than the one being accessed by PaperClip. Each device connected to the computer has a device number associated with it. PaperClip only knows that it is going to communicate with device 4 or device 8 or whatever. Each device connected to the computer must have a unique device number. Commodore disk drives are set to be device 8 when shipped from the factory. Printers are usually device 4. You may want to connect more than one disk drive to the computer. Since there cannot be two device 8's, one of the disk drives must have its number changed, usually to 9. When PaperClip starts running it assumes that the disk drive is device 8. If your disk drive has a device number other than 8, you will have to change PaperClip's disk device number before you can use PaperClip with the drive.

There are also a number of errors which may be generated by the disk drive itself. When a disk error occurs, the message **Disk Status:** followed by an error message will be printed on the tab line and the save function will be aborted. Check to make sure that there is no write-protect tab on the diskette. This is a small tab which covers a notch on the edge of the diskette jacket. If there is a tab over the notch the disk drive cannot store any information on the diskette. Also make sure that the drive door is closed properly and that the diskette is inserted correctly. If you cannot determine the cause of the error, consult the disk drive manual for a list of possible errors and their causes.

## Saving Text On Cassette Tape

To save a document on cassette tape:

Press: CTRL
Press: **Z**
Press: **N**
Press: RETURN

Before pressing RETURN, position the tape in the tape drive. Make sure there are no buttons pressed on the tape unit and that it is correctly connected. When you press RETURN you will be told to press the record and play buttons on the tape drive. PaperClip will wait until the buttons are pressed. Be sure to press the right ones. Once the RETURN key is pressed, the command must be carried through. *There is no way to abort the SAVE once it has started.*

The file will be stored on the tape with the name specified. Remember that it will overwrite any information which was already there. Also, at the end of each file, an End of Tape marker will be written. The reason for this will be explained in a moment. It is recommended that only one file per tape side be used. This will cut down on the number of files which "disappear".

## Loading Text From Disk

Press: CTRL
Press: L
Type: **document name**
Press: RETURN

Both slots on a dual disk drive will be searched to find the file. If the message **Disk Status: 62, file not found,ØØ,ØØ** is displayed on the tab line then you probably entered the document name incorrectly. PaperClip does offer the ability to do a "screen read", which reduces the chance of misspelling a document name. This is covered under Disk Directories.

All the errors associated with saving a document also apply to loading with the exception of the "file exists" error.

Assuming that no errors occur, the document will be loaded into PaperClip. You will see the text as it is coming in. It will be loaded in starting at the first line in memory. *Any text previously in memory will be erased.*

## Loading A Sequential File

Press: **CTRL**
Press: **J**
Type: **document name**
Press: **RETURN**

You will be asked whether the file is on diskette. If you answer **Y** then the file will be read from the disk drive. If you answer **N** then you will be asked to press the play button on the cassette unit. Once the play button is pressed, the computer will start searching for the file. The computer will stop searching when it either finds the file or encounters an end of tape marker. If it finds the file, it will start to read it into memory. If the End of Tape marker is encountered the load operation will quit. The reason for writing the End of Tape marker in the first place is to provide some protection in case the file name was misspelled, otherwise PaperClip would end up searching the tape forever with no way to abort the function.

## Verifying A Sequential File

PaperClip can verify the integrity of data stored as a sequential file. Verifying means comparing the text in memory with text stored in a file on either tape or diskette. Although it can be used with either tape files or diskette files, the verify function is primarily designed for use with the cassette tape unit. There are two reasons for this. First, cassette tape is not a very reliable storage medium. Second, the verify function can be used to skip past files already on tape before saving the text in memory.

To verify a file on tape or diskette:

Press: **CTRL**
Press: **U**
Type: **document name**
Press: **RETURN**

The verify function behaves like the load function (**CTRL J**), except that instead of bringing the text into memory, it will compare the text with that in memory. Once the file has been verified, either the message **Verify error.** or **Verify ok.** will be printed on the tab line. **Verify error.** indicates that the file contents were not the same as the text in memory.

# Disk Directories

## Displaying Disk Directories

| | |
|---|---|
| Press: CTRL | Press: **Ø** [view a directory of drive Ø] |
| Press: CTRL | Press: **1** [view a directory of drive 1] |
| Press: CTRL | Press: **2** [view a directory of both drives] |
| Press: CTRL | Press: **3** [load a directory into text memory] |

The screen will clear and a message such as **Disk Drive #Ø:** will appear at the bottom. Immediately following the colon will be the "name", the "ID", and the "format type" of the diskette. Usually the format type will be "2A". For further information regarding the format of diskettes, consult the disk drive manual.

Pressing any key other than CTRL will cause the directory to scroll up the screen. Release the key to pause, press **CTRL** to stop. This will halt the directory scroll, but will not return you to text. PaperClip will remain in Directory Mode (explained momentarily). When the last entry in the directory has displayed, the directory scroll will stop but PaperClip will still be in Directory Mode.

Each entry in the directory will consist of three components. The first is the file name. The second is the file type. The file type **prg.** is used to indicate text files or programs. The PaperClip program will be a **prg** file. Another type of file is **seq** which is a "sequential file". The CTRL **Z** function creates this type of file. The third component of a directory entry is the number of blocks used by file. The disk drive stores all data in blocks.

The last entry will read **BLOCKS FREE** = followed by a number. This number is the number of unused blocks on the diskette. When this number reaches Ø, the diskette is full and no more information can be stored on it.

If an asterisk appears next to the number of blocks occupied by a particular file, this means that the file was never finished properly, usually the result of trying to store a file on a diskette which did not have enough room, for instance, there were only 24 blocks free and the file required 26 blocks. If this happens, the diskette should be validated immediately (see the appendix Common Disk Commands). If it is not validated, there is a possibility of losing information when storing a file. *Under no circumstances should you attempt to scratch the file.*

If you choose to view both directories (**CTRL 2**) or to load a directory into text memory (**CTRL 3**), you will be asked **File Name?**. Enter a file name which follows the conventions outlined in the disk drive manual. For example,*specifies all the files; **a***specifies all the files that begin with the letter "a"; * =**s** specifies all the sequential files. Both these commands can be used with single disk drives.

Specifying CTRL **3** to load a directory into is useful to edit, sort, or print a directory. Be forewarned that this PaperClip will load the directory starting at the current position of the cursor. Thus, if your cursor is not at the bottom of the document, some infomation will be lost.

## Directory Mode

When a directory is finished, PaperClip will not return to Edit Mode. Instead the directory will be left on the screen and PaperClip will be in Directory Mode. No text can be entered. The cursor can be moved, and you can scroll up or down through all the entries that have been displayed. PaperClip will retain up to 3ØØ directory entries to scroll through. While in Directory Mode, certain control functions will not work, primarily those which would alter text.

Directory Mode is indicated by the presence of the letters ***DRC*** on the Status Line. To exit Directory Mode, press: **CLR/HOME** twice in succession. The text will be brought back onto the screen.

## Screen Reading

To avoid making typing errors when entering document names and such, PaperClip allows you to read a phrase from the screen into a command. Suppose you wish to load a document named "sample document" and it is in the directory listing on the screen. Simply move the cursor to the beginning of the name — in this case the "s" in "sample". Now,

Press: **CTRL**
Press: **L**

When you are asked for the file name, instead of entering the name,

Press: **RUN/STOP**

The name **sample document** will be copied into the tab line. Reading the file name in this way is known as "screen read". Any time you are asked for information in the tab line, pressing the **RUN/STOP** key will automatically copy whatever name is next to the cursor. Many people place a comment line (see the section on formatting) at the beginning of their files containing the name of the file. Saving the file then becomes simplified, just move the cursor to this name and then use the screen read function.

# Moving Text

## Inserting And Deleting Lines

To delete an entire line from text memory,

Press: **CTRL**
Press: - [minus sign]

The line under the cursor will be deleted and all text lines below the cursor will be moved up. To insert a blank line into text memory,

Press: **CTRL**
press: +

All text under and below the cursor will be pushed down and a blank line will be inserted. A **RETURN** (<-) will automatically be placed at the beginning of the new line. To insert more than one line into text at the current cursor position,

Press: **CTRL**
Press: **I** [Insert]

You will be asked how many lines you wish to insert. A response of **∅** (zero) will abort the function. The maximum number of lines which can be inserted is 255. Enter the required number of lines and press: **RETURN**. The specified number of lines will be inserted at the current cursor position.

## Line Ranges

One of the most powerful capabilities of PaperClip is its ability to allow you to move sections of text about to suit your needs. Suppose for example, you have a paragraph within a document which is out of place, and you want to move it to another place in text. Move the cursor to the first line of the paragraph you wish to move. Now,

Press: **CTRL**
Press: **R** [range]
Press: **RETURN**

This is the "Set Range" function. The line which the cursor was on should now be displayed in reverse field. Use the **(shift) CRSR⇒** and ⇓ **CRSR** keys to illuminate the entire paragraph that you want to move. When you have set the paragraph, press the **RETURN** key. This will terminate the Set Range function.

The Set Range function specifies a Line Range which can then be moved, deleted, copied, etc.

## Transfer Range

To transfer the Line Range to some other part of text, move the cursor to the line where you wish the Range to be moved to. Now,

Press: **CTRL**
Press: **T**
Press: **RETURN**

This will delete the Line Range and insert it at the current cursor position.

## Delete range

To delete the Line Range from text altogether,

Press: **CTRL**
Press: **D**
Press: **RETURN**

The entire Line Range will be deleted.

## Copy Range

To copy a Line Range, position the cursor where you want the copy to appear,

Press: **CTRL**
Press: **C**

Enter the number of copies you want (up to 255 copies of a range may be made) and Press: **RETURN**.

The copy function behaves differently from the transfer function in two ways. First, the Line Range is not inserted at the current cursor position, therefore text already there will be overwritten by the range when it is copied. Second, the old range is not deleted and still exists, so multiple copies can be made.

## Moving Text from File to File

There will be times when you wish to take a paragraph and move it from one document and place it in another. Often, this is the result of wanting to insert some space into text when text is already full or nearly full. Suppose that you want to place a new paragraph right before the last paragraph in a file. You determine that the new paragraph will require about ten text lines, only to discover that there are only five lines left in text memory and this is not enough. To add your paragraph, you must take the last paragraph and move it to another file.

Although it sounds complicated, the procedure is simple. The section you wish to move is saved onto the disk under a temporary name, then deleted from the file it came from. Next, it is appended (inserted) into the file you want it in at the current cursor position.

Move the cursor to the beginning of the section of text which you wish to move. Now, set a Line Range over the section and strike the **RETURN** key. Next,

Press: CTRL
Press: **Q**

and proceed as in a normal text save operation. The only difference is that this time only the line range will be stored on the diskette. The file name that you specify can be a temporary one. This file will contain the contents of the Line Range so that they can be apppended to another file.

Once the Line Range has been saved,

Press: CTRL
Press: **D**
Press: RETURN

to delete it from text. Save the original document on the diskette after doing any other editing required.

What you now have is your first file with the new paragraph inserted, and the line range stored under a temporary file name. The contents of the temporary file are to be added to another file.

## Appending Text On Disk To Text In Memory

To do this, load in the file to which the text is to be added. Move the cursor to the place where you wish to insert the contents of the temporary file. Now,

Press: CTRL
Press: **A**

and proceed as in a load command. There will be one difference this time. Instead of the incoming file overwriting the text already in memory, it will be inserted into memory at the current cursor position.

## Erasing Text

To erase text,

Press: CTRL
Press: **E**
Press: RETURN

At this point you will be asked one of two questions. If there is a Line Range set then you will be asked **Erase range only?** Answering **Y** to this question will erase only the contents of the Line Range. Answering **N** will erase all text from the current cursor position to the end of text.

The difference between erasing a Line Range and deleting a Line Range is that erasing merely overwrites the text with spaces. Deleting leaves text continuous while erasing a Line Range leaves a gap within text.

If there was no Line Range set, you will be asked **Are you sure?** Answering **Y** will delete all text after the cursor. Answering **N** will abort the function.

## Phrases

Although the ability to move Line Ranges offers a great deal of power when editing text, Line Ranges can be inadequate for certain applications. The most common instance is when you have a sentence or group of sentences which you feel should be moved. The problem with Line Ranges is the fact that they are a group of *lines*. The sentence you want to move may not start at the beginning of a line and stop at the end of a line. For this reason, the Set Phrase function is available.

To set a Phrase, move the cursor to the first character of the group of characters (for instance, a sentence) that you wish to move. Now,

Press: **CTRL**
Press: **P**
Press: **RETURN**

The character under the cursor will now be highlighted. Use the cursor keys to highlight the phrase you wish to edit. While setting a phrase, pressing the **S** key will highlight an entire sentence, and the **W** key will highlight words. You are not allowed to set a phrase if the cursor is beyond the end of text memory. Also, a Phrase is confined to one paragraph only and **RETURN** (←) characters cannot be included in the phrase. A phrase cannot be set if a ← occurs to the left of the cursor.

When the sentence or phrase you wish to edit has been highlighted on the screen,

Press: **RETURN**

Now move the cursor to the place in text where you wish to insert the phrase. Next,

Press: **CTRL**
Press: **M** [Move]
press: **RETURN**

A copy of the phrase will be inserted at the current cursor position.

The phrase is not deleted automatically after it is moved. This allows the flexibility to insert it many times into text. To delete a phrase from memory,

Press: **CTRL**
Press: **K** [Kill]
Press: **RETURN**

The phrase will be deleted from text.

If you simply want to delete a phrase, and not move it,

Press: **CTRL**
Press: **Y**
Press: **RETURN**

Now, set the phrase as shown above, then press **RETURN**. The phrase will be deleted. If you change your mind while you are setting the phrase, you can press **CTRL** to abort the function.

## Setting Case In A Phrase

It is also possible to convert a Phrase to all upper-case or all lower-case letters. To set the case of a phrase, set the phrase, then,

Press: CTRL
Press: **(shift) K**

You will then be asked if you wish to change the Phrase to all upper-case letters. Answering **Y** will force all letters within the phrase to upper-case. Answering **N** will change all letters to lower-case. Pressing CTRL will abort the command.

## Common Phrases

Quite often during the creation of a document a phrase will appear many times. To save typing time and reduce errors, these common phrases can be predefined in the text and can be called. Put the cursor on a blank line. The line above *MUST* have a RETURN (<-) character on it.

Press: ✓
Type: **a=phrase**
Press: RETURN

On the screen this should look like:

**✓a=phrase<-**

In the above example **a** can be any alphabetic key either upper or lower case and **phrase** is the common phrase required to be defined. The phrase cannot exceed one screen line in length including the setup characters. An example follows.

Press: ✓
Type: **t=This is a common phrase.**
Press: RETURN

The above defines unshifted **t** to equal the phrase **This is a common phrase.**

Any time during you require the phrase "This is a common phrase."

Press: ESCAPE [the <- key on the top left of the keyboard]

The prompt **KEY?** will appear on the tab line. Press: **t** and the predefined phrase will be typed starting at the current cursor position. All alphabetic characters, unshifted and shifted, may be used for common phrases.

# Searching And Modifying Text

## Global Files

Often, the document you are creating will be too large to fit into PaperClip. The solution to this problem is to split the document into several pieces and store them on disk separately. The document is then chained together and referred to as a Global File. A Local File is the text that is currently in PaperClip. Several of the following functions can operate either Globally or Locally.

Suppose we have four files called "File #1", "File #2", "File #3", and "File #4". These files comprise one document with each file containing one part.

Chaining files together is accomplished by placing a link at the end of each file in the chain, with the exception of the last file. This link indicates the next file in the chain. On the last line of each text segment the ✓ is placed in column 1, followed by the letters **nx** and a colon. After the colon is the name of the next file in the chain, followed by the **RETURN** (←) symbol.

In our example the last line in File #1 would read:

✓**nx:File** #2←

This tells PaperClip that "File #2" contains the next segment of the document. The last line in File #2 would be ✓**nx:File** #3←. The end of File #3 would have ✓**nx:File** #4←. There would be no link at the end of File #4 since it is the last file in the chain.

## Searching Text

To search text for a specific character, word or phrase:

press: **CTRL**
press: **F** [Find]

The tab line will prompt **Search String?**. Enter the Search String and press: **RETURN**. The word processor will then start searching the file in memory for the string in question. If the Search String is found, then the computer will return to Edit Mode with the cursor blinking at the end of the string in text.

To search for the next occurrence of the Search String:

press: **CTRL**
press: **H** [Hunt]

The cursor will then reappear at the end of the next occurrence of the Search String. The difference between the **find** and **hunt** is that the **hunt** command does not ask for the Search String. It uses the previous Search String. The **find** command is used when you want to change the Search String.

If a file link (the ✓**nx:** directive) is encountered at the end of a document then the top line on the screen will prompt **Fetch next file?**. Press: **Y**, press: **RETURN** will load in the next file in the chain and carry on with the search. *Remember that this will erase the document in memory.*

Press: **CTRL** or Press: **N**
Press: **RETURN**

will abort the search.

## Searching And Replacing Text

Another useful feature is the ability to search for a given Search String and replace all occurrences with another string. Suppose you have written a document which in several places refered to "Frederick P. Spnutt". Unfortunately , you have consistently misspelled the name as "Splutt". The problem is easily rectified with the Search and Replace function.

## Local Search And Replace

To enter the Search and Replace function,

Press: **CTRL**
Press: @

You will then be asked for the Search String. Enter the Search String and press: **RETURN**. Next you will be asked for the Replace String. Enter the Replace String and press: **RETURN**. This is the string which will replace the Search String.

The next question will be **More? N**. During a Search and Replace, you can specify up to five Search Strings and five Replacement Strings. If you wish to specify another Search String and another Replace String then,

Press: **Y**
Press: **RETURN**

otherwise

Press: **RETURN**

The next question will be **Query before replacement? N**. If you answer **Y** then each time the Search String is encountered, you will be asked whether or not you want it to be replaced. If you press: **RETURN** then the Search Strings will automatically be replaced.

The next question will be **Global? N.** If you want the Search and Replace to operate over a Global File then,

Press: **Y**
Press: **RETURN**

otherwise

Press: **RETURN**

The Search and Replace operation will start. If no query before replacement was specified then each occurrence of the Search String will be replaced by the Replace String. If query was specified, then when the Search String is encountered, the cursor will appear at the end of the Search String. The prompt at the top of the screen will be **Replace?**.

Press: **Y** or **N**
Press: **RETURN**

to indicate if you want the replacement. If you Press: **CTRL** the operation will abort.

## Global Search And Replace

Follow the instructions for a Local Search and Replace until the status line asks the question **Global? N**

Press: **Y**
Press: **RETURN**

You will then be asked to supply the name of the first file in the chain. Also, you will have the option to have a query before each new file in the chain is called in. Global Search and Replace will now begin with the first file in the chain. If there is a file link at the end of text then the next file will be called in. If a query before the file call was specified then you will be asked whether or not the file is to be called in. Sometimes you may wish to go back and change the text still in memory instead of continuing with the Search and Replace.

When the Search and Replace operation for the file in memory is completed, it will be stored onto the disk. The Global Search and Replace will continue until it is either aborted or the last file has been searched. If during a Global Search and Replace no replacements are made (no occurrences of the Search String were found) then the file will not be saved back to the disk.

A Search and Replace operation can be aborted by pressing: **CTRL**. The operation may not abort immediately since the **CTRL** key will not be sensed until either the Search String or the end of text is encountered.

## Specifying Search Strings

When you use the Search, Hunt, or Search and Replace commands, PaperClip searches for strings according to criteria that you enter. For instance, if you enter **the** as a search string, PaperClip will find all occurrences of these letters, wherever they occur, regardless of case. (In this case it would match **The**n, clo**the**, and o**ther**.) However, there are ways to specify more complex search strings.

1) A backslash character (\) can be used to match any character. For example, a search string of **t\e** will match any occurrence of the letter "t", followed by any character, followed by the letter "e". On a Commodore 64 computer, you enter a backslash character by pressing the **£** key. Because the backslash will match any character, it is called a "wild card" character.

2) A question mark (?) can be used to match any alphabetic character. For example, a search string of **t?e** would match **the** but not **t,e**.

3) An apostophe (') can be used to specify that whatever follows must be matched exactly. For example, a search string of **'the** would match **the**, but not **The**. If you want to match a character that has a special use in a search string, you can preceed it by an apostrophe. For example, to match a question mark, enter a search string of **'?**.

4) To match strings at the beginning or end of a word, you can use the left and right square brackets ([ and ]) respectively. For example, a search string of [**the** would match **the**n but not clo**the**; a search string of **the**] would match clo**the** but not **the**n. The two brackets may be combined. For example, a search string of [**the**] would find all the occurrences of the word **the** all by itself.

5) Screen Reading can be used when entering either the Search String or the Replace String. Pressing the **RUN/STOP** key will cause whatever text is under the cursor on the screen to be entered into the tab line. Pressing **(shift) RUN/STOP** will automatically reproduce the previous Search String or Replace String.

*Note:* Be careful when using a wild card character in a replace string. The corresponding character will not be replaced.

# Miscellaneous Commands

## Rapid Scrolling

It is possible to scroll through text at a faster speed than that offered by normal scrolling.

Press: **CTRL**
Press: **CRSR⇓** or **(shift) CRSR⇑**

This will accelerate the scrolling.

Scrolling will continue until the cursor key is released. While you are holding down the **CRSR** key, pressing or releasing the **shift** key will reverse the direction of the movement. While the **shift** key is down, the screen will scroll up. When the **shift** key is released, scroll will be downwards.

Pressing **SPACE** during the rapid scrolling will pause scrolling. Scrolling will resume when **SPACE** is released. Rapid scrolling is useful for moving through a document quickly.

## Global File Copy

This function works only with a dual disk drive. Global Files can be cumbersome when it comes time to copy them from one disk to another. It is possible to have PaperClip copy Global Files for you. Although there is no appreciable savings in time by doing this, it does mean less typing and you are not tied to the computer while the copy proceeds.

To copy a Global File from one drive to another,

Press: **CTRL**
Press: **G** [Global]

The top of the screen will display the message **Copy tab line(s)? Y.** Not copying the tab lines will make the disk files compatible with some other wordprocessors. If you want the tab line copied with the document (normal operation) then

Press: **RETURN**

or if not,

Press: **N**
Press: **RETURN**

Enter the name of the first file in the Global chain. Next enter the drive number where the files are. That's all there is to it. PaperClip will load in each file in turn and store it on the other drive. If the files are being read from Drive Ø, they will be written on Drive 1, and vice versa. Using the Global File copy causes the files to be copied in order.

## Setting Device Numbers

As mentioned earlier, some disk drives may have a device number other than 8. When PaperClip first starts running, it assumes that the device number for the disk drive is 8 and the device number for the printer is 4. To change the device number of the disk drive,

Press: CTRL
Press: **(shift) $**

You will then be asked for the device number of your disk drive. Enter the number and press RETURN. You cannot enter a device number less than 4.
To change the device number for the printer,

Press: CTRL
Press: **(shift)** #

Enter the device number for the printer and press RETURN.

## Exiting From PaperClip

To exit from PaperClip and return to Basic,

Press: CTRL
Press: **X** [EXIT]
Press: RETURN

PaperClip will ask you if you are sure you want to quit. Responding with either CTRL or **N** and RETURN will cancel the request and PaperClip will continue operation. Pressing **Y** and RETURN will quit PaperClip and return operation to Basic. *Double check to make sure that you have saved the text in memory before exiting the program. Any text not saved will be lost.*

## Issuing Disk Commands From PaperClip

Commodore disk drives are capable of executing many functions without help from the computer. PaperClip can issue these commands directly to the disk drive.

The sequence to issue a direct command is -

Press: CTRL
Press: > [greater than symbol]

The tab line at the top of the screen should clear and display:

```
>________________________________________________________________
```

PaperClip is now waiting for a command to send to the disk drive. PaperClip will not interpret or check what you type here. The command will simply be sent to the drive when you press RETURN. The commands that Commodore disk drives understand, while powerful, are also cryptic. They obey a rigid syntax with no flexibility for extra spaces or typing errors.

## Formatting A Diskette.

Before a disk drive can store anything on a diskette, the diskette must be "formatted". Formatting the diskette means that the disk drive maps out the entire surface of the diskette in small sections. These sections are called blocks. When you view the directory of a diskette (such as the PaperClip diskette) the last line of the directory will show how many blocks are free. These free blocks indicate how much space is left on for information storage.

If we wanted to format a diskette with a name of "Freddy" and an ID code of "6Q", the command would be:

| | |
|---|---|
| Press: **N** | |
| Press: **Ø** | [drive number where the diskette is] |
| Press: : | [colon] |
| Type: **Freddy** | [diskette name, 16 letters max] |
| Press: , | [comma] |
| Type: **6Q** | [ID code, must be two symbols] |

The top line should look like this:

**>nØ:Freddy,6Q**

Press: **RETURN**

The disk drive should start with a tapping sound, then settle down to a quiet whirring. After a minute or so the drive light will go out indicating that the command has been completed.

A list of some other commands is in the appendix Common Disk Commands.

## Reading The Disk Drive Error Message.

If you find a red light rapidly flashing on the front of the drive (dual drives will have a red light in the center which does not flash), then the disk drive was unable to carry out your command. To find out exactly what went wrong, enter the following:

| | |
|---|---|
| Press: **CTRL** | |
| Press: < | [the less than symbol] |

The tab line will display the error message from the disk drive. This will also cause the error light on the disk drive to extinguish. Disk error messages are of the form:

number, message, number, number.

The message should give a good indication of the problem, although many of the messages are brief. For a list of some of the error messages see the appendix Common Disk Errors.

## Loading An Alternate Character Set (Commodore 64 & 128)

PaperClip has the ability to load alternate characters sets. A character set is a table containing the images of every symbol that can be displayed on the screen. When you start PaperClip, the standard Commodore 64 characters set is used. If you need to have PaperClip display other characters, French for instance, then it will be necessary to use a different character set since the standard one does not have French characters. To load another character set, the command is:

| | |
|---|---|
| Press: **CTRL** | |
| Press: **↑** | [up arrow] |
| Type: **name** | [this will be the name of the character set] |
| Press: **RETURN** | |

As PaperClip loads the new set, the characters on the screen will change to reflect the new shapes. The PaperClip diskette contains several character sets, distinguishable by the 9 block size. There is the standard set, a set derived from the CBM computers (mainly for use with video monitors), and a French set. Any character set editor can be used to create a character set for PaperClip. Information on which characters are defined as multilingual is in the appendix Multilingual Specifications.

## Changing Character Sets (PET and CBM)

If you have a multilingual character generator installed, the following command can be used tc turn it on or off.

Press: **CTRL**
Press: **↑** [up arrow]
Press: **Y**
Press: **RETURN**

The multilingual character generator is available from your PaperClip dealer as an option.

## Using Multilingual Characters

The following keys are defined as multilingual.

**Ø 1 2 3 4 5 6 7 8 9** + £ = @ **u** : ; , / *****

To enter a multilingual character,

Press: **ESCAPE**

followed by the key.

## Outputting Formatted Text To Disk

PaperClip can send output normally destined for a printer to the disk drive. This can be used to store text for transmission via telecomunications or input to a typesetter, etc. The command is as follows:

Press: **CTRL**
Press: **#**
Press: **8** [the disk drive device number]
Press: **RETURN**

When you you tell PaperClip to output, all information normally sent to the printer will be directed to a disk file. PaperClip will ask for the file name just before output starts.

## Creating a Table of Contents

PaperClip offers the ability to easily create a Table of Contents. Here's how it works. At the beginning of your document set up a command that looks like this:

**✓ tf:Ø:contents←**

You can use any name that you wish for contents. Remember to specify the drive number. This will open a sequential file on disk into which PaperClip will place the individual table entries. The ✓**tb** directive is used to add entries to the file:

**✓tb:entry←**

This will cause the word **entry** to be placed into the file followed by the page number where the entry occurred. Since this file is created as a sequential file, it can be loaded in using **CTRL J**, formatted, and printed out.

Using it is easy. Place ✓**tb** directives wherever a section heading occurs within your document (and anywhere else you wish). Then do a Video Output of the document. This will cause the Table of Contents file to be created with all the entries and their respective page numbers. Then load this file in and set up any formatting directives needed. Now you can print the table of contents along with the rest of the document.

During multiple-copy output the table of contents file is only created on the first pass.

## Writing the PaperClip128 'Boot Sector'

PaperClip128 can be automatically started up when the computer is turned on. This process is called 'booting'. To 'boot' a program, a special file must be stored on the diskette. There is only one spot on the diskette where a 'boot sector' may be placed. If that sector is already in use, PaperClip128 will inform you that it can't write the boot sector.

Press: **CTRL**
Press: ' [with shift held down]
Press: **Y**

BOTH PaperClip128 and the 'boot sector' must be present on the same diskette for PaperClip128 to automatically start.

# Formatting Printed Output

## Formatting The Text — Overview

Paperclip is a post-formatted word processor. The term post-formatted means that the text is not formatted as you type it in. The raw text in memory does not exactly match what appears upon the printed page.

When PaperClip is printing, each line is scanned for a space where it can break up the line. PaperClip will not break up a line in the middle of a word. If it can't find a space anywhere in the line, an error will occur.

It is important that you do not edit your text so that it looks nice and readable on the screen during editing. Do not split up lines on your own. When PaperClip formats text for the printer it will come out properly.

## Formatting Directives — Overview

We mentioned earlier that the ✓ character was a special character. The ✓ character precedes every Formatting Directive. All formatting instructions for setting margins, page length, etc., appear on lines which begin with the ✓ character.

Formatting Directives can appear anywhere within text as long as the line upon which they appear begins with the ✓ character and the line previous has a **RETURN** (<-) on it. Directives only affect the text which appears afterwards. If you don't set margins until after the first paragraph, the first paragraph will not be affected by those margins.

## Setting The Left Margin

The directive for setting the left margin is ✓**lm**. This is followed by the number for the left margin setting.

Press: ✓<br>
Type: **lm12**<br>
Press: **RETURN**

This should look like this: ✓**lm12** <-

This sets the left margin to column 12. The left margin can be set from 1 to 25Ø.

## Setting The Right Margin

Setting the right margin is done in almost exactly the same way as setting the left margin. The directive is ✓**rm**. Set the right margin to 7Ø as follows:

Press: ✓<br>
Type: **rm7Ø**<br>
Press: **RETURN**

This should look like this: ✓**rm7Ø**<-

This sets the right margin to column 7Ø. The right margin can range from 1 to 25Ø. *The right margin must be set greater than the left margin.*

## Setting The Page Length And Paging

During output of a multiple page document, it is usually desirable to have a margin at the top of the page and at the bottom. This feature is known as "paging". After so many lines have been printed on one page, the printer will then "page", going to the top of the next page before resuming printing. Two things need to be known for paging to be done properly. The first of these is the physical page length. The page length is given as the maximum number of lines which can be printed on one page. On standard sized 11 inch paper printed at 6 lines per inch, this number is sixty-six (6 X 11). To tell PaperClip that the maximum number of lines is sixty-six, the ✓**pp** (printer **p**age) directive is used:

Press: ✓
Type: **pp66**
Press: RETURN

This should look like:

✓**pp66<-**

PaperClip assumes two things when it finds a ✓**pp** directive. First, it assumes that the paper is currently at the first line on the page. Second, it assumes that it will page after sixty-six lines (or whatever number follows the ✓**pp** directive). Usually you want some white space between pages so in almost every case the ✓**pp** directive is followed by the ✓**pg** (**p**a**g**e) directive. This is the paging directive that tells the word processor how many lines are to be printed before paging is done.

Press: ✓
Type: **pg55**
Press: RETURN

This should look like: ✓**pg55<-**

This tells PaperClip that the page is 66 lines long and that paging should occur after 55 lines have been printed. This will leave an eleven-line gap between one page and the next. Remember, these directives should only be used at the top of a page. The number in the ✓**pg** directive cannot be greater than the number used in the last ✓**pp** directive.

## Multiple Directives On One Line

Having each directive use one line wastes a lot of space. It is possible to have more than one directive after the ✓ symbol. Placing a *colon* after a directive instead of the RETURN (<-) allows more than one directive to be placed on a line. Of course the last directive in the string must be followed by RETURN.

Press: ✓
Type: **lm1Ø:rm7Ø:pp66:pg55**
Press:RETURN

This would show on the screen as: ✓**lm1Ø:rm7Ø:pp66:pg55<-**

## Relative Margins

The previous discussion on setting the left and right margins dealt with *fixed margins*. In longer documents it is useful to be able to adjust the margins without having to look back for the exact previous setting. This can be accomplished using *relative margins*. For example:

Press: ✓<br>
Type: **lm1Ø:rm7Ø**<br>
Press:RETURN

This will set a left margin of 1Ø and a right margin of 7Ø and fix the position of the margins.

Press: ✓<br>
Type: **lm+1Ø:rm–12**<br>
Press: RETURN

This will add 1Ø to the current fixed left margin placing the left margin at 2Ø and subtract 12 from the current right margin placing it at 58.

Press: ✓<br>
Type: **lm+Ø:rm+Ø**<br>
Press: RETURN

This sets the margins back to the original fixed margins of 1Ø and 7Ø.

Fixed margins are defined by ✓**lm** or ✓**rm** followed by a number. Relative margins are defined as ✓**lm** or ✓**rm** followed by a plus sign or a minus sign followed by a number. The relative margins are not accumulative. Thus, if **lm+1Ø** is followed later in text by **lm+5**, the margin will adjust from the original fixed margin and add 5 to the fixed left margin. Relative margin settings are cancelled whenever a fixed setting is issued.

## Forced Paging

Sometimes you may wish to have a certain part of a document start at the beginning of a page. Charts, tables, appendices and indexes are examples of this. For this purpose, the ✓**fp** (**f**orced **p**age) directive can be used. For example:

Press: ✓<br>
Type: **fp**<br>
Press: RETURN

On the screen ✓**fp**← will be displayed. This will cause paging to be performed even if printing has not reached the bottom of the page. The ✓**fp** directive will have no effect if paging has just occured.

The ✓**fp** command can be used in a second way. If a number follows the ✓**fp** directive, paging is performed only if the number of lines remaining on the current page is less than the number following the ✓**fp** directive. This can be used before a section heading to avoid having the heading printed at the bottom of one page and the section itself starting at the top of the following page.

The ✓**fp** directive cannot be placed within a string of directives and it must be followed by RETURN (←).

## Centering

The centering directive is used to turn automatic centering on or off. When centering is turned on, all text printed will be centered between the current margins. The centering directive is ✓**cn** (center), followed by a number. The number **Ø** (zero) will turn centering off. The number **1** will turn centering on.

To turn centering on:

Press: ✓
Type: **cn1**
Press: RETURN

This should look like this: ✓**cn1**←

To turn centering off:

Press: ✓
Type: **cnØ**
Press: RETURN

## Justification

When text is printed, it is normally left-aligned. That is, the left margin will be even while the right margin will be jagged since the lines will be of varying lengths. Justification will produce text with both margins even. This is done by inserting extra spaces between words to fill out the line. The ✓**ju** (justification) directive is used to turn justification on and off in the same way that centering is turned on and off.

To turn justification on:

Press: ✓
Type: **ju1**
Press: RETURN

To turn justification off:

Press: ✓
Type: **juØ**
Press: RETURN

## Right Alignment

The right-alignment directive allows you to do just the opposite of left-alignment. Using ✓**ra** (right alignment) causes text to be printed with an even right margin and a jagged left margin. Right-alignment forces text against the right-hand margin.

To turn right-alignment on:

Press: ✓
Type: **ra1**
Press: RETURN

To turn right-alignment off:

Press: ✓  
Type: **raØ**  
Press: RETURN

The **cn**, **ju**, and **ra** directives can all be placed in a string of directives. Centering will take precedence over justification. Justification in turn takes precedence over right-alignment. In other words if both justification and centering are specified then text will be centered instead of being justified.

## The Margin Release

Sometimes it is desirable to have the first line in a paragraph printed to the left or right of the left margin. For example, if you had a document with each paragraph numbered you might want to have the numbers appear to the left of the body of the text. The paragraph numbers would then appear in the margin.

The ✓**ma** (**ma**rgin release) directive releases the left margin on the following line. You may print to either the left or right of the left margin. The **ma** characters are followed immediately by a plus or minus sign and a number. Suppose we had the following string of directives:

**✓lm1Ø:rm7Ø:ma-3←**

The actual margins would be 1Ø and 7Ø. But the first line of the paragraph immediately following the directive would have margins of 7 and 7Ø. The **ma-3** means that the next margin is to be three characters less than the normal left margin. Obviously in the above example you could not have had **ma-12** since the left margin is normally 1Ø. *The margin number resulting from any margin command must not be less than 1.* ✓**ma** will override any ✓**ai** setting already in effect.

## Automatic Indentation

The directive ✓**ai** (**a**utomatic **i**ndentation) behaves very similar to the ✓**ma** directive except that it will remain in effect for every paragraph which follows it. In that way you can have the first line in each new paragraph adjusted automatically.

For instance, to indent every following paragraph, the command is:

Press: ✓  
Type: **ai+5**  
Press: RETURN

In order to turn off automatic indentation

press: ✓  
Type: **ai+Ø**  
press: RETURN

Note that ✓**ai** will have no effect if centering is turned on. If automatic indentation is not explicitly turned off, it will remain in effect for the rest of the document, even global and external. Therefore, be sure to turn it off when you don't need it.

## Printer Spacing

Printer output is normally single-spaced. The spacing directive allows you to set the spacing, and must specify a whole number (1, 2, 3, etc.). To set the spacing use the ✓**sp** (**sp**acing) directive:

Double spacing:

Press: ✓
Type: **sp2**
Press: RETURN

Triple spacing:

Press: ✓
Type: **sp3**
Press: RETURN

## Forcing Blank Lines

A RETURN (<) symbol by itself on a line will cause a blank line to be printed by the printer. However, if you want to have several blank lines, placing several RETURNs within text can use up a lot of space. The ✓**ln** (**l**i**n**es) directive will print as many blank lines as you want.

Press: ✓
Type: **ln5**
Press: RETURN

will cause the printing of 5 blank lines. If the printer is currently at the top of a page the command will be ignored. If paging occurs while these lines are being printed, they will not be carried over to the next page. That way text on the next page will always start at the top of the page. The lines directive cannot be placed within a string of directives.

## Comments Within Text

Comments may be placed within text file using the ✓**cm** (**c**o**m**ment) directive. A comment is a string of characters which is not to be printed. The syntax of the comment directive is different from directives discussed so far. A colon is placed after the ✓**cm** characters. Whatever characters follow the ✓**cm** will be ignored during output. The end of a comment is signified by the RETURN (<) character.

Press: ✓
Type: **cm:This will NEVER be printed.**
Press: RETURN

A good idea is to place the document name in a comment on the first line of each document. This way the "screen read" feature can be used to fetch the document name whenever the file is to be saved.

## Headers And Footers

Paperclip offers the ability to have both a header and a footer printed on each page of the output. A header is simply a line which will be printed at the top of each new page. It may contain the page number, title, author's name, or whatever you choose. A footer is essentially the same except that it is printed at the bottom of the page. The syntax for setting up a header is:

Press: ✓
Type: **hd*N*:left part,middle part,right part**
Press: RETURN

"*N*" is the number of lines above the main text body the header will appear. If "*N*" is 1 then the header will appear immediately above the text body.

The header itself consists of three parts separated by commas. The left part will appear against the left margin; the middle part will be centered over the page; and the right part will be right-aligned against the right margin. All three components do not need to be present. Comma's must be entered for the missing components. Some headers may just be the title centered in the middle. In that case you simply type a comma, then the title, then **RETURN**. Suppose you wanted the header to be a title centered three lines above the main text on the page. It would be set up like so:

Press: ✓
Type: **hd3:,Title**
Press: RETURN

The footer is set up the same as the header. Instead of ✓**hd** (header), the ✓**ft** (footer) directive is used. However "N" this time is the number of lines from the bottom of the page where the footer will appear.

You should be careful with headers. They cannot be part of a string of directives. A comma can't be part of one of the header components since it will be regarded as a separator. Use the user-defined characters to get around this problem (see the section on Sending Special Characters).

Be careful where you put the header directive. Since the header is only printed at the top of a page make sure you place the directive before any text that will be printed on that page. If you put it after the first paragraph in a document, the first page will not have the header printed. If you do want to have the header start printing on the second page and not the first then place the directive partway through the first page.

## Paging

Automatic page numbering is available. The command to print the page number is <>. This must appear as a component either in the header or footer. For example, suppose we want to have a footer which will place the page number in the bottom right-hand corner of each page. This footer is to be three lines from the bottom of the page:

Press: ✓
Type: **ft3:,,page <>**[the less than sign followed by the greater than sign]
Press: RETURN

Three lines from the bottom of each page, PaperClip will print **page N** where **N** is the current page number. Remember that the two symbols must be together. < > will simply cause < > to be printed.

## Setting The Page Number

You can set the page number to any number you wish using the ✓**p#** directive. Setting the page number to twelve would be done like this:

Press: ✓  
Type: **p#12**  
Press: RETURN

Being able to set page numbers can be useful if you want to start printing a document part-way through and still have the page numbers correct.

## Margins For Footers And Headers

You may wish to have set different margins for the header and footer from main text and from each other. Whenever the page margins are set using the ✓**lm** and ✓**rm** directives, the header and footer margins are set as well. This is not always convenient, so it is possible to set the header and footer margins separately using the ✓**hl** and ✓**hr** directives for the header left and header right margins. Suppose we have a document whose margins are normally 1Ø and 7Ø, but for one small section of text we want to change them to 2Ø and 6Ø. However we don't want the header margins to change. This could be done as follows:

Press: ✓  
Type: **lm2Ø:rm6Ø:hl1Ø:hr7Ø**  
Press: RETURN

Remember that each time the margins are changed using ✓**lm** and ✓**rm** the header margins will change as well. ✓**hl** and ✓**hr** are only in effect until either the text or header margins are changed (unless "margin lock" is turned on).

## Margin Lock

As mentioned above, the header and footer margins will change with the ✓**lm** and ✓**rm** directives. The ✓**hl** and ✓**hr** directives are changed by setting ✓**lm** or ✓**rm**. You may not always want this to occur. You may wish to *lock* the margins so that the header and footer margins are not affected by ✓**lm** and ✓**rm**. To do this use the ✓**ml** (**m**argin **l**ock) directive.

Press: ✓  
Type: **ml1** [the number one]

Press: RETURN

The ✓**ml1** directive will lock the header and footer margins. After that, manipulation of ✓**lm** or ✓**rm** will not affect the header and footer margins. Margin lock is turned off by:

Press: ✓  
Type: **ml0** [the number zero]  
Press: RETURN

## Vertical Positioning

The ✓**vp** (vertical **p**ositioning) directive is used to cause printing to begin part-way down the page instead of at the first line. The syntax of the ✓**vp** directive is:

Press: ✓
Type: **vp1Ø**
Press: RETURN

This is an example of a format line.

✓**pp66:pg6Ø:vp1Ø<**

The page length is set to 66, with paging to occur after 6Ø lines are printed. The **vp1Ø** means that 1Ø blank lines are to be printed at the top of each page. This means that only 5Ø lines of actual text will be printed out before paging occurs — the other 1Ø being taken up by the 1Ø blank lines generated by the **vp** directive. Be careful that you don't set the **vp** value equal to the **pg** value (that would mean that no text would be printed out, only headers and footers).

## Printer Offset

The ✓**of** (**of**fset) directive is used to offset the output on the printer. If text is not properly positioned on the paper, it can be shifted to the right without adjusting margins or moving the paper. To offset the printing ten spaces to the right:

Press: ✓
Type: **of1Ø**
Press: RETURN

## The Pause Command

You can cause the output to the printer to pause and a message to be printed on the top line using the ✓**ps** (**p**au**s**e) directive. A common use of this feature is to stop printing so the operator may change a print wheel on a daisy-wheel printer. An example follows:

Press: ✓
Type: **ps:Change to Courier 72 print wheel.**
Press: RETURN

When, during printing, output reaches this point, PaperClip will pause and the message **Change to Courier 72 print wheel.** will be printed on the top line of the screen. Pressing any key except CTRL will re-start the output. CTRL will abort output.

## Changing The Pitch

Some printers offer the ability to select the pitch of the print. "Pitch" refers to the number of characters per inch that are printed. Standard pitches are 1Ø-pitch (pica type) and 12-pitch (elite type). The pitch of the printer can be set using the ✓**pt** (**p**i**t**ch) directive:

Press: ✓
Type: **pt12**
Press: RETURN

This will set the pitch to twelve characters per inch. There are four different pitch settings allowable. They are 1∅, 12, 15, and an optional pitch setting. Numbers of 1∅, 12, or 15 in the ✓**pt** directive will select the respective pitch. Any other number will select the optional pitch. The actual codes which are sent to the printer to set the pitch are contained in the printer file.

## Changing Pitch - Headers And Footers

Pitch setting for the header and footer follow the same form as text pitch. The directives are as follows

Press: ✓
Type: **ph12**
Press: RETURN [for the header]
Press: ✓
Type: **pf12**
Press: RETURN [for the footer]

## Line Spacing

The ✓**ls** (line spacing) directive is similar to the ✓**pt** command. It allows you to select the number of lines printed per inch on the printer. It differs from the ✓**sp** directive since the ✓**sp** directive only determined the number of blank lines to be generated after each line. You were still working with the same number of lines on the page. The ✓**ls** directive allows you to actually change the spacing between lines.

The usual values are 6 and 8 lines per inch. Like the ✓**pt** directive it also allows an optional line spacing. Again this is determined by the printer file. An example of setting the line spacing to 8 lines per inch would be:

Press: ✓
Type: **ls8**
Press: RETURN

## Sending Special Characters To The Printer

There may be times when you want to send special characters to the printer to select modes on your printer which are not normally available through PaperClip. This can be done using the user-definable characters. There are nine characters labelled 1 through 9. For example, to set user-defined character 6 equal to ASCII value 27 the formatting directive would look something like this:

Press: ✓
Type: **6=27**
Press: RETURN

To place a user-defined character within text,

Press: CTRL
Press: ; [semi-colon]

This will produce a ↑ symbol on the screen. Then enter the number (1 to 9) of the character you defined. If the character does not generate a character on the printer then it is "non-printing". Characters 6 to 9 are considered to be non-printing and will not be counted during justification or centering. If you want to send down a two-character sequence to print just one character on the printer, then set one of the characters to be non-printing (6 to 9) and use a printing (1 to 5) number for the other one.

You may remember that in the discussion of headers the comma was designated as the character which separated the header/footer components. However, suppose you want to have a comma in one of the components. For example, suppose the header is to have the date as the left component and the page number as the right component, and no middle component. It might look like this on the screen:

**✓hd3:December 21, 1984,,Page <>←**

The problem is that the comma in the date will be interpreted as a separator and the year 1982 will be centered on the page. The trick is to use one of the user defined characters. Define it as being equal to the comma's ASCII value (usually 44). Then place it in the header at the place where you want your comma to appear. Since it is not a comma, it won't be interpreted as a separator. But it will print as a comma during output.

**✓4=44←**

**✓hd3:December 21 ↑ 4 1984,,Page <>←**

# Linked Files

## Global File Link

The Global File link directive ✓ **nx** has already been explained in the section on Searching and Modifying Text. There are in fact two other ways of linking files together. To review, an example of a Global File link is:

Press: ✓  
Type: **nx:document name**

## Non-Specific File Link

This is really just a simplified version of the ✓ **nx** directive. Use the ✓ **lk** directive which is not followed by a file name. During output, when the ✓ **lk** directive is encountered, you will be asked what the next file in the chain is to be. In this way you can create a document from segments on the diskette without having previously loaded each in and linked it to the next. An example of a non specific file link follows:

Press: ✓  
Type: **lk**  
Press: **RETURN**

## External File Print

PaperClip can directly transfer the contents of a disk file to the printer. This is not the same as printing text, because PaperClip will not analyze the data it recieves from an external print file. The data will simply be read from the disk, a byte at a time, and sent to the printer. For example, if your printer has a bit image mode, then it would be possible for you to create a disk file that, when sent to the printer, would create an elaborate heading on the page. Setting up the file is up to you, but once you have it, PaperClip can send it. This type of file is usually created by a graphics package.

For example, to send the contents of the file "special stuff", the directive is:

✓ **ep:special stuff←**

## External File Link

Another way of linking files is through the use of the external file directive ✓ **ex**. This directive does not have to occur at the end of a file. During output, when the ✓ **ex** directive is encountered, the internal text pointers are saved and the external file is called in and printed. The external file may itself link to other files with the ✓ **nx** or ✓ **lk** directives, but not with the ✓ **ex** directive. Only one level of external files is allowed.

When the last file in the chain of files called by the ✓ **ex** directive is finished, the control file (the one with the ✓ **ex** directive in it) will be recalled and the text pointer reset. Output of that file will continue where it left off.

External File links are especially useful when you have a large number of document segments which you may want to print in any order. Simply set up the control file as a series of external file links. By rearranging the order of these links you can change the order of the output. The nice thing about it is that you only have to change one file, and it can be a local file. An example of an External file link follows.

This will print **Hello**, followed by the contents of "file #1", followed by the contents of "file #2", followed by **The end**.

**Hello←**
**✓ex:file #1←**
**✓ex:file #2←**
**The end←**

# Embedded Format Characters

You may have noticed that there were no directives for underlining or boldface print. This is due to the fact that the directives must be placed between paragraphs. Boldface and underlining are controlled through the use of special symbols within the text to turn special functions on and off. Embedded format characters can appear *anywhere* in the text.

## Escape Characters

All of the embedded format characters are generated using the **ESCAPE** key. When the key is pressed the prompt **Key?** will be displayed at the top of the screen. A non blinking cursor will be waiting for a response.

The **ESCAPE** key functions in a similar way to the **CTRL** key, except that the **ESCAPE** key is used to produce the embedded characters. The **CTRL** key is intended for executing functions, although some of the embedded characters can be generated using the **CTRL** key. In all discussion, only the **ESCAPE** key will be mentioned. Refer to the appendix Summary Of Commands to find out the **CTRL** equivalents.

## Underlining

To turn underlining on:

> Press: **ESCAPE**
> Press: **(shift) [**

This will produce a box on the screen one quarter the size of the cursor. This is the *underline-on* symbol.

To turn underlining off:

> Press: **ESCAPE**
> Press: **(shift) ]**

This will produce a box on the screen one quarter the size of the cursor. This is the *underline-off* symbol.

All characters between the underline-on and underline-off symbols will be underlined on the printer.

## Boldface Print

To turn boldface on:

> Press: **ESCAPE**
> Press: **(shift) (**

This will produce a half sized **L** symbol on the screen. This is the *boldface-on* symbol.

To turn boldface off:

> Press: **ESCAPE**
> Press: **(shift) )**

This will produce a half sized reverse **L** symbol on the screen. This is the *boldface-off* symbol.

All text between the boldface-on and boldface-off symbols will be printed in boldface type.

## Italics Print

To turn italics on:

Press: ESCAPE
Press: **(shift)** <

This will place an inverted half **L** symbol on the screen.

To turn italics off:

Press: ESCAPE
Press: **(shift)** >

This will place a reversed inverted half **L** symbol on the screen.

These sequences produce the Italic Print formatting characters. Many printers do not have an *Italic* print font. In those printer files the *Italics* on and off sequences turn on some other type of special printing.

## Superscript And Subscript

To superscript a single character

Press: ESCAPE
Press: **(shift)** #

This will produce a horizontal bar at the top of the screen line. Follow with the character you require superscripted.

To subscript a single character

Press: ESCAPE
Press: **(shift)** $

This will produce a horizontal bar at the bottom of the screen line. Follow with the character you require subscripted.

Only the first character following the superscript or subscript symbol will be affected. If you require whole words or phrases to be superscripted or subscripted then use the following:

To turn on superscripting

Press: ESCAPE
Press: **(shift)** %

This will produce a vertical bar on the screen.
To turn off superscripting,

Press: ESCAPE
Press: **(shift)** ‘ [apostrophe]

This will also produce a vertical bar on the screen. All characters between the superscript begin and superscript end symbols will be superscripted.

To turn on subscripting,

Press: ESCAPE
Press: **(shift)** !

This will produce a vertical bar on the screen.
To turn off subscripting,

Press: ESCAPE
Press: **(shift)** " [double quotes]

This will also produce a vertical bar on the screen. All characters between the subscript begin and subscript end symbols will be subscripted.

Both superscripting and subscripting are automatically turned off at the end of a paragraph.

## The Shifted Space (Hard Space)

There may be times when you want to force two words to appear on the same line. Normally you cannot be sure that this would happen. During output, the two words could become separated if the first was at the end of a printer line.

Press: **(shift)** SPACE

will produce a horizontal bar on the screen — a shifted space. It will print as a space on the printer, but it is recognized by PaperClip as being different from a space and PaperClip will not break words joined with a shifted space.

Another situation where the shifted space is useful is centering. Suppose you have to center a number of lines of text which are all of different lengths, yet you wish these lines to have a straight left margin. Make all of the lines the same length by placing shifted spaces at the end of the shorter lines.

## The Conditional Hyphen

One of the problems in using a post-formatted word processor is that you never know just where on the page a given word will occur. This can lead to problems with long words which you would certainly hyphenate if you knew that they would overlap the end of a line. You obviously don't want the hyphen if the word is to appear in the middle of a line. The answer is to use the conditional hyphen.

To produce a conditional hyphen,

Press: ESCAPE
Press: -

If the word appears at the end of a printed line a hyphen will be printed. If it appears in the middle of a line, the hyphen will not print.

## The Breakpoint

The breakpoint character is produced by,

Press: **CTRL**
Press: **(shift)** !

It is similar to the conditional hyphen but it *does not print* when it breaks a word. It allows you to have a line broken in the middle of a word. The breakpoint will be ignored if it occurs in the middle of a line.

## The Underline Character

There may be times when you wish to have a number of spaces underlined on the printer. An example might be a signature line on a contract. Printers which have an "underline mode" generally underline only from the first printable character to the last printable character (spaces are considered non-printable characters).Unless there are printable characters both before and after the spaces then they won't be underlined.

You can force the printer to produce the underline character with,

Press: **ESCAPE**
Press: . [period]

This will produce a graphics character on the screen, but when it is encountered during output, the underline character will be sent to the printer. Before using this method turn the underlining on and off to check that your printer does underline.

# Outputting The Text

## Video Output

The text entered into PaperClip is formatted as it is output to the printer. PaperClip also allows you to output formatted text to the screen. This feature is used for checking a document to see that it has been set up properly before sending it to the printer. The screen will show only the left-most portion of the page. If you have a 4Ø-column screen, the right-hand side of the document will be off the screen.

If you want to see the right side of the document, PaperClip allows you to use horizontal scrolling during video output, to view the entire document width. To do this while you are viewing a formatted document, press either the **CRSR⇒** or **(shift) ⇐CRSR** key. The video display will stop and the display window will shift in the direction you have indicated until you release the key. You can now press any key to restart the display.

PaperClip 64 & PaperClip 128 allow several options for video output. PaperClip 64 can display 8Ø columns at once on the screen without horizontal scrolling. PaperClip 128 can display 4Ø , 8Ø , 16Ø or 32Ø columns!

To change video output modes, hold down the **shift** key, and press **f8**. This must be done in edit mode *before* you select video output. When you switch display modes, PaperClip will display a message on the top line of the screen, showing you the new mode.

To display video output,

Press: **CTRL**
Press: **V** [Video]

The Status Line message Video Output will be displayed. The following questions will be asked before video output starts:

**Fill variable blocks? N** — Questions at the top of the screen generally have default answers if they are just yes or no questions. For this question the default answer is "No" so press: **RETURN**. Variable blocks will be discussed in the section on Form Letters and Variable Blocks.

**Page? 1** — PaperClip allows you to start the output to Video with any page number. The default answer is page 1. If you want to start at page one press: **RETURN**. If you want to start at any other page then enter the starting page number and press **RETURN**.

**Global? N** — PaperClip files can be linked together using the **✓nx** directive. If **RETURN** is pressed then the text in memory will be output to the screen. If **Y** and **RETURN** are pressed then the name of the first document segment will be requested. The specified file will be loaded, and output to the screen will start. When the file link is encountered PaperClip will automatically load in the next file and continue the video output. CAUTION: When the next file in the chain is loaded into the text memory it will erase the current text.

The screen will clear and the formatted document will start to scroll up the screen. Press any key to pause the scrolling: press again to restart. At the end of each page a line will appear and the the scrolling will pause. Press any key to restart the video output. The Video Output may be aborted by pressing the **CTRL** key.

PaperClip has an additional feature which allows horizontal scrolling in the Video Output mode. Any time that the screening is scrolling, if the **CRSR⇒** key is pressed the vertical scrolling will stop and the screen will scroll horizontally. Holding **(shift) ⇐CRSR** will scroll text to the right. Pressing any other key (except the **CTRL** key) will restart the video scrolling vertically.

## Output To Printer

NOTE: Make sure your printer file has been loaded in!

To send a document to the printer:

Press: **CTRL**
Press: **O**

The following questions will be asked.

**Continuous output? N** — Press **RETURN** if you want the printer to pause at the end of each page. If you are using single-sheet paper then you should specify discontinuous output. Pressing **Y** and **RETURN** will provide continuous output and the printer will not stop between pages.

**Fill variable blocks? N** — Variable blocks will be discussed in detail in the section on Form Letters. For now, press: **RETURN.**

**Number of copies? 1** — Pressing **RETURN** will print one copy of the entire document. Any number from 1 to 255 can be specified. If you require more than one then enter the number of copies you want and press **RETURN**. If you select more than one copy, PaperClip will repeatedly send the document to the printer until the required number of copies have been printed.

**Page? 1** — Pressing **RETURN** will start printing at page one.If you require printing to start at another page then enter the page number and press **RETURN.**

**Global? N** — If your file is not linked to other files and it is currently in memory then press **RETURN**. If your document is composed of files linked with the **✓ nx** directive then press **Y** and **RETURN**. If you answered **Y** to global then you will have to enter the name of the first file in the chain. CAUTION: The first file in the chain will be loaded into the text memory, erasing the document previously in memory. Remember to save the contents of the memory before starting a global output.

After this last question, output to the printer should begin. While output is being sent to the printer, it will also be printed on the screen. Text on the screen will be identical to that which is printed out on the printer except that any text which appears underlined, in boldface, or in italics will be displayed in reverse-field on the screen. If you asked for discontinuous output, when PaperClip pauses at the end of a page, pressing almost any key will start it going again. The exceptions are described below.

At any time during output, printing can be paused by pressing any key on the keyboard. The printer may keep going as some printers have large storage areas to hold the text before printing it. PaperClip will pause immediately, but the printer could still be emptying its storage area.

Output can then be re-started by pressing any key except the **CTRL** key. The **CTRL** key will cancel the output function altogether.

## Direct-Key Output

It is possible to start output without having to answer any questions.

Press: **CTRL**
Press: **(shift) O**
Press: **RETURN**

This will start the printer output immediately. All of the questions which were asked when you pressed **CTRL** and then pressed **O** have been answered automatically with the default answers. The output will be discontinuous; variable blocks will not be filled; one copy will be printed; output will be local.

## Switching Between Video And Printer Output

If output is continuous then there will be no pausing between the pages. This is best when you are using fan-fold paper or roll-paper in your printer. However, single-sheet paper will require that you select discontinuous output.

When output is discontinous the output will pause at the end of each page. This allows you time to position a new sheet of paper before printing starts up again. When the pause at the end of the page occurs, pressing almost any key will cause output to start up again. Pressing **CTRL** will abort output.

There are three other options: If you press **V** then the output will become video output. If you Press **P** subsequent output will be sent to the printer. If you Press **R** the previous page (the one just displayed) will be printed. You can use the **R** option to preview a page in Video Output before printing it.

You can switch between printer output and video output only between pages. Thus, during continuous output, you will not be able to change the output destination.

The ability to change from video output to printer output has advantages. It allows you to print certain pages of a document at your discretion. If only one part of a document requires corrections, there is no need to reprint the entire document, only the page where the changes have occurred. Use video output until the page you want comes up and change to printer output.

## Switching Between Continuous And Discontinuous Output

When printer output first begins, you have the option to select either continuous or discontinuous output. While output is occurring you can change this setting. As mentioned earlier output can be paused by pressing any key. There are however two exceptions to this. Pressing **C** will not pause output, instead it will cause output to be continous from that point on. Pressing **D** will cause output to be discontinuous. Even if continuous output is specified, it can be changed. You can then make use of the ability to change from video to printer output.

## Restarting Output At The Top Of The Page

During output (video or printer) you may see an error. Pressing **CTRL** will break the video output and return to the text mode. You can now make the correction. When you have made the changes,

Press: **CTRL**
Press: **(shift) P**
Press: **RETURN**

This will restart the output at the top of the current page and you can carry on with the editing. Files with the corrections may be saved before restarting the output. The above applies to any output. If you are working with a global, external link, or non-specific linked file, save the changed text before restarting output.

**NOTE**: Restart will not work properly if any changes are made to pages **previous** to the page which was halted.

# Form Letters

## Form Letters And Variable Blocks

One of the more common business tasks is the mailing of form letters.

An example of a form letter is:

> Febuary 26, 1984.
>
> Dear Ms. Dianne MacLeod,
>
> Our records show that the amount overdue on your account is $526.5Ø. Please submit this amount in order to keep your valued account in good order. If this amount has been paid then disregard this notice.
>
> Sincerly,
>
> Alicia DeSoto

In this example the text is similar for anyone whose account is outstanding. The only information which changes from letter to letter is the name and the amount. PaperClip allows a document to be created with "holes" where variable data can be entered from another source. These holes are referred to as "variable blocks".

There are three stages to printing form letters. They are: creating the form letter, creating the variable data, and printing out the letters.

## Creating A Form Letter

As the form letter is created, places where variable data is to be filled in later are marked with a special symbol to designate the variable block. To enter a variable block into text,

Press: **CTRL**<br>
Press: **B** [Block]

A marker is placed on the screen. The marker covers two character positions and marks both the start and the end of the variable block. Using the above example the form letter will look like the following:

> February 26, 1983.
>
> Dear ■,
>
> Our records show that the amount overdue on your account is $■. Please submit this amount in order to keep your valued account in good order. If this amount has been paid then disregard this notice.
>
> Sincerly,
>
> Alicia DeSoto

In the above example we have two varible blocks in the form letter, where the name and amount will go. Once the letter is created we would save it on a diskette.

## Creating A Variable Data File

The variable data to be used in a form letter is stored in a separate file on disk. To set up a file of variable data first erase the screen. We will create data for our example letter. With the cursor on line 1 column 1, type in the following:

Type: **Annie Ablative** press: RETURN
Type: **12.36** press: RETURN
Type: **Mr. A. Galapagos** press: RETURN
Type: **85.26** press: RETURN
Type: **Mrs. Begonia** press: RETURN
Type: **789.23** press: RETURN

For our example we will use three names and amounts. Notice that each datum is followed by RETURN. The RETURN marks the end of each item.

The file must now be saved on the diskette. *Variable data must be saved as a sequential file.*

Press: CTRL
Press: **Z**
Type: **data file name**
Press: RETURN
Press: RETURN
Press: **Ø** (or **1**)
Press: RETURN

The data file will be stored on the disk.

## Form Letter Output

Recall the form letter saved previously. In this example we will display the output on the screen.

Set the left margin to 2 and the right margin to 38. (✓**lm2:rm38**)

Press: CTRL
Press: **V**

This initiates video output.

The screen displays **Fill variable blocks? N**.

Press:**Y**
Press: RETURN

The screen displays **Variable file name?**.

Type: **data file name**
Press: RETURN

The disk drive becomes active making sure that the data file is on the diskette.

The screen displays **Global? N**.

Press: RETURN

The screen will clear and display the first letter. Pressing a key will display the next letter, then the last letter. The top of the screen will display the message **Out of variable data.**.

To output a Form Letter to the printer

Press: CTRL
Press: O

To the question **Fill variable blocks? N** answer,

Press: **Y**
Press: RETURN

Enter in the name of the variable data file.

## Other Variable Block Functions

There are a number of variable block functions which can be performed while editing. They are as follows:

Each of these commands should be preceeded by Press: CTRL

**(shift)F** — This will move the cursor to the next available variable block in text. If no block can be found then the message **Out of variable blocks.** will be printed.

**(shift)B** RETURN — This will find the next variable block in text and insert data from the current variable data file. If no file is opened, you will be asked for the file name.

**(shift)V** RETURN — This will fill all variable blocks in text from the current file. Again you will be asked to supply a file name if no file is open.

**(shift)N** RETURN — This will empty all variable blocks in text. All variable blocks must be empty before you can use CTRL **(shift) V** RETURN.

**(shift)Z** — This allows you to change from one variable data file to another. PaperClip will ask you for the new file name. The new file will then be opened and the old one closed.

## Multiple Data Items Per Line

Data files are saved using the sequential save: CTRL **Z**. If you want to edit this file it must be loaded using CTRL **J** (sequential file load).

In the above example we saw that each datum occurred on a separate line and that the RETURN marked the end. Besides RETURN you can also use ESCAPE followed by ✓ as the end marker. Using this symbol allows multiple data on a line, thus allowing more data per file.

For instance, the first two items from the previous example could be entered as follows:

Type: **Annie Ablative**
Press: ESCAPE
Press: ✓
Type: **12:36**
Press: RETURN

It should look like this on the screen:

**Annie Ablative £12.36←** [note that the end-marker is displayed as the **£** character]

Now only half as many lines are needed in the variable data file to hold all the information. This method can be used to increase the number of entries a single data file can hold. Many items can be put on each line. We recomend that you use one line for each letter you wish to fill out. In the section on sorting you will see that the data file can be sorted. The maximum length of any single data item is 25Ø characters, including spaces.

If you are using variable blocks to hold addresses, you must have one variable block for each line of the address. Sometimes the data will have fewer items than the number of variable blocks in your form letter. If this happens, you must enter blank lines in the variable data list to make up the missing entries.

Ocassionally, you may have a data list which contains more data than you really need. For instance, suppose you have a list of names and amounts. Later, you wish to send a note, but don't want the amounts included in the letters. You can cause PaperClip to ignore a data item as follows. When you come to the place in the document where the unwanted item will appear, place a variable block after a comment ( ✓ **cm:**). When PaperClip fills the variable blocks from the data file, this block will be filled as well. When PaperClip outputs the document, the comment field with the data item is skipped over and not printed.

# Advanced Editing Features

These commands are more specialized, designed for use in a business environment, but if you deal with numbers you will be interested.

## Numeric Tabs

Regular tabs are poor for handling of columns of numbers. There are two reasons for this. First, most people want columns of numbers to line up properly. This can lead to a lot of fussing and fiddling when you are working with regular tabs. Second, the embedded formatting characters are "non-printing". When you have them in text among columns, your columns will not line up on the printer.

Numeric tabs offer a partial solution to the problem. A numeric tab is set or cleared as follows:

Press: **CTRL**
Press: **N**

There are two characteristics which make numeric tabs unique. One difference is that numeric tabs move. A regular tab set at column 2Ø will always be at column 2Ø. A numeric tab set at that postion might be almost anywhere near column 2Ø. A numeric tab is not screen oriented; it is printer oriented.

Suppose you have a line with no embedded formatting characters on it. The character which will print on the 2Øth column of the printer will be on the 2Øth column of the screen during editing.

But suppose the next line is to be printed in boldface and underlined. That means that there will be two embedded format characters at the beginning of the line. It also means that the character that will print in column 2Ø on the printer is no longer in column 2Ø on the screen. It is in column 22. Suppose the first two characters on the line are the underline-begin and boldface-begin characters. Neither of these will cause a character to be printed on the printer, they merely select modes of output. That means that the first text character that will print on the printer will be found right after these formatting characters. That will be in column 3. But it will print in column 1 on the printer. Therefore, the character that will be printed in column 2Ø on the printer will be found in column 22 on the screen.

If you had set a numeric tab it would have automatically moved over two columns to account for the fact that there are two non-printing characters on the line. It would still be at printer column 2Ø where it was first set. When setting numeric tabs, it is important that there be no embedded format commands to the left of the cursor.

The numeric tab can be useful when entering columns of information. Set your numeric tabs before entering text. The numeric tab will appear as **N** on the tab line and is accessed with the tab key like regular tabs. Up to 16 numeric tabs can be active at one time.

## Numeric Mode

The second unique feature of numeric tabs is referred to as *Numeric Mode*. Whenever you tab to a numeric tab, you enter Numeric Mode. This will be indicated by ***NMR*** on the Status Line.

While in Numeric Mode, numbers entered will be pushed to the left of the tab setting. This means that your numbers will line up for you neatly without having to go through a lot of work. When a non-numeric character is typed in, numeric mode is cancelled. The following characters are considered numeric:

**Ø 1 2 3 4 5 6 7 8 9 - + ( , $**

PaperClip will stay in Numeric Mode until you type a character other than the ones above. Pressing **CTRL** will cancel numeric mode. Numeric Mode is also cancelled if there is not enough room to insert the characters.

The delete key will function in Numeric Mode. It will simply draw in any of the above characters, deleting them. The shifted space is included in the list of characters that will be forced out to the left. That way you can enter your numbers in the now-fashionable metric style.

There won't be any problems with typing decimal points since they will turn Numeric Mode off. Set your numeric tabs at the columns where you want your decimal points.

To move the cursor directly to the next numeric tab, bypassing any regular tab settings, use the following

Press: **ESCAPE**
Press: **TAB**

# Columns

A powerful feature of PaperClip lies in its ability to manipulate columns of text: numbers, names, anything. This makes it ideal for business situations where preparing budgets, financial reports, etc., are quite common. PaperClip is flexible when it comes to entering and editing columns of data.

## Setting A Column

Suppose you have a chart within text which looks like this:

| | | |
|---|---|---|
| **name1** | **price1** | **date1** ← |
| **name2** | **price2** | **date2** ← |
| **name3** | **price3** | **date3** ← |
| **name4** | **price4** | **date4** ← |

To set a Column, move the cursor to the top left corner of the column you wish to edit.

Press: **CTRL**
Press: **(shift) C**
Press: **RETURN**

The Status Line should display the message Set Column. The character that was under the cursor should now be in reverse-field. Press the **CRSR⇒** key. At first this will seem identical to the setting of a phrase. Press the **CRSR ⇓** key. You can now use the cursor keys to draw a highlighted box over the entire column. When the column is highlighted press **RETURN** to set the column. You must press **RETURN** to finish setting the column.

Pressing **S** while setting a column will move the right-hand side of the column over to the right hand edge of the text. The very last column will not be included since there MUSTbe a **RETURN** at the end of each line in a column.

## Moving A Column

Type in the above chart in order to be able to follow this example. There are five spaces between columns.

Suppose we decide that column 1 is in the wrong place. We want it between column 2 and column 3. Move the cursor so that it is over the **n** of the first entry in column 1.

Press: **CTRL**
Press: **(shift) C**
Press: **RETURN**

We will now set a Column over column 1. Press: **CRSR ⇓** three times. This should produce a bar one character wide so that the **n**'s in all of column 1 are now be highlighted. Press: **CRSR⇒** until the whole column is highlighted. Press: **CRSR⇒** five more times as we want to move the spaces as well, then press: **RETURN**

Move the cursor so that it is blinking over any of the **d's** in column 3.

Press: CTRL
Press: **(shift) M**
Press: RETURN

You should see column 1 move in between column 2 and column 3.

Moving a column can only be done horizontally across the page. When you are moving a column, what actually occurs is that for each line in the column, the column is inserted at the current cursor postion, then deleted from where it was.

## Deleting A Column

To delete a Column that was previously set,

Press: CTRL
Press: **(shift) D**
Press: RETURN

Text to the right will be shifted over to replace the deleted column.

## Erasing A Column

To erase a previous set column from text,

Press: CTRL
Press: **(shift) E**
Press: RETURN

The difference between erasing and deleting a column is that erasing *replaces the column with spaces* — no other part of text is affected.

## Inserting Spaces Before a Column

When you are constructing a chart you may discover that you want to open up space within the chart for another column of data. Set a column over the first columns of the chart entries you want moved to the right.

Press: CTRL
Press: **(shift) I** [the letter I]
Press: RETURN

Then enter the number of spaces which you wish to have inserted and press: RETURN. The spaces will be inserted in front of the column you set on each line. Be careful that you don't insert enough spaces to cause any line in the column to expand onto the next line.

## Replicating A Column

You can take any column and produce a copy of it within text. Set the column that you want to copy. Then move the cursor to where you want the upper-left hand corner of the copy to be placed.

Press: CTRL
Press: **(shift) R**
Press: RETURN

The column will be duplicated in the new position. The old column is not erased.

## Shifting A Column

The Move Column function is not appropriate in all situations. It performs a series of inserts and deletes to move the column from one place to another. If you want to take an existing column and move it in any direction without affecting the text around it, then use the Shift Column function. To use this function, set your column: move the cursor to the new upper left-hand corner for your column.

Press: CTRL
Press: **(shift) S**
Press: RETURN

You can move a column up, down, right, and left without affecting the rest of text, unless you shift the column over some text.

## Adding A Column

PaperClip can perform addition and subtraction within text. To add a column of numbers, use the Column Set function to indicate the desired column. Move the cursor to the position in text where you wish the answer to be printed. The result may appear anywhere in text. The decimal point will line up where the cursor is positioned.

Press: CTRL
Press: **(shift)** =
Press: RETURN

PaperClip will add the column and print the total.

Negative numbers are accomodated in two ways. If any occurence of the negative sign is found in a number then the number is assumed to be negative. The negative sign may be in front of the number, behind it, or in the middle of it. The other character which is used to interpret a negative number is the left parenthesis, (. In this way numbers can be enclosed within parentheses to indicate that they are negative. Remember to make sure when you set the column that you include all negative signs. If they are not included in the column, then the number will be interpreted as being positive. The same holds true for numeric digits and decimal points. They must be included or they will be ignored.

The only characters that the Column Addition examines are numbers and the decimal point. For example, all of these are equivalent in value for PaperClip:

**123.45**
**$123.45**
**123.45%**
**1$23.45**
**123a.45**

Whenever you add a column it is possible for an *Overflow Error* to occur. PaperClip is capable of 38 digit precision during addition. When an *Overflow Error* occurs it indicates that the number would require 39 digits or more.

One more note about precision: since numbers are stored internally in decimal (as opposed to binary) there will *NOT* be addition errors of the type normally associated with microcomputer arithmetic.

## Setting The Decimal Point

The decimal point can be fixed to any one of 39 positions. To set the decimal point,

Press: **CTRL**
Press: . [a period]
Press: **RETURN**

You will be asked for the number of digits that are to be to the right of the decimal point. This number must fall into the range Ø to 38. When the result of an addition is printed, only the specified number of digits will be printed to the right of the decimal point. Should the result of the calculation contain more digits to the right of the decimal point, the number will be rounded.

There is one other setting that you can enter for the decimal point. The number 255 will cause floating-point printing of the results. This means that the number of digits to the right of the decimal point will depend upon the number of digits following the decimal point in the numbers that were added. PaperClip defaults to floating-point addition.

## Adding A Row

PaperClip can also add up a row of numbers — adding being done across rather than down.

Use the Set Column function to define the row of numbers you want to add. The number of lines in the Column doesn't matter since only the first row in the Column will be checked. Use the Set Delimiters function to set up the field delimiters. (This is explained in the next section: Advanced Column Functions). One field should hold one number.

Move the cursor to the place where you wish the result to be put and,

Press: **CTRL**
Press: **(shift) H**
Press: **RETURN**

You will be asked if you want leading spaces in the fields to be skipped, in case you are using spaces as delimiters. The row will then be added and the answer placed at the current cursor position.

## Changing The Line Length

In order to perform operations with columns each line must end with **RETURN**. Each row in a chart must occupy only one line in text. This can lead to a number of problems. Many business applications require large charts which could easily take up 1ØØ columns or more.

Earlier in the manual we explained how vertical scrolling allowed the screen to act as a window into text which could be moved up and down. Horizontal Scrolling gives you the ability to move this window left and right within text.

To use Horizontal Scrolling, you must change the line length.

Press: **CTRL**
Press: **(shift) L** [Length]
Press: **RETURN**

The message in the Status Line is **Line Length** and the message at the top of the screen is **New line length?**. Enter the line length you need. The maximum is 25∅, the minimum is the size of the screen your computer has. An 8∅-column display has a minimum length of 8∅; a 4∅-column unit has a limit of 4∅.

Press: **RETURN**

The screen then displays **Are you sure?**. *Changing the line length erases all text. Be sure that all text is saved.*

Now you have a system which is capable of handling charts and tables up to 25∅ columns wide. Not only will the column functions work properly, but your chart will be easier to read since there are no wrap-around effects.

## Horizontal Scrolling

To invoke scrolling simply use the cursor keys to move the cursor across the screen. Horizontal Scrolling will occur ten columns from the edge of the screen.

When the cursor reaches the end of the screen line it will move down to column 1 on the next line. Note that the tab line is also scrolled. When the tab line is scrolled the numeric tabs are not visible although they are still active.

Whenever a file is saved on disk using **CTRL S**, the tab line and the line length are saved with it. When you load a file back in, the line length will be set to what it was when you saved the file. *Appending a file* (**CTRL A**) will cause PaperClip to reformat the incoming text to the current line length,allowing you to change the line length of a document. The sequential file load (**CTRL J**) will also reformat a file to the current line size.

# Advanced Column Functions

## Sorting A Column

One of the most powerful column functions is the ability to sort the contents of a column. Sorting will be done only on text within the column you set using CTRL **(shift) C**.

To sort a column, first set up your column within text. The column should contain all the data that is to be sorted. All text outside the column will not be affected by the sort.

Sorting is performed based on "fields". A field is a distinctive part of each "record". In this case the records are the rows in the column. When sorting occurs, entire records will be exchanged although the sorting will be based on the fields within each record. A record is permitted to have up to 16 fields. Erase the screen and type the following list:

| | |
|---|---|
| Type: **Chuck Wagon** | Press: RETURN |
| Type: **Ben Dover** | Press: RETURN |
| Type: **Terry Dactyl** | Press: RETURN |
| Type: **Rose Ecrusion** | Press: RETURN |
| Type: **Sal Amander** | Press: RETURN |
| Type: **Perry Scope** | Press: RETURN |
| Type: **Holly Hocks** | Press: RETURN |
| Type: **Al Uvial** | Press: RETURN |

Press: CLR/HOME
Press: CLR/HOME

The cursor should be over the **C** in **Chuck**.

Press: CTRL
Press: **(shift) C**
Press: RETURN

The **C** should be highlighted.

Press: **S**

The whole line is highlighted.

Press: CRSR ↓ [hold it down until it stops]
Press: RETURN

The column is defined.

Press: CTRL
Press: **(shift) A**
Press: RETURN

Status Line reads **Sort Column**, tab line reads **Field number?**.

Press: **2**
Press: RETURN

We are specifying that we want to sort by last name, that is, the second field. The prompt at the top of the screen is **More?**. We have two people with the same last name so we will set up a "secondary sort" on the first name.

Press: **Y**
Press: RETURN
Press: 1
Press: RETURN

We have specified that the secondary sort will be field one. The top of the screen displays **More? N**.

Press: RETURN

The top of the screen displays **Ascending order? Y**.

Press: RETURN

We want our list in ascending order. If we wanted our list in descending order we would press N and RETURN.

The screen prompts **Ignore leading spaces in fields? Y**.

Press: RETURN

The list is now sorted into the order specified.

## Delimiters

In the above sort example the computer knew the difference between field one (first name) and field two (last name) because the SPACE between the fields acted as the field separator or *delimiter*. If no delimiter is specified then SPACE is the default delimiter.

Here is another list, of book titles, cost, and retail prices:

| | | |
|---|---|---|
| Basic Basic | 21.25 | 29.25← |
| Computers | 5.25 | 9.95← |
| The Joy of Computing | 18.5Ø | 24.95← |

If we want to sort the list based on the book title only it would be straightforward. The problem arises if we require a secondary sort or we want to sort the list based on the cost price. The third field in line one is the cost, the second field in line two is the cost and the fifth field in line three is the cost. Remember that PaperClip uses spaces for field separation unless otherwise specified.

In order to sort the list properly we need to specify a distinctive delimiter character. The following will show how we can set this up.

Type in the list. Make sure that each line ends with a RETURN and does not wrap around.

## Filling A Column With A Delimiter

Press: CLR/HOME

Move the cursor to the space before 21.25 in line 1.

Press: CTRL
Press: **(shift) C**
Press: RETURN

Hold down CRSR ⇓ until you reach the bottom of the column.

Press: RETURN
Press: CTRL
Press: **(shift) W**

Status Line reads **Delimiters** and the prompt is **Delimiter character?**.

Press: **(shift)** SPACE
Press: RETURN

The Column will fill with the shifted SPACE character.

## Setting The Sort Delimiter

Press: CTRL
Press: **(shift) Q**

Status line reads **Set Delimiters** and the prompt is **Delimiters?**.

Press: **(shift)** SPACE
Press: RETURN

You can enter up to sixteen delimiter characters, one for each field. In our case we specified the shifted SPACE to separate fields one and two; PaperClip assures that SPACE's separate the other fields.

## Sorting Columns Of Numbers

Set a column over the entire area using **(shift) C**.

Press: CTRL
Press: **(shift) A**
Press: RETURN

We are now in the sort column mode. The prompt is **Field number?**.

Press: **2**
Press: RETURN
Press: RETURN
Press: RETURN

To sort by field number 2. To the question **More? N** answer **N** — we do not require subsorting. To the question **Ascending order? Y** accept the default answer for ascending order. The screen now prompts **Ignore leading spaces in fields? Y**.

Press: **N**  
Press: **RETURN**

If we specify that leading spaces in the fields are to be ignored then the **5.25** would have been listed as the most expensive book and would be at the bottom of the list. We will not ignore the leading spaces in each field. The list will be compared as **21.25, Ø5.25, 18.5Ø** and will sort properly.

*If you need to sort columns of numbers then you must not ignore leading spaces when sorting.*

Study the example and try different options — if you understand the reasoning you will be able to sort even complicated lists.

# Appendices

# Summary Of Commands

## CONTROL FUNCTIONS

| | |
|---|---|
| CTRL | CTRL |
| ESCAPE | ← |
| ✓ | £ |
| TAB | RUN/STOP |

## MISCELLANEOUS

| | |
|---|---|
| Caps Lock | ↑ or CAPS |
| Switch Composite/RGB Display | 4Ø/8Ø |
| Change Line Length | CTRL **(shift)** L |
| Change Screen Color | |
| Text | **(shift) f2** |
| Background | **(shift) f4** |
| Border | **(shift) f6** |
| Cursor to Top Left of Screen | HOME |
| Cursor to Beginning of Text | HOME HOME |
| Cursor to End of Text | **(shift)** RUN/STOP |
| Delete One Character | DEL |
| Insert One Blank | **(shift)** INST |
| Load Character Set | CTRL ↑ |
| Quit PaperClip | CTRL X |
| Rapid Scrolling | |
| Down | CTRL CRSR ⇓ |
| Up | CTRL **(shift)** CRSR ⇑ |
| Screen Read | RUN/STOP |
| Switch 4Ø /8Ø /16Ø 32Ø Char Mode | **(shift) f8** |
| Switch Insert Mode | C= |

## FILE HANDLING

| | |
|---|---|
| Load | CTRL L |
| Save | CTRL S |
| Scratch | CTRL > S **Drive** #:[filename] |
| Append | CTRL A |
| Directory -- Disk Ø | CTRL Ø |
| Directory — Disk 1 | CTRL 1 |
| Directory - Both | CTRL 2 |
| Load Both Directories | CTRL 3 |
| Global Copy | CTRL G |
| Verify | CTRL U |
| Load (sequential) | CTRL J |
| Save (sequential) | CTRL Z |

## SPELLING CHECKER

| | |
|---|---|
| Start | CTRL **(shift)** Y |
| (when a word is not found) | |
| skip word | f1 |
| skip and ignore | f3 |
| replace word in text | f5 |
| add word to user-directory | f7 |

## RANGES

| | |
|---|---|
| (group of lines) | |
| Set | **CTRL R** |
| define with: **CRSR ⇓** | |
| **(shift) CRSR ⇑** | |
| **RETURN** (end definition) | |
| Copy (replaces target) | **CTRL C** |
| Transfer | **CTRL T** |
| Delete | **CTRL D** |
| Erase (replaces with blanks) | **CTRL E** |
| Save To A File | **CTRL Q** |

## PHRASES

| | |
|---|---|
| (group of characters within a paragraph) | |
| Set | **CTRL P** |
| define with: **S** (sentence) | |
| **W** (word) | |
| **CRSR** keys | |
| **RETURN** (end definition) | |
| Move (does not delete phrase) | **CTRL M** |
| Kill (deletes phrase) | **CTRL K** |
| Set and Delete | **CTRL Y** |
| Convert to Upper or Lower Case | **CTRL (shift) K** |

## COLUMNS

| | |
|---|---|
| Set | **CTRL (shift) C** |
| define with: **CRSR** keys | |
| **S** (extend to right edge of text) | |
| **RETURN** (end definition) | |
| Move | **CTRL (shift) M** |
| Shift (replaces target) | **CTRL (shift) S** |
| Delete | **CTRL (shift) D** |
| Erase (replaces with blanks) | **CTRL (shift) E** |
| Insert Spaces | **CTRL (shift) I** |
| Replicate (replaces target) | **CTRL (shift) R** |
| Add | **CTRL (shift) =** |
| Add a Row | **CTRL (shift) H** |
| Set Decimal Point | **CTRL .** |
| Sort | **CTRL (shift) A** |
| Create Delimiter | **CTRL (shift) W** |
| Set Delimiter | **CTRL (shift) Q** |

## LINES

| | |
|---|---|
| Insert One Line | **CTRL +** |
| Insert Multiple Lines | **CTRL I** |
| Delete One Line | **CTRL -** |
| Erase to End of Text | **CTRL E** |
| At the End of a Line: | |
| Break With Hyphen | **ESCAPE -** |
| Break Without Hyphen | **CTRL (shift) !** |
| Do Not Separate Two Words | **(shift) SPACE** |

## SEARCHING

| | |
|---|---|
| Specifying Search String: | |
| Match Any Character | £ |
| Match Any Alphabetic Character | ? |
| Match What Follows Exactly | ’ |
| Match At Beginning Of Word | [ |
| Match At End Of Word | ] |
| Match A Word | [word] |
| Screen Read | RUN/STOP |
| Recall Previous Search String | (shift) RUN/STOP |
| Find String | CTRL F |
| Hunt (for previous search string) | CTRL H |
| Search And Replace | CTRL @ |
| To Stop The Search | CTRL |

## OUTPUT

| | |
|---|---|
| To Printer (with options) | CTRL O |
| To Printer (no options) | CTRL (shift) O |
| To Video Screen | CTRL V |
| Restart Output | CTRL (shift) P |
| During Output (between pages:) | |
| Continuous | C |
| Discontinuous | D |
| To Printer | P |
| Restart At Last Page | R |
| To Video Screen | V |
| Load Printer File | CTRL W |
| Set Printer Device Number | CTRL (shift) # |
| Send Special Character | CTRL ; or ESCAPE & |
| Set For RS-232 | CTRL (shift)" |

## TABS

| | |
|---|---|
| Set | (shift) CLR/HOME |
| Clear | (shift) CLR/HOME |
| Tab | TAB or RUN/STOP |
| Clear All | CTRL (shift) CLR/HOME |
| Set (numeric) | CTRL N |
| Clear (numeric) | CTRL N |
| Tab (numeric) | ESCAPE TAB or ESCAPE RUN/STOP |

## SPECIAL PRINTING

| | |
|---|---|
| Boldface | |
| Begin | ESCAPE ( or CTRL ( |
| End | ESCAPE ) or CTRL ) |
| Italics | |
| Begin | ESCAPE < |
| End | ESCAPE > |
| Subcripting | |
| Begin | ESCAPE ! or CTRL 9 |
| End | ESCAPE" or CTRL / |
| One Character | ESCAPE $ or CTRL 6 |

| | |
|---|---|
| Superscripting | |
| Begin | ESCAPE % or CTRL 7 |
| End | ESCAPE ' or CTRL 8 |
| One Character | ESCAPE # or CTRL 4 |
| Underlining | |
| Begin | ESCAPE [ or CTRL [ |
| End | ESCAPE ] or CTRL ] |
| One Character | ESCAPE . |

## DISKS

| | |
|---|---|
| Issue Disk Command | CTRL > |
| Format New Disk | **N[drive]:[diskname],[id]** |
| Scratch A File | **S[drive]:[filename]** |
| Duplicate (entire disk) | **D[dest drive]=[source drive]** |
| Validate | **V[drive]** |
| Rename File | **R[drive]:[newname]=[oldname]** |
| Copy (transfer all files) | **C[dest drive]=[source drive]** |
| (only specified files) | **C[drive]:[dest]=[drive]:[source]** |
| Display Error Message | CTRL < |
| Set Disk Drive Device Number | CTRL **(shift) $** |

## VARIABLE BLOCKS

| | |
|---|---|
| Define | CTRL **B** |
| Find Next | CTRL **(shift) F** |
| Empty All | CTRL **(shift) N** |
| Fill Next | CTRL **(shift) B** |
| Fill All | CTRL **(shift) V** |
| Save Data | CTRL **Z** |
| Change Filename | CTRL **(shift) Z** |
| Data Separator | RETURN or ESCAPE ✓ |

# Summary of Directives

## Directives

(* = must be on a line by itself)
(**N** = a number)

### MISCELLANEOUS

| | |
|---|---|
| * ✓ **cm:anything** | comment within text |
| * ✓ **lnN** | print N blank lines |
| ✓ **ofN** | offset printer output N to print |
| * ✓ **ps:message** | pause and print message on the tab line |
| ✓ **digit=value** | set digit equal to ASCII value |
| ✓ **letter=phrase** | set letter equal to phrase |

### SETTING UP

| | |
|---|---|
| ✓ **lsN** | set line spaceing to N lines per inch |
| ✓ **pgN** | set paging after N lines |
| ✓ **ppN** | set physical page length to N lines |
| ✓ **ptN** | set pitch to N characters per inch |
| ✓ **p#N** | set current page number to N |
| ✓ **spN** | set line spacing to N full lines |
| ✓ **vpN** | leave N lines at the top of each page |

### LEFT MARGIN

| | |
|---|---|
| ✓ **lmN** | set left margin at column N |
| ✓ **lm+N** | reset left margin N spaces to right |
| ✓ **lm-N** | reset left margin N spaces to left |

### MARGIN RELEASE

| | |
|---|---|
| ✓ **ma+N** | release to N columns right of left margin |
| ✓ **ma-N** | release to N columns left of left margin |

### RIGHT MARGIN

| | |
|---|---|
| ✓ **rmN** | set right margin at column N |
| ✓ **rm+N** | set right margin N spaces to right |
| ✓ **rm-N** | set right margin N spaces to left |

### CENTERING

| | |
|---|---|
| ✓ **cnØ** | turn centering off |
| ✓ **cn1** | turn centering on |

### INDENTATION

| | |
|---|---|
| ✓ **ai+N** | indent each paragraph N to right |
| ✓ **ai-N** | indent each paragraph N to left |

### JUSTIFICATION

| | |
|---|---|
| ✓ **juØ** | turn justification off |
| ✓ **ju1** | turn justification on |

### RIGHT ALIGNMENT

| | |
|---|---|
| ✓ **raØ** | turn right alignment off |
| ✓ **ra1** | turn right alignment on |

## FORCE NEXT PAGE

| | |
|---|---|
| *✓ **fp** | force next page unconditionally |
| *✓ **fpN** | force paging if less than N lines remaining |

## HEADERS

| | |
|---|---|
| ✓ **hdN** | set up header N lines above text |
| ✓ **hlN** | set left margin of header to column N |
| ✓ **hrN** | set right margin of header to column N |
| ✓ **mlØ** | unlock header margins |
| ✓ **ml1** | lock header margins |
| ✓ **phN** | set header pitch to N |

## FOOTERS

| | |
|---|---|
| ✓ **flN** | set left margin of footer to column N |
| ✓ **frN** | set right margin of footer to column N |
| ✓ **ftN** | set footer N lines from bottom |
| ✓ **pfN** | set pitch to N |

## FILES

| | |
|---|---|
| ✓ **ep:FILE** | transfer data from external file to printer |
| *✓ **ex:FILE** | external file link (one level only) |
| *✓ **lk** | non-specific Global file link |
| *✓ **nx:FILE** | specific Global file link |

## TABLES OF CONTENTS

| | |
|---|---|
| *✓ **tb:entry** | add entry to table of contents file |
| *✓ **tf:FILE** | open table of contents file |

# Summary of Error Messages

**No Line Range set.**
A Line Range function was selected and no Line Range had been set. Set the Line Range and select the function again.

**No Phrase set.**
A Phrase function was selected and no Phrase had been set. Set the Phrase and select the function again.

**Column not set.**
A Column function was selected and no Column had been set. Set the Column and select the function again.

**Error: Device not present.**
When an attempt was made to send information to the disk drive or printer (for example, saving a file), the device did not respond and was deemed to be "not present". Make sure that the disk drive and printer are turned on and connected to the computer properly. Also make sure that the device numbers of the devices are the same as the ones which PaperClip is using.

**Error: Timeout on IEEE.**
When the computer tried to get information from the disk drive (for example, loading a file) the disk drive did not respond in time. This is a "timeout". It usually is an indication that the device is not present but there may be other problems. Follow the procedures for the "Device not present" error above.

**No text to save.**
An attempt was made to save a text file when there was no text in memory. This will not apply to the Line Range Save.

**Out of text memory.**
When doing an insert function of some sort and it was necessary to insert a line, this error will be generated if the last line of text is used. In other words no line could be inserted without destroying text. This can also occur during a file load or file append if the incoming file cannot fit into the text memory.

**No tab stops set.**
The tab key was pressed when there were no regular or numeric tabs set. Set up your tabs beforehand.

**No search string.**
The Find function, Hunt function, or Search and Replace function was called and the search string did not contain any characters. A "null" search string cannot be searched for. Set up your search string properly.

**Verify error.**
The text in memory is not the same as that in a sequential file.

**File not found (EOT).**
During a tape operation, a file search was in progress when an End of Tape marker was encountered. Check to make sure that the file you want is in fact on the tape.

**Out of Variable Data.**
Either **CTRL (shift)B** or **CTRL (shift)V** was executed and no more data could be retrieved from the current variable file. If variable data was being read during output then this would indicate the end of output.

**Out of Variable Blocks.**
Either **CTRL (shift)F** or **CTRL (shift)B** was executed and there were no variable blocks found after the cursor position.

**Overflow error.**
During a Column Add or a Row Add, the running total exceeded 38 digits. Even if fixed-point addition is specified, floating-point addition is used internally and fixed-point is only handled when the result is to be printed out. This is a rare error.

**Output Errors.**
Whenever an error is generated during output the cursor will be left at the place in text where the error was encountered.

**Format error.**
During output, PaperClip could not make the line fit between the margins. There is no break in the line — no spaces, breakpoints, or conditional hyphens were found. Check margin settings.

**Syntax error.**
A formatting directive could not be interpreted. Check to make sure that the directive is entered properly.

**Margin error.**
The margins have been set to an illegal value. The left margin must always be less than the right margin. The ✓ **ma** directive must not make the left margin less than 1. Check to make sure that your margins are set up properly and that the ✓ **ma** and ✓ **ai** values are not too large.

**Paging error.**
The total number of lines established by adding up the ✓ **hd**, ✓ **ft**, and ✓ **pg** values must not exceed the total number of lines on the page set by the ✓ **pp** directive. Also, the ✓ **vp** value must be less than the ✓ **pg** value. Check all paging directives, notably the ✓ **hd** and ✓ **ft** values.

**Header error.**
This error can occur when either the header or footer is to be printed. It is usually caused by the header overflowing the margins. Another cause has to do with the centering of the central portion of the header/footer. If it cannot be centered properly without overlapping either the left portion or the right portion then this error will be generated.

**Illegal quantity error.**
A number greater than 255 was encountered when trying to evaluate a number for a directive. Change the number.

**Output Terminated.**
Output will be aborted if the printer does not respond when PaperClip tries to print. This error is usually the result of a **Device not present** error and is only generated when printing is first started.

# Unpacking PaperClip

The PaperClip package you received should contain the following:

PaperClip diskette (compatable with your drive, see below)
PaperClip key (with plug that fits Control Port 1, Port 2 on C-128)
PaperClip warranty and registration card

There are two types of PaperClip diskettes. Those labeled **1541, 2031, 4040** can be used with Commodore disk drive models 1541, 2031, 4040 and any other drives which are read compatable. Those labeled **8050** can be used with the Commodore SFD 1001, 8050 and 8250 model drives. Please make sure the PaperClip diskette you have is compatible with your disk drive. If you are not sure check with the dealer where you bought PaperClip. It is important that the diskette and disk drive are compatible for PaperClip to work.

The minimum equipment needed to use PaperClip is:

Commodore 64 or 128 computer with suitable display (color or B&W)
Commodore disk drive or C2N cassette unit (see below)
Commodore or other manufacture printer (with suitable interface)

PaperClip can be used with a Commodore 64 computer without a disk drive, however, a disk drive is needed to initially set up PaperClip. Also, some functions of PaperClip can only be used with a disk drive. We recommend that a disk drive be purchased if you intend to do more than an absolute minimum of wordprocessing.

If you are using a 2031, 4040, 8050, 8250 or other Commodore IEEE disk drive (in other words, not a 1541) then please note that PaperClip has only been tested and proven compatable with the following IEEE interfaces:

| | |
|---|---|
| The **BusCard** | - Manufactured by Batteries Included |
| The **C-64 Link** | - Manufactured by Richvale Telecomunications |

If you have any other IEEE interface connected to your system please check with your dealer to make sure it is compatable with PaperClip before proceeding.

## Will your printer work with PaperClip?

There are many printers and printer interface combinations which may be connected to a Commodore 64/128. Most of them will work with PaperClip. Since it is impossible to predict all possible combinations of computer, printer and printer interface, we cannot guarantee that a given combination will work. The following should help you decide if your particular equipment will function properly with PaperClip.

If all your equipment (including any interfaces and cables) is manufactured by Commodore then you will not have any trouble. At the time of printing, only two printers were made by Commodore for the Commodore 64 computer. They are the 1525 and 1526 printers. Both of these work properly with PaperClip.

If you have a Commodore manufacture IEEE printer (4023, 8023, etc) the IEEE adapter should be one of the units listed above.

Non-Commodore printers can be divided into three groups: RS-232, Centronics parallel and Serial bus connection.

### RS-232

These usually connect to the RS-232 port on the left rear corner of the Commodore 64 computer. Most printers connected in this way should have little difficulty in working correctly with PaperClip. For technical information on how PaperClip uses the RS-232 port refer to the Appendix: RS-232 - Printer Port. We have tested PaperClip with the following interfaces:

| | |
|---|---|
| Data20 Printer Interface | - Data20 Corporation |
| 1011A | - Commodore Business Machines |
| 232-1 | - Batteries Included |

### Centronics Parallel

This will be a cable connected to the RS-232 user port on the left rear of the Commodore computer. In most cases this cable will be part of a product which provides some other function, such as an IEEE interface or Basic language extension. For technical information on how PaperClip uses the Centronics Parallel port refer to the appendix Parallel Printer Port. The following cables have been tested with PaperClip:

| | |
|---|---|
| C-64 Link | — Richvale Telecomunications |
| Vic Tree | — Skyles Electric Works |
| 2ØØ1-1 | — Batteries Included |

**Serial bus**
Interfaces which connect to the standard serial communications bus on the Commodore 64 (not the RS-232 port) fall into this group. These interfaces convert the Commodore serial bus to the communications protocol required by a particular printer. These interfaces may be used without difficulty with PaperClip provided they meet one requirement: the interface must be capable of transmitting *ALL* data sent to the printer with *NO* character conversion of the data being sent. PaperClip must be able to send any combination of data to the printer without interference. The following interfaces have been tested with PaperClip:

| | |
|---|---|
| Cardprint | - Cardco Inc. (type in commands below before loading PaperClip)<br>(OPEN 4,4,25 : PRINT#4 : CLOSE4) |
| MW 3Ø2 | - Micro World (set switches for pass-through mode) |
| BusCard | - Batteries Included (set switches for pass-through) |

## How to begin using PaperClip on your Commodore 64.

*This section will assume that you have a simple computer system consisting of a Commodore 64 computer, a Commodore 1541 single disk drive and a printer.* If you do not own a disk drive, please refer to the appendix Putting PaperClip On Cassette, which describes the process of converting PaperClip for use with only a tape drive.

Plug the PaperClip key, (the small plastic box with a connector on one end) into Control Port 1 (Port 2 on C-128) on the right hand end of the Commodore 64 computer. Make sure the key is inserted all the way in. Turn on the Commodore 64 and display (TV or monitor). Check that all the keys on the keyboard function as usual. If some keys no longer work properly see the appendix Solving Problems.

Right now you should make a copy of the original PaperClip diskette. (See the appendix Copying A Diskette for instructions.) Store the original PaperClip diskette in a protected place away from the rest of your diskettes.

Now, using the copy, load PaperClip as explained in the Getting Started section at the beginning of this manual.

# Putting PaperClip on Cassette

If you do not own a disk drive, PaperClip may be used with a Commodore C2N cassette unit. To convert PaperClip for tape, use the following procedure. You will need a Commodore 64 computer, 1541 disk drive and C2N casette unit.

PaperClip must be merged with a printer file before being saved on tape if you wish to have PaperClip print. If you do not have a printer then the merge operation may be omitted. Refer to the appendix Using The Merge Prtfil Program to choose the correct printer file, and the appendix Choosing A Printer File for instructions on merging a printer file with PaperClip.

Put the PaperClip diskette into the disk drive. Make sure the label is up. Refer to the disk drive manual if you are unsure about how to insert the diskette. Close the door on the drive.

Type: **LOAD"Ø:*",8**
Press: **RETURN**

The disk drive should start whirring and the *RED* drive lamp should light. After a minute or so, the screen should read:

```
LOAD"Ø:*",8

SEARCHING FOR Ø:*
LOADING
READY.
```

If you have any difficulty loading PaperClip refer to the appendix Solving Problems.

Put a cassette tape into the C2N unit. The tape should be fully rewound and have a playing time of at least 1Ø minutes per side.

Type: **save"paperclip64"**
Press: **RETURN**

The Message: **PRESS PLAY AND RECORD ON TAPE** should appear on the screen. Press the RECORD and PLAY buttons down together.

The screen will go blank. After about 8 minutes the display should reappear:

```
WRITING PAPERCLIP64
READY.
```

Turn the tape over, rewind it, and save PaperClip again on the other side.
Store the original PaperClip diskette in a protected place.

## To use PaperClip from tape.

Type: **load"paperclip64"**
Press: **RETURN**

Press the PLAY key on the C2N cassette unit. The screen will go blank for a minute or so. When the display reappears, it should look like this -

```
LOAD PAPERCLIP64
SEARCHING FOR PAPERCLIP64
FOUND PAPERCLIP64
```

Press the Commodore logo key (lower left corner). The screen will go blank again, this time for about 8 minutes. When the display returns,

Type: **run**
Press: **RETURN**

# Utility Programs

There are several programs supplied on the PaperClip diskette which are useful for general housekeeping. All the utility programs require the use of a disk drive.

**backup64** — This program allows you to use a Commodore 64 computer and a single slot disk drive (1541 or 2Ø31) to make duplicate copies of the PaperClip diskette or any other diskette. See the appendix Copying A Diskette for detailed instructions on how to use this program.

**backup64k** — This is an expanded version of backup64. It may not work with all Commodore 64 computers since it sensitive to devices such as IEEE adapters. However, if you can use it you will be able to make backup copies with fewer exchanges of the diskettes. Standard Commodore 64 computers with a 1541 disk drive should have no problems with this program. Try it on your system — if the resulting diskette is completely normal, then use this program.

**backup 2Ø31** — This program allows you to use a Commodore PET or CBM computer and a single slot disk drive (2Ø31) to make duplicate copies of the PaperClip diskette or any other diskette. See the appendix Copying A Diskette for detailed instructions on how to use this program.

**merge prtfil** — This program allows you to define a specific printer file as the default entry in PaperClip. It works by reading in the original PaperClip program, then the specified printer file, combines the two, then saves the modified copy of PaperClip on another diskette. This new copy of PaperClip will not require that the printer file be retrieved with the **CTRL W** command before printing. See the appendix Using the **merge prtfil** Program for instructions on defining a default printer file.

**printer setup** — If you have a printer which can not be used with any of the existing printer files, then this program can create a printer file tailored to your particular printer. Please note that few printers cannot be used with one of the existing printer files. Before you can use this program you will have to be familiar with the control codes and sequences of your printer. See the appendix Using the **printer setup** Program for detailed information on how to set up a printer file.

# Copying A Diskette With A Single Drive

If you have a model 1541 or 2Ø31 single disk drive use the method outlined below to make a copy of any diskette, including the PaperClip64 diskette. Owners of a Commodore 64 computer should use the program on the PaperClip64 diskette called **backup64**. If you have a PET or CBM computer use the program on the PaperClip diskette called **backup 2Ø31.** (Use C-128 Computers in 64 mode.)

Reset the computer (turn it off and back on).

Put the PaperClip64 diskette in the drive and close the door.

Load the appropriate program as follows:

Type: **load"backup64"**,8 (**load"backup2Ø31",8** for PET or CBM)
Press: RETURN

When the program has loaded, remove the PaperClip master diskette and insert a blank diskette into the drive.

Type: **run**
Press: RETURN

The program will ask you for a name and an ID code, any name up to 16 characters long may be used, and any 2 characters may be used for the ID code. (If you are making a backup copy of the original PaperClip64 diskette, use a name of **paperclip** and an id code of **xx**.)

The program will format the destination diskette (the blank one you just put in). It will then request you to put the source (original) diskette in the drive (the PaperClip diskette if you are making a PaperClip backup). The program will read as much of the diskette contents into memory as possible, then ask that you exchange the diskette with the destination diskette. The information stored in the computer is put on the new diskette, and the program will repeat the process until all information on the original diskette has been transferred to the new diskette. Now you have two copies of the diskette.

Take care of your diskettes. If one should become damaged, all information (your documents) stored on that diskette will be lost. Make backup copies of all important diskettes frequently. One day you will be glad you did! See the appendix Diskette And Cassette Care for instructions on caring for your diskettes.

# Using The merge prtfil Program

Normally, the printer file for your printer must be retrieved with the **CTRL W** command before printing for the first time each sesion. If you always use the same printer file you can use the **merge prtfil** program to define that printer file as the default printer file. Once you have done this it is no longer necessary to retrieve the printer file with the **CTRL W** command before printing. PaperClip will automatically work with that printer file. You can still retrieve other printer files, but PaperClip will always *start* with your printer file.

This program works by reading in the original PaperClip program, then the specified printer file, combining the two, then saving the modified copy of PaperClip on another diskette.

**merge prtfil** can be used to merge any printer file with any copy of PaperClip64, PaperClip or PaperClip Expanded, using either a Commodore 64 or a PET/CBM computer.

Proceed as follows:

Put a copy of the PaperClip diskette in the disk drive.

Type: **load"merge prtfil*",8**
Press: **RETURN**
Type: **run**
Press: **RETURN**

The first question will ask for the name of the original PaperClip program on the PaperClip diskette. Enter the program name exactly as it appears in the diskette directory and press **RETURN**.

Enter the drive number where the PaperClip diskette is (if you have a single drive, enter **Ø**) and press **RETURN**.

**merge prtfil** will load the PaperClip program into the computer.

Next it will ask for the name of the printer file you wish to have as the default. Enter the printer file name exactly as it appears in the diskette directory and press **RETURN**.

**merge prtfil** will fetch the printer file and combine it with the PaperClip program.

The modified PaperClip, with your printer file as the default, is ready to be stored on a diskette.

Remove the original PaperClip diskette from the disk drive and store it. Insert a *new* diskette (either blank or formatted) into the drive.

**merge prtfil** will ask for the drive number where it is to store the new PaperClip program. If you have a single drive, enter **Ø**.

If the diskette on which you want to save the new copy of PaperClip is already formatted, answer **N** to the next question. If the diskette is not formatted, answer **Y** and **merge prtfil** will ask for a diskette name and an ID code. Enter up to 16 characters for the diskette name and press **RETURN**. Enter two characters for the ID code and press **RETURN**.

Now enter the name you wish to call the new copy of PaperClip. Use a meaningful name.

**merge prtfil** will store the new copy of PaperClip on the diskette. From now on, all you have to do is load and run your copy of PaperClip. Unless you change printers, you will not have to use the **CTRL W** command.

# Creating A Printer File

## What Is A Printer File?

The printer file provides a translator between PaperClip and your printer. The printer file customizes PaperClip for your particular printer. When you tell PaperClip to print, it must send information to the printer for setting line spacing, printing text, doing underlining, etc. The exact codes it sends out to the printer are held in an area of memory called the "printer file buffer". The contents of a printer file are held here. The printer file buffer serves as a translation table between PaperClip's internal text format and the manner in which your printer processes information. When you type **CTRL W** you cause PaperClip to transfer a printer file from the disk into this buffer. The buffer is separate from the text area. That's why no visible action occurs when you execute **CTRL W**.

## Why use printer files at all?

Of the thousands of printers on the market only a few use the same protocols for controlling various functions. Let's look at a typical example of a word processing function: underlining. How do you go about underlining? The common way to do it on most letter-quality printers is to print out a character of the text, followed by a backspace character, followed by the underline character. The backspace character simply causes the printer to back up over the previous character.

So why doesn't PaperClip do all underlining like that if it's so simple? While this is true for most letter-quality printers, many dot-matrix printers don't have a backspace character. While letter-quality printers usually print character by character, dot-matrix printers usually print entire lines at once and therefore cannot do backspacing. So how do you do underlining on a dot-matrix printer?

Some dot-matrix printers have an "underline mode" which can be turned on and off. The printer will then do the underlining for you. You simply turn it on before your underlined text and off afterwards. The printer does the rest. So how do you turn it on and off? That's the whole reason for having printer files. To turn built-in printer functions on and off, you must send down some sort of character sequence which in essence says "turn underlining on" or "start printing in italics" or something. These character sequences vary greatly from printer to printer. All of the printers on the market have varying degrees of intelligence (built-in functions). Some letter-quality printers have "underline modes" as well. The printer files tell PaperClip which codes to send to the printer to get it to do something. They also tell PaperClip how some functions, like underlining, are implemented on that particular printer.

## What's in a printer file?

To understand the relationship between the contents of a printer file and your printer, we can look at how PaperClip communicates with a printer. When PaperClip wants the printer to print something, say the letter "A", it will send a number to the printer. All information is transmitted to and from PaperClip as numbers. We may see them as letters or symbols, but inside they are dealt with as numbers. When the printer receives this number it will print out whatever character corresponds that number. Not all numbers actually cause something to be printed on the paper. The number which causes the paper to advance forward is an example of this.

There is usually a one-to-one correspondence between the numbers PaperClip sends and the characters which the printer prints. A standard "character set" called ASCII — American Standard Code for Information Interchange — is used by most printers. ASCII is a standard arrangement of the alphabetic characters which the printer can produce. Each character is seen by the computer as a number. For example, the number which corresponds to "A" is 65. So an ASCII printer will print an "A" whenever it receives the number 65. All of the other printer characters have been assigned numbers in the ASCII character set, including the carriage return and backspace characters.

When a printer file is loaded into the buffer, it will contain all of the numbers the computer can send to print various characters. PaperClip *MUST* know which number corresponds to which character in order for it to operate your printer properly. This is the basic function of the printer file.

## The Null Character

The first number in the printer file is the "null" character. It does not have anything to do with actual printer control, but rather with how the data is arranged in the remainder of the printer file. Any of the numbers in the printer file may range from Ø to 255. Usually, they will contain a number which corresponds to a character to be sent to the printer. When PaperClip is sending numbers to the printer, it first compares each number with the "null character" number. if the number to be sent to the printer is the same as the null character number, then PaperClip won't send the number.

The null character is simply the number which indicates to PaperClip "I am not a character to be sent to the printer". The null character number will *NEVER* be sent to the printer.

The next 128 numbers in the printer file contain the translation table for all the characters which you see on the screen: letters, numbers, etc. When PaperClip wants to print an "A", it can pick a number from the corresponding place in the table and send it to the printer.

The last 256 numbers in the printer file contain the various control sequences for turning on and off functions such as underlining, boldface, italics, paper feeding, etc.

## Making a printer file

To get a close look at just what goes into a printer file, and also to see how one is created, follow the example below. Using the Basic language, **LOAD** and **RUN** the program **printer setup** from the PaperClip diskette.

This example creates a printer file for an Epson MX8Ø TYPE III (Graftrax) printer. It doesn't matter whether you have this printer or not. This is just an exercise in creating a printer file.

The first question the program will ask is **Load previous file?**. If you anwer yes, you will be able to edit a previously set up printer file.

If you answer no then you will be asked for the "null" character for this file. Remember that whatever number you give will designate to PaperClip the number which corresponds to "no character to send". Be careful when you select this. Choose a number which will *NEVER* be sent to the printer. Remember that a printer with a 7-bit interface (a bit is a digit in a binary number) can only receive numbers from Ø to 127. A number like 255 would be a good value for the null character for such a printer since it is a number which the printer could never process.

Since the MX8Ø printer is an ASCII printer, we can save a lot of time by starting off with the standard ASCII character set and then adding to it. Answer "yes" to the **Load previous file** question.

You will now be asked for the name of the printer file we wish to edit. Type in **True ASCII**. The True ASCII printer file is set up to drive a dumb ASCII printer—one that cannot do underlining or anything fancy, just print text and that's all. The MX8Ø printer is an intelligent printer, so we'll take the True ASCII printer file and "smarten it up".

The next question is **ASCII value for "@"?** On the line below will be a question mark, followed by a number, followed by the cursor. The number is the value currently assigned to the "@" character in the printer file.

If you tap the cursor-down key, you will be asked **ASCII value for 'a'?**. Typing the cursor-down key allows you to skip forward in the printer file. Typing the cursor-up key will step backwards. To change the number assigned to a character, delete the old number with the delete key, then enter the new number and press **RETURN**. The new number will be accepted and you will skip forward to the next question. Numbers must be in the range Ø to 255.

If you type the reverse-off key (**CTRL** with the **Ø** key on the Commodore 64), you will move to the second section of the printer file—the control section. Striking reverse-off while you are in this section will move you back to the character section. Since we don't have to modify the character set in any way for our MX8Ø printer, we can go on to the control section. If you are still in the character section, type reverse-off.

## Carriage Returns and Line Feeds

The first question in the control section is **Carriage return (line feed)?** Before we can answer this question, an understanding of line feeds is in order. Whenever PaperClip has finished sending a line of text to the printer, it will then send a "carriage return" character. This signals the printer that PaperClip has finished that line and is about to start another.

Different printers react differently to this character. Some printers will automatically advance the paper to the next line. These are refered to as "auto-line feed" printers since they feed the paper forward automatically every time they receive a carriage return. Other printers don't move the paper automatically. In order to get them to advance the paper, we must send down the "line feed" character.

Sometimes, PaperClip can make better use of printers which don't automatically advance the paper. Since the paper doesn't move when the carriage return character is sent, PaperClip can do multi-pass printing, printing multiple times on the same line. PaperClip does this for underlining if the printer does not have a backspace character or an underline mode.

Since the MX8Ø is generally configured as being an auto-line feed printer, we should enter the value 13 to the question *Carriage return (line feed)?* 13 is the number which corresponds to a carriage return. PaperClip will send this number to the printer as a carriage return. It also knows that this character will cause the printer to automatically do a line feed. If the MX8Ø printer was not auto-line feed then we would enter the null character value here to indicate that there is no character that would do a carriage return AND a line feed.

The next question is **Carriage return (no line feed)?** Here PaperClip needs to know what the number for a carriage return without a line feed is. Since the MX8Ø is auto-line feed, enter the null character value here. The null character value in the True ASCII printer file is 255. Type in 255 and strike return. If the MX8Ø printer had not been auto-line feed then we would have entered 13 here for the carriage return character which did not generate a line feed.

The next question is **Line feed character?** This is normally used in conjunction with the *Carriage return (no line feed)* and really doesn't make a difference to our auto-line feed MX8Ø. However it's good practice to give all the information you can, so let's enter it anyway. The standard line feed value is 1Ø. Enter the number 1Ø and press **RETURN**.

After this, you will be asked **Backspace character?** This is the number which will cause the printer to do a backspace. Our MX8Ø printer does have a backspace character. Enter the value 8, which is the standard backspace code. Again, if our printer did not have a backspace character, we would enter 255 (or whatever the null value is).

The next question is **Underline character?** The underline character on the MX8Ø has a value of 95. Even though we will be using the MX8Ø's built-in underline mode for underlining, it pays to give it this value anyway.

If we were to stop now and save this printer file as it is, we would find that PaperClip could do underlining and double-striking on our MX8Ø printer. We have all the basic characters in place: the character set itself including the carriage return setup, the backspace character and the underline character. In fact, this file should work on any ASCII printer with a backspace and underline character. But the printer can do more than that—things like superscripting and subscripting.

The next question is **Printer initialize characters?** This time we need to know just what to send to get the printer to initialize (reset). Usually this requires a sequence of several characters. Since more than one character will be needed, we can enter a five character (or less) sequence. Right now only the first number should be on the screen. If you type the cursor-right key, the next in the series will appear, and so on until you have five numbers on the screen. The cursor-right key always moves you to the next number in a multi-character series. The cursor-left key always moves you to the previous number.

The characters to reset the MX8Ø are as follows: **ESCAPE**,"@". **ESCAPE** is a very important ASCII character, like backspace or carriage return. It usually doesn't print anything and that's why it has a name.

So what do we enter for **ESCAPE** "@"? The value for Escape is 27, and the value for "@" is 64. So we want to enter the numbers 27 and 64. Do this now. Enter each number, press **RETURN**, and then enter the next number. Enter the null value (usually 255) for the last three entries in the five-character sequence. This will tell PaperClip that we are only dealing with a two-character initialization sequence.

You will now be facing another question: **Secondary address?** This question must be answered every time we enter a five-character sequence. Instead of using an **ESCAPE** code, some printers use what are called "secondary addresses". Different things happen when you send numbers to different secondary addresses. For the most part, only Commodore manufacture printers use these. A few printer interfaces, such as the Cardprint, use secondary addresses. One secondary address is used for setting lower case, another for line spacing, etc.

Our printer doesn't use secondary addresses, so we enter the number Ø as our secondary address. If you enter the null character value for the secondary address, PaperClip will assume a secondary address of Ø. Secondary addresses must be in the range Ø to 31. *Note:* PaperClip always sends all *text* to secondary address Ø.

You will now be asked **Line begin characters?** This five-character sequence is sent to the printer just before each line of text. Some printers require that a specified code preceed any line of data. This is where to put it. On Commodore 2Ø22 and 2Ø23 printers, we send a 17 here to put the printer into upper and lower case. Our MX8Ø doesn't need anything like this, so just type cursor-down to skip to the next question.

You will now be asked **Underline begin characters?** This sequence of characters is sent to the printer whenever underlining is to be turned on. The MX8∅ printer does have an underline mode, so we should enter the characters which turn this mode on. They are Escape,"-",1. So the numbers we want are 27, 45, and 1. Enter these numbers. Again, enter the null character value for the remainder of the five-character sequence. Enter ∅ for the secondary address as well.

The next question is naturally **Underline end characters?** This sequence is sent whenever underlining is to be turned off. The sequence of characters which do that on our MX80 printer are Escape,"-",∅. Enter the numbers 27, 45, and ∅. Pad out the remaining numbers with the null character value and enter a value of ∅ for the secondary address.

After this comes **Boldface begin characters?** This sequence is sent whenever boldface print is turned on. The MX80 printer has an emphasized print mode which will serve our purpose. It is turned on with Escape,"E". Enter the numbers 27, and 69 plus the three null values.

The next question is **Boldface end characters?** Here we want to turn our emphasized print off. The characters to do this are Escape,"F". Enter the numbers 27 and 7∅.

The next question is **Underline begin (no line feed)?** If PaperClip cannot do underlining using an underline mode (the *Underline begin* question), it will try to do it using the backspace and underline characters. If there is no backspace character, PaperClip then checks to see if it can do a carriage return without a line feed. If it can, it will send that character, then do a second pass on the line using this sequence of characters. This is called multi-pass underlining. Since our MX8∅ printer has an underlining mode, we don't need to worry about this series of characters. Type the cursor-down key to go to the next question.

After this comes **Underline end (no line feed)?** This sequence is sent just at the end of multi-pass underlining. Again, type the cursor-down key to go to the next question.

We now have **Boldface begin (no line feed)?** and **Boldface end (no line feed)?**These are virtually identical to the previous two questions except that they pertain to multi-pass doublestriking or boldface. On the first pass, text is printed in normal mode. On the second pass it is printed in emphasized mode. We don't need to worry about these questions since we have already handled boldface print on our MX8∅III. Type the cursor-down key twice to skip both questions.

The next question is **Superscript begin characters?** This is a five-character sequence that will be sent to the printer whenever superscripting is turned on. On letter-quality printers, this would often be a "negative half line feed" sequence — a sequence of characters which would cause the printer to roll back the paper by half a line. Our MX8∅ printer cannot do negative half line feeds, but it does have a superscripting mode, so we'll use that here. The superscripting mode is turned on with ESCAPE,"S",∅. So enter 27, 83, and ∅ as our numbers. You can see why ∅ would have been a poor choice for the null character value because this is the second time we have had a character sequence with a ∅ in it.

After this comes **Superscript end characters?** This sequence is sent to the printer whenever superscripting is to be turned off. On our MX8∅, this sequence is ESCAPE,"T". This will turn off the MX8∅'s superscript mode. Enter 27, and 84 as the first two numbers. Now we're not done yet. A close inspection of the MX8∅ manual reveals that whenever we turn super or subscripting on in the printer, the printer also turns on the "double-printing mode". Yet turning superscripting off will does not turn this mode off. So we must do it. To turn off the double-printing mode, the characters are ESCAPE,"H". So next two characters in our sequence are 27, and 72. This will get the printer back to the state it was in before we turned the superscript mode on.

The next question is **Subscript begin characters?** This too is a five-character sequence which is sent to the printer whenever subscripting is turned on. To turn on subscripting in our MX8∅ printer we must send the characters ESCAPE,"S",1. So enter the numbers 27, 83, and 1.

After this we will be asked **Subscripting end characters?** This sequence is sent whenever subscripting is turned off. To turn off the subscripting mode in our printer, the characters are also ESCAPE,"T". So enter the numbers 27, and 84 for this question as well. As with superscripting, the printer automatically invokes double-print mode during subscripting, so we must turn it off with ESCAPE,"H" here as well. Enter 27, and 72 for the last next two characters in the five-character sequence.

The next question is **Italics begin characters?** This five-character sequence will be sent to the printer whenever italics printing is turned on in text. Our MX8Ø does have an italics character set so we should enter the character sequence that will start printing in italics. This is done with the character sequence ESCAPE,"4". So enter the numbers 27, and 52. We do not enter a value of 4 for the second number but rather the value which corresponds to the character "4" which is 52.

Next we will be asked **Italics end characters?** This sequence is sent whenever italics printing is turned off. To turn off italics printing in our MX8Ø printer we must issue the character sequence ESCAPE,"5". So enter the numbers 27, and 53.

The next question is **Value to add for italics characters?** Some printers do not have an italics character mode, but rather have the italics characters as an "offset" of the standard character set. In normal seven-bit ASCII, the actual printing characters are in the range 32 to 126. If you go back to the character section of the printer file you can see just where the various characters fall into this range. In seven-bit ASCII no number greater than 127 would have any meaning to the printer. Suppose then that to get the printer to print an italic character, as opposed to normal printing, we just add 128 to the number assigned to the normal character. So normal characters would have the range 32 to 126 and italics characters would have the range 16Ø to 254. In this case the value 128 would be the "offset" to get italics characters. This is what the question is asking. It wants to know what number it should add to the standard character values to get the italics characters. For our MX8Ø this number is 128. But we don't need to do that since in the previous two questions we told the word processor how to turn on the italics characters mode. In this mode, the printer looks after the offset all on its own. So we really don't need to do anything here. Enter a value of Ø. If the MX8Ø printer did not have an italics character mode, we would have entered 128 so that PaperClip could access the italics characters. This question only requires one number as an answer since it is not a character sequence but rather information for the word processor to use.

You will now be asked **Pitch = 1Ø characters per inch?** As you remember, there are four available pitches in PaperClip, set with **✓pt1Ø**, **✓pt12**, **✓pt15** and **✓**ptXX where XX is any number other than 1Ø, 12, or 15. PaperClip wants to know what numbers it must send to the printer in response to the **✓pt1Ø** directive. Our MX8Ø does not actually have a pitch command as such, but it does have standard, condensed, and enlarged character modes. So let's use those for our pitch settings. It would make sense that **✓pt1Ø** would cause the printer to print in the standard character mode. So whenever **✓pt1Ø** is issued, we will want to cancel any previously set printing modes. To cancel the enlarged character mode, we must send down the number 2Ø. To cancel the condensed character mode, we must send the number 18. So for this question we should enter the numbers 18 and 2Ø.

The next question is **Pitch = 12 characters per inch?** This sequence will be sent in response to the **✓pt12** directive. Since the MX8Ø does not have a print mode that prints at 12 characters per inch, we will skip this question for the moment. Type the cursor-down key to go to the next question.

For **Pitch = 15 characters per inch?**, we can put the printer into the condensed character mode. This is done by sending the number 15 to the printer. But before we do that, we should also cancel the enlarged character mode, just to make sure. Otherwise we could end up in the enlarged condensed character set which the MX8Ø also has. Our printer should respond the same way every time the **✓pt15** command is issued. We can cancel the enlarged character set by sending the number 2Ø to the printer. So enter 2Ø, and 15 as our numbers for this question. This will cancel the enlarged character mode and put the printer into the condensed character mode.

The next question is **Optional pitch?** This sequence of characters is issued if a ✓ ✓**pt** value of other than 1Ø, 12, or 15 is selected. Here, we will have our MX8Ø start printing enlarged characters. Before we do that, we must make sure to cancel the condensed mode as well. So our numbers will be 18 to cancel the condensed mode, and 14 to enable the enlarged mode.

After this comes the question **Line spacing = 6 lines per inch?** This is a five-character sequence which is sent to the printer whenever the **✓ls6** directive is encountered. Glancing through the MX8Ø printer manual we find that to set the line spacing, we must issue the two character sequence, ESCAPE,"2". Remember that this is the character "2" and not the number 2. The ASCII value for "2" is 5Ø. So we want to send the numbers 27, and 5Ø. Enter these numbers for this question.

The next question is **Line spacing = 8 lines per inch?** This character sequence is sent in response to the **✓ls8** directive. To do this on our MX8Ø printer we must send ESCAPE,"Ø", that is the numbers 27 and 48. Enter these two numbers.

You will now be asked **Optional line spacing?** This is sent whenever PaperClip encounters an ✓ **ls** value other than 6 or 8. We must decide what our optional line spacing is going to be. We could make it anything we want, but a standard line spacing is 4 lines per inch, which is the equivalent of 1 ½ line spacing on a typewriter. So we want to set the line spacing to 4 lines per inch for our optional line spacing. If we check at the printer manual we see that there is no preset way of telling the printer "give me 4 lines per inch". We had nice short character sequences for the previous two line spacings, since we could take advantage of the built-in line spacing settings of the printer. What do we do if the printer has no direct line spacing command for 4 lines per inch? Well, closer inspection of the manual shows that there is a more general way of setting the line spacing. The printer has a "stepper motor" which is used to advance the paper. Whenever the printer does a line feed, the stepper motor advances the paper a certain number of steps. When we send Escape,"2" down, the printer figures out how many steps that should be to get 6 lines per inch. The same thing applies to our 8 lines per inch setting. But we can also select the number of steps ourselves. We can tell the printer to do however many steps we want. So just how many steps correspond to 4 lines per inch? Well, the printer does 72 steps per inch. And 72 divided by 4 is 18. So we want to tell the printer to do 18 steps for every line feed. After four line feeds the printer will have done 18 steps 4 times, which is 72, which will have moved the paper one inch. The sequence to tell the printer to do 18 steps is **ESCAPE**,"A",n where 'n' is the number of steps we want. In this case our 'n' is 18. So enter the numbers 27, 65, and 18 in response to this question.

We could also have used this method of setting the line spacing for the previous two questions. 'n' values of 12 and 9 would have corresponded to line spacings of 6 and 8 lines per inch respectively.

After this comes **Microspacing for boldface (using bs)?** This ominous looking question has to do with how PaperClip will do boldface print using the backspace method. Double-strike printing is done by typing a character, doing a backspace, then typing the character again. Boldface print is done a little bit differently. The printer will print a character, do a full backspace, go forward just a tiny little bit (a microspace), then re-type the character. In this way, the two strikings of the same character are not quite over top of each other. This produces a fatter and darker character which is much more effective than simple double-striking. So what PaperClip wants to know here is how to set the pitch as small as possible, so it can do the microspace. Normally the pitch would be in the neighbourhood of 1Ø characters per inch. But if it is set at something like 1ØØ characters per inch, the actual spacing would be very tiny. In the case of our MX8Ø printer we don't need to worry about this since we are using the emphasized print mode in the printer which essentially does the same thing for us automatically. It causes each character to be printed twice, slightly offset. Type the cursor-down key to skip to the next question.

Our next question is **Forced 8-bit characters begin?** This is actually refering to the printer interface. As mentioned earlier, ASCII is a 7-bit character code, meaning that only numbers Ø to 127 have any meaning to the printer. Many printers are 8-bit, so the number range is increased to 255. Remember our discussion of italics printing in the MX8Ø? The extra bit is used to select the italics character set (as well as a few other things). So what's the problem? Well, the computer can send 8-bit numbers to the printer, and the printer can receive and process 8-bit numbers. So far so good. The problem lies in the interface (communication connection) between the computer and the printer. Many interfaces can only handle 7 bits, which effectively eliminates any possibility of transmitting an 8-bit number to the printer. So how do we get around this problem? Well, when a 7-bit number is sent to the printer, the eighth bit will always be a "Ø". The same thing would occur when we send an eight bit number over a 7-bit interface—the interface would force the eighth bit to be Ø. In eight bit binary, any number greater than 127 has a "1" for the eighth bit. So we must find some way of telling the printer to ignore what the interface tells it for the eighth bit. Many printers have a way of "forcing" the eighth bit to be one. So here's what the computer would do when it wants to send an eight-bit number over a seven-bit interface: it would tell the printer that the following character has a 1 in the eighth bit; send the lower seven bits of the charcter; then cancel the "force eight bit" mode. The printer would take the seven-bit number it got from the computer and set the eighth bit to be one. It would then process the number. So this question is asking how to tell the printer to force the eighth bit to be a "1". If the word processor ever has to send an number with a "1" for the eighth bit, it will send this sequence of characters down first. Fortunately we don't have to worry about this since all of the numbers we have entered so far have been less than 128, or only seven bits. Type the cursor-down key to skip to the next question.

Now we are asked **Forced 7-bit characters begin?** This sequence will be sent after any eight-bit character to cancel the *Forced 8-bit character* mode. Type the cursor-down key to skip to the next question.

The next question seems a bit odd at first: **Number of line feeds per line?** Whenever PaperClip has finished printing a line, it needs to know how many carriage returns (and line feeds if necessary) to send to the printer before it can start printing a new line. Remember when we entered the numbers which would tell the printer to set the line spacing to 6 lines per inch? Suppose we had entered the numbers which would actually set the line spacing to twelve lines per inch.

If that were the case, we would need to do two carriage returns (and linefeeds) to effectively get six lines per inch. So what, you might ask? What's the point in doing all this? Why would you ever want to set the line spacing to a multiple of what you want? That comes into play with the next question. For the moment, enter 1.

The next question is **Superscript mode?** If this is ∅, then PaperClip will behave as described up until now. However, if you enter 128, PaperClip will do "multi-pass" super and subscripting. Normally, when a carriage return is sent to the printer, the paper will advance one line. But if you change the line spacing character sequences so that you actually get three times as many lines per inch (i.e. ✓ls6 would actually give you 18 lines per inch), the paper would advance only one third of a line in response to a carriage return. And you would need to give a value of 3 in the previous question (number of line feeds per line). To move the paper one full line PaperClip would need to send 3 carriage returns instead of 1. This allows us to do some fancy tricks with the printer. Suppose that instead of printing a line and then doing three carriage returns, PaperClip did this: print all superscript characters, do a carriage return, print the main line of text, do a carriage return, print all the subscript characters. This would enable you to do super and subscripting even if the printer didn't have a super/subscript mode or could not do "negative half line feeds". This is what PaperClip will do if you respond with 128 to this question. It will do three passes per line, one each for superscripts, main text, and subscripts. All you have to do is set the line spacing so that it is some multiple of the desired spacing (i.e. some multiple of 6 and 8) and set the "number of line feeds" number to the correct value.

The next question is Microspacing on (SPS)? **SPS** refers to "semi-proportional spacing". This has to do with how PaperClip will do justification on your printer. Normally PaperClip justifies text by "padding it" with extra spaces to make the right-hand margin line up. This is not always best since some words will have more than one space between them, while others will have only one space. Semi-proportional spacing is a method of justification where the spaces between all the words on a line are equal. This is done by inserting "microspaces" instead of whole spaces between words. Remember that a microspace is just a very tiny space. What this question is asking is how to turn microspacing on so it can do the semi-proportional spacing. In the case of our MX80 we don't have a microspacing mode so we will skip this question. Type the cursor-down key to go to the next question.

We are now asked Microspacing off (SPS)? PaperClip now wants to know how to turn microspacing off. Normally this would probably be done by just setting the pitch back to normal. But there may be other ways of doing it. For example, we could actually use the "bit image graphics" mode to do microspacing, but unfortunately this makes the MX8∅ tediously slow. Again, type cursor-down to skip the question.

After this we're asked Semi-proportional spacing mode? This is a number which will tell the word processor whether or not it can do semi-proportional spacing, and just how it should be done. If semi-proportional spacing cannot be done, then this number should be ∅. If it can be done, then it must be at least 128. Add a value of 64 if the "microspacing mode" must be turned on for each microspace. In other words, if there is a gap between two words which has three microspaces in it, can we just turn microspacing on and send out three spaces, or must we turn it on each time we want to do one microspace? Again, add a value of 1 to this if the pitch setting should be re-set after microspacing is done. We should enter a value of ∅ for our MX8∅ printer.

The next question is Number of microspaces per space (pt1∅)? PaperClip needs to know how many microspaces make up a regular space at the **✓pt1∅** setting. The next three questions will also ask the same thing for the other three pitch settings. Again, we can skip these questions since we will not be doing semi-proportional spacing on our MX8∅ printer.

After this comes the question **Space character (SPS)?** This is the character that the computer will send to the printer to do a microspace while the printer is in microspacing mode. If you are controlling the microspacing by simply setting the pitch to a very small setting, this character would be the same as a normal space. But if you are using some sort of graphics mode, it might be something altogether different. We can skip this question too.

The next two questions refer to multi-lingual character printing. They will both be covered in the section on multi-lingual characters. Type the cursor-down key twice to skip these questions for the time being.

After this comes **Byte postion for 'space' character?** This refers back to the five character sequence that turns microspacing on to do semi-proportional spacing. Presumably one of those numbers indicates just how many microspaces are to follow, or possibly sets the size of the microspace. The answer to this question should be a number from ∅ to five indicating just which of the numbers in the sequence sets the microspacing size. An answer of ∅ indicates that none of the bytes controls the microspacing size.

The next question is **Offset for spacing byte (SPS)?** This is an offset value which will be added to the number of microspaces about to be done in order to set the microspacing properly. To get an idea of just what all this means, let's take a look at a typical example of doing semi-proportional spacing on a Nec Spinwriter (55ØØ series or 77ØØ series). The character sequence on this printer to change the pitch to the smallest value (microspacing) is **ESCAPE**,"]","A". So what we could do when we wanted to output three microspaces is send this sequence, followed by three spaces, followed by the normal pitch setting. That's easy enough. But is there a faster way? Yes, why bother sending three spaces? Why not send only one? To do this we would need to make our new microspace equal to three of the old microspaces. Can we do this? Of course we can, just by changing the pitch setting to make it a little bit bigger. The character sequence this time would be **ESCAPE**,"]","C". So how can we get the computer to do this kind of thing automatically—to set the printer up so it only has to send one larger space instead of a lot of little ones? Well, as you've probably noticed, it is the third character in our sequence which sets the actual size of the microspacing, so our answer to "Byte position...." is 3. We want to alter the third byte when we change the spacing. But we also need to know the offset which determines what this third character is. Now then, the ASCII value for "A" is 65. And the ASCII value for "C" is 67. So 65 corresponds to one microspace, and 67 corresponds to three microspaces. Clearly, 64 corresponds to Ø microspaces and is our offset value. So for PaperClip to generate the third character in the sequence, it takes the number of microspaces it wants to send and adds that to the offset value, 64.

So, if we want to do one microspace, the computer would send down **ESCAPE**,"]". Then it would add 1 to the offset value 64, giving a value of 65. It would then send this number to the printer. So the effective sequence would be **ESCAPE**,"]","A", since 65 is the number for an "A" in ASCII.

The next question will be **Secondary address for text?** This is the secondary address that the main text will be sent over. On non-Commodore printers, this will be Ø. Valid ranges for secondary addresses are Ø to 31. A secondary address greater than 127 will be interpreted as "no secondary address". If the text secondary address is equal to the null character value, a secondary address of Ø will be used. PaperClip64 users who are using a Cardco interface to hook up their printer, might want to change this to 25 to "lock" the interface into its invisible mode. If they do so, they will not have to type in the Basic open statement before running Paperclip.

## Multi-lingual Character Sets

To produce multi-lingual characters on the printer, the corresponding codes must be included in the printer file as well. Go back to the character section, and skip forward until you are asked for the number of the first multi-lingual character. If you are using the French character ROM, this character will be an "e" with an accent on it. PaperClip handles multi-lingual characters slightly differently than normal characters. It treats them as two characters overprinted on top of one another. Suppose your printer doesn't have a character for an "e" with an accent, but does have the accent character by itself. Here's how you would do it. In response to the **ASCII value for 'e'** question, enter the ASCII code for an "e" without an accent. You will now be asked "Second character?" What the word processor wants to know now is what character it should over-strike the first one with. Now enter the code which produces the accent character. When PaperClip prints, it will first send down the code for the "e", followed by a backspace, followed by the second character — the accent. This requires that your printer must be able to do a backspace. Of course, if your printer actually has an "e" with an accent, then simply enter that code in response to the first question and enter the 'null' character value in response to the "Second character?" question. PaperClip will check to see that the "Second character" is not the 'null' character before it sends the backspace. Remember that you can still do multi-lingual characters even if your printer does not have a backspace, provided that all of the "Second character" values are the null character value.

You will remember that we skipped over two questions before. They were **Multi-lingual character set on?** and **Multi-lingual character set off?** These are five-character sequences which are sent to the printer immediately before multi-lingual character printing is done, and immediately afterward. Some printers (such as the Nec Spinwriter) cannot normally access special characters like accents, etc. A special sequence must be sent to gain access to these, usually swapping some existing standard characters out (making them temporarily unavailable). These two sequences will select and deselect multi-lingual character sets on printers like this. They are similar to the "Italics begin" and "Italics end" sequences.

Once you have your printer file set up, you will want to save it on a diskette. This is done by typing the reverse-on key. (On the Commodore 64, hold down **CTRL** and press the **9** key). You will then be asked for a name for your printer file and the drive number on which you want to save it. The printer file will then be written onto the diskette so that you can load it into PaperClip using **CTRL W** or merged with PaperClip using the **merge prtfil** program.

## Summary of Printer File Control Sequences

### Null Character

This is the value against which PaperClip will compare all charaters being sent to the printer. If they match, then the character will not be transmitted. For example, if the null character had a value of 65, which corresponds to the ASCII value for "A" then the number 65 would *NEVER* be sent to the printer. Whenever an "A" was encountered in text, it would be ignored. Similarly any values of 65, which were encountered in control sequences would also be ignored. The null character is the value which is used to mean "no character". It usually has a value of 255.

### Carriage return (line feed)

If this character is not equal to the null character then it will be sent to the printer at the end of each line of text that is printed. PaperClip will expect this character to cause the printer to print out any text it has received and advance the paper. The number of carriage returns that PaperClip will send out is determined by the value in the "Number of carriage returns per line" question.

### Carriage return (no line feed)

If the "Carriage return (line feed)" character is equal to the null character then PaperClip will send out this character along with the "Line feed" character at the end of each line. PaperClip will expect that this character will cause printing to be done and that the "Line feed" character will cause the paper to be advanced. This character may also be sent without the "Line feed" character in order to do more than one pass on the same line for such things as boldface printing, underlining, etc.

### Line feed character

This character is sent along with the "Carriage return (no line feed)" character if the "Carriage return (line feed)" character does not exist (i.e. is equal to the null character value).

### Backspace character

This character is sent to the printer for doing such things as underlining and boldface printing. PaperClip will expect the printer to do one backspace when this character is sent. This character is also used for the multi-lingual characters. It will be sent after the first character in the multi-lingual character sequence is sent, providing that the second character is not equal to the null character value.

### Underline character

This character is used in both the backspace-style and multi-pass underlining operations. If PaperClip determines that it must do backspace-style underlining (i.e. there is no valid "Underline begin" sequence) then PaperClip will send this character followed by a backspace for each character that is to be underlined.

### Printer initialize character

This sequence is sent at the beginning of each output and at the beginning of each subsequent copy in multi-copy output. Its basic function is to reset or initialize the printer.

### Line begin characters

This sequence is sent out at the beginning of each line. In the printer files for the Commodore 2022 printer, this sequence is used to tell the printer to go into upper/lower case mode.

### Underline begin characters

This sequence is sent out whenever underlining is turned on within text. If this character sequence is not valid (all null characters) then PaperClip will check to see if there is a valid backspace code and a valid underline character. If there is then it will do underlining that way. If there isn't then it will check to see if it can do a carriage return without a line feed. If so it will do multi-pass underlining.

### Underline end characters

This sequence is sent to the printer whenever underlining is turned off within text.

### Boldface begin characters

This sequence is sent whenever boldface printing is turned on within text. If this sequence is invalid, PaperClip will do checks similar to underlining to see if it can do backspace-style or multi-pass boldface print.

### Boldface end characters

This sequence is sent whenever boldface printing is turned off within text.

**Underline begin (no line feed)**
If PaperClip is doing multi-pass underlining this character sequence will be sent before the second pass — the one which prints the underlines — is done.

**Underline end (no line feed)**
This sequence is sent at the end of the second pass in multi-pass underlining.

**Boldface begin (no line feed)**
This sequence is sent before the second pass in multi-pass boldface printing.

**Boldface end (no line feed)**
This sequence is sent out at the end of the second pass in multi-pass boldface printing.

**Superscript begin characters**
This sequence is sent whenever superscripting is turned on within text. It can be used to invoke a superscript mode or to back the paper up by part of a line.

**Superscript end characters**
This sequence is sent whenever superscripting is turned off within text. It is also sent after a single character has been printed if single-character superscripting had been turned on.

**Subscript begin characters**
This sequence is sent whenever subscripting is turned on within text. It can be used to invoke a subscript mode or to advance the paper by part of a line.

**Subscript end characters**
This sequence is sent whenever subscripting is turned off within text. It is also sent after a single character has been printed if single-character subscripting had been turned on.

**Italics begin characters**
This sequence is sent whenever italics printing is turned on within text.

**Italics end characters**
This sequence is sent whenever italics printing is turned off within text.

**Value to add for italics characters**
This value is added to the ASCII codes for every character being sent out if italics printing is turned on.

**Pitch = 1Ø characters per inch**
This sequence is sent at the beginning of each line if the ✓**pt1Ø** directive has been issued.

**Pitch = 12 characters per inch**
This sequence is sent at the beginning of each line if the ✓**pt12** directive has been issued.

**Pitch = 15 characters per inch**
This sequence is sent at the beginning of each line if the ✓**pt15** directive has been issued.

**Optional pitch**
This sequence is sent at the beginning of each line if a value other than 1Ø, 12, or 15 has been used with the ✓**pt** directive.

**Line spacing = 6 lines per inch**
This sequence is sent at the beginning of each line if the ✓**ls6** directive has been given within text.

**Line spacing = 8 lines per inch**
This sequence is sent at the beginning of each line if the ✓**ls8** directive has been given within text.

**Optional line spacing**
This sequence is sent at the beginning of each line if a value other than 6 or 8 has been used with the ✓**ls** directive.

**Microspacing on for boldface (using bs)**
This sequence is sent during backspace-style boldface to get the pitch as small as possible. That way the second striking of the character would be slightly offset from the first leaving an impression that is thicker and darker. The pitch sequence corresponding to the current pitch setting will be sent immediately afterwards.

**Forced 8-bit characters begin**
This sequence is sent to the printer immediately before any character with a 1 (one) in the eighth bit. It will precede any character with an ASCII value greater than 127.

**Forced 7-bit characters begin**
This sequence will follow the transmission of any character with an ASCII value greater than 127.

**Number of line feeds per line**
This value tells PaperClip how many carriage return and line feed sequences it must send to advance the paper by one line.

**Superscript mode**
If this value is 128 and the number of line feeds per line is greater than 1, PaperClip will do separate passes for superscripts, text, and subscripts. It is recommended that 3 be used for the number of line feeds per line if multi-pass super and subscripting are going to be done.

**Microspacing on (SPS)**
If PaperClip is going to do justification using semi-proportional spacing, this sequence will be sent to the printer to set the pitch as small as possible before microspacing is done.

**Microspacing off (SPS)**
This is sent after microspacing for semi-proportional justification.

**Semi-proportional spacing mode**
This value tells PaperClip if it can do semi-proportional justification. A value of 128 indicates that it can. Add 64 if the "microspacing on" sequence must be issued for each microspace. Add 1 if the pitch must be reset after microspacing.

**Number of microspaces per space (pt1Ø)**
This is the number of microspaces that make up one regular-sized space when the ✓**pt1Ø** directive is in effect.

**Number of microspaces per space (pt12)**
This is the number of microspaces that make up one regular-sized space when the ✓**pt12** directive is in effect.

**Number of microspaces per space (pt15)**
This is the number of microspaces that make up one regular-sized space when the ✓**pt15** directive is in effect.

**Number of microspaces per space (optional pitch)**
This is the number of microspaces that make up one regular-sized space when the optional pitch is in effect.

**Space character (SPS)**
This single character is issued for each microspace during semi-proportional justification.

**Multi-lingual character set on**
This sequence is sent immediately before multi-lingual characters are printed.

**Multi-lingual character set off**
This sequence is sent immediately after multi-lingual characters are printed.

**Byte position for 'space' character**
This value (ranging from Ø to 5) indicates which character in the "microspacing on (SPS)" sequence actually determines the size of the microspace.

**Offet for spacing byte (SPS)**
If the "byte postion for 'space' character" is not Ø then this value will be added to the number of microspaces that need to be done and placed in the "microspacing on (SPS)" sequence before it is sent to the printer.

## Character Sequences Sent for Each Line of Text

The following character sequences will be sent in the order shown when a line of text is transmitted from PaperClip

1. The appropriate line spacing sequence will be sent. Two IEEE unlisten/listen sequences may be issued immediately before and afterwards if the line spacing sequence has to be transmitted to a secondary address other than Ø.
2. The line begin sequence will be sent. Two IEEE unlisten/listen sequences may be issued immediately before and afterwards if the line begin sequence has to be transmitted to a secondary address other than Ø.
3. The appropriate pitch sequence will be sent. Two IEEE unlisten/listen sequences may be issued immediately before and afterwards if the pitch sequence had to be transmitted to a secondary address other than Ø.
4. The offset and left margin will be sent.
5. Underlining, boldface, italics, superscripting, and subscripting begin sequences will be sent if these functions have been turned on within text.
6. The line of text will be transmitted.
7. Underlining, boldface, italics, superscripting, and subscripting end sequences will be sent if these functions have been turned on within text.
8. If multi-pass underlining or boldface have to be done then the second and third passes, if necessary, will be done.
9. Paper will be advanced to the next line.

In addition, the intialize printer and appropriate line spacing sequences are sent at the start of each document.

# Choosing A Printer File

## What Is A Printer File?

One of the many problems with producing output from a word processor is the fact that there are no standards about what printers can do, or how you tell them to do it. Although many of the features available with printers are fairly common, the way these features are invoked can vary widely.

Underlining is a good example of just how this can vary. Some printers have an "underline mode". When this mode is active, all printed text is underlined. But the actual code sequences used to turn "underline mode" on and off vary from printer to printer. Other printers require that the backspace character be used to underline text. You simply print the character, backspace, then print the underline character. It's all very simple. But some printers don't have a backspace character....

It is because of the lack of standards among printers that printer files have been devised. A printer file is a small table which contains specific information about your printer. PaperClip uses the information in the printer file to handle the formatting of text. The printer file contains information such as: the code for a backspace; the code for the underline character; how underlining is turned on; etc.

On the PaperClip diskette there are printer files for most of the popular printers. Most comon printers have been covered, so it is likely that there is one on the PaperClip diskette that is correct for your printer. If you happen to have a printer which is not on the list, check with your PaperClip dealer. Although printers vary widely in terms of control codes, it is quite possible that a printer file intended for another printer will work equally well with yours. For example, the Diablo printer has been around for some time, so other printer maunufacturers have copied the Diablo codes. This makes their printers easier to sell and it allows PaperClip to use the Diablo printer file with those printers.

If you discover that your printer is not listed with any of the printer files, and there is no printer file that is close enough to work, you can use the **printer setup** program to create a printer file with the characteristics of your particular printer. See the appendix Creating A Printer File for instructions.

If you build a printer file, and it works well, please send it to us at **Batteries Included**. There are probably others like you that would appreciate finding the printer file already on the diskette. We cannot, however, provide any royalties, credit or reimbursement for creating a PaperClip printer file.

## Choosing A Printer File For Your Printer.

Before any printing can be done, the appropriate printer file for your printer must be selected. Do not worry, if you choose an incorrect printer file neither your printer nor your computer will be harmed. The printed output may be unreadable, but that's all. Remember, the printer file is PaperClip's link with your printer. If the printer file is wrong, PaperClip will not be able to correctly control the printing process.

The PaperClip diskette comes with a number of printer files for use with a wide variety of printers. The printer files are two-block entries with a file name of the form:

**WWWW-X-YYY-ZZ**

**WWWW** represents the name of printer for which the printer file is intended. For example, a printer file that starts with **MX8∅** is intended for an Epson MX8∅ printer.

**X** will be either the letter **P** or **A**. This indicates what type of character representation the printer uses. **A** means that the printer file will generate *True ASCII* output. *Most* non-Commodore printers use this. **P** indicates that the printer file will generate *Pet ASCII*. The Commodore dot matrix printers such as the 4∅22 and 8∅23 models use this character representation.

**YYY** will be either **ALF** or **NLF**. This indicates how the printer file expects the line-feed mechanism in the printer to behave. PaperClip will send a Carriage Return character to the printer at the end of every printed line. This will cause the printer to print that line. At this point, printers which have an AUTO LINE-FEED mode will perform a line-feed operation and advance the paper one line. The ALF printer files expect the printer to do this. For printers which do not do a line-feed automatically, PaperClip must also send the line-feed character to the printer to advance the paper. The **NLF** files will cause this character to be sent.

In some cases PaperClip can actually do more on a printer which requires an **NLF** printer file. Since the printer does not advance the paper after printing, it is possible to do multi-pass printing — printing more than one pass on the same line. Underlining and boldface print can thus be done on printers which otherwise could not.

Some printer files will also have **SS** at the end of the file name. This indicates that the printer file will do super and subscripting on that particular printer using a very small line spacing. The line-feed would then cause the paper to advance by say, a third of a line instead of a whole line. On the first of these "thirds of a line" the superscripts would be printed, on the next third the main text, on the last the subscripts. The reason for providing separate **SS** files is that they may print more slowly than normal files because the line spacing is so tiny. You would effectively have three times as many lines per page. Many printers can produce super and subscripting with both the **SS** and the regular printer files. In these cases the **SS** entry will print *full size* characters while the non-**SS** entry will produce *smaller* super and subscript characters.

The following chart should help you find the right printer file for your printer. In many cases there will be two or four entries for the same printer. For the most part, the difference will be that one of the printer files expects the auto-line-feed to be turned on, another expects it to be turned off, one will do full size superscripting and subscripting, another will do small superscript and subscript characters, and so on. Once you find the right set for your printer, try each one out. One of them will probably work just right.

Quite often several printers will be able to use the same printer file. Therefore, the printer file which works with your printer may have a name taken from another printer. For instance, the printer file called **83ØØp-a-alf** is named that because it was made for the Commodore 83ØØP letter quality printer. It turns out that Diablo Corporation made the 83ØØP printers for Commodore. So the same printer file can be used for: Diablo printers, C. Itoh printers, TEC-F15 and most other letter quality printers.

Once you have chosen a printer file, you can test it out with PaperClip as follows:

Make sure a copy of the PaperClip diskette is in the drive.

Press: **CTRL**
Press: **L**
Type: **printer test**
Press: **RETURN**

The printer test document contains examples of every type of text PaperClip is capable of producing. Many printers will not be able to pass the test perfectly. They will simply ignore the commands they do not understand.

Press: **CTRL**
Press: **Ø** [the digit Ø]

Scroll the directory until the printer file you want to try comes into view. If the cursor is *not* flashing, press **CTRL**.

Move the cursor to the first letter of the printer file name.

Press: **CTRL**
Press: **W**
Press: **RUN/STOP** [the file name will appear on the tab line]
Press: **RETURN**

PaperClip will retrieve the printer file from the diskette. Now we are ready to test the printer file. Make sure the printer is ready to print, has paper in it, etc. If you are using an unusual interface, it is critical that it be correctly set up to pass all data *straight through without interference*.

Press: **CLR/HOME**
Press: **CLR/HOME**
Press: **CTRL**
Press: **(shift) O**
Press: **RETURN**

The printer should start printing. If things go wrong, such as the paper starts to feed continuously, turn the printer *off*. PaperClip will report the error, but no damage will occur. Check the interface settings (if any), load the printer file again, and try again. If it still doesn't work, you probably don't have the right printer file.

If the printer does print without trouble, let it finish. Now, check the printed output. See if each line is printed the way it is described. For instance, the line which reads **This should be printed in 12 pitch.** should be printed in 12 pitch. Note each place the printed output does not print as it should. If there are any other printer files which you think might work, try them. Choose the printer file which gives you the best overall success. If none of the printer files produce much luck, contact your PaperClip dealer.

The following list describes most of the printer files on the PaperClip diskette. Choose the one which matches your printer most closely.

## Printer Reference Chart

**1525-p-alf**
— Commodore 1515 and 1525 VIC printers — 5 and 1Ø pitch print only — italics produces reverse field print — no underline, bolface, super or subscripting possible

**1526-p-alf**
**1526-p-alf-ss**
— Commodore 1526 printer
— ss entry allows super and subscripting

**2Ø22-p-nlf**
**2Ø22-p-nlf-ss**
— Commodore 2Ø22 and 2Ø23 printers
— 5 and 1Ø pitch only, italics produces reverse field print
— ss entry allows super and subscripting

**4Ø22-p-nlf**
**4Ø22-p-nlf-ss**
— Commodore 4022, 4Ø22P and 4Ø23 printers
— 5 and 10 pitch only, italics produces reverse field print
— ss entry allows super and subscripting

**553Ø-a-alf**
**553Ø-a-nlf**
— NEC Spinwriters 553Ø, 773Ø, 353Ø, etc
— all functions supported except italics

**64ØØ-a-alf**
**64ØØ-a-nlf**
— Commodore 64ØØ letter quality printer
— all functions supported except italics

**737-a-alf**
**737-a-nlf**
— Centronics 7ØØ series printers
— all functions supported except italics

**8Ø23-p-nlf**
— Commodore 8Ø23P printer
— 5, 1Ø and 15 pitch, italics produces reverse field print

**8Ø23-p-nlf-ss**
— ss entry allows super and subscripting

Note 1: Pseudo letter quality mode is invoked as follows. Set pitch to 15, place a carriage return on the next line, then set the pitch to 6.

**83ØØp-a-alf**
**83ØØp-a-nlf**
— Commodore 83ØØ P, DPS 11Ø1 letter quality printer
— TEC F15 and some C. Itoh Starwriters
— Also most Diablo and Qume type printers
— all functions supported except italics

**daisy-a-alf**
**daisy-a-nlf**
— Daisywriter letter quality printer

**f1Øs-a-alf**
**f1Øs-a-nlf**
**f1Øs-p-alf**
**f1Øs-p-nlf**
— TEC F1Ø letter quality printer - see Note 2
— Also most C. Itoh Starwriters
— all functions supported except italics
— p entries are same but transmit in PET ASCII

Note 2: TEC F1Ø must be set in *Serial print mode* for these printer files to work properly. *This is not to be confused with RS-232 serial data transmission to the printer.* On some units this is controlled by pin 1 on the right hand set of DIP switches. These switches can be found just inside the front lip of the printer when the lid is open.

**fx8Ø-a-alf**
**fx8Ø-a-alf-ss**
**fx8Ø-a-nlf**
**fx8Ø-a-nlf-ss**
— Epson FX8Ø printers
— ss entries allow full size super and subscripting
— all functions suported in all files

**gemini-a-alf**
**gemini-a-alf-ss**
**gemini-a-nlf**
**gemini-a-nlf-ss**
— Star Micronics Gemini 1Ø, 1ØX, 15 and 15X printers
— ss entries allow full size super/subscripting
— all functions supported in all files

Note 3: Gemini 1Ø and 15 units we worked with would not allow small super and subscript characters to be mixed with wide pitch print. Use the ss entries if this combination is necessary.

| | |
|---|---|
| **mt16Ø-a-alf-dp**<br>**mt16Ø-a-alf-wp**<br>**mt16Ø-a-nlf-dp**<br>**mt16Ø-a-nlf-wp** | — Mannesmann Tally MT 16ØL printers<br>— wp entries support high quality print mode<br>— dp entries support draft quality mode<br>— all functions supported<br>— italics produces proportional print<br>Note 4: Units we worked with would not allow super and subscript characters to be printed in boldface.<br>Note 5: Proportional print mode ignores margin settings. |
| **mx8Ø-a-alf**<br>**mx8Ø-a-alf-ss**<br>**mx8Ø-a-nlf**<br>**mx8Ø-a-nlf-ss** | — Epson MX8Ø I, all Epsons without Graftrax<br>— alf entries do not underline or boldface<br>— only ss entries support super and subscript printing<br>— all functions supported except italics and 12 pitch |
| **mx8Øiii-a-alf**<br>**mx8Øiii-a-alf-ss**<br>**mx8Øiii-a-nlf**<br>**mx8Øiii-a-nlf-ss** | — Epson MX8Ø III, MX1ØØ III, Epsons with Graftrax Plus<br>— all functions except 12 pitch supported<br>— ss entries allow full size super and subscripting |
| **mx8Øp-p-nlf**<br>**mx8Øp-p-nlf-ss** | — Epson MX8Ø 'PET' converted printers (ESSNA)<br>— may not work with all versions of printer<br>— italics produces reverse field print |
| **mx82iii-a-alf**<br>**mx82iii-a-alf-ss**<br>**mx82iii-a-nlf**<br>**mx82iii-a-nlf-ss** | — Epson MX82 III, Epson MX82 with Graftrax Plus<br>— all functions supported by all files<br>— ss entries allow full size super/subcripting |
| **mx1ØØ-a-alf-ss**<br>**mx1ØØ-a-nlf-ss** | — Epson MX1ØØ II, MX8Ø II, MX82, Graftrax Epsons<br>— alf entry does not underline or boldface<br>— all functions supported except italics and 12 pitch |
| **oki82-a-alf**<br>**oki82-a-nlf** | — Okidata 82 and 82A printers<br>— 5, 1Ø, 12 and 15 pitch print<br>— only nlf entry allows boldface and underline |
| **oki84-a-alf**<br>**oki84-a-nlf** | — Okidata 84 printer (some Okidata 92 printers)<br>— all functions supported in all files<br>— italics produces pseudo letter quality proportional printing |
| **oki92-a-alf**<br>**oki92-a-alf-ss**<br>**oki92-a-nlf** | — Okidata 92 printer (some Okidata 84 printers)<br>— all functions supported in all files<br>— ss entries allow full size super and subscripting |
| **oki92-a-nlf-ss** | — italics produces pseudo letter quality proportional printing |
| **oliv-a-alf**<br>**oliv-a-nlf** | — Olivetti typewriter printers |
| **olym-a-alf**<br>**olym-a-alf-ss**<br>**olym-a-nlf**<br>**olym-a-nlf-ss** | — Olympia typewriters and printers, ESW 1Ø1, etc<br>— ss entries allow super and subscripting |
| **pc8Ø23-a-alf**<br>**pc8Ø23-a-alf-ss**<br>**pc8Ø23-a-nlf**<br>**pc8Ø23-a-nlf-ss** | — NEC PC8Ø23A, TEC 851ØA and some C.Itoh Prowriter printers<br>— ss entries allow super and subscripting |
| **tp1-a-alf** | — Smith Corona TP1 printer, any ASCII printer with backspace and underline characters |
| **ttx1Ø14-a-alf**<br>**ttx1Ø14-a-nlf** | — TTX printer model 1Ø14 |
| **Pet ASCII**<br>**True ASCII** | — transmits Commodore PET ASCII, no control codes<br>— transmits standard ASCII, no control codes |

# Solving Problems

## Keyboard does not work with the PaperClip key installed

If some keys do not work when the PaperClip key is plugged in, but do work with the PaperClip key removed, there are two possible causes. One is that the PaperClip key is defective and must be replaced. If this is the case then your PaperClip dealer should be able to replace the key. The second cause could be a fault in the electronics of the Commodore computer. If there is any problem with the joystick or paddle controls, the Paperclip key will not work. The PaperClip key uses circuitry in the Commodore which is *not* used by the keyboard, joysticks or paddles. It is possible that an otherwise functioning computer will not work with the PaperClip key installed. In this case the computer will have to be repaired before you can use PaperClip.

## The PaperClip program does not load

**Screen displays FILE NOT FOUND error.**

Make sure the diskette is inserted in the drive correctly and the drive door is closed. Check that the drive is correctly connected to the computer. The label on the diskette should have a number which matches the disk drive model (for instance, the diskette with 1541/2Ø31/4Ø4Ø matches the 1541 disk drive). Make sure the LOAD command was typed properly (see below). Make sure that there are no extra spaces typed in the command.

**Screen displays SYNTAX ERROR.**

SYNTAX ERROR means that the computer does not understand your command. Do not press the **(shift)** key while entering the LOAD command. Make sure the **SHIFT LOCK** key is not down. Only the quote" symbols need to be shifted. There should be no extra spaces in the commands. Enter the entire command and try again. Refer to the computer owners manual for more detail about entering commands.

**Screen displays LOADING but the READY message does not appear.**

It can take up to two minutes to load PaperClip. If the *RED* error light on the drive starts to flash (1541) or stays on (4Ø4Ø), then the disk drive is having difficulty reading the information from the diskette. Press: **RUN/STOP**, remove and re-insert the diskette, and enter the command again. If the error recurs, contact your PaperClip dealer. The PaperClip diskette, or your disk drive, may be faulty.

## The PaperClip program loads, but does not start properly.

**After typing RUN and pressing RETURN, nothing happens.**

PaperClip can take a minute or so to get rolling — give it time. Check that the PaperClip Key is installed in the correct Control Port. The key must be inserted before loading PaperClip. Remove any other devices from the control ports. Turn the computer off and on and try again. If PaperClip will not start after two or three trys, contact your PaperClip dealer. The fault could lie with the computer, the disk drive, the diskette, the PaperClip key, or a number of other things.

PaperClip for the PET and CBM computers should start up immediately. If it does not, check that the PaperClip chip was installed correctly. Make sure the computer has 31743 BYTES FREE when it is turned on.

**Screen fills with asterisks (*) and will not clear.**

You have a PaperClip Expanded chip and a standard PaperClip diskette, or the reverse. The chip and diskette must match. Contact your PaperClip dealer.

## Printer output is garbled or printer does not print

Make sure your printer can print correctly using the BASIC language. If the printer can't be used without PaperClip, it is unlikely that PaperClip will be able to print properly. If you are using a non-Commodore printer, make sure the interface (the cable or device that connects between the computer and the printer) is compatable with PaperClip. If you do not have one of the units listed in the Getting Started section, check the interface manual. It is critical that the interface be set correctly before PaperClip will be able to print properly. PaperClip must be able to send *ANY* data to the printer without the interface performing *ANY* conversions or interpretations of the data. The interface should simply pass all data to the printer unchanged.

Make sure you have loaded a printer file before trying to print. Without a printer file PaperClip will not be able to control the printing process. The printer file matches PaperClip to your printer. Refer to the appendix Choosing A Printer File.

# RS-232 Printer Port — Commodore64

## Setting Paperclip for RS-232 Output

To set PaperClip to transmit to an RS-232 printer connected to the user port, use the following command. (Interfacing to the user port is explained below.)

| | |
|---|---|
| Press: **CTRL** | |
| Press: **(shift)"** | [quotation marks] |
| Press: **N** | |
| Press: **RETURN** | |

The question **RS-232 baud rate?** will appear. The baud rate is the speed at which data will be transmitted to the printer. Allowable baud rates are: 5Ø, 75, 11Ø, 135 (134.5), 15Ø, 3ØØ, 45Ø, 6ØØ, 12ØØ, 18ØØ, 24ØØ, 36ØØ, 48ØØ, 72ØØ, or 96ØØ. Enter the baud rate that you wish to use with your printer and press **RETURN**.

The next question is: **Word length (5 to 8 bits)?** This is the size in bits of each character that will be transmitted to the printer. It will normally be either 7 or 8 bits. Answer with the word length that you wish to use.

The next question is: **Form of parity (N/E/O/M/S)?** Parity is an error checking system. The forms of parity are None, Even, Odd, Mark, and Space. Enter the letter corresponding to the form of parity you wish to use.

The next question is: **Do you want x-line handshaking?** Answer yes or no depending upon whether you want handshaking. If you want to use a high baud rate (print speed), handshaking will be necessary so that the printer can control the transmission of data from Paperclip. (This is explained below.)

### Interfacing RS-232 Printers with Paperclip

RS-232 printers are interfaced to the Commodore 64 via the "user port" located in the left rear of the computer. When Paperclip is doing RS-232 output it uses its own software, not the internal RS-232 software in the C64. The voltage levels on this port are different from those used by an RS-232 device so an RS-232 interface is needed.

When PaperClip sends data to an RS-232 printer, it uses 5 active lines used as follows:

Data Terminal Ready (DTR)
— used by PaperClip to send a signal to the printer indicating that PaperClip is ready for communication.

Data Set Ready (DSR)
— used by the printer to send a signal to PaperClip indicating that the printer is ready for communication.

Request To Send (RTS)
— used by PaperClip to send a signal to the printer indicating that PaperClip is ready to send data.

Clear To Send (CTS)
— used by the printer to send a signal to PaperClip indicating that the printer is ready to recieve data.

Transmitted Data (XMT)
— used by PaperClip to send the actual data to the printer.

Here is a table of reference data for each of these lines, showing which pin is used by PaperClip (via the user port), and which is used by the RS-232 interface (via the connector).

| | **user port** | **direction** | **RS-232 connector** |
|---|---|---|---|
| DTR (Data Terminal Read) | E | → | 20 |
| DSR (Data Set Ready) | L | ← | 6 |
| RTS (Request To Send) | D | → | 4 |
| CTS (Clear To Send) | K | ← | 5 |
| XMT (Transmitted Data) | M | → | 2 |

For example, the data on the XMT line, leaves the user port via pin M, enters the RS-232 interface, and then leaves the interface via pin 2 on the RS-232 connector. From there, it is sent to the printer.

The simplest way to print data is to use only the XMT line to send the data. This works fine at low baud rates. However, if you want to send data at a fast rate, you have to ensure that you don't send it faster than the printer can handle. To do this, the printer will use the CTS line to indicate when it can receive more data, and PaperClip will use the RTS line to indicate when it wants to send more data. This system — using two extra lines to control the flow of data — is called *Multi-Line Handshaking* or *X-Line Handshaking*.

Here is a summary of how PaperClip prints a document:

```
Set DTR and RTS on.
REPEAT (until the whole document is printed)
    Check DSR line—if FALSE, abort output
    Check CTS line—if FALSE, wait until TRUE
    Send data to printer
```

While Paperclip is waiting for the CTS line to go TRUE, it monitors the DSR line and the CTRL key.

Some interfaces (such as Commodore's Vicmodem) invert the meaning of the handshaking lines. If you request X-Line Handshaking, you will be asked a second question, "Is this a Vicmodem-type interface?" If your interface inverts the handshaking lines, answer yes. (Versions of Paperclip previous to 64D assumed a Vicmodem-type interface.)

Be aware that if your printer is set up as a Data Terminal and not a Data Set, you will have to cross-connect wires on the RS-232 connector to ensure proper communcation with PaperClip.

The RS-232 interface is more complex than a parallel interface, and is one of the most difficult ways for a novice to interface a printer. The best way to do it, is to first get the printer working at a low baud rate (5Ø) without any handshaking. Use your printer manual to find out the characteristics of your printer. If you don't know the word length it will probably be 7 or 8. If many characters are misprinted, check the parity setting.

Once you get your printer printing properly without handshaking, change Paperclip so that it uses handshaking, if it can with your printer. Do not attempt to raise the baud rate until you have it working with handshaking. A typical result of no handshaking at a fast baud rate is only the first line of the document will print properly.

Whatever system you set up, you can then try raising the baud rate to find the fastest that will work with your system. However, 96ØØ baud may not work as it approaches the limits of software-driven handshaking. (The screen display may flicker. This is normal for 96ØØ baud).

# Parallel Printer Output — Commodore64 & 128

## Setting PaperClip for Parallel Printer Output

Before PaperClip can send text to a Centronics Parallel type printer connected to the RS-232/user port you must enter the following command:

| | |
|---|---|
| Press: **CTRL** | |
| Press: **(shift)"** | [quotation marks] |
| Press: **Y** | |
| Press: **RETURN** | |

PaperClip will now send all printer output to the parallel printer connected to the RS-232/user port.

To return output to the normal printer channel, you must set the Printer Device Number.

Before using PaperClip with the parallel printer port, test the port and cable according to the cable manufacturers instructions. When actually using PaperClip, *do NOT use any software supplied by the cable manufacturer.* PaperClip will control the port directly.

The pin connections are as follows:

| **Signal** | **6526** | **User Port** |
|---|---|---|
| Data lines | PBØ to PB7 | pins C, D, E, F, H, J, K, L |
| Strobe (active low) | PA2 | pin M |
| Printer busy (or ACKNLG) | FLAG | pin B |

It is important that the printer be turned on after PaperClip is running in order to make sure that PaperClip starts printing correctly. If turned on before, there is the possibility of PaperClip not being able to send data to the printer until it is turned off and back on.

# Using PaperClip With 8Ø Columns

PaperClip is compatable with some 8Ø column displays available for the Commodore 64 computer. Other than initial startup, there are no limitations imposed by the use of an 8Ø column display.

The following units have been tested with PaperClip. Inclusion in this list is *NOT* to be interpreted as an endorsement of any of these products. Be sure to see them in operation *WITH* PaperClip before deciding on a purchase.

*Batteries Included — B.I.-8Ø:8Ø Column Display Adapter*

*Data2Ø — Video 8Ø pak*

Follow the normal PaperClip start up sequence except for the actual LOAD command. Use the commands below to load the PaperClip program for use with 8Ø column displays.

Type: **load"paperclip64?-8Ø",8** (for Spellpack versions, **load" pclip 64?-8Ø-s?",8**
Press: **RETURN**

Wait until the program is loaded, then:

Type: **run**
Press: **RETURN**

# Multilingual Specifications

Within PaperClip several keys have been defined as multilingual. When one of these keys is pressed, a corresponding symbol will appear on the screen. The symbol that appears on the screen when a key is pressed is defined by the "character set". PaperClip has the ability to change character sets.

With PaperClip for the PET and CBM computers the character set is contained in a chip called a "character generator". This chip may be replaced with a multilingual character generator available from any PaperClip dealer. The new chip, once installed, does not affect normal operation of the computer in any way. Until a command is issued with PaperClip, the new symbols will not appear. See the section Changing Character Sets, for details on how to invoke the extra symbols.

PaperClip64 & 128 can retrieve a character set from disk and use it. See the section Loading An Alternate Character Set for details.

It is possible to build your own character set for special PaperClip applications. You need a type of program called a "character set editor", which is available from several sources. See your Commodore dealer for information on availability. Using such a program, you may define the shape of any symbol that can be displayed on the screen. Every character has a number, or value, that when "poked" on the screen causes that symbol to be displayed. Each multilingual key is preceeded by pressing **ESCAPE**, then the key. The table that follows lists each multilingual key and the corresponding "poke" value.

| Key | Poke Value | Key | Poke Value |
|---|---|---|---|
| Ø | 1Ø5 | + | 117 |
| 1 | 92 | - | 28 |
| 2 | 95 | = | 127 |
| 3 | 1Ø2 | @ | 115 |
| 4 | 96 | u | 93 |
| 5 | 97 | : | 12Ø |
| 6 | 98 | ; | 119 |
| 7 | 1Ø4 | , | 118 |
| 8 | 94 | / | 1Ø2 |
| 9 | 113 | * | 1Ø7 |

Before you can print any multilingual character, you will have to define a printer file that tells PaperClip how to print the special characters on your printer. See the appendix Creating a Printer File, for details on how to do this.

# Common Disk Commands

**Note:** All disk commands must be issued from disk command mode. Disk command mode is set by pressing **CTRL** followed by >

(see your disk drive manual for more detail)

FORMAT (NEW):

**ndr:name,id**

**dr** is the drive number (Ø for single drives)

**name** can be any name up to 16 characters

**id** can be any two characters

example: **nØ:documentd,d1**

Causes the disk drive to map out the diskette surface so that it can be used to store information. *This command will erase any previous information stored on the diskette!*

SCRATCH:

**sdr:name**

**dr** is the drive number (Ø for single drives)

**name** is the name of the file being erased

example: **sØ:documents**

Erases the file called **name** from the drive **dr**. *Once a file has been erased, there is no way to recover the information.*

DUPLICATE

**ddx=dy**

**dx** is the destination drive number

**dy** is the source drive number

example: **d1=Ø**

Can only be usd with the dual slot drives. Causes the diskette in the source drive to be duplicated onto the diskette in the destination drive. *Caution: this command will erase any data previously on the destination diskette.* If you are using a model 4Ø4Ø disk drive, take care because the duplicated diskette will have the same ID code as the original diskette.

COPY:

**cdx=dy**

**dx** is the destination drive

**dy** is the source drive

example: **c1=Ø**

Can only be used with the dual slot drives. Causes the files on the diskette in the source drive to be copied onto the diskette in the destination drive. Any files already existing on the destination diskette are not altered or erased. The destination diskette must be formatted before issuing the COPY command.

VALIDATE:

**vdr**

**dr** is the drive number containing the diskette

example: **vØ**

Causes the disk drive to trace through every file listed in the diskette directory, checking for correct end markers, etc. Any files on the diskette which are not correctly stored will be removed. The BLOCKS FREE count and the diskette BAM (see your drive manual for details) are updated. If an error is encountered, the diskette is left unchanged. This is the only correct way to remove directory entries marked with an asterisk (unclosed files).

# Common Disk Error Messages

Your disk drive manual will have a list of disk error messages and their causes. A few of the more common error messages and their causes are listed here.

SYNTAX ERROR
The disk drive did not understand your command. Make sure you do not enter any extra spaces or characters in a disk command.

READ ERROR
The disk drive found an error on the diskette surface. Try issuing the command again. If it still does not work use another diskette. This error will also show up if you omit the ID code from the command to format a diskette. Make sure the diskette you are trying to use has been formatted.

WRITE PROTECT ON
A command to save, alter or erase data was issued but the write protect notch on the diskette is covered. The write protect notch must be uncovered before the command can succeed. Note: if you are using a 4Ø4Ø disk drive, remove all diskettes and switch the drive off and back on before proceeding.

WRITE FILE OPEN
The data file being referenced was not properly saved. There might have been a READ ERROR, or the diskette was removed from the drive before the save was complete, etc. In any case, the data in this file is not recoverable. Do not erase (SCRATCH) this file, use the VALIDATE command to remove it.

FILE EXISTS
You have attempted to create a file with the same name as one which already exists.

DISK FULL
There is no more room on the diskette. If this message is generated during a SAVE of text, the document in memory has *NOT* been saved. You must VALIDATE the diskette before using it again. To SAVE the document in memory, use a diskette with more free blocks.

DRIVE NOT READY
This message will be generated for many reasons. A few are: the diskette in the drive is not formatted, there is no diskette in the drive, or the diskette is not inserted in the drive correctly.

# The PaperClip Diskette

The PaperClip and PaperClip64/128 diskettes have similar contents. The standard PaperClip will have some entries are missing. In general, the following applies to all PaperClip diskettes.

The PaperClip diskette comes with a large number of entries in the directory. They are divided up as follows:

**paperclip64?** - This is the main PaperClip program. PET and CBM versions have one program which works with both 4Ø and 8Ø column displays.

**paperclip64?-8Ø** - This version of PaperClip64 is designed for use with 8Ø column display adapters.

**paperclip128?** - Version for Commodore 128 computers

**paperclip64?-SC** - PaperClip64 with Spellpack

**paperclip128?-SC** - For Commodore 128 computers - Includes Spellpack option

**pclip64?-80-5S** - This version of PaperClip with Spellpack is designed for use with 8Ø column display - adapters.

**sample document** - This is a short piece of text used in the tutorial section.

**form letter** - Sample of a form letter. This is the example from the section Form Letters And Variable Blocks.

**data file** - Information to be inserted in the form letter.

**global part one** - First part of a two part document illustrating the use of global files.

**global part two** - Second part of the global document.

**printer test** - This file is used to exercise all possible printer functions.

**printer setup*** - Using this program allows you to create a printer file for your printer. This is necessary if your printer cannot use any of the supplied printer files.

**merge prtfil*** - This program can be used to combine a printer file with the PaperClip program. This allows the use of PaperClip without having to load a printer file each session before printing.

**defaults*** - This program allows you to permanently change some of the default parameters in PaperClip64, such as: default printer port and device number, RS-232 paramaters, and default screen colors.

**backup64** - Backup copies of any diskette can be made with a single drive using this program.

**backup64k** - This is an expanded version of **backup64**. While it is not usable with all Commodore 64 systems, it works faster.

**backup 2Ø31** - Only on the standard PaperClip diskette. Allows a PET or CBM computer to make backup copies of any diskette using the 2Ø31 disk drive.

**character sets** - The files that follow this are special character sets which provide multilingual symbols for the screen.

**standard 64** - This is the standard Commodore 64 character set.

**french 64*** - This is a French character set for the Commodore 64.

**printer files** - The entries which follow are printer files.

**1525-p-alf** - This printer file is for the Commodore 1525 printer. Each printer file has been designed for a different printer. See the appendix Choosing A Printer File.

# Diskette And Cassette Care

## Diskette Care

Floppy diskettes are enclosed in a protective jacket and supplied with a protective envelope. Handled properly, they are very reliable. If you mistreat them, they will reward you with loss of data, confusing errors, and general frustration.

*NEVER* turn the disk drive power on or off with a diskette inside. *NEVER* remove a diskette while then *RED* drive lamp on the front panel is lit. Wait until the drive motor stops before removing a diskette.

Follow the guide below and you should have little trouble.

1 - Don't touch the shiny parts with *ANYTHING*. Keep diskettes in their envelopes when not in use. Data is stored on the *BOTTOM* side of the diskette, so *NEVER* put it down without protection.

2 - Don't force a diskette into the drive — it should slide in easily. Don't bend, scratch, fold or otherwise exercise the floppy diskette's floppiness.

3 - Keep the diskette away from magnetism. The diskette's memory comes from a careful arrangement of tiny magnetic particles on the smooth surface. Any exposure to magnetic fields can scramble the data. Keep diskettes away from television sets, air conditioners, computer power packs, loudspeakers, etc.

4 - Be gentle with diskettes. Don't use them as coasters, bookmarks, notepads, cushions, etc. Write on the labels *BEFORE* you stick them on. Attach them so they don't cover any of the openings in the diskette jacket.

5 - Watch the temperature. Intense heat or cold will cause your diskettes grief. If you're comfortable, so are your diskettes.

6 - Never use the reverse side of a diskette. The read/write head is on the bottom surface of the diskette with a pressure pad riding on the upper surface. This pressure pad is abrasive compared to the smooth glass surface of the head. When a diskette is flipped over, the pressure pad rides on and scratches the data storage area. Also, the inside surfaces of the diskette envelope continually clean the diskette as it revolves. While rotating in the reverse direction dirt held in the envelope is pulled back onto the surface of the diskette. These two factors guarantee that eventually part or all of the data stored on the diskette will be lost.

7 - *COPY THEM!* Always make backup copies of any diskette with important data. Diskettes are cheap — the time it would take to re-create the lost data is worth far more. If the most you ever want to lose is 4 hours worth of work, then make a backup copy every 4 hours. If you don't, one day you will pay for all the time you thought you saved.

8 - When you do make copies, use at least three diskettes. Have one as your master, which you use every day. Make your first and second copies using the other two diskettes. When you want to make the next backup, *copy from the master to the oldest backup*. In other words, always copy from the master to the diskette holding the most out-of-date information. This three disk system is called the grandfather system. From time to time, when the master diskette starts to wear out, copy it and throw the old one away. It is better to throw out an old diskette than to have an old diskette throw out your new data.

## Cassette Care

Cassette care is similar to diskette care. Keep cassettes away from magnetism, dirt, heat, and abuse. Keep them stored in their cases *FULLY REWOUND*. Always press STOP between functions. Clean the cassette machine periodically. If you have problems, clean it and try again. Refer to the owners manual for details. Use alcohol and a cotton swab. *NEVER* use solvents or abrasive tools. Use a demagnetizer occasionally to remove residual magnestism.

When saving text on tape, *SAVE IT AT LEAST TWICE*. Tapes can lose data for no apparent reason, and you should be prepared for this. If it is a long document, save it as you write, don't wait until you are finished. Save only one item on each side of a tape. This wastes tape, but it reduces the chance of losing a lot of text at once. Make sure there is enough tape to store the text. If it runs off the end, you will have to start over.

# The Spelling Checker

## Overview Of The Spelling Checker

If you are using Paperclip with a built-in spelling-checker, you can check a complete document with a single command.

Paperclip with a spelling checker comes with two disks. The first, the **program-disk**, is the same as the standard Paperclip disk described in the manual. However, it contains one extra program, described below.

The second disk is the **dictionary-disk**. The dictionary on this disk is divided into a **core-dictionary** and a **user-dictionary**. The core-dictionary contains approximately 15,ØØØ words. The user-dictionary comes empty. You can add words to it to build your own custom dictionary. Each dictionary is stored as a set of "modules", each of which holds all the words of a particular length, from two letters to twenty-two letters.

The program works by checking each word of your document against one or both dictionaries. It keeps a list of every word that it can't find. It then displays each such word, which you can leave as it is or change. If the word is a new word, you can save it in the user-dictionary.

## How To Use The Spelling-Checker

The dictionary-disk is accessed frequently. This increases the wear on the diskette. Before you use the spelling-checker, you *MUST* make a backup copy of the dictionary-disk. Use the backup program that comes with PaperClip. Then put the original disk in a safe place. You should do this again if you add many words to the user-dictionary.

To check the spelling of a document, start PaperClip and load the document you want checked. To start the spelling-checker, press: **CTRL (shift) Y**. Except where noted, pressing the **CTRL** key at any time, will return you to edit mode.

The spelling-checker will tell you to put the dictionary-disk in the disk drive. (Use your backup-disk.) If you are using a single disk drive, you will have to remove your document-disk. If you are using a dual disk drive, you can place the dictionary-disk in either drive. After you have placed it in the drive, press **RETURN**.

The spelling-checker will check the disk drive to see if the dictionary-disk is present. If it is not, the program will wait until you press the **RETURN** key again, at which time it will repeat the process.

Once the spelling-checker has found the dictionary-disk, it will ask if you want to check the user-dictionary. The spelling-checker first checks all words against the core-dictionary. If a word is not found, and you have told it to check the user-dicionary, it then looks for it in the user-dictionary.

After you have made your choice, the word **Checking:** will appear on the top line of the screen, followed by each word as it is checked. If a word is found, a checkmark will be displayed beside it.

When the checking operation has finished, the message **Not found** will appear on the top line, followed by a word which was not found. This word will also be highlighted in the document.

You now have four options:

Press: **f1** — to skip to the next word
Press: **f3** — to skip to the next word, and ignore any further occurrences of the current word
Press: **f5** — to correct the spelling of the word
Press: **f7** — to add the word to the user-dictionary

Notes:

— If you choose to replace the current word, the new word will *NOT* be checked for spelling. Be careful not to replace one spelling mistake with another.

— If you want to be able to add new words, then you *MUST* answer "yes", when you are asked: Check user dictionary?

— After you have added a new word to the dictionary, you can select one of the other three options.

— Be sure to make a backup copy of the dictionary-disk if you add many new words.

— Occasionally, the message **Updating dictionary** will be displayed on the top line. This means that the program is saving information pertaining to words that have been added to the user-dictionary.

## Example of Using the Spelling Checker

To illustrate the use of the spelling-checker, let's check the spelling in one of the documents on the Paperclip disk. Place a copy of the Paperclip program-disk in the drive and load PaperClip and run it. Next, load the file called "sample document" using **CTRL L**.

Start the spelling-checker by typing **CTRL (shift) Y**. When the program asks you to do so, place a *copy* of the dictionary-disk in the drive, and press **RETURN**. At this point, the disk drive light will come on as the program verifies that the dictionary-disk is present. When the drive light goes off, the display should ask **Check user dictionary?** If it does not, check to make sure that the disk in the drive is, in fact, the dictionary-disk, and try again.

As mentioned earlier, your answer to this question controls whether or not the user dictionary will be searched. Since the user dictionary is empty when the dictionary-disk is shipped, there is no point in having the spelling-checker check it for this example. Answer **N** to the **Check user dictionary?** question.

Once you have done this, the message **Checking** will appear on the top line and the words from the document will be displayed along-side as they are checked. During this time, the dictionary-disk will be accessed almost constantly.

After about a minute, the disk activity will stop and the message **Not found: movedd** will appear on the top line of the screen. The word **movedd** will be highlighted in text as well. In this case, the spelling-checker has found a legitimate spelling mistake and not an unusual word which is not in the dictionary.

To correct this mistake, we want to replace this word with the correct one. To do this, press the **f5** key. The spelling-checker will then ask **Replace with?** It wants to know with what word you wish to replace **movedd**.

Press the **CTRL** key. You will see that the program will return to the **Not found** message, rather than edit mode. This is the only time when pressing the **CTRL** key does *not* put you back into edit mode. This allows you to change your mind about replacing a word, without having to start all over again.

Now, press the **f5** key to return to the replacement. The spelling-checker should now be asking **Replace with?** Enter the word **moved** and press **RETURN**. The word **movedd** in the text will be replaced with **moved**, and the spelling-checker will display the next misspelled word.

The top line now displays **Not found: neverbe**. This is the second mistake Press **f5** and enter the correct words: **never be**. The program will now move on to the third spelling error, **spellung**. Correct this word in the same way.

At this point, there are no more mistakes, so PaperClip will now return to edit mode. If you were checking one of your own documents, you would now save the corrected version.

Notice that the spelling-checker was able to detect three different types of spelling errors: **Movedd** is a typing error, **neverbe** is two words running together, and **spellung** is incorrect spelling. However, the program cannot examine context. If the error is, in fact, a legitimate word — say, **beer** instead of **bare** — it will not be detected.

## Time Needed to Check a Document

The time needed to check a document depends heavily on the type of disk drive. An IEEE drive (the Commodore 2Ø31, 4Ø4Ø, 8Ø5Ø, and so on) will run about three times faster than a serial bus drive (the Commodore 1541).

Most of the time is spent reading the dictionary. This time will be about the same no matter how large the document. Thus, a very large document takes much less time *per word* to check.

If you want to test this, load **sample document**, and set a range over the entire document. Now, copy it 19 times to create a file that is 2Ø times as large. Invoke the spelling-checker by pressing **CTRL (shift) Y**, and specifying not to check the user-dictionary. You will see that it does not take 2Ø times longer to check all of this new, larger document.

## The Maintenance Program

On the program-disk, you will find a program called **maintenance**, used to update the dictionaries.

To use **maintenance** load it from the disk and run it like any BASIC program. After a short wait, the screen will clear and a menu will be displayed. These are the options:

**View Dictionary** **0024**— to display a sorted listing of the user-dictionary on either the screen or the printer

**Add Word** **0024**— to add words to the user-dictionary

**Delete Word** **0024**— to delete words from the user-dictionary

**Repair Dictionary** — If a dictionary-disk becomes messed up due to disk errors or garbled data, this option can be used to partially correct the problem. It can replace any core- or user-dictionary module with an empty module. (A "module" is a part of a dictionary.) This will allow the spelling-checker to run properly, but the replaced module will have to be rebuilt. This option can also be used to rebuild the pointer file for a user-dictionary module. Of course, it would be preferable to use a backup copy of the dictonary-disk.

**Update Core Dictionary** to transfer the contents of a user-dictionary module into the core-dictionary. This allows you to check new words, without using the user-dictionary.

**Exit Program** — to stop the program. It is important to use this option, rather than pushing the stop key. A dictionary module may have been updated but not yet saved on the disk. By stopping the program properly, you insure that nothing is lost.The maintenance program is written in BASIC with a little machine langauge, and it is not fast. If you want to add a lot of words to the user-dictionary, it is faster to do it from within PaperClip. Simply type your words as text and then check them (using the user-dictionary). After each new word is not found, you can add it to the user-dictionary by pressing the **f7** key.

# Appendix U — Notes For Commodore 128 Owners

PaperClip128 has been designed to take full advantage of the advanced features of the Commodore C-128 computer system. The following list some changes, enhancements and tips you should be aware of.

* The 'key' provided with this program should ALWAYS be inserted into Control Port 2 (the one closest to the back of the computer) when using PaperClip128. Insert the key into the computer BEFORE turning it on.

* To load PaperClip128 type **DLOAD "paperclip128?"** and press **RETURN**
When the screen displays 'READY' type **RUN** and press **RETURN**
(Note: If the diskette contains the PaperClip128 'boot sector' then PaperClip128 will automatically load and run when the computer is turned on or reset.)

* PaperClip128 can use either the Commodore 17Ø2 (composite video) type display or the Commodore 19Ø2 (RGBI video) type display. (Any standard monitor can be substituted for the 19Ø2 type RGBI display by using the **'BI C128 Monochrome adapter')** To switch displays, simply press or release the **4Ø/8Ø DISPLAY** button on the keyboard. PaperClip128 will instantly switch displays, blanking the unused one. If you have only one display, set the **4Ø/8Ø DISPLAY** switch before starting up PaperClip128. (There is one limitation reguarding display switching: The editing line length MUST be set to 8Ø columns or more before PaperClip128 will be able to switch displays. This is automatic if PaperClip128 starts up with the 4Ø/8Ø DISPLAY switch depressed. Otherwise, simply set the editing line length to 8Ø columns (or more) using the command **CTRL (shift) L** (see pages 75 & 76.)

* Video output has been made much more flexible. To change the number of columns displayed use **(shift) F8.** PaperClip128 can output 4Ø , 8Ø or 16Ø columns on a composite monitor. The RGBI monitor can show 8Ø , 16Ø or 32Ø columns.

* PaperClip128 also makes use of some of the extra keys on the C-128 keyboard.
ESC - same as LEFT ARROW for entering special characters and functions
TAB - same as RUN/STOP for tabbing
CAPS - same as UP ARROW for entering uppercase text
Individual cursor keys - use either these or keys in main keyboard area
ENTER - same as RETURN

* PaperClip128 can be automatically started up when the computer is turned on. This process is called 'booting'. To 'boot' a program, a special file must be stored on the diskette. This 'boot sector' is placed on the diskette by PaperClip128 with the command **CTRL (shift)'**. There is only one spot on the diskette where a 'boot sector' may be placed. If that sector is already in use, PaperClip128 will inform you that it can't write the boot sector.

# Index

# NEED HELP?

Having problems? Have you got questions about Batteries Included software? Call the **BI Customer Support Hotline**. We'll be glad to help you in any way we can. Because this is a **free service** we must ask that you **keep your questions brief and to the point**. Since a limit of 5 minutes of free service is permitted per call, prepare your query carefully and have pertinent materials (e.g. your manual) on hand when you phone; there will be an additional charge for calls over five minutes. Our hours are 9am to 4pm (est), Monday to Friday. Best time to call — mornings. Worst time to call — midday. If our lines are busy, please be patient. Do not call our switchboard; they cannot patch you through or take messages for customer support. If you would rather not phone, or your question or problem is not urgent, please do write. We'll be glad to send you a personal reply.

**Please note: customer support is a free service for registered owners only.** Please make sure you have sent in your registration card.

BATTERIES INCLUDED

**Customer Support (416) 881•9816**

**30 Mural Street Richmond Hill Ontario Canada L4B 1B5**

Printed in Canada