SwyftCard™

Using SwyftCard on the Apple //e: a guide, glossary, and reference manual

SwyftCard by *Jef Raskin, James Winter, Terry Holmes,*
MinoruTaoyama, Jonathan Sand,
John Bumgarner, David Lavond,
Paul Baker, Jim Straus, and Renwick Curry

Manual by *David Alzofon and Jef Raskin*

Graphics by *Suzanne West*

First Edition

Information Appliance Inc.

Dedication

This manual is dedicated to all SwyftCard users. Unfortunately, it was written by all-too-human beings. We have tried to make it complete so that everything you need is included. This made it larger than we wanted. To save you time and to make the manual small and unintimidating, we tried to make it concise. This made it less explicit than we wanted. We have tried to make it enjoyable, but our sense of humor may not match yours. We have tried very hard to make it accurate, but are sure that an error or two has slipped in.

We know SwyftCard works: this manual was written and edited in the SwyftCard system running on Apple //e's. We printed the drafts with the PRINT commmand, and calculated various values with the CALC command. We used the DISK command and never lost any text. The SEND command was used to transmit the text to a typesetting system, which then made up the version you hold in your hand.

If you encounter a problem, or you have a suggestion that would improve the manual or the product, drop us a line. If you include a self-addressed stamped envelope we'll respond directly back to you.

— David, Jef and Suzanne

Credits

Movies have the fine tradition of giving credit (or blame, in the case of some) to nearly everybody who worked on the project. We can do no less:

David Alzofon wrote, tested, and edited the manuals and the on-line tutorial. Paul Baker honcho'd the parts and circuit boards through production. Steve Bowling, the president of the company, did what a president does (a lot). His assistant Patti Okoniewski manages to be helping everywhere at once.

Terry Holmes did the initial internal software design and the actual programming of the early versions of SwyftCard, and insisted on making everything run as fast as possible. Richard Krause designed the labels and did manufacturing coordination; Jonathan Sand transformed the prototype software into the final product, adding features and fine tuning. Dr. David Lavond programmed during the final phase and completed the documentation.

Jim Straus (who banished the company picnic and substituted an annual gourmet lobster fest – not at company expense) and Mino Taoyama contributed much to design and debugging; Mino's main contributions were the hardware design of the SwyftCard itself, and the design and implementation of the internal disk routines and ProDOS converter.*

Ralph Voorhees, while mainly working on other projects, kept reminding us that our customers will not put up with the usual computer nonsense; Sharmon Scipio* kept the paychecks coming and the books in order; John Bumgarner* did some coding of the early versions and was a constant source of inspiration, as was the ever-effervescent Scott Kim.*

Collaborators who are not currently employees include Dr. Ren Curry, Brian Howard, and Michael Raskin, who helped early on with the design and human factors testing. Ren's contribution to the cursor design was especially noteworthy. Deb Clifford worked on the early versions of the on-line tutorial; Charlie Springer contributed his technical know-how, and Cerafin Castillo was our electronics tech; Ron Genise* did testing and teacher training. For a time Mike Scott, Sonya Hoag, Claire Llewelyn, Bana Witt, Camillo Santomero, and Tina Bracewell ably assisted us. Bruce Cooper extensively tested an early prototype in a publication environment and provided helpful suggestions about formatting.*

Mike Ivanitsky and Kevin Thurber tested printers, and Suzanne West of West and Moravec did the graphics.

A company is fueled by the money of its investors: we thank them all. The company would not have been possible without the legal advice of James E. Lewis and the work of corporate secretary Linda Blum*. We also thank Gayla Newsome of Orange Micro in Santa Clara for allowing us to test the printer set-ups.*

A special note of gratitude is extended to the many people who tested our systems, and to the San Jose and Fremont School Districts for letting us test SwyftCard in an educational environment.

The company grew from discussions between Dr. James Winter who invented the DISK command and contributed greatly to the design and implementation of the software, and Jef Raskin*, who created the concepts that underly SwyftCard and its implementation, founded Information Appliance, and who is now Chairman and C.E.O of the company.*

**An asterisk marks a founder of Information Appliance Inc.*

The SwyftCard Manual

Table of Contents

Appendices

Chapter I

Getting Started

Introduction

SwyftCard™ Meets Apple //e™

When you plug the SwyftCard into Slot 3 in the interior of your Apple //e and turn on the power, your computer will be ready to help you

- produce written material
- store and find information (information retrieval *)
- do arithmetic
- write short programs, and, if you have a modem,
- send and receive information over phone lines

You will *not* have to remove your SwyftCard in order to run most Apple software.

SwyftCard Is Not Just a Word Processor

If it seems that it is, that's because nearly everything you do with a computer consists of typing to it and having it type back at you. With the exception of graphic or audible output, everything done with a computer is some form of word processing. Much of this manual explains methods of typing and modifying text. This is a useful tool in its own right. But, as you can see from the list above, you can — without leaving the editor — do calculations and (if you are so inclined) write programs; you can communicate with other SwyftCards, computer bulletin boards and commercial data services; and you can move information to and from and among disks.

SwyftCard is perfect for instantly retrieving correspondence, names, addresses, and telephone numbers. You can, for example, find a name anywhere in your text in the blink of an eye. The same goes for a phrase, a word, or a part of a word. And it doesn't matter to SwyftCard whether your text is neatly organized or a disorganized mess.

** If these or any other terms appearing in the manual are new to you, please consult the glossary.*

How to Learn SwyftCard

The next four pages of the manual will get you started and ready to do the on-screen tutorial. After you've done the tutorial (most people take about 90 minutes) you will know how to type and edit text, how to use SwyftCard's unique cursor, how to do information retrieval, and how to use each of SwyftCard's five commands. The rest of the manual is for your reference: it covers the material presented in the tutorial, and provides additional details, such as how to initialize your printer and alter the format of your printout. Keep the Reference Card (back of the manual) by the computer for quick consultation.

Part 1: Equipment Checklist

What You Need

Besides the SwyftCard, you need

- an Apple //e (regular or enhanced)
- an 80-column card (either standard or extended)
- a video monitor suitable for 80-column display
- a disk drive with controller (you really only need one disk drive when using SwyftCard!)
- blank disks on which to save your work
- adhesive-backed labels for your Apple //e keyboard (these came in the SwyftCard package)

To Print Out the Things You Write

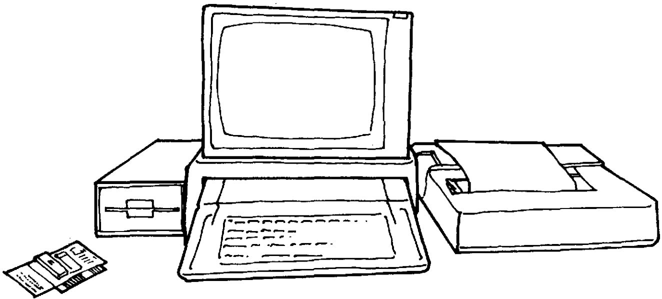
To put your words on paper you will need a printer, printer interface, printer cable, and, of course, some paper.

For Telecommunications

To send and receive information over telephone lines you will need an Apple Super Serial Card, a modem, and modem cable.

Slot Assignments for Apple Cards

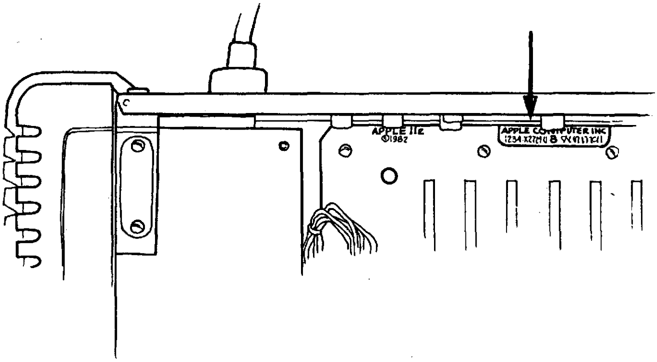
If you're going to use a printer, a printer card should be in Slot 1; if you're going to use a modem, an Apple Super Serial Card should be in Slot 2; SwyftCard goes in Slot 3; the disk controller goes in Slot 6. Additional cards in other slots will usually not affect the operation of SwyftCard.



What you need. The printer, the modem, and their interfaces are optional.

Logic Board

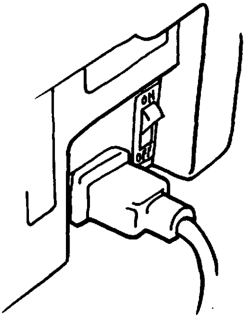
If SwyftCard doesn't work, check to see if your Apple has a REV-A logic board. If it does, have it changed to a REV-B logic board, which can be done for free at your dealer.



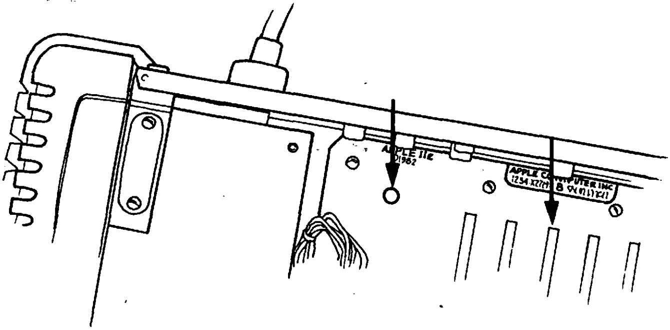
Where to look to see if you have a REV-A logic board

Part 2: Install the SwyftCard

Switch off the power switch to your //e and remove the cover. Check the little red light in the back left corner — this light should be off if the power is off. Do not unplug the computer: the ground connection helps drain static electricity which could damage the computer, and there are no dangerous voltages on the inside of the computer when it is off.

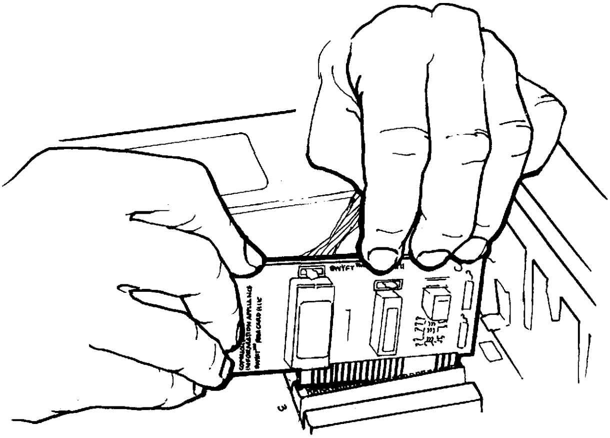


Turn the power switch OFF, leave the system plugged in.



The power light inside the computer should be off. Notice the location of Slot 3.

SwyftCard can only fit one way, with the long portion toward the keyboard. Seat the fingers of the SwyftCard in Slot 3 (the fingers are the metal strips that look like the teeth of a gold comb). Then gently but firmly push the SwyftCard into the slot until it touches bottom. With the SwyftCard snugly installed, replace the cover and push down in back to latch the cover.

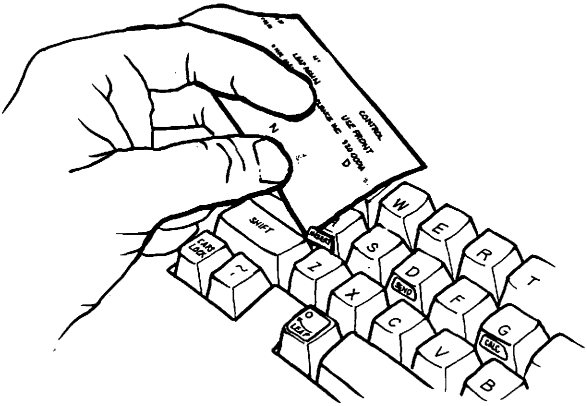


Installing SwyftCard in Slot 3

Install other cards according to the instructions in their manuals.

Part 3: Label the Keys

Remove each label from its backing and stick it to the indicated key. The two LEAP key labels, and the PAGE, LEAP AGAIN, and USE FRONT labels go on top of their keys. Our favorite technique for applying them is shown in the illustration. First we move the label so that it sticks to the lower right corner of the label carrier sheet and hangs off to the side. This helps hold it in place as we smooth it onto the key.



Applying labels

Key	***Label***	***Where it goes on the key***
ESC	PAGE	Top
TAB	LEAP AGAIN	Top
CONTROL	USE FRONT	Top
A	INSERT	Front
D	SEND	Front
G	CALC	Front
L	DISK	Front
N	PRINT	Front
Open-Apple	LEAP (open circle)	Top
Solid-Apple	LEAP (solid circle)	Top

Part 4: Start Up

How to Start the Tutorial

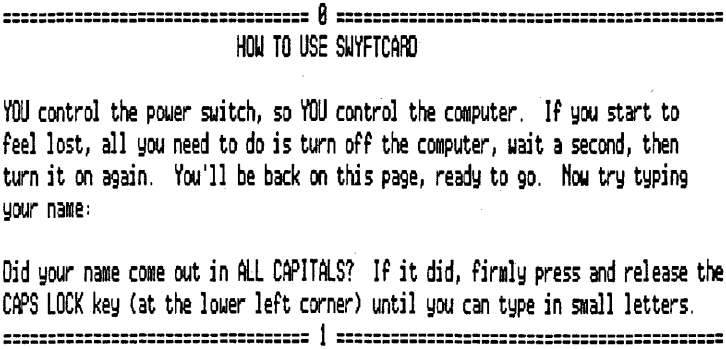
Place the tutorial disk in the disk drive (the label of the disk should be facing up and toward you). Close the disk drive door. Turn on your video monitor. Now turn on the power to your computer. The disk drive will clackety-clack and whirr. When it quiets down you will see the first screen of the tutorial.

The on-screen instructions will guide you from there. The tutorial disk is write-protected. That means that nothing you type can harm it, so don't worry about mistakes.

Fresh Starts for the Tutorial

The tutorial has no error messages programmed in. That's because we have one simple remedy for all problems: *Turn off the computer, wait a couple of seconds, and turn it on again.* There's no need to struggle with a mistake — forget it and start over.

***CAUTION:** Turning off the computer to get a fresh start only works when you have saved your typing on a disk. The tutorial is saved on the tutorial disk, so it comes back when you turn the computer on again. If it wasn't saved, it would be gone forever. So, when typing your own things, be sure to save them before you turn off the computer. How to save things is explained in the tutorial and in Part 11: The DISK Command.*



```
================================= 0 ==========================================
                         HOW TO USE SWYFTCARD

YOU control the power switch, so YOU control the computer.  If you start to
feel lost, all you need to do is turn off the computer, wait a second, then
turn it on again.  You'll be back on this page, ready to go.  Now try typing
your name:

Did your name come out in ALL CAPITALS?  If it did, firmly press and release the
CAPS LOCK key (at the lower left corner) until you can type in small letters.
================================= 1 ==========================================
```

The first screen of the tutorial.

While the tutorial is self-explanatory, you can, if you like, read the brief guide which follows.

What's in the Tutorial?

In easy steps the tutorial illustrates how to type, edit, move the cursor, highlight, and use the commands. This is roughly the same order the information is presented in the manual, making it easy to use the tutorial and the manual together.

The lessons are brief (about two paragraphs each), and usually include an exercise. Look for the word "Example." This tells you that what follows is something for you to do. In general, don't do anything unless it's an example.

If you find yourself wondering why things work the way they do, just keep moving — the answer is probably around the corner.

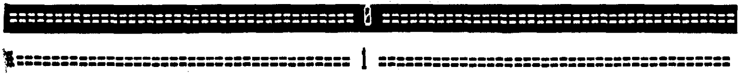
There are about 60 of these short lessons. They are numbered. If you keep track of the number of your screen, it will help you get back in case you get lost. The tutorial explains how.

Starting to Write Something of Your Own (If You've Already Done the Tutorial, or You're Just Rarin' to Get Goin')

Make sure there is no disk in the drive. Turn on your monitor and let it warm up a few seconds. Turn on your Apple. The drive will whirr and the screen will come to life. You'll see the copyright notice for a second or two while the drive runs. Then things will quiet down and two rows of equal signs will appear across the top of your screen. The first line has a zero at the center, the second line has a one. You are ready to roll. Put an unused disk in the drive if you intend to make a permanent copy of what you write (see Part 11: The DISK Command for more information).

```
SwyftCard 1066 //e  Copyright 1985  Information Appliance Inc.
```

Copyright notice. It appears for only a few moments.



Flashing object

When you see this, you are ready to use SwyftCard.

Bleeps While Doing the Tutorial

A bleep is an audio warning signal from your Apple (Part 21 covers them in detail). If you're doing the tutorial and the Apple starts to bleep every time you tap a key, it means that there is no more room to type anything. This very rarely happens, but since it is possible, here's what to do: Turn off the computer, wait a couple of seconds, then turn the computer on again. The tutorial will start over with plenty of free space for typing.

Once the tutorial is on the screen, you can get back to the lesson you were working on by holding down the LEAP FORWARD (Solid-Apple) key and typing the number of the lesson. Make a mental note of that number before you turn off the computer.

Note: *This is not the usual, recommended way of getting more room when you run out. We discuss the proper technique in Part 11, "When You Run Out of Room."*

Chapter II

Editing

Part 5: Typing

Characters and Text

When you type, you create *characters* and *text*. Characters are individual letters, punctuation marks, symbols, and so forth. Text is what they add up to: all the characters you will be able to see on the screen. When we write "Text" with a capital "T," we mean everything of yours that is in the computer at one time.

The Cursor and the Highlight

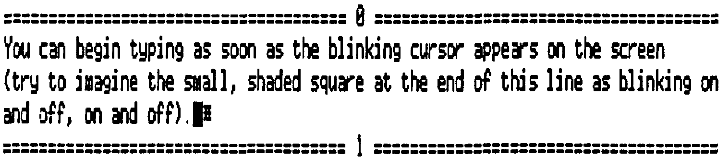
The cursor is the blinking object on the screen. It usually appears to the right of a solid, unblinking rectangle called the highlight. The cursor and highlight mark your place on the screen when you are typing.

What They Do

The cursor shows you where the next character you type will appear. The highlight shows you what will be erased if you tap the DELETE key.

When You Can Type

You can begin typing as soon as you see the blinking cursor on the screen.



```
==================================== 0 ====================================
You can begin typing as soon as the blinking cursor appears on the screen
(try to imagine the small, shaded square at the end of this line as blinking on
and off, on and off).█▒
==================================== 1 ====================================
```

Where the Type Will Appear

Your typing will appear on the screen at the position of the cursor. As you type, the cursor will move ahead, indicating at all times where your typing will next appear.

How to Type Capital Letters

Hold down a SHIFT key and, while you hold it down, tap the letter key. We say "tap" because if you do not quickly release the letter key, you will get many letters. That is why in writing this manual we have been careful to use the word "tap" when we mean a brief tap, and "hold" when you are to hold the key down.

When There Are Two Symbols on One Key

To type the lower symbol, simply tap the key. To type the upper symbol, hold down one of the SHIFT keys and tap the key of that symbol. For example, to type an asterisk (*), hold down a SHIFT key, and, while holding it, tap the "8" key.

Note: *The keys labeled ESC/PAGE, TAB/LEAP AGAIN, CONTROL/USE FRONT keys are exceptions. Holding SHIFT will not affect these keys, whose functions are discussed elsewhere.*

The Labeled Keys

The letter keys we asked you to label do not type any differently than the non-labeled keys. Holding a SHIFT key and typing a "D", for example, results in a capital D, not a "SEND". The labeled keys will be explained as we go.

Word Wrap

When a line fills up, the computer automatically begins a new one without ever breaking a word. This is called word wrap. Because of word wrap, you do not have to tap the RETURN key at the end of lines. This saves a lot of thought and effort compared to a typewriter. Word wrap also works in reverse. If you remove part of what you have already typed, the material ahead of the part you took out will move back to fill in the gap.

Spaces in Word Wrap

Typing a space at the very end of a line will trigger word wrap. The space and the word preceding the space move down to the beginning of the next line. This is done to keep the left edge of the text even. If the space alone was moved down, the left edge would look ragged.

Typing in the Middle of Something

What you type is always inserted; it never replaces what is already there. The letters to the right of the cursor, including the letter at the cursor itself, are pushed out of the way when you type. Line ends adjust automatically.

Cast of Characters

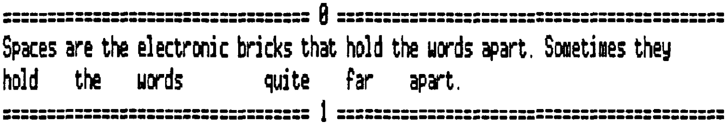
This is a complete list of the 97 characters available in SwyftCard:

Lowercase letters:	a b c d e f g h i j k l m n o p q r s t u v w x y z
Uppercase letters:	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Numerals:	0 1 2 3 4 5 6 7 8 9
Punctuation:	, . ; : ' " - ? !
Special symbols:	@ # $ % ^ & * ( ) _ = + [ ] { } \ | < > / ' ~ '
Format Characters:	space, return, page

Format Characters

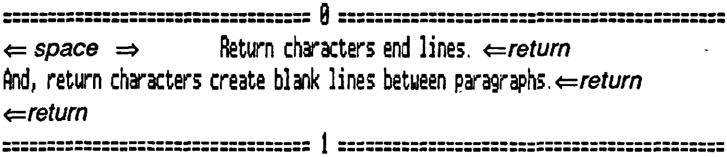
The following three characters — called *format characters* — determine the spatial arrangement of text on the screen and in the printout. As far as SwyftCard is concerned, format characters are no different than letters, punctuation, and numerals.

Space: An electronic separator the width of one letter; produced by the SPACE BAR. Separates words on the screen.



```
================================ 0 ========================================
Spaces are the electronic bricks that hold the words apart. Sometimes they
hold    the    words        quite    far    apart.
================================ 1 ========================================
```

Return: Ends the line you are typing and moves the cursor to the beginning of the next line. Invisible except when included in the highlight portion where it appears as a shaded spot.



```
================================ 0 ========================================
⇐ space ⇒         Return characters end lines. ⇐return
And, return characters create blank lines between paragraphs.⇐return
⇐return
================================ 1 ========================================
```

Page: Produced by the PAGE key (marked ESC on the Apple keyboard). Forces the ending of a page (see Part 6, "Page Characters).

```
================================== 0 ==================================
⇓ end of page 1
================================== 1 ==================================
```

Underlining

Unfortunately, the Apple //e cannot show underlined text on the screen. Thus we have to use a trick, a trick that allows you to get underlined text from your printer and at the same time see which text is underlined on the screen.

The trick is to type an underline character *instead* of a space in front of the word or words you want to underline, then type another underline character *instead* of a space after the word or words you want to underline. Everything *between* the two underline characters will be underlined in the printout; the underline characters themselves will print as spaces.

This means that you cannot underline a single character in the middle of a word. Underlining a word at the beginning of a line is no problem: make the underline character the last character of the preceding line.

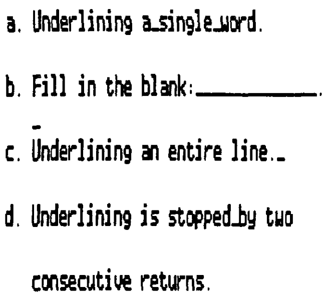
To underline the last word of a sentence, type the underline character *instead* of the first space after the period. The period will be underlined, but this is generally acceptable.

To print a continuous line (with most printers), as when creating a form, type as many underline characters in a row as necessary. The first and last will become spaces, but the rest will print as underline characters.

An isolated underline character can be obtained by typing three underline characters in a row.

Two return characters in a row will turn off underlining. This stops your whole text from being underlined in case you forget to turn the underlining off. A page character will also turn off underlining, but a page break will *not*.

Underlining:
Screen appearance



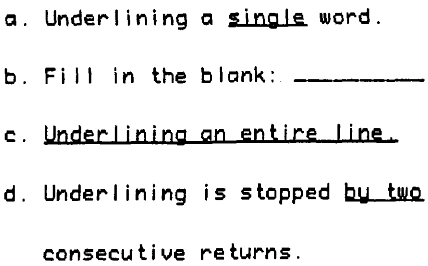
a. Underlining a_single_word.

b. Fill in the blank:__________

c. Underlining an entire line._

d. Underlining is stopped_by two consecutive returns.

Underlining:
Printout appearance



a. Underlining a single word.

b. Fill in the blank: __________

c. Underlining an entire line.

d. Underlining is stopped by two consecutive returns.

We feel that this underlining technique is not truly satisfactory, but it is a compromise between the design limitations of the Apple //e and our desire to keep the arrangement of text on the screen looking the way it will look on the printer. If the underline characters disappeared when printed, rather than being replaced by spaces, then items that lined up on the display would no longer line up in the printout. Before underlining will work with your printer, you may have to follow the instructions in Part 17: Detailed Control of Your Printer.

Other Features of the Keyboard

DELETE Key Erases material from the screen (see Part 9: DELETE). You can always tell what will be erased when you press DELETE — it's whatever is in the highlight (the solid unblinking rectangle).

CAPS LOCK An alternate-action switch that, when engaged, switches all letter keys to capitals. Numerals, symbols, and punctuation are unaffected. (This key, if left depressed, may interfere with cursor movement. See Part 7, "Capital Letters in Leap Patterns".

TAB Acts as if you had pressed the SPACE BAR enough times to reach the nearest preset tab stop to the right of the cursor. Tab stops are at columns 5, 10, 15, 20, ..., 75. Pressing TAB adds space characters to the text; it does not just move the cursor across the screen.

USE FRONT Press and *hold* USE FRONT (marked CONTROL on the Apple keyboard) when you want to use the actions shown by the labels on the *fronts* of the letter keys. Note that you cannot type any letters or move the cursor while USE FRONT is held down. The influence of the USE FRONT key ends as soon as it is released.

Keys You Can Use While USE FRONT Is Held Down

Key	*Label*
L	DISK
A	INSERT
G	CALC
D	SEND
N	PRINT
Z	No label (see Part 18: The SEND Command)

Autorepeat If you press and hold DELETE or any character-producing key for more than half a second, it will repeat at a rate of 15 characters per second. Keys that do not autorepeat include USE FRONT, SHIFT, CAPS LOCK, and the LEAP keys (Open-Apple and Solid-Apple).

The Arrow Keys Located in the lower righthand corner of the keyboard, these are the cursor control keys for the Apple system. They are disengaged while you are in the SwyftCard system. They become operational again when you use Apple software. We do not use them in SwyftCard because leaping is a much faster, more efficient way of moving the cursor.

Part 6: Pagination

Page Breaks and Page Numbering

Unless adjusted otherwise, SwyftCard allows for 54 printed lines per page. When you type enough lines to fill a printed page, SwyftCard automatically starts a new page and — by displaying a single-dashed line with the page number at center — shows you on the screen where the paper page produced by the printer would end. The number will print at the bottom center of the page, but the dashed line does not print. Pages numbers 0 and 1 will not normally print (this can be changed). Page numbering can be changed (see Part 16: Formatting the Printout).

Page Characters

You can force a page to end and a new page to begin at any point in the text by tapping the PAGE key (marked ESC on the Apple keyboard). This inserts a page character into the text. When sent to the printer, the page character causes the printer to advance the paper, print the page number at the bottom center of the page, and then advance to the top of a new page.

The on-screen symbol of the page character is a double-dashed bar with a number at center. Both the double-dashed bar of a page character and the single-dashed line of a page break may be thought of as representing a perforation in a physical page of paper.

```
---------------------------------- 25 ----------------------------------
```

Page Break

```
================================== 26 ==================================
```

Page Character

> *A page break is inserted when you type enough lines to fill a printed page. A page character will force a page to end at any point you desire. Think of the dashed line as representing the perforations in a printed page.*

Uses of Page Characters

Page characters are generally used to begin a new document. The page character ensures that the next thing you type will be at the start of a new page and will remain in that position, regardless of future changes you make to the text that comes before the page character. Once the initial page character has been typed, pagination is automatically taken care of by page breaks.

Page characters can also be used to force a page to have less text on it. For example, you might wish to leave half a page blank for an illustration. To do this you would type a half a page of text and then type a page character.

A page character can also eliminate a page break that falls at an inconvenient spot — in the middle of a paragraph, for example. Place the page character above the split paragraph. This eliminates the page break and moves the whole paragraph to the top of the next page.

Differences Between Page Breaks and Page Characters

Unlike page characters, page breaks are not characters in the text, just indications of where a new page will automatically start because the preceding page was filled. You might think of them as mileage markers — signs of going a certain distance, or filling a certain amount of text.

Because page breaks are not characters, you cannot erase them. Also, because SwyftCard's cursor moves only to characters in the text, you can move the cursor to a page character, but not to a page break (cursor motion will be fully explained in Part 7).

Note: *The first and last page characters — numbered 0 and 1 on a new screen — are exceptions: though you can move the cursor to them, you cannot erase them.*

If you insert new text above a page character, the page character moves out of the way with the rest of the displaced text. If you insert new text above a page break, the page break stays where it is while the text that is pushed out of the way flows around it. (Remember that it doesn't cost anything to take a bit of time and try these features out on your own system.)

Maximum Number of Page Breaks

SwyftCard permits no more than 200 page breaks or page characters per Text. In practice, this 200-page maximum is rarely even approached, as SwyftCard's storage capacity permits between 15 and 20 densely typed pages per disk.

This does not mean, however, that page numbers cannot exceed 200. The page numbering of individual disks can be adjusted (see Part 16, "Changing Page Numbers"). A single document may span several disks: pages 1-20 on Disk A, pages 21-40 on Disk B, pages 41-60 on Disk C, and so on.

Part 7: Cursor Motion

Leaping

SwyftCard's cursor — the flashing object on your screen — is capable of moving anywhere in the text almost instantly. The term *leap* has been adopted to describe this new mode of travel. A leap is extremely fast, averaging less than half a second no matter what the distance.

Creeping

The cursor may be moved short distances by creeping. Creeping moves the cursor character-by-character. Though usually much slower than leaping, creeping is often convenient for short moves.

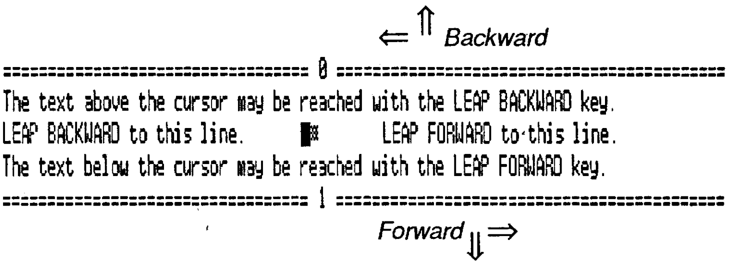
The LEAP Keys

Open-Apple (left side of SPACE BAR) and Solid-Apple (right side of SPACE BAR) control the movement of the cursor. Open-Apple is called LEAP BACKWARD, Solid-Apple is called LEAP FORWARD.

Pressing and *holding* LEAP BACKWARD readies the cursor for a leap backward to any character in the text. Pressing and *holding* LEAP FORWARD readies the cursor to leap forward to any character in the text.

Forward and Backward — Where They Are in Relation to Cursor

The line of text to the right of the cursor and all lines below the cursor are "forward." The line of text to the left of the cursor and all lines above it are "backward."



```
================================ 0 ========================================
The text above the cursor may be reached with the LEAP BACKWARD key.
LEAP BACKWARD to this line.     █*      LEAP FORWARD to·this line.
The text below the cursor may be reached with the LEAP FORWARD key.
================================ 1 ========================================
```

Forward and backward in relation to the cursor.

How to Creep

To creep forward, tap repeatedly on the LEAP FORWARD key. Each tap advances the cursor one more character.

To creep backward, tap repeatedly on the LEAP BACKWARD key. Each tap moves the cursor backward one more character.

Creeping does not autorepeat. If you find yourself wishing that creeping would autorepeat, this only means that you are creeping when you should be leaping.

How to Leap

First choose the exact character you want to leap to. This is called the target character. Any character among the cast of 97 (see Part 5, "Cast of Characters") may be used as a target, including a space, a return, or a page character.

Hold down the appropriate LEAP key, and, *while you hold it down,* tap the key of the target character. The cursor will leap in the direction indicated by the LEAP key to the nearest instance of the target character.

The Importance of Letting Go

After the cursor reaches the target character, let go of the LEAP key. You cannot type text or start a new leap until you let go.

Also, *don't let go until you are satisfied with where the cursor has landed.* Until you let go, you have the option of undoing the leap, that is, leaping back to where you were (see "Undoing a Leap").

Patterns — How to Reach Any Target in One Leap

Patterns are the characters (or the character) you type while holding down a LEAP key. A pattern may be up to 31 characters in length, much longer than you are likely to need. In practice, most patterns are three or four characters at most.

Patterns permit you to leap directly to a target.

The first character in the pattern is the target character. The characters that follow establish exactly which instance of the target character you want.

To use a pattern to leap somewhere, hold down the appropriate LEAP key, type the target character, then continue to type the characters that follow the target character in the text, just as if you were typing that part of text all over again. Stop typing and let go of the LEAP key when the cursor reaches the target.

For example, if you type "ab" while holding down LEAP FORWARD, the cursor will leap ahead to the nearest instance of an "a" followed by a "b". If you now type an "o" (without letting go of the LEAP key), the pattern becomes "abo". The cursor will leap in the same direction to the nearest instance of an "abo" pattern in the text (unless the cursor was already on the "a" of "abo", in which case it won't move).

Note that no matter how lengthy the pattern becomes, the cursor still leaps to the first character. If you typed the pattern "about", the cursor would leap to that nearest "a" which is followed by the letters "bout".

Nonexistent Patterns

If the pattern cannot be found, the cursor will return to its starting place. This may happen so fast that it looks as if it hasn't moved at all.

The Most Common Mistake In Leaping

The most common mistake is to try to repeat a leap by tapping the key of the character you want to leap to again (and again). If you want to leap again to the same character use the LEAP AGAIN key (while you continue holding down the LEAP key).

Tapping again and again on the same character key in an attempt to repeat a leap usually results in a nonexistent pattern (something like this: "yyyyyyy"). The cursor leaps back to where it started from and stays there. You wonder why it isn't moving, even though you are tapping and tapping.

It's because you're making *the most common mistake in leaping.*

Undoing a Leap

To undo a leap, continue to hold the LEAP key and change the pattern so that it won't match anything in text. Adding x's to the pattern is one way. Another way is to tap a bunch of keys at once. The cursor will return to its starting place because the nonexistent pattern cannot be found.

Again, the importance of *not* letting go: You must undo the leap *before* you let go of the LEAP key.

And again, the importance of letting go: You must let go of the LEAP key after the leap is undone. Only then can you type or begin a new leap.

Correcting Errors in a Pattern

As long as a LEAP key is held down, characters can be deleted from a pattern. One press of DELETE removes one character from the pattern. New characters can be typed to replace those that have been deleted.

The LEAP key must be held down continuously while deleting and correcting a pattern.

As you erase characters, the pattern grows shorter. Consequently, the cursor retraces its steps. If you delete all the characters in the pattern, the cursor will return to the point where it stood when you first pressed and held the LEAP key. This is a way of fixing a typing error you make while leaping, and it is also another way you can undo a leap.

LEAP AGAIN

Holding down a LEAP key changes the TAB key to the LEAP AGAIN key. While holding a LEAP key, each tap on the LEAP AGAIN key causes the cursor to leap again to the same pattern. This way you don't have to retype a pattern when you want to leap to it again.

As soon as you release the LEAP key, the LEAP AGAIN key goes back to being a TAB key.

If you tap LEAP AGAIN repeatedly, you will eventually find every instance of a pattern in the text. When you have found every instance of a pattern, the cursor will return to where it was when you first held down the LEAP key.

You can use LEAP AGAIN to circle the text continuously in the same direction, as if circling a racetrack. You know you've made a complete tour of the text when the cursor returns to its starting point — the place it was when you first held down the LEAP key.

LEAP AGAIN autorepeats. Pressing and holding LEAP AGAIN for more than half a second causes the cursor to move rapidly from one instance of a pattern to the next.

Pattern Memory

Once a pattern is used, it is retained until you press and hold down a LEAP key and type another pattern.

To Use the Same Pattern to Leap the Cursor in Either Direction

Let go of the LEAP key you are using. Press and hold down the other LEAP key. Tap the LEAP AGAIN key. Because the pattern is saved, you can leap again in either direction by using the LEAP AGAIN key.

To Reuse the Same Pattern

Hold down either LEAP key and tap LEAP AGAIN. The last pattern used in leaping is still available after you have typed something or used the DELETE key to delete character-by-character.

Capital Letters in Leap Patterns
— or, a handy thing, so long as the CAPS LOCK key is not stuck down.

Lowercase letters in a leap pattern match both uppercase and lowercase letters in the text. This means that you can look for an instance of a word without having to worry if it starts a sentence or is capitalized for some other reason. But uppercase letters in a leap pattern match ONLY uppercase letters in the text. By using a capital letter in a leap pattern you can look for, say, the word "Leap" at the beginning of a sentence, ignoring all the words "leap" in the middle of the sentence.

This explains why leaping seems not to work if the CAPS LOCK key is left in its down position: it makes the patterns you type come out all caps, so they do not match any lowercase letters.

The following chart shows what SwyftCard will find when you use capital letters in a leap pattern.

Leap Pattern	*What It Will Find*
me	me, Me, mE, ME
Me	Me, ME
ME	ME

Leaping Word-by-Word

Words are usually separated by spaces. To leap from word to word, leap to a space (hold down a LEAP key and tap the SPACE BAR), then use the LEAP AGAIN key. Press and hold the LEAP AGAIN key to engage autorepeat if you wish to move quickly from word to word (though still space-by-space).

Leaping Sentence-by-Sentence

Sentences generally close with a period. To leap from sentence to sentence, leap to a period (hold down a LEAP key and tap the period key), then use the LEAP AGAIN key.

Another approach: Most typists separate sentences with two spaces, so you can use two spaces as a pattern for leaping from sentence to sentence. This way you don't leap over sentences that end with a question mark or exclamation point. Hold down a LEAP key and tap the SPACE BAR twice, then use the LEAP AGAIN key.

Leaping From Paragraph to Paragraph

Paragraphs are generally separated by two returns. To leap from paragraph to paragraph, hold down a LEAP key, tap the RETURN key twice, then use the LEAP AGAIN key. This leap allows you to "scroll" through the text — quite quickly, if you hold down the LEAP AGAIN key while also holding down a LEAP key, engaging autorepeat.

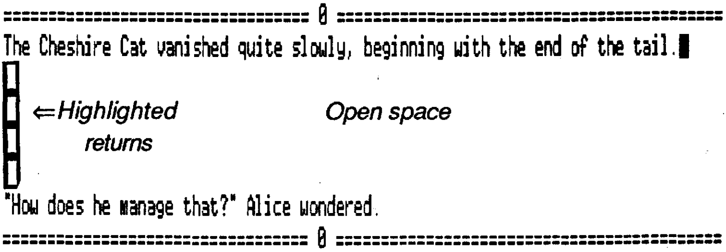
Leaping From Document to Document

Documents generally begin with a page character. To leap from document to document, hold down a LEAP key, tap the PAGE key, then use the LEAP AGAIN key.

Leaping to Open Space on the Screen

Open space means blank parts of the screen, such as beyond the ends of lines or between blocks of text.

Open space is created by a return character or characters along the left margin. You can see the returns if they are in the highlight (Part 8: Extending the Highlight describes how to make the highlight cover more than one character).

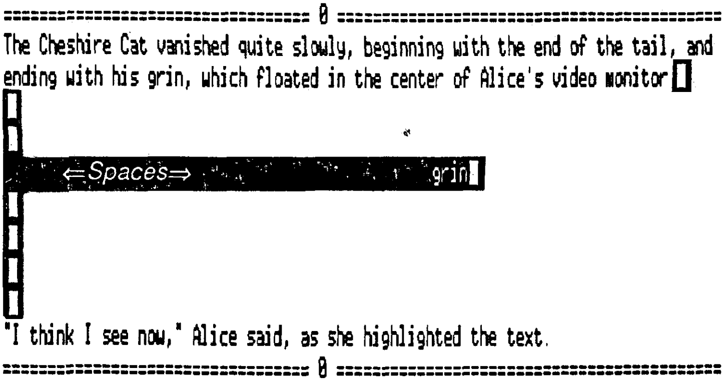


```
================================ 0 ========================================
The Cheshire Cat vanished quite slowly, beginning with the end of the tail.

"How does he manage that?" Alice wondered.
================================ 0 ========================================
```

Open space between lines of text. The returns are highlighted.

The open space to the right of the returns has no characters in it; therefore you cannot leap to it. If you wish to move the cursor into the open space, leap to a return next to the desired line. Use the TAB key or the SPACE BAR to type enough spaces to move the cursor to the middle of the screen.



```
================================= 0 ==========================================
The Cheshire Cat vanished quite slowly, beginning with the end of the tail, and
ending with his grin, which floated in the center of Alice's video monitor



    ⇐Spaces⇒                                    grin




"I think I see now," Alice said, as she highlighted the text.
================================= 0 ==========================================
```

Text that appears in the middle of open space is held there by space characters and returns.

Note:*The space characters and returns in this screen illustration are highlighted so you can see the structure of the text.*

Part 8: Extending the Highlight

The highlight — the bright, solid rectangle that accompanies the flashing cursor on the screen — can be extended to include two or more characters at once. (The highlight normally covers only one character.)

A contiguous portion of the text that includes two or more characters is called a *chunk*. A highlighted chunk of text can be manipulated in a variety of ways by using SwyftCard's commands.

To Highlight a Chunk of Text (Two or More Characters)

Leap the cursor from one end of the desired text to the other, then press both LEAP keys at once. You do not have to let go of one key before you press the other. (After letting go of both LEAP keys you can still highlight the text covered in your last leap until you type something or you leap again.)

A highlighted chunk of text remains highlighted until you type something, use the DELETE key, or leap again.

You can leap from the first character of the chunk to the last, or from the last to the first — the direction you leap is unimportant.

Cover the Chunk in One Leap

When you let go of the LEAP key, the boundary of the highlight is set. Therefore, use only one leap (one pattern) to cover the text you wish to highlight. It's OK to use LEAP AGAIN while leaping across the chunk of text because LEAP AGAIN does not force you to let go of the LEAP key.

The Exact Boundaries of an Extended Highlight

The highlight will extend from the character that was inside the highlight at the beginning of the leap to the target character at the end of the leap.

Highlighting Recently Typed Text

If you've typed some text since the last leap or creep, and you press both LEAP keys at once, the highlight will extend over the text you've just typed.

How to Undo an Extended Highlight

Tap either of the LEAP keys. The highlight will collapse back to the size of a single character.

If you tap LEAP BACKWARD the cursor and highlight will collapse on the first character of the text that was highlighted.

If you tap LEAP FORWARD the cursor and highlight will collapse on the last character of the text that was highlighted.

In practice, it is usually unnecessary to undo the highlight — all you have to do is begin typing. The highlight will collapse and the new characters will appear on the screen at the position of the cursor.

Rehighlighting

If you undo a highlight by tapping a LEAP key and then press both LEAP keys at once, the highlight will reappear exactly as before. As soon as you type or leap again, you lose the option to rehighlight.

After you invoke the PRINT, SEND, or CALC commands, you can rehighlight the text simply by pressing both LEAP keys at once.

Adjusting the Boundaries of the Highlight by Creeping

Sometimes you find that you have highlighted a chunk of text that is off at one or both ends by just a little bit. This can happen due to a leaping error, but usually it is because some other target was more convenient than the exact end of the chunk you wanted.

To adjust one end or the other of the highlight, collapse the highlight to the left or right end (by pressing and releasing the left or right LEAP key) and creep the cursor to the right or left until it is at the right place. Then press both LEAP keys to rehighlight the chunk you wanted.

Autoextending Highlight

Three of the commands — PRINT, CALC, and SEND — permit you to highlight automatically, without pressing both LEAP keys at once beforehand.

If you use any of these three commands after a leap, the text you covered in the last leap will automatically highlight and the command will be executed.

If you use any of them after typing something, the highlight will extend to everything you've typed since the last leap and the command will be executed. This can be very handy, especially with SEND.

To find out more about these commands, see Parts 12, 15, and 18.

Typing and Highlighting a Chunk

If you press both LEAP keys at once, then type some text, and then again press both LEAP keys at once (or allow the PRINT, CALC, or SEND commands to automatically highlight for you), the text you just typed will be highlighted.

Here's a practical example — how to write yourself a quick reminder: Press both LEAP keys at once (so that typing will start a brand new chunk), then type, "Remember to buy two quarts of milk!" Then use the PRINT command (which automatically highlights what you just typed), then press both LEAP keys at once (to rehighlight what you typed), and finally tap the DELETE key to erase the message.

Part 9: DELETE

To Delete Single Characters

Tap the DELETE key. DELETE erases whatever character is inside the highlight. Always. If the character you want to erase is already in the highlight, just tap DELETE. If not, move the cursor to the character you want to erase and tap DELETE.

DELETE is an autorepeating key. If you want to erase a string of characters rapidly, hold down DELETE (with or without autorepeat, this process is called character-by-character deletion).

Use character-by-character deletion with care — once deleted, these characters are gone forever.

Wide Cursor and Narrow Cursor

When the cursor and the highlight are both on the same character we call the cursor "narrow." The cursor becomes narrow whenever you press a LEAP key, leap, or creep somewhere.

As soon as you begin to type, or a command inserts characters into the text, the cursor and the highlight split apart. We call a cursor "wide" when the flashing cursor and the highlighted character are next to one another.

Backward Delete

A wide cursor deletes to the left. This is called backward delete. Backward delete resembles backspacing on a typewriter.

Forward Delete

A narrow cursor deletes to the right. This is called forward delete. The cursor does not move when you forward delete. It stands still while vacuuming up the characters to its right.

Why Two Delete Directions?

Now that we've told you about forward and backward delete, you should feel free to forget all about them most of the time. That's because DELETE generally does what you want it to do without any planning.

When you make a mistake while typing, the error is to the *left* of the cursor and DELETE backspaces to the left. When you LEAP to something you want to delete, you naturally leap to its *beginning*, putting the text you want to delete on the right side of the cursor. And that's the way DELETE works after you leap — to the right, automatically.

To Delete Two or More Characters at One Keystroke

Highlight the text, then tap the DELETE key. The highlighted text will vanish and the remaining text will close in to fill the gap.

To Bring Back the Deleted Chunk

Use the INSERT command: Press and hold USE FRONT, and, while you hold down USE FRONT, tap the "A" key.

Unlike a deleted chunk, text that has been deleted character-by-character cannot be retrieved.

How Much You Can Delete

As much as you have. There is no other limit on the size of a delete.

Part 10: The INSERT Command

How to Insert New Text by Typing

Move the cursor to the place you want to add material and begin typing. The character on which you place the flashing cursor will be pushed ahead of the new text.

How to Move a Chunk of Text Using the INSERT Command

Extend the highlight and DELETE the chunk. Leap the cursor to the new location, then use the INSERT command (press USE FRONT A). The deleted chunk of text will reappear at the new position of the cursor. The moved chunk is left highlighted to make it easy to move again.

To Make a Single Copy of a Chunk of Text

Extend the highlight and DELETE the chunk. Then use the INSERT command (press USE FRONT A). A DELETE immediately followed by an INSERT leaves the original copy exactly in place. Then leap to the location in the text where you want the copy to go. Use the INSERT command to insert the copy.

To Make Multiple Copies of a Chunk of Text

After deleting the chunk, use the INSERT command repeatedly. You can place the multiple copies where the original was located, or (by leaping before each INSERT) at desired locations in the text. If you hold USE FRONT A down for more than half a second, autorepeat takes over (autorepeat is slower for the INSERT command; expect a delay between copies).

Lifetime of the Deleted Chunk

The deleted text remains in memory, available for INSERT, until you highlight and delete another chunk. Only the most recently deleted chunk is available (and taking up space). Each time you delete a chunk, the previously deleted chunk is gone forever. If you save a the text on a disk, the most recently deleted chunk is also saved.

Search and Replace

It sometimes happens that you want to replace some or all occurrences of a word with some other word. There is a relatively easy way to do this. Say that you want to change "Smith" to "Jones" wherever it occurs in a letter:

1. Start by highlighting and deleting the replacement word as one chunk (character-by-character deletion will not do): Type "Jones" once, leap backward to "J", highlight the word by pressing both LEAP keys, then press the DELETE key.

2. Now leap to the word you want to replace and delete it character-by-character: Leap to "Smith" and press the DELETE key five times.

3. Use the INSERT command: Press USE FRONT A and "Jones" pops out on-screen.

   The rest is very easy. Just

4. LEAP AGAIN

5. Use the DELETE key

6. Use the INSERT command

Repeat steps 4, 5, and 6 for each instance of "Smith" you want to change. If there is an instance you want to leave alone, just LEAP AGAIN over it to the next instance.

Like a lot of what you can do with SwyftCard, it is easier to do it than to read the explanation. Try it once or twice.

Chapter III

Saving and Calculating

Part 11: The DISK Command

The Golden Disk Rule

Everything will work fine if you follow this simple rule: Always use the DISK command (USE FRONT L) just before you take a disk out; always use the DISK command just after you put a disk in.

The manuals that came with your Apple //e explain the mundane rules about taking care of disks, which we paraphrase as follows:

> Don't use them as coasters,
> Don't put them in toasters,
> Don't leave them near magnets,
> Don't tear them to fragments.

SwyftCard Needs Only One Disk Drive

If you have more than one disk drive, the DISK command will use Drive 1. If you're not sure which drive is Drive 1, open the Disk drive doors, use the DISK command (hold down the USE FRONT key, and, while you hold it, tap the L key), and see which red light comes on.

The Text

The Text, as we have said, is all the text that you have, all that you can work with at one time. Each Text has room for over 40,000 characters. That translates to roughly 38 double-spaced pages of average density, or 14 pages of really dense single-spaced text. You can store one Text on a disk. To put it another way, you can have as many Texts as you have disks.

Writing on Sand in the Windblown Desert

Such is the nature of your on-screen text. It is fun to see your ideas appear on the screen, and to edit and print them. But — unless you use the DISK command — that text is lost as soon as you turn off your Apple, and nothing can bring it back. A disk provides relatively safe and permanent storage. Thus you need the DISK command to store information on diskettes while the Apple's power is off, and to extend the amount of text that you can work with.

Storing Information Between Sessions

Just before you turn off your Apple, use the DISK command to save your work. After the save, the cursor blinks at a faster rate (the "happy" cursor). If you see a happy cursor, you know your text is safe.

The first time you save a Text you should have a brand-new (blank) disk in the drive. If the disk in the drive is not a blank, SwyftCard will not save on it. Instead, it will give you an audible warning signal — a "bleep" — and do nothing. This protects the Text on your disk from accidental ruin, since you wipe out what is on the disk whenever you save what is on the screen.

SwyftCard also recognizes which Text goes with which disk, so don't change disks before you use the DISK command. If the disk in the drive is different than the one you started with, you will get a bleep. Again, this is a safety feature: saving your text on the wrong disk would destroy the text on the disk.

After you save, you can safely turn off the computer. Leave the disk in the drive if you like. When you come back and turn on your Apple again, the screen will come back just as though you had never turned it off. The cursor will be in exactly the same place; even the last chunk you erased will still be in storage, ready to be recalled with the INSERT command.

What If the Power Goes Off by Accident?

Suppose your pet rhino trips over the power cord to your Apple, or a persevering woodpecker finally topples the power pole down the street. Out goes the electricity to your computer, and with it goes all the work you've done since the last time you used the DISK command and saved your Text. Thus it's good to cultivate the habit of using DISK every few minutes: when you stop to think, or when you stand up to stretch your legs — whenever it won't interfere with the flow of your work. It takes only a few seconds.

Switching From One Text to Another

Suppose you have one disk with a list of phone numbers and addresses, and another disk with letters on it. You arrive in the morning, put the phones disk in the drive, and turn on your Apple.

The phone numbers and addresses appear on your display and you add a new name to the list. Now you want to look at the letters disk.

Here's what you do: Use the DISK command. This will save the new name on the phones disk. Take out the phones disk and put in the letters disk. Use the DISK command again. After a few seconds your letters will show on the display.

The DISK Command Protects Your Work

What if you forgot to use the DISK command before taking the phones disk out of the drive? You put the letters disk in the drive and use the DISK command. This puts SwyftCard into a quandary. If it brings in the Text from your letters disk, it will wipe out and lose the new name and address you typed in. If it saves the text from your phones disk onto the disk in the drive, it will wipe out and lose your letters.

So instead of acting, SwyftCard sounds a bleep and does nothing. This reminds you to put the phones disk back in the drive, use the DISK command (which saves the new phone number and address), and then switch the disks.

After you've done all that and you use the DISK command again, SwyftCard thinks, "Aha! All the work on the phones disk is saved, so it is safe to bring in the Text from the letters disk." And that's what it does.

Protecting the Contents of a Diskette

By putting a patch of adhesive-backed material called a write-protect tab over the notch in the side of a disk, you can protect the disk's contents from being changed. A supply of write-protect tabs usually comes with a new box of disks.

Write-protection is especially useful for disks you want to refer to without the risk of changing them accidentally.

Other Bleeps When Using the DISK Command

If you use the DISK command to try to save your text onto a write-protected disk, SwyftCard will bleep to let you know that it can't do what you want. The fix? Take off the write-protect tab (if you really want to change the disk), or find another disk on which to save your Text.

There are two other reasons SwyftCard may bleep when you use DISK: an empty disk drive, or an open disk drive door. What to do? Put in a disk and close the door. Use the DISK command again.

Summary of What to Do If You Get a Bleep When You Use the DISK Command

1. Is there a disk in the drive? If not, put one in and close the door, try DISK again.

2. Is the disk drive door open? If it is, close it, and try DISK again.

3. Have you switched disks (e.g. gone from "phones" to "letters") without saving any changes you made first? Switch back to the first disk and use the DISK command, then go ahead.

4. Have you switched disks and you don't want to save any changes you made in the text of the first disk? Delete the entire Text so that there are ONLY the two page characters 0 and 1 left on the display and then use the DISK command. SwyftCard will bring the second disk on-screen without a bleep. (An empty text is illustrated at the end of Part 4.)

   Here's how to delete the entire Text: Leap to the first page character, number zero. Let go of all keys. Hold LEAP BACKWARD (the left LEAP key) and tap the LEAP AGAIN key (TAB) once. This leaps you to the last page character in the text. Now press both LEAP keys at once. The entire Text highlights. Press the DELETE key. Done.

   CAUTION: *It is very important that you delete the entire text, leaving not a single character. If you use the DISK command with, say, a couple of space characters left, you will save the few characters you have, wiping out the old version of the text on the disk — the text you wanted to get back.*

5. Is there a large chunk of text highlighted, a chunk that's too large to be copied up (see next section) to the new text? Tap a LEAP key to collapse the highlight to a single character and then try the DISK command again to load in the new text without attempting to copy up the chunk that won't fit. Alternatively, highlight a smaller chunk that *will* fit and try the DISK command again.

Force Loading —
Or, What to Do When Your Kid Bangs on the Keyboard and You Want to Go Back to the Way It Was

To discard the text on-screen and reload the text on the disk, follow the steps given in 4 above.

Another way to do it is to hold down the CONTROL (USE FRONT) and Open-Apple keys, and, while holding both of these keys, press and release the RESET button at the upper right keyboard (see Part 20, "Overriding the DISK Command" for a complete discussion of this technique).

Something You Don't Want to Do When Using the DISK Command

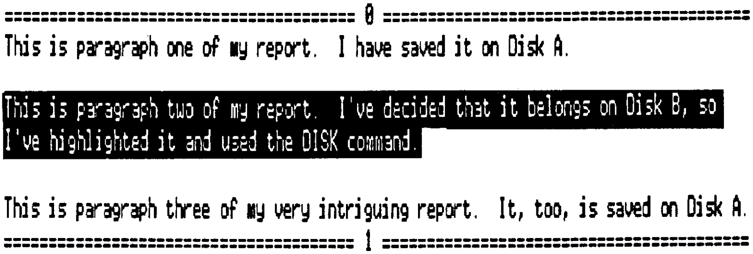
When you use the DISK command, the red in-use light on your disk drive will come on. This is an indicator that the drive is running. *Don't open the disk drive door while this light is on.* It could damage your text. If the disk drive door is left open and you use the DISK command, no harm done; but *don't close the door while the red in-use light is on.* This, too, could damage your text.

Moving a Chunk From Disk to Disk to Disk — Copy Up

When you change Texts, if a chunk (i.e. two or more characters) was highlighted in the old Text, it is inserted (and left highlighted) at the location of the cursor in the new Text. We say that the chunk of text was "copied up" from the old to the new Text.

The following illustrations show how to copy up. In (1) you see a paragraph of text highlighted. The DISK command has just been used to save the text onto Disk A. When we replace Disk A with Disk B, we're ready to copy up. In (2) we've used the DISK command again to load the Text from Disk B — and notice, the highlighted paragraph has been inserted into the new text.

The paragraph from Disk A has been inserted at the position of the cursor in Disk B. If if you didn't want it here, you could tap the DELETE key, move the cursor, then use the INSERT command.

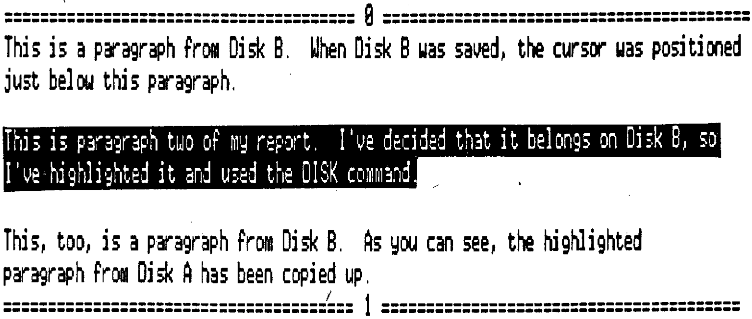


```
==================================== 0 ======================================
This is paragraph one of my report.  I have saved it on Disk A.

This is paragraph two of my report.  I've decided that it belongs on Disk B, so
I've highlighted it and used the DISK command.

This is paragraph three of my very intriguing report.  It, too, is saved on Disk A.
==================================== 1 ======================================
```

To copy up the middle paragraph, we highlight it, then use the DISK command to save it on Disk A. Put Disk B in the drive and use the DISK command again — you see the result below:



```
==================================== 0 ======================================
This is a paragraph from Disk B.  When Disk B was saved, the cursor was positioned
just below this paragraph.

This is paragraph two of my report.  I've decided that it belongs on Disk B, so
I've highlighted it and used the DISK command.

This, too, is a paragraph from Disk B.  As you can see, the highlighted
paragraph from Disk A has been copied up.
==================================== 1 ======================================
```

The highlighted paragraph from Disk A has been inserted into the text from Disk B. You can now save the combined text. This illustrates the copy-up technique: Any text left highlighted on the screen will be inserted into the next Text you load with the DISK command.

Thus the DISK command can be used to automatically copy a chunk of text from one Text to another. The chunk is copied — not removed — from the old Text.

If the chunk is too large to fit into the new Text, then SwyftCard bleeps and leaves you in the old Text. You will have to copy up less text or throw away something from the new Text.

If you accidentally copy up a chunk of text, don't panic. It is very easy to get rid of it from the new Text. Since it is highlighted, just press DELETE.

Using Belt and Suspenders, or Making a Backup Disk

Many people feel more secure having a backup disk — a second, identical copy of their work — as an insurance policy against possible loss or destruction. If one disk gets munched, then there's always the other.

Use a blank disk for the backup. Put the blank in the drive immediately after you use the DISK command to save onto the first disk. Use the DISK command again. This will create your backup copy.

As far as SwyftCard is concerned, the original and the copy are *identical.* You can use the DISK command any number of times on one disk, then put the second disk in the drive, use the DISK command, and SwyftCard will save the Text on the second disk.

The most important thing to remember is not to do any typing or other operation between the two uses of the DISK command. In other words, the DISK command *has* to be the last thing you did to the first disk before you save on your backup.

You can make a second copy at any time onto a blank (fresh out of the box) diskette.

How to Make a Used Disk Into a Blank Disk

This is a deadly command, since it destroys whatever is on the disk. After you apply this command, the disk is "new" as far as SwyftCard can tell. First type the dreaded incantation:

```
CALL 3600
```

Then highlight it and use the CALC command (USE FRONT G). The disk drive whirrs, the deed is done. You can now use the DISK command to save onto this disk. SwyftCard will treat it like a blank.

Once executed, CALL 3600 is irrevocable. Fortunately, it is not likely to happen by accident.

Making a Backup Disk After CALL 3600

1. Have the Text you want to save on the display (let's call this Text the "original").

2. Put the used disk (the one you want to wipe clean) in the drive. Type and highlight CALL 3600 and use the CALC command. Thisdisk is now a blank.

3. Take the blank disk out of the drive, press DELETE to get rid of the "CALL 3600" that was left highlighted on the screen.

4. Put the original disk in the drive and use the DISK command.

5. Put the blank disk in the drive and use the DISK command.

This method satisfies the rule about not typing or doing any other operation between uses of the DISK command when making backup disks.

Running Apple Software, or, So You Want to Play a Game

SwyftCard will not stand in the way of any standard Apple program disk you may want to run. Just put it in the drive and use the DISK command. As long as you've saved any changes in the SwyftCard disk you've been using, the Apple program will operate as though SwyftCard wasn't there.

To get back to using SwyftCard after using some other program, remove the Apple program disk, place a SwyftCard-generated disk in the drive, then press and hold the CONTROL and Open-Apple keys, and, while holding them, press and release the RESET key at the upper right corner of the keyboard. If you do not have a SwyftCard-generated disk, you may have to turn the computer off and then on again to get back into SwyftCard.

When You Run Out of Room

You can tell when this has happened since it's the only time that SwyftCard will bleep EVERY TIME YOU TAP A CHARACTER KEY.

This always seems to happen at the most inconvenient moment. But it's easy to fix. You're usually in one of two situations when you run out of room: either you don't mind throwing out some of the text, or you want to save everything you've got and add more to it.

Let's consider the first case first.

When You Run Out of Room and Can Throw Something Away

If you can throw away a few chunks of text, there's no problem. Highlight and delete them and you have some more room. If there is only one chunk to be thrown away and you highlight and delete it, this WILL NOT GET YOU BACK THE ROOM THIS CHUNK TOOK UP! That's because it is still stored, waiting to be INSERTed again.

In two steps, here's how you avoid this problem and get back the room you need:

1. Highlight and delete the text that you want to throw out.

2. Highlight and delete two characters (character-by-character deletion will not do — leap from character to character, press both LEAP keys, press DELETE).

Now INSERT only has two characters to remember, as it has thrown away the big chunk you deleted first. You now have more room. You can retype the two characters you just highlighted and deleted.

When You Run Out of Room and Want to Preserve Everything

The trick here is to keep part of your text on the ORIGINAL disk, and move the rest of the text onto the NEW disk. The way this is done is to make TWO disks with the entire Text, and then erase the *first* part of the text from the NEW disk, and the *second* part of the text from the ORIGINAL disk. Here's the recipe:

1. Save the text on the ORIGINAL disk by using the DISK command.

2. Remove the ORIGINAL disk from the drive and put in the blank NEW disk.

3. Highlight and DELETE the first part of the text.

4. Use the DISK command to save the remaining second part of the text on the NEW disk.

5. Remove the NEW disk and set it aside.

6. Highlight and DELETE the complete text, so there is nothing left (not even *one* character).

7. Put the ORIGINAL disk in the drive.

8. Use the DISK command to bring in the complete text.

9. Highlight and DELETE the second part of the text.

10. Use the DISK command to save the remaining first part of the text on the ORIGINAL disk.

This is not the fastest way to do this, but it's the safest, since at all times there is a copy of both parts around, and any mistake can usually be caught and corrected. Once the ten steps are done, you may have to use the technique described in "When You Run Out of Room and Can Throw Something Away" to clear INSERT of the big chunk of text.

A faster, though not as safe, technique the authors have used is this: Just make a COPY by putting a blank disk in the drive and using the DISK command. Now delete roughly half of the text (and get rid of the stuff that INSERT is holding on to). Continue working. Later, when you get around to using the COPY disk, you have to remember to delete the other part of the text. It is this having to remember what to delete and the chance that you can lose track of what you are doing that makes us recommend the ten-step method given above.

You can also split a text by using COPY UP, so long as the disk you want to use for the NEW disk already has some text on it.

How to Split a Text Onto Two Disks When the Text Is Not Full

Use steps one through ten above. The method works fine even if the text isn't full.

A Strange Case: The Text Is Empty, Yet You Get a Bleep on Every Character You Type

A rather interesting and paradoxical situation can happen if you've filled up the text and never deleted a chunk, or if you have done an INSERT, gotten a bleep and continued typing until the text is full.

Suppose you now highlight the entire text and delete it. You will see just the two page characters on the screen, just as when you first turn on your Apple with SwyftCard. It looks empty, but every character you type still causes a bleep because the text is really full — *of deleted characters.*

How to get rid of this jammed text? Use the INSERT command to bring back the deleted text. Use character-by-character delete to remove a few characters. Now delete the whole remaining text. You have room to delete a chunk of two characters now. Type them, highlight them, and delete them.

Part 12: The CALC Command

CALC is used when you need to do arithmetic.

How to Do Calculations

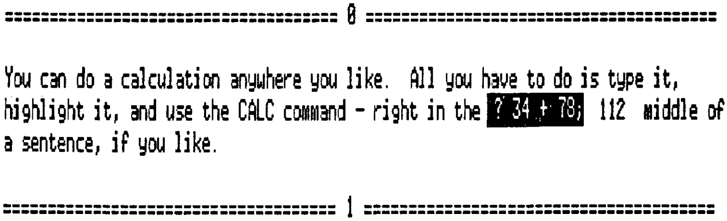
To do a calculation, such as adding 34 and 78, think of asking SwyftCard,

```
How much is 34 plus 78?
```

Abbreviate "How much is?" to a question mark and type the formula (followed by a return or space so that the answer, when it appears, will be easy to read). Highlight the formula (including the question mark) and use the CALC command (USE FRONT G). The answer will appear in your text just after the problem:

```
? 34 + 78; 112
```

The problem will remain highlighted so that you can delete it, leaving only the answer in your text. A return character is put in after the answer. If you do not want a return after the answer, type a semicolon after the arithmetic expression. For example, ? 34 + 78; will not be followed by a return.



```
==================================== 0 ======================================

You can do a calculation anywhere you like.  All you have to do is type it,
highlight it, and use the CALC command - right in the ? 34 + 78; 112 middle of
a sentence, if you like.

==================================== 1 ======================================
```

Using the CALC command.

CALC Is an Autoextending Command

When using CALC, you do not have to press both LEAP keys at once in order to highlight the text.

If you use the CALC command after a leap, the text you covered in the leap will highlight automatically and the command will be performed.

If you type something after a leap, after a creep, or immediately after extending the highlight, and then use the CALC command, the highlight will automatically extend to what you have typed and the command will be performed.

Calculations You Can Do

Operation	*Symbol*
add	+
multiply	*
subtract	-
divide	/
exponentiate	^

Multiplication is symbolized by an asterisk to avoid confusion with the letter "x". You may also use parentheses in expressions.

The Rest of the Details of Arithmetic...

are in the AppleSoft BASIC manual. When you use the CALC command you are actually writing a BASIC "PRINT" statement. You don't need to know this to perform calculations on SwyftCard, but if you are interested in pursuing the subject, the BASIC manual will show you how to use functions such as LOG, SIN, COS, and TAN. You can write whole BASIC programs in the SwyftCard editor, LIST and RUN them. Even graphics is available — but how to write programs is beyond the scope of this manual.

Part 13: CALC Command Miscellany

In addition to its major use — doing calculations — CALC can perform a few odd jobs, such as telling you how much more you can type before you run out of room, or storing brief stock phrases.

How Much Room Is Left?

To get the answer, type and highlight

```
?RO%
```

and then use the CALC command. ("RO" stands for "ROom," and the percent character is required by BASIC.)

This will tell you exactly how many characters you can type before the Text is full. If the value comes out negative, this means you have more than 32,768 characters left. A negative answer usually means you have so much room that you don't really care about the exact figure, but if you must know, just type and highlight

```
?65536 + RO%
```

and use the CALC command.

Stored Messages

CALC can store phrases that you use often but don't like to retype time and again. For example, suppose that you had to include a legal phrase many places in a document. Type

```
L$ = "unless the party of the first part gives prior written notice."
```

just once, highlight it, and use the CALC command (the highlight will not collapse, but the message will be recorded). Now, whenever you type, highlight, and use the CALC command on the expression

```
?L$
```

the phrase you set up will automatically be inserted into the text (without quotation marks). In the example, we used "L" for "legal."

You can have up to 26 phrases stored as letters of the alphabet (A$, B$, C$, ... Z$), as long as the total does not exceed the space allocated for BASIC. Each phrase can be up to 80 characters in length, must be typed between quotes, and must not contain any returns.

Those of you conversant with Apple's BASIC language will be able to find many other applications, including ways of storing more and longer phrases.

Part 14: BASIC in the Swyft Environment

Mixing programming and text editing can be very powerful. For example, to make a calendar for the month of January, you might write the following program

```
10 FOR I = 1 TO 31
20 PRINT "JANUARY "; I
30 NEXT I
RUN
```

If you then highlight it (including the RUN) and use the CALC command you will get a calendar that begins

```
JANUARY 1
JANUARY 2
JANUARY 3
JANUARY 4
JANUARY 5
JANUARY 6
```

You can even LIST the program by typing the word "LIST" and using the CALC command on it. Almost all BASIC commands work as usual. However, the size of the BASIC program is limited to about 900 bytes (in its compacted internal form). So SwyftCard is not suitable for developing large programs.

Use of BASIC can be dangerous to your text. If used incorrectly or with certain values CALL, POKE, and PEEK can cause you to lose your text. Graphics — both regular and high resolution — will work, but they, too, may clobber your text. Graphic images are stored in the same place as text. High resolution images will usually not clobber anything if you have only a few thousand characters in your text (a full page or two).

You can try anything safely by doing a DISK command and taking out your disk before you run a BASIC program that might affect your text. Check out your text after running the program, and if everything is OK, then use the DISK command to save your work.

While doing simple arithmetic is generally quite safe, running BASIC programs may wreak havoc with your text. We therefore recommend that you not keep BASIC programs on disks that also have important text on them.

If running BASIC does cause the system to "hang," try CONTROL RESET. This will often get things up and running, and usually doesn't cost you (see note below) the loss of your text. Backup disks are a *great* idea when experimenting with the limits of BASIC.

Any programs that you write are run in the same manner you do calculations: leap from one end of the program to the other, then use the CALC command. The highlight autoextends.

Note: *While a BASIC program is running, the traditional CONTROL C (or USE FRONT C) can be used to interrupt it. You may have to use it a few times to stop a program. The computer may insert the word "BREAK" into your text when you do this. A neater way to stop the program is to press and hold the CONTROL key, and, while you hold it, press and release the RESET key (located in the upper right corner of the keyboard).*

Warning: *Don't allow a BASIC program to cause the text to consist of more than 256 pages. This will cause you to lose your text.*

Chapter IV

Printing and Communicating

Part 15: Printing

Printer Preliminaries

It is very easy to print with SwyftCard. However, in the name of progress, every printer manufacturer likes to do things their own way, and as a by-product of all this progress nearly every printer requires a custom setup. So while it is very easy to print with your SwyftCard, it may or may not be easy to set up the printer and the printer interface.

Details on how to set SwyftCard for certain printers may be found in Part 17. Your printer may be easy to set up. If it is not, you may have to use the interface card and printer manuals to figure out how to set it up. Part 17 explains the different settings required. If you have purchased the equipment from a dealer, it should be in the dealer's interest to make sure that it works.

Note: *Be sure that the printer interface card is in Slot 1 of your Apple.*

How to Print

You can print whatever is highlighted. Actually, you can just leap from first character to last (or last to first) of what you want to print, then use the PRINT command (USE FRONT N). That's it. You do not even have to highlight the text — the highlight will extend automatically.

You can print as little as one character, a part of a word, a few words, or as much as you want: a paragraph, a letter, a chapter of a novel, or the entire Text. You can start printing in the middle of a line, or even in the middle of a word, and you can choose the ending point with equal flexibility.

If you begin the highlight in the middle of a line, then your printer will start to print in the middle of a line. The way it looks on the screen gives a very good indication of how it will look on paper. The highlight even vanishes from the screen line-by-line at the same rate the text goes to the printer.

The page breaks and page characters in your text will correspond to the perforations between sheets of paper in your printer (top and bottom margins will be inserted). You may have to experiment to find just the right spot to position your paper before printing starts so that the top and bottom margins come out equal.

If your printer requires that you print one page at a time, just highlight one page at a time and print it. To make it easier to do this, go through the text and type a page character one line above each page break (the page break will disappear when you type a page character above it). After each page prints out, leap forward to the next page character and use the PRINT command.

Printers and Form Feeds

A form feed is a special signal you can send to some printers that tells them to move to the next page. Page breaks and page characters send a form feed to the printer, causing it to roll out the rest of the paper page it's working on and get ready to start a new page.

However, if the first character in the chunk of text you've highlighted is a page character, then that particular form feed is not sent to the printer. This saves a lot of paper, as it means that you won't lose a sheet of paper if you highlight from one page character to another when you are about to print (see "How to Print a Document" below).

If your printer requires you to eject a blank sheet of paper before you start printing, just put two page characters at the beginning of what you want printed and include both of them in your extended highlight.

How to Print a Document

In the tutorial we recommended that you begin a document by typing a page character, then let the page breaks appear automatically as you typed the text. This means that many documents will have only two page characters: one at the beginning and one at the end. To print the document, leap from one to the other, then use the PRINT command.

If the document has several page characters in it, use the LEAP AGAIN key to help the cursor leap from first to last, then use the PRINT command.

How to Print the Entire Text

There's a trick to this: Leap to the first page character in the Text, the one with a number zero in it. Let go of all keys. Press and hold LEAP BACKWARD, then tap the LEAP AGAIN key. Let go of all keys and use the PRINT command.

When you tap the PAGE key you will "hop the back fence" and leap to the last page character in the text. You've leaped from the very beginning to the very end, so when you use the PRINT command, SwyftCard will highlight and print the entire Text.

Getting the Page Numbers Right

SwyftCard automatically leaves page one unnumbered, then prints a number for page two and higher at the bottom center of each page. In Part 16 we explain how to play with pagination in detail. But without ever reading that section you can easily get any document to begin with page one: just move the document to the front of the text.

Here's how: Delete the document, leap the cursor to the page character with the number zero in it, and use the INSERT command. The inserted text (your document) will be highlighted, so you can use the PRINT command as soon as it appears on-screen.

Emergency Stop Print

To halt a print that is in progress, press and hold the USE FRONT key, and, while you hold it down, press and release the RESET button at the extreme upper right of the keyboard. Do *not* press either the Open-Apple or Solid-Apple keys while you do this. The unprinted text will remain highlighted. To resume printing, use the PRINT command. Some printers may require that you move the cursor back to the nearest page break before you begin to print.

Part 16: Formatting the Printout

SwyftCard automatically formats your text and printout with standard margins, single-spaced lines, and 66-line pages. This chapter tells you how to adjust the format. Format adjustment is another application of the CALC command. Remember that the autoextending highlight feature is available with the CALC command (see Part 8, "Autoextending Highlight").

Line Width

Both the screen and printer are normally set up for 80 characters per line. The Apple display is unable to show more than 80 characters per line, so it is impossible to set line width at greater than 80 characters. To set the line width to a lower number, say 66 characters, type

```
WI%=66
```

Highlight this formula, then use the CALC command. After a second or two, the lines on-screen will narrow to 66 characters and the entire Text will be automatically reformatted, including pagination. The highlight remains extended so that you can tap the DELETE key and erase the formula from your text.

You can set the line width to any value from 16 to 80. Values too large or too small are set to the maximum or minimum values respectively. "WI" stands for "WIdth."

Narrowing the line width increases the number of pages in the Text. Don't narrow the line width if this would result in more than 200 page breaks: it could destroy your text.

Left Margin (of the Printer)

The left margin is normally set at the left edge of the printer. It can be moved to the right a given number of characters, say 16, by typing, highlighting, and using the CALC command on

```
MA%=16
```

This does not affect the computer's display in any way. "MA" stands for "MArgin."

The whole body of printed text moves to the right on the page when you increase MA%, thus you may wish to reduce WI% when you increase MA%. For example, if you set MA%=12 you may want to reduce WI% from 80 to 68 to keep the same right-hand margin.

Top and Bottom Margins

The top margin is set by typing, highlighting, and using the CALC command on an expression giving the number of blank lines you want in the top margin. If you wanted six blank lines above the printout on each page, you would type

```
AB%=6
```

and use the CALC command.

Most printers print six lines per inch, so this gives the standard one-inch top margin. SwyftCard automatically gives six-line margins. "AB" stands for "ABove."

Similarly, typing

```
BE%=6
```

and using the CALC command sets the bottom margin to six blank lines (BE% is automatically set at 6 when SwyftCard starts up). "BE" stands for "BElow."

When you increase the margins, the number of lines per page automatically decreases to compensate; when you decrease the margins, the number of lines per page automatically increases. SwyftCard assumes that there is room for a total of 66 lines per page from the top edge to the bottom.

Number of Lines Per Page

SwyftCard normally allows for 66 lines per page (top to bottom). A lesser number, say 34, may be selected by typing

```
PL%=34
```

highlighting this expression, and then using the CALC command. "PL" stands for "Page Length."

Setting PL% to a large number results in unpaginated text (maximum value is 255). When reducing the page length, make sure that you don't get more than 200 page breaks in the text as a result (you may lose your text).

Lowest Page Number

The first page of a document or letter is customarily unnumbered; numbering typically commences on page two. SwyftCard does this automatically (PF%=2 when you start up). You might desire something different, like having the page numbering start with "1". In that case you would type and highlight

```
PF%=1
```

and then use the CALC command.

If you wanted page numbering to start on page 3, you'd use PF%=3, and so on. You can use a very high value, such as PF%=200, to get rid of all printed page numbers (the maximum value for PF% is 32767). Though the numbers will not be printed, they will still appear on your display. "PF%" stands for "Print First."

Changing Page Numbers

Consecutive page numbering is possible for very long documents that sprawl over many disks. For example, say that the first disk has page numbers 1 through 22. How would you get the second disk to start with page 23? You would set the number on the first page character to 22 (the first page character is normally "0"; that is, PA%=0 when you start up). That way the bottom of the first page will have the desired value of 23. Simply type and highlight

```
PA%=22
```

and then use the CALC command. "PA" stands for "PAge."

When the text on one disk includes many documents, only the first document starts with page one. All other documents will start out with some other page number. One way to alter the page numbering of such documents is to move them to the front of the text, as explained in Part 15. Another way is to move the zero page break ahead in the text. This is done by resetting the page numbering to a negative number.

For example, suppose you have a letter that begins on page ten. You'd prefer that this letter began on page one, so you type and highlight

```
PA%=-9
```

and use the CALC command. This reduces page character 9 to 0. Page 10 is now page 1 (the page numbering on your display will change, so you can check to see if you've got it right).
Some documents may have negative page numbers when you use this technique of changing the page number. This is OK. Since the normal setting of PF% prevents page numbers less than two from printing, these negative numbers simply will not print.

To go back to normal numbering, type and highlight

```
PA%=0
```

and then use the CALC command. "PA" stands for "PAGE."

Changing the Vertical Position of the Page Number

The position of the page number is normally dead center, three lines from the bottom of the page. While the page number will always be centered, it can be moved up or down. The lowest it can be is zero, the highest is limited to two less than the number of lines in BE%, the bottom margin. If you wanted to move the page number to the fourth line from the bottom of the page, you would type

```
LP%=4
```

highlight it, and use the CALC command. You cannot change the horizontal position of the page number. "LP" stands for "Line of Page." (LP%=3 is normal.)

Double-spacing

To make your text print double-spaced, type and highlight

```
SP%=2
```

and then use the CALC command. The new setting will not noticeably affect the appearance of the text, which remains single-spaced on the screen. The page breaks and page characters will move closer together because you have half the number of lines per page.

To return to single-spacing, type, highlight, and use the CALC command on

```
SP%=1
```

This is the normal value for SP%.

SP% can be 1 or 2. Values less than 1 will be changed to 1; values greater than 2 will be changed to 2.

The screen may jump when you use the CALC command to change the line spacing. "SP" stands for "SPacing."

Use caution in changing to double-spacing if the text includes more than 100 pages.

Questioning the Printer Values

If you want to know how many characters you have per line, type

```
?WI%
```

highlight it, and use the CALC command. If you want to know what number the first page character is set to, type

```
?PA%
```

highlight it, and use the CALC command. All the values described in this chapter may be questioned in this manner.

Resetting the Format Values to Normal

The standard default values for each of these settings can be found in the List of Features (default values are normal values). To adjust any setting back to the standard default value, type the expression, an equal sign, and the default value, then highlight it and use the CALC command.

For example, to set the line width back to the normal value of 80, you would type

```
WI%=80
```

and use the CALC command.

Adjusting Several Format Values at Once

You can adjust several values at once by typing a colon between them. For example, to reset line width, page numbering, and top margin, you would type

```
WI%=80:PA%=0:AB%=6
```

and use the CALC command.

Part 17: Detailed Control of Your Printer

Though most printers require some setup, SwyftCard often works reasonably well without it. Test the result of using the PRINT command first. If you get the wrong font or inappropriate margins — or if all you get is garble, lots of printing on the same line, double-spacing (when you didn't ask for it), or nothing at all, then read on.

We cannot cover the installation of your printer, printer interface, and printer cable, nor how to load the paper. We leave that up to the printer and interface card manuals. Please read them carefully or have your dealer or an experienced friend give a hand.

Interface Cards

We tested parallel printers with the Grappler+ and the Apple parallel card; serial printers with the Apple Super Serial Card. The switches on the Grappler+ card were set as follows:

1	2	3	4
off	*on*	*on*	*on*

Super Serial Card switches were set as follows (*dn* is down or off, *up* is up or on)

	1	2	3	4	5	6	7
SW1	*dn*	*dn*	*dn*	*up*	*dn*	*up*	*up*

	1	2	3	4	5	6	7
SW2	*dn*	*dn*	*dn*	*up*	*dn*	*dn*	*dn*

If there is a switch on the parallel card which supplies a "line feed" for each "carriage return" (sometimes called "Auto LF with CR") it should be on. If this gives you double-spacing (extra space between lines of print), then it should be off.

Printer Switches

Printers often have a multitude of switches, often in places that require disassembly to get to. We have designed SwyftCard to work with most printers with *the switch settings as they come from the factory.* So don't alter their settings unless you find some reason that makes it necessary. If you have a serial printer with a baud rate setting, set it for 9600 baud.

Printer Initialization

If the printer does not operate satisfactorily when you use the PRINT command, then follow the instructions in this and the following sections.

Initialization is done by means of sending a message to the printer telling it how you want it to work. You make up the message and prepare it to be sent to the printer by typing the message, highlighting it, and then using the CALC command. Then, whenever you use the PRINT command, the message is sent out to the printer followed by the text you want printed.

You can type the printer initialization message *wherever you happen to be in the text* — in the middle of a paragraph if you like. Just highlight it and use the CALC command.

In the next few paragraphs we will give the settings necessary for many printers. You can use these just as they are. Then we will explain how these settings are created. This will be of interest if you just want to know more, or if you want to make changes or use a printer we have not listed.

How to Use the Printer Initialization Message

Type and highlight the message given for your printer, then use the CALC command: this sets up the printer messages. Since new printers come on the market all the time, and old printers continue to be useful, we may not have listed your printer brand or we may list the right brand but the wrong model. It would not hurt to try some of messages listed here, especially if one of them is for your brand but a different model.

Note: *if your printer keeps on printing over and over on the same line, then change the part of the messages below that say*

```
PR$=CHR$(0)+
```

into

```
PR$=CHR$(1)+
```

and leave the rest the same as shown.

Printer Settings

Group 1: Printers Not Requiring Any Set Up

Epson FX-80, FX-85, FX-80+, FX-100, RX-100, FX-185,
Spectrum LX-80, JX-80
Brother M-1009, Brother Twinriter 5 (DP Dot Matrix mode)
Panasonic P-1092
Star Micronics SD-10, SD-15, SG-10, SG-15, SR-10, SR-15, SB-10
Hewlett Packard ThinkJet
Okidata 82, 83, 84, 92, 93, 182, 192, 193 (IBM mode, draft quality)
Smith-Corona D-100, D-200, D-300
IBM Graphics Printer

If you want to change back to using one of the printers above after working with some other printer, type, highlight, and use the CALC command on

```
PR$=CHR$(0)+CHR$(0)+CHR$(5)+CHR$(27)+CHR$(77)+CHR$(27)+CHR$(108)+CHR$(8)
US$=CHR$(27)+CHR$(45)+CHR$(49)
UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)
```

Group 2: Printers Requiring Set Up

Brother 2024

```
PR$=CHR$(0)+CHR$(10)+CHR$(5)
US$=CHR$(27)+CHR$(45)+CHR$(49)
UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)
```

Epson LQ-1500, SQ-1500

```
PR$=CHR$(0)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(33)
US$=CHR$(27)+CHR$(45)+CHR$(49)
UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)
```

Fujitsu Dot Matrix 24

```
PR$=CHR$(0)+CHR$(0)+CHR$(5)+CHR$(27)+CHR$(77)+CHR$(27)+CHR$(108)+CHR$(8)
US$=CHR$(27)+CHR$(69)
UE$=CHR$(1)+CHR$(27)+CHR$(82)
```

Diablo 630
Xerox Advantage D25
Brother HR-25, HR-35, HR-15XL, HR10, Twinriter 5
(WP Daisywheel mode)
DTC 380 (and other DTC)
Panasonic KX-P3151
Octave Systems (DTC) Laser Printer

```
PR$=CHR$(0)+CHR$(4)+CHR$(5)
US$=CHR$(27)+CHR$(69)
UE$=CHR$(1)+CHR$(27)+CHR$(82)
```

Juki 6100, 6300

```
PR$=CHR$(0)+CHR$(6)+CHR$(5)
US$=CHR$(27)+CHR$(69)
UE$=CHR$(1)+CHR$(27)+CHR$(82)
```

Apple Laserwriter

```
PR$=CHR$(1)+CHR$(0)+CHR$(0)
US$=CHR$(27)+CHR$(69)
UE$=CHR$(1)+CHR$(27)+CHR$(82)
```

Qantex 7030

```
PR$=CHR$(0)+CHR$(0)+CHR$(5)+CHR$(27)+CHR$(91)+CHR$(54)+CHR$(115)
US$=CHR$(27)+CHR$(91)+CHR$(52)+CHR$(109)
UE$=CHR$(1)+CHR$(27)+CHR$(91)+CHR$(48)+CHR$(109)
```

Mannesman Talley 160L

```
PR$=CHR$(0)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(91)+CHR$(53)+CHR$(121)
US$=CHR$(27)+CHR$(91)+CHR$(52)+CHR$(109)
UE$=CHR$(1)+CHR$(27)+CHR$(91)+CHR$(48)+CHR$(109)
```

NEC Spinwriter (all models)

```
PR$=CHR$(0)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(35)+CHR$(65)+CHR$(27)+CHR$(73)
US$=CHR$(27)+CHR$(45)
UE$=CHR$(1)+CHR$(27)+CHR$(39)
```

C-Itoh (Apple parallel printer)

```
PR$=CHR$(0)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(69)
US$=CHR$(27)+CHR$(88)
UE$=CHR$(1)+CHR$(27)+CHR$(89)
```

Apple Imagewriter, Apple Scribe

```
PR$=CHR$(1)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(69)
US$=CHR$(27)+CHR$(88)
UE$=CHR$(1)+CHR$(27)+CHR$(89)
```

Data Products 8050, 8010

```
PR$=CHR$(0)+CHR$(8)+CHR$(0)+CHR$(30)
US$=CHR$(27)+CHR$(102)+CHR$(49)+CHR$(36)
UE$=CHR$(1)+CHR$(27)+CHR$(102)+CHR$(48)+CHR$(36)
```

Hewlett-Packard LaserJet

```
PR$=CHR$(1)+CHR$(0)+CHR$(0)
US$=CHR$(27)+CHR$(36)+CHR$(100)+CHR$(68)
UE$=CHR$(1)+CHR$(27)+CHR$(38)+CHR$(100)+CHR$(64)
```

Qume Sprint 5

```
PR$=CHR$(0)+CHR$(12)+CHR$(5)
US$=CHR$(95)+CHR$(8)
UE$=CHR$(0)
```

Qume Sprint 9, 11, 11 Plus

```
PR$=CHR$(0)+CHR$(12)+CHR$(5)
US$=CHR$(27)+CHR$(73)
UE$=CHR$(1)+CHR$(27)+CHR$(74)
```

Integral Data Systems 480

```
PR$=CHR$(0)+CHR$(8)+CHR$(5)+CHR$(30)+CHR$(27)+CHR$(82)+CHR$(44)+CHR$(49)
PR$=PR$+CHR$(44)+CHR$(36)
US$=CHR$(95)+CHR$(8)
UE$=CHR$(0)
```

Setting Up for Other Printers and New Features of the Printers Already Listed (a section that you should skip over unless you really need it, or are fascinated by inner workings and messy details)

There are four parts to the printer initialization. The first part is control of the printer in general: its line spacing, left margin, and top margin. The second part controls the style of printing (where there is a choice). The third part is how underlining is begun, the fourth part is how underlining is finished.

Control of the Printer in General

Each item in the printer message is just an ASCII code. Because of the way BASIC is designed, each code has to be put between the parentheses in the formula CHR$( ). Then all the codes are strung together one after the other by means of plus signs (as was done in the listings above). The resultant message packet for the printer is called PR$ (PR stands for "PRinter;" and the dollar sign is required by BASIC.)

The first code just tells SwyftCard whether your printer needs a line feed for each carriage return in order to separate lines: "0" means none is needed, "1" means that it is needed. Quite often there is a switch on the printer for this very same purpose, and sometimes there is another switch on the interface card for this purpose again!

If there is a problem, the best procedure is to make sure that the interface card and the printer's switches are both off, and then play with the first code in PR$ until it works right. There are only two choices, so it won't take long.

The second code is the number of spaces the printer will skip over before it begins to print each line. This is best determined by experiment, and by where you like the print to fall on the paper.

The third code is the number of blank lines you want the printer to eject before printing starts. This adjustment accounts for the different distances between the print head and the paper tear-off bar on various printers. It, too, is best decided by experiment and your personal preference.

Example: If your printer keeps on printing on the same line, and the line of print needs to be moved over 5 spaces to the right and 7 lines down from the starting position, then your PR$ would begin

```
PR$=CHR$(1)+CHR$(5)+CHR$(7)
```

This is often enough. These three values are used by SwyftCard to generate the appropriate returns and spaces, which are sent to the printer. The three values themselves are *not* sent to the printer. But some printers require more, so please read on...

Control of Printing Style

Some printers will give you different print qualities (print size, draft or correspondence quality) or styles (italics, Gothic, bold). You set the style by a number of characters *sent to the printer* each time you use the PRINT command. These characters form the rest of the PR$ message that we started above.

You have to first determine what characters the printer requires (from the printer manual), and then translate what it says into ASCII codes.

For example, if it says that for italics you have to send the printer ESC, I, T, then you would translate this into

```
CHR$(27)+CHR$(73)+CHR$(84)
```

***Note to BASIC experts:** you can, of course, use CHR$(27) + "IT" in this example, but for the sake of uniformity we use the CHR$ technique throughout.*

It is easy to find the ASCII codes for any symbol you can type. You can ask SwyftCard: Just put the symbol (letter, number, space, whatever) in between the quotes of

```
?ASC(" ");
```

highlight it, and use the CALC command. The only other one you usually need to know is that ESC translates to 27. If "control" characters are called for, then CONTROL A is 1, CONTROL B is 2, and so on through CONTROL Z, which is 26.

Once you have found the desired print quality code in the manual and translated it into its ASCII equivalent, then you would add it on to the general printer controls. In our example you would have

```
PR$=CHR$(1)+CHR$(5)+CHR$(7)+CHR$(27)+CHR$(73)+CHR$(84)
```

And you are done with the PR$ part.

Controlling Underlining (or Local Font Selection)

If you use underline characters in your text, you have to tell your printer how to handle them. When an underline character (_) is encountered during printing by the PRINT command, SwyftCard sends an "underline start" message to the printer. Unfortunately, each different printer requires its own form of the message. This message is called "US$" (which stands for "Underline Start"; the dollar sign is required by BASIC). The underline character in your text is printed as a space, thus keeping the arrangement of text the same on the printer as it is on the screen.

When a second underline character (or when two returns or a new page) is encountered during printing by the PRINT command, an "underline end" message is sent to the printer. This message is called "UE$" (which stands for "Underline End"). This underline character, if present, is also turned into a space upon being printed.

US$ is a message that tells the printer that underlining will start with the next character. The message is built up just as for PR$, based on information from the printer manual, translated into ASCII codes, using CHR$ as before.

The first code in UE$ tells SwyftCard whether the printer is of the kind that underlines a group of characters all at once (in which case the code is CHR$(1)) or if it is the kind that underlines a character at a time (in which case the code is CHR$(0)). The rest of UE$ is a message that tells the printer to stop underlining. If your printer settings are not included in the list in Part 17, you will need to translate them from your printer manual.

You can use the underline character in text to turn on or off any enhancements that can be coded in five characters. Not only underlining, but any other print feature can be used: italics, boldface, letter quality printing, or even graphics that can be operated by start and stop codes. Though it would be quite a tedious process, you could even print out portions of your text (say, a paragraph at a time), changing the US$ and UE$ messages in between so that you get underlining here, boldface there, and italics in yet a third place in the same document.

Debugging Your Printer

If you've tried everything and you still can't get your printer to work right, try turning off the printer and the computer and starting all over. Sometimes a printer will "remember" the codes you sent it earlier (perhaps while you were trying to get it to work, or while you were using some other program), and will not correctly accept the new information until it has been turned off and on again.

If all else fails, read the manuals, and see if your printer dealer will help. In rare cases the problem will be a malfunction of the printer, the cable, or the cards in your computer. In this case, only a hands-on test by a repair person will solve the problem.

How to Start a New Disk Without Having to Retype Printer Initialization

The following steps allow you to get a new disk ready to go without having to retype the printer initialization message:

With the text of an already initialized disk on the screen, remove the disk from the drive and insert a blank disk.

Highlight and delete the entire text. Type a few spaces, and then use the DISK command.

The printer initialization will be saved on the blank disk along with the Text (a few blank spaces).

Part 18: The SEND Command (Electronic Communications)

The SEND command allows you to communicate electronically either through cables locally or to anywhere in the world over the telephone network. You can communicate with data services, bulletin boards, friends, or any "on-line" system. This command is also your key to sharing information with other Apples (whether they have a SwyftCard or not), and computers both small and large of other brands, so long as they have a standard modem and communications capabilities.

Hardware Requirements

The SEND command assumes that you have an Apple Super Serial Card in Slot 2. While you can communicate directly between two SwyftCard-equipped Apples with serial cards (if they are close enough for you to run a cable between them) this section primarily deals with communications over telephone lines, which means that you also need a modem connected to your serial card.

The modem must, of course, be connected to a telephone line. The modem manual should explain how.

Setting Up the Super Serial Card

There are two banks of switches, labeled SW1 and SW2 on the Apple Super Serial Card. They should be set as follows ("*dn*" means the switch is down or off, "*up*" means up or on):

SW1	*1*	*2*	*3*	*4*	*5*	*6*	*7*
	up	*dn*	*dn*	*up*	*up*	*up*	*up*
SW2	*1*	*2*	*3*	*4*	*5*	*6*	*7*
	up	*dn*	*dn*	*up*	*dn*	*up*	*dn*

The only other setting is a black block with a triangle on it. The triangle should have its point up, pointing toward the word "modem." The Super Serial Card manual explains the use of each switch.

Setting Up the Modem

Apple modems have no settings. The Hayes Smartmodem switches (behind the front panel) may be set:

1	2	3	4	5	6	7	8
dn	*up*	*dn*	*dn*	*up*	*dn*	*up*	*dn*

the Hayes manual explains the use of each switch.

The SEND Command Itself

Compared to setting up, the SEND command is simplicity itself. Type and edit the text you want to send, highlight it, and use the SEND command (USE FRONT D).

When you are having an electronic conversation (that is, where you and the device you are communicating with work alternately, as in a normal human discussion) it is even easier: just type what you want to send, and use the SEND command. It is not necessary to highlight since everything you typed since the last received message will be highlighted and sent automatically when you use the SEND command.

If you use a LEAP key in the course of typing your response, you will have to highlight that response before you send it. Use of character-by-character delete is OK: it will not affect the autoextending highlight.

You can type even while a message which is coming into the text. It will probably be easier to read what you are typing if you type a return first.

Note: *Leaping while a message is coming in is not recommended. As soon as you let go of the LEAP key, the incoming text will begin to spout from where the cursor has landed. This can be confusing if you leap while typing a response to an incoming message.*

The above instructions are all you need to know if you are communicating between two SwyftCard-equipped Apple //e's. Working with other systems sometimes requires a bit more effort.

Systems That Require a Return (or Something Else) at the End of Every Line

Many non-SwyftCard systems require a return at the end of every line. You can do this by simply typing a return at the end of every line before you SEND, but it gets tiresome to have to give both a return and a SEND, and you lose the advantage of automatic word wrap. There's an easier way: if you type, highlight, and use the CALC command on

```
LE%=29
```

then a Return will be sent at the end of every line, and at the end of the SEND as well. This is what most systems expect.

If you do include a return, then SwyftCard will use that return and not add one of its own. When you use LE% in this way, and the system you are communicating with asks you to type a return, just use the SEND command without typing anything. To turn off this feature, type, highlight, and use the CALC command on

```
LE%= -1
```

Expert's Use of LE%
(this section can be ignored by almost everybody almost all of the time)

LE% can be used to send any one or two characters at the end of every line. To send one character, set LE% to the ASCII code for that character. To send two characters (e.g. CR and LF, with ASCII codes 13 and 10) you set LE% to the sum of the first character and 256 times the second character. In the case of CR and LF, you would type, highlight, and use the CALC command on

```
LE% = 13 + 256 * 10
```

If the last character in the line matches either character of LE%, that character is not sent, so you do not get doubled characters.

Internally, SwyftCard uses ASCII 29 for a return, rather than ASCII 13. In communicating with the outside world, of course, the standard value of 13 is used. Thus, if LE% is set to 13, you will get two returns for each one in the text.

Dialing

If your modem will dial for you, read your modem manual to see what message you must send to make it do this. For most Hayes-compatible modems it is important to first set LE%=29. Then you can dial a phone number — say, 555-1212 — by typing, highlighting, and using the SEND command on

```
ATDT 555-1212
```

By putting a dialing code in front of every phone number, your phone number list and SwyftCard can be turned into an automatic dialing machine! Just leap to the name, highlight the dialing code and the number, and use the SEND command. When you hear the phone on the other end ringing, pick up the handset and talk.

Answering the Phone

This, again, is a function of your modem and not of SwyftCard. If you use the settings shown above for a Hayes modem, or you are using an Apple modem, the modem will automatically answer the phone.

Setting Baud Rates
(useful information for many people, though not all)

For each of the following baud rates, type, highlight, and use CALC on the expression on the right. ("SE" stands for "SEnd.")

300	SE%=5641	*(default baud rate)*
1200	SE%=6153	
2400	SE%=6665	
4800	SE%=7177	
9600	SE%=7689	
19200	SE%=7945	

The baud rate you choose depends on the requirements of the system you are communicating with, and what your modem can accept. You should set the baud rate before you begin communications. When you save your disk, the baud rate and all other settings are saved, so that you may never have to do it again.

SwyftCard normally works at 300 baud, with 1 start bit, 1 stop bit, 8 data bits, and no parity. It doesn't matter if you don't understand what any of these things are, just that they match what the service you are using expects. We have found that the choice we have made will work with most services and bulletin boards, and you can ignore the whole business if it works for you the way it comes.

Setting Two Stop Bits and Other Fine Points

(a section for experts only — most people can and should skip this)

The BASIC variable SE% is used to initialize the Super Serial Card for its communication protocol. This operation is analogous to initializing the printer with PR$. By changing the value of SE% and modifying the modem protocol you can communicate with a wide variety of data services.

Here's how to find the value for almost any settings you might want:

First, find the number for the baud rate:

Rate	*Number*	*Comment*
153,600	0	yes, we *do* mean over one-hundred thousand baud
50	1	
75	2	
110	3	
134.58	4	
150	5	
300	6	default
600	7	
1,200	8	getting popular
1,800	9	
2,400	10	
3,600	11	
4,800	12	
7,200	13	
9,600	14	
19,200	15	

Now determine the number for how many bits in each word:

Bits	*Number*
8	0
7	32
6	64
5	96

and add 16 (the 16 sets up the baud rate generator — it's not an option).

Next, determine the number for the desired stop bits:

Stop bits	*Number*
1	0
2	128

Choose your baud rate, bits per word (plus 16), and stop bits. Add their numbers together, then multiply by 256 and add 9 to the result of the multiplication (sounds like the start of a boring magic trick, no?).

Now determine what parity you will use, and find the appropriate number:

Parity	*Number*
none	0
odd	32
even	96
mark	160
space	224

Add this last number and you have the value for SE%. Example: The default is 300 baud, which gives us a number 6 from the table for baud rates. We are using an 8-bit word, so we add 0 (from the Bits/Number table), plus 16 for the baud rate generator to get 22. One stop bit adds zero, so we multiply 22 by 256 and add 9, giving us 5641. We are done since the default is no parity.

Unfortunately, we have one more step to do if the calculated value is greater than 32767. For example, if we wanted two stop bits we would have come out with 38409. So we subtract 65536 from any value we get over 32767. In this case we would calculate 38409 - 65536 and would type, highlight and use the CALC command on

```
SE%= -27127
```

All this is made much easier by having the CALC command available to do the arithmetic in the first place.

Designer's note: *We could have given you a long menu to run through to choose all these parameters, but by doing it the way we have we give experts direct access to the registers of the Serial Card's 6551 chip. This would otherwise have required menus with choices like, "Do you want the Mark Parity bit transmitted and the Parity check disabled?" and so on. Also, experts may note that if they add 13 instead of adding 9 to the product above, they will be able to send a break by setting SE%. To stop the break signal you will have to reset SE%. Fortunately, most systems no longer require breaks.*

Garbled Electronic Communication

If occasional characters are garbled, it is probably due to a poor phone connection. If you are getting total gibberish, the most likely problem is that your setting of SE% doesn't match the other person's setting.

Dealing With the Outside World — a Hint or Two

If you are using a data service or bulletin board, it may have rules of its own that have nothing to do with SwyftCard, and this manual cannot help you there. Some of those external services have messages that will guide you through. We will only give one set of hints that we have found useful — hints which work most of the time: Starting with an empty text, first set LE%=29 as explained above, then delete it. Now dial (by hand or with the modem) the telephone number of the service you want to use. We hope your modem has a speaker so that you can listen to the dialing, the ringing, and then hear two tones, one after the other. If information does not appear on your display within a few seconds, delete everything so that the text looks empty and use the SEND command two or three times. This often gets things started.

Lastly, once you get on, remember that you can leap to instructions and command tables that you have seen go by on your screen; you can even print them out. Sometimes the cursor will stop blinking after a print. Just hold down the CONTROL key, and, while you hold it, press and release the RESET button and everything will be fine.

***Personal note from the authors:** We've run into a lot of services and bulletin boards that confused us with their commands, so don't think that it's you if you have trouble. Many data services are very poorly designed, and nobody can figure them out without some help or a lot of experimentation.*

Control Codes

Some bulletin boards or data services might ask you to use "Control Characters." For example, they might say, "To access the next message, type CONTROL Y." Sometimes they use a carat (^) instead of the word "control," so that the instruction might be "type ^Y". To do this with SwyftCard, press and hold the USE FRONT key, and, while you are holding it, tap first the letter "z" and then the letter asked for — a "y" in this example.

If you are asked to type an escape code sometimes called ESC), press and hold the CONTROL (USE FRONT) key, and,.while holding it, tap first the letter "z" and then the ESC key (labeled "PAGE" in the SwyftCard system).

You do not use the SEND command to send control characters. Use the technique we have just explained.

Receiving Information From the Modem

When information comes into the SwyftCard, it acts just as if another person were typing. All the things you can do with text that you create can be done with text you get from the modem. On the other hand, text can become full when receiving just as when typing, and the same thing happens, the Apple bleeps and tries to tell the system sending information to stop.

We say "tries to tell" because not all systems use the same method. But SwyftCard uses the most common technique, where a special code means to stop sending, and another special code means to start sending again. If incoming information does fill up text, do what you would do if typing had filled up text (see Part 21, "Bleep When You Are Typing," and Part 11, "When You Run Out of Room").

Stopping Incoming Text and Restarting It

Text coming from another SwyftCard or from most data services can be interrupted by a CONTROL S. As explained above, press and hold CONTROL, and while holding it tap "z" and then "s".

To let the text continue, you use a CONTROL Q: press and hold CONTROL, and, while holding it, tap "z" and then "q".

This use of CONTROL S and CONTROL Q is sometimes called by the mysterious name *XON-XOFF protocol.*

How to Wake Up the Person at the Other End of a Conversation

You can try to make the terminal (or a SwyftCard-equipped Apple //e) at the other end bleep by sending a CONTROL G. Similarly, if SwyftCard gets a CONTROL G, it makes your Apple beep. It is customary to send three or more beeps for this purpose (hold CONTROL and type: zg zg zg).

Using Your SwyftCard as a Message Center

If your modem can answer the phone (auto-answer) then — if you leave your system turned on and the modem hooked up — people can leave messages for you by sending text from their system. However, they cannot operate your computer remotely through SwyftCard or get any information out of it — it is a very secure system.

You should tell those people who might send you messages that, unless they send a CONTROL E to your SwyftCard, they will get no reply nor will they have any way of confirming that your system is working. When they do send a CONTROL E, your SwyftCard will automatically send back

```
SwyftCard 1066 //e
```

(the number may be different if they have a different version of SwyftCard). This way they know that your system is operating correctly.

Other Uses of SEND

The SEND command may be used with serial devices other than modems. For example, by attaching a Votrax speech synthesizer to the Apple Super Serial Card, you can have the messages that you send spoken for you.

Chapter V

Problem Solving and Information Retrieval

Part 19: Stopping

To Stop Printing

Press and hold the USE FRONT key, and, while you hold it, press and release the RESET key. The unprinted text will remain highlighted. To begin printing again from the point you left off, use the PRINT command.

To Stop a BASIC Program

Press and hold the CONTROL (USE FRONT) key, and, while you hold it, press and release the RESET key. The program will halt and the text generated by the program will be highlighted. If you now use the CALC command, the program will begin to run again *from the beginning.*

Certain errors in the typing of calculations, printout format changes, or printer initialization strings can cause the Apple to bleep. An on-screen message, "?SYNTAX ERROR", pops out when you make such a mistake. If you accidentally use the CALC command on a large chunk of highlighted text, the computer may start hiccuping:

```
?SYNTAX ERROR          (bleep!)
?SYNTAX ERROR          (bleep!)
?SYNTAX ERROR          (bleep!)
?SYNTAX ERROR          (bleep!)
```

CONTROL RESET will stop the hiccups, or you can just wait for them to stop.

Stopping an Incoming SEND

When communicating with another SwyftCard-equipped Apple or with most data services, the following techniques will stop an incoming send:

It is usually possible to temporarily stop an incoming transmission by holding down the CONTROL key and pressing and holding the Z key for as long as you want it to stop.

To temporarily stop a transmission without having to continuously hold down a key, hold down the CONTROL key, and, while you hold it, tap first Z and then S. (To restart a transmission that you have stopped, hold the CONTROL key and tap first Z and then Q).

If XON-XOFF does not work for you, refer to the manual for the system with which you are communicating.

Stopping an Outgoing SEND

Use CONTROL RESET.

Note: *This will stop the outgoing SEND, but it will cause your modem to drop its carrier, usually causing the other person's modem to hang up.*

Part 20: Emergency Restart

Frozen Cursor

If the cursor stops blinking and does not respond when you type, press and hold the CONTROL (USE FRONT) key, and, while you hold it, press and release the RESET button at the upper right corner of the keyboard. You will *not* lose any text. The cursor freezes during the time a BASIC program is running. An error in a BASIC program, certain BASIC statements, or starting a SEND or PRINT when the modem or printer is turned off can also cause the cursor to freeze.

The cursor will stop moving but continue flashing when the text is full. This is not the same as a frozen cursor. When the text is full, the Apple will bleep every time you tap a key (see Part 21, "Bleep When You Are Typing," for the remedy).

Overriding the DISK Command

Normally we trust the DISK command to save text or load a new disk as needed. You can sidestep the DISK command and its safety features if you wish, but not without risk: if you haven't saved the text on-screen, you may lose valuable changes when you force SwyftCard to load the disk in the drive. We call this procedure *restarting*. You should avoid using it. If you begin to rely on this technique instead of the DISK command, eventually you will clobber yourself with it — that is to say, you'll accidentally lose some valuable text. A better procedure is described under "Force Loading" in Part 11.

Restarting consists of starting the disk drive and loading whatever information is on the disk: the drive whirrs, the on-screen text vanishes, and SwyftCard loads the disk in the drive. If there is no disk in the drive, SwyftCard will start up with an empty text: page characters 0 and 1, the cursor and highlight, and that's all (this screen is illustrated at the end of Part 4).

Here's how it goes: Press and hold the CONTROL key and Open-Apple, and, while you hold both these keys, press and release the RESET button at the upper right corner of the keyboard. Then release CONTROL and Open-Apple.

You can also override the DISK command by turning the power to your Apple off and then back on.

WARNING: *Whenever you override the DISK command the on-screen text is lost forever. If you don't intend to lose the on-screen text, make very sure that you first have the text safely stored on a disk.*

Part 21: Bleeps

A bleep is an audible signal from the Apple. It signifies a refusal or inability to carry out a command, or it is a warning.

Bleep When You Are Typing

1. The most common reason by far is that you have run out of room. This can happen by typing characters, or by tabbing (which is the same as typing spaces). A bleep sounding on every keystroke and the cursor not moving is another sign.

2. An attempt to add a page character to the text when there are more than 200 page breaks or page characters already present will cause a bleep.

If the problem is lack of room, you can sometimes get more by highlighting a couple of characters and deleting them. Remember that SwyftCard saves the last chunk of text you deleted, and this saved chunk consumes space. By deleting two characters you gain back the space (less the room for the two characters).

In general, the best thing to do is to start another disk. If what you are working on is just one item of many, or you need room to edit the rest of the text, use copying up to transfer part of the Text to a fresh diskette (this procedure is explained in Part 11, "When You Run Out of Room").

Bleep When You Use the INSERT Command

When you INSERT a chunk of text and the system bleeps, this warns you that there is no longer room to INSERT the same chunk again. In fact, even if you deleted enough text character-by-character so that the chunk last INSERTed would seem to be able to fit, you will find that there is nothing to insert (but nothing was lost — you INSERTed the chunk into your text, so it is available for further use.)

If you use INSERT to make additional copies, you are creating additional text, just as if you had typed the new material, and you may run out of room. See "Bleep When You Are Typing."

Bleep When You Use the DISK Command

1. There is no diskette in the drive. Oops — put one in.

2. The drive door is open. Close the door and try DISK again.

3. You've put a new disk in the drive without saving the changes in the text on-screen. If SwyftCard does a save, you will lose whatever is on the new diskette. If it brings in the Text from the new diskette, then you will lose your changes. SwyftCard doesn't want to lose any text no matter what, so it bleeps in frustration.

   You can do one of two things: a) Put back the diskette that belongs with the on-screen text and use the DISK command to save the changes, then put in the new disk and use the DISK command to bring it on-screen. Or, b) if you are absolutely sure you don't care about saving your changes, put the new disk you want to load in the drive, then — if you're absolutely certain you don't want to save the changes — DELETE everything (leaving only page characters 0 and 1 on the screen), and use the DISK command.

4. The disk is write-protected and SwyftCard tried to do a save. Take out the disk, and if there is a write-protect tab remove it.

5. The disk is damaged.

Bleep When You Copy Up

If there isn't enough room in the receiving Text for the material you are trying to copy up, SwyftCard will bleep. The fix is to copy up less, or to make more room by deleting some information from the receiving Text.

Bleep When You Use the CALC Command

1. A calculation can generate too many characters to fit in the computer, just as typing can. Running a BASIC program with the CALC command can do the same thing. In either case, SwyftCard will bleep. Split the Text onto two disks, or make room by deleting part of the text.

2. Certain errors in the typing of calculations, printout format changes, or printer initialization strings can cause the Apple to bleep. An on-screen message, "?SYNTAX ERROR", also pops out

when you make such a mistake. If you think you've committed a simple typing error, just start over. Make certain you have typed the expression exactly as it is written in this manual.

This bleep can also occur if you press the CALC command by mistake — when you intended to PRINT or SEND, for example.

When You Use the SEND Command

1. If the person sending you text sends more than SwyftCard's capacity to receive and ignores SwyftCard's request to stop sending, you get a bleep. Every character that overflows the limit will cause a bleep, and since the characters are sent at a high rate of speed by the modem, the bleeping sounds something like an alarm clock going off.

   The bleeping can be stopped by pressing and holding the DELETE key until the transmission quits, or by a single press of DELETE once the transmission is over.

2. If the sender sends a CONTROL G, the Apple will bleep. The method of sending a CONTROL G from SwyftCard is to hold down the CONTROL key and tap first Z and then G.

Part 22: Information Retrieval and Other Applications — a Few Ideas

"Information retrieval" is a five-dollar term that means nothing more than being able to find what you want after you've tucked it away somewhere. The technique of leaping is not only a means of moving the cursor, but of moving the cursor to a particular item in the Text — hence leaping *is* information retrieval.

Suppose you have a text consisting of people's names and telephone numbers, like this:

```
678/526-1234  Harold DaSilva, 34 Dusty Rd.          San Palinicia, CA
777/352-9666  Theodore Chiesa, 14900 Eightfold Way  Sinkerville, TX
292/446-2956  Peter's Pizza, 12 Della St.           Oleeklee, OK
892/501-1841  Patricia Lewis,  1200 Appellate Ct.   Tavistock, CT
292/101-8632  Livestock Protective Assoc.           Two Hats, OK
                Avery Thixill, pres.  292/102-2342
                M. Violet Blanc, treasurer 292/446-1202
                Michael Lovkovich, cowhand
498/184-3330  Fandango Frank, 5622 Dress Circle     Wobston, MA
            Manuel Crozier Heintz, 82 Camino Cerrado
292/615-9235  Kim Mascoli,  11206 Lois Ln.          Mantis, OK
555-1252      Doctor Fenton Farnsworth
398/783-8114  David Barnett, 6266 Warp Drive        Stony Shoe, MA
292/152-6945  Calixto Skelton, 626 Wrong Way        Spoon River, OK
```

and so on (these numbers and names were made up). Say that you keep such a phone list in the Text you are currently using. Whenever you want to look up somebody's number, all you have to do is leap to their name. Notice that you can leap to their first name or last name, or to their street address or street name, or even their town or state. With at most a few taps of the LEAP AGAIN key, you are sure to find them.

You can use such a list in other ways, too. For example, to find all the people you know in Oklahoma, leap to "OK" and use the LEAP AGAIN key. To find your doctor's phone number, just leap to "doctor".

Let's say that you had a few letters in the same Text as your phone numbers. You could find the letters by leaping to "Dear", and with each press of the LEAP AGAIN key you would be able to look at another letter. It won't take more than a second or two to find the letter you're looking for.

Even faster — at least most of the time — is to look for a key word *inside* the letter. If you want to see the letter you wrote a while ago about a new brand of hula hoops you discovered in a local store, you could leap to "hula" or "hoop", and there would be the letter you wanted.

Similarly, if you had typed in a lot of recipes, you could not only look up a recipe for, say, chile con carne by leaping to the name of the recipe, but if you had picked a dozen zucchinis in your garden, you could search for all recipes with zucchini in them just by leaping to "zucchini" and using the LEAP AGAIN key.

Clever arrangement of items can make certain kinds of retrieval easier. Take the phone list, for example: a leap to "Livestock Protective Association" will simultaneously give you the names of the officers. Similarly, a leap to any of the people who work for the Association will turn up the Association phone number, plus the names of all the other principals of the group. Notice that with SwyftCard, information does not have to be complete: not all names have addresses, nor need they all have phone numbers. You might remember only a first name, or part of a last name — SwyftCard doesn't care. In fact, you don't even have to line up all the names and numbers as we've done here. They can be almost helter-skelter on the page and you can still find them.

SwyftCard can be used to do inventories, mailing lists, and many other applications, all without programming or learning anything beyond typing, leaping, and elementary editing.

"Mail Merge" Applications

If you want to send the same letter to a number of people, you might start by arranging their names and addresses like this:

```
Paul Conroy
21 Easy St.
Soy, NY 12534
Dear Paul,
\
Andrew Johnson
1066 Swyft Wy.
Sundapp, ND 73453
Dear Andrew,
\
```

and so on. Then you would type the letter, highlight it, and delete it. Next you'd leap to a backslash (\), delete the backslash, and use the INSERT command. This inserts the letter under the address and greeting. Highlight and print the letter, then do the same thing for each address. It's a lot easier than doing it by hand. If you had to do it for a hundred letters every week, this would be too tedious, and you'd probably be well advised to get a conventional "mail merge" program. The SwyftCard <—> ProDos converter might be used to move your address list to a ProDos file for this purpose.

We'd like to show one more application that uses BASIC and information retrieval. Say that you'd like to make up a calendar for April — which hath 30 days. If you know BASIC, you might write a little program, like this:

```
10 PRINT
20 FOR I = 1 TO 30
30 PRINT "APRIL  "; I
40 NEXT I
RUN
```

highlight it, and use the CALC command. You will get:

```
  APRIL  1
  APRIL  2
  APRIL  3
  APRIL  4
```

...and so on, up to...

```
  APRIL  30
```

and now by leaping to any given date, you can make entries in your calendar — or find out what's happening on a date you've already filled in. And you can, for example, find out when your friend's birthday is (assuming you've put it into the calendar) merely by leaping to your friend's name. You could find all birthdays by leaping to the word "birthday" and using LEAP AGAIN.

With experience you will find more and better uses for SwyftCard. The wealth of applications is endless.

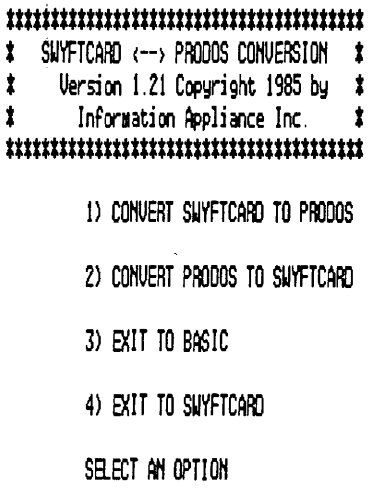
Appendices

Appendix A: ProDOS Converter

The ProDOS converter program is included in the SwyftCard pack, on the back of the SwyftCard Tutorial disk. If you wish to convert some SwyftCard text to a ProDOS text file, you must first highlight the text you wish to convert and then use the DISK command. If you wish to convert a ProDOS text file to SwyftCard text, you will need a disk to store the converted SwyftCard text.

When you are prepared, place the SwyftCard/ProDOS converter disk in Drive 1 and use the DISK command (if you are using SwyftCard). If the computer is off, the SwyftCard ProDOS converter will run when the system is turned on. If you are running other Apple //e software, re-boot the system using the proper procedure for the program you are running.

After a few seconds you will see the SwyftCard/ProDOS main menu:



```
***************************************
*  SWYFTCARD <--> PRODOS CONVERSION   *
*    Version 1.21 Copyright 1985 by   *
*      Information Appliance Inc.     *
***************************************

         1) CONVERT SWYFTCARD TO PRODOS

         2) CONVERT PRODOS TO SWYFTCARD

         3) EXIT TO BASIC

         4) EXIT TO SWYFTCARD

         SELECT AN OPTION
```

Select the option you need by typing a number from 1 to 4. No return is required. You can get back to the main menu by pressing the ESC (PAGE) key when any input is asked for. Here is what the four options do:

1. Convert SwyftCard to ProDOS

This option will convert the highlighted text on a SwyftCard disk to a ProDOS text file. We assume you have a SwyftCard disk with the text you want to convert to a ProDOS text file highlighted. As soon as you type the "1", the SwyftCard-to-ProDOS screen will appear and ask you to insert the SwyftCard disk in Drive 1 and press the RETURN key. Do so.

The next screen will ask you for a pathname. Type the name of the file to which you want to transfer the converted SwyftCard text. When the conversion is complete, the display will return to the main menu.

Since the SwyftCard text is stored on the disk located in Drive 1, the pathname you specify cannot be a disk in Drive 1. If you have a single drive system and an extended 80-column card, you can transfer the SwyftCard text to the volume /RAM — for example, to a file named /RAM/TEMP. To store the text on a ProDOS disk, you will have to transfer the converted file from /RAM/TEMP to another disk.

The conversion process does not affect the SwyftCard disk in any way.

If the text to be converted is too long for the resulting file to fit on the volume specified in the pathname, an error message will appear and as much of the text as will fit in the ProDOS file will be transferred.

If the ProDOS file you name does not exist, it will be created. If it already exists, the SwyftCard text will replace the contents of the file.

2.Convert ProDos to SwyftCard

This option will convert a ProDOS text file to a SwyftCard disk. When the "2" is typed, the ProDOS-to-SwyftCard screen will appear, asking that you insert a blank disk in Drive 1. Any non-write-protected disk will do; the converted SwyftCard text will be written to the disk. Once you have inserted the disk in Drive 1, press the RETURN key.

WARNING: *The disk will not be checked! Anything it contains will be lost.*

Next, the screen will ask for a pathname. Type the pathname of the ProDOS text file that you want converted to a SwyftCard disk. When you press the RETURN key, the conversion will take place and the screen will return to the main menu screen.

To complete the conversion, choose option 4, "EXIT TO SWYFTCARD". When you have returned to the SwyftCard environment, use the DISK command so you can see the converted text. Now type and highlight

```
CALL 3600
```

highlight what you have typed, and use the CALC command. This clears the disk. Tap the DELETE key to eliminate the CALL 3600 from the text. Now use the DISK command to save. The conversion is complete.

If the disk is write-protected or the ProDOS file is too large for the SwyftCard (more than 40K bytes) an error message will appear. Tap the ESCAPE (PAGE) key to return to the main menu, and repeat the process with a non-write-protected disk, or with a smaller ProDOS file.

3. Exit to BASIC

This option returns the Apple //e to the BASIC prompt with ProDOS running.

4. Exit to SwyftCard

This option runs the SwyftCard system. The system will begin the normal SwyftCard startup. If the converted SwyftCard disk is still in Drive 1, the SwyftCard text will be empty. The DISK command will bring in the converted ProDOS text file (see 2, above).

To convert AppleWorks or DOS 3.3 files to SwyftCard, first convert them to a ProDOS text file. See Apple's manuals. Apple's ProDOS Filer and ProDOS DOS 3.3 converter programs are included on the SwyftCard/ProDOS converter disk.

Appendix B: Theory of Operation

This Appendix is divided into three sections: interface, software, and hardware. The sections are not system documentation, but guides to the spirit of design in each area.

SwyftCard User Interface Theory of Operation

The paradigms used in SwyftCard were invented to cure a host of problems shared by almost all current systems, most of them small enough in their own right, but which taken together make learning and using conventional software far more time consuming than necessary: and make using computers a frustrating and annoying process.

We have always wondered why, for example, you have to format disks — isn't the computer smart enough to see if a disk isn't formatted and do it if necessary? We find cursor control keys far too slow, and when you consider the number of auxiliary commands they require (move to next/previous word, sentence, paragraph, page, move to beginning/end of line, document, file...) we find them too complex. The mouse, we find, is small improvment since it takes your hands away from the keyboard, and uses up much screen space for menus, scroll bars, and the rest of the associated mouse apparatus. We are annoyed when we are put through menus instead of being able to do what we want right now, and we are puzzled by the huge number of commands in most systems. We hate disk systems that allow you to lose work through trivial human error. We are amazed that many word processors can't keep up with human typing speed.

SwyftCard shows that, with proper design, all these questions and bothers, and many others that have plagued us for years, can be answered and fixed. And it works on an inexpensive computer with only one disk drive, with minimum memory requirements. Our product does what most people need, without operating system, expensive price tag, or bells and whistles.

The major design principles include a few innovations, and many examples of applying what we have learned from the work of others.

1. The cursor LEAP concept, whose average time to target is about one third that of the most advanced method in common use up to now: the mouse.

2. The cursor design itself, which shows both where what you type will appear and where delete will operate. The cursor also collapses upon being moved so that you do not have to aim "one off" if you want to delete.

3. Finding a very small set of fundamental operations that allow you to easily accomplish a wide range of tasks.

4. Eliminating the operating system and allowing all operations to be performed directly and immediately from the editor without having to go into different modes.

5. The elimination of modes in general, which makes habit formation easy since you do not have to think about what state the system is in to figure out what a command will do. This property is called "modelessness."

6. *Not* providing many ways to do a task — again so that you do not have to think about alternate strategies when you are about to do something. This, too, aids in habit formation. We call this principle "monotony."

7. The emphasis on habit formation is itself a fundamental principle of the design, and one often overlooked by others. We consider it important that after a brief period of learning, a user should not have to think about the system while using it.

8. The DISK command, which simplifies the usual complexities of a DOS (Disk Operating System) into a simple command. It also provides protection against most common mistakes that on other systems would cause loss of data. This kind of command is possible due to the technique of making one disk correspond to one text.

9. The emphasis on speed of operation proportional to frequency of use (often done tasks must be very fast, seldom done tasks can be slow).

10. What you see is what you get — the way it looks on-screen is the way it prints on paper. (This was violated for underlining due to a l imitation in the Apple display hardware.)

11. Noun-verb design of commands. You always specify what you are going to work on first (which gives you time to make sure you are right and to make corrections), and *then* you give the order as to what to do. Some systems work the other way around, or — what is even worse — mix the two styles.

12. It is very hard to louse yourself up or clobber something you are working on. Not impossible, but hard enough to do that it is not likely to happen by chance.

13. The inclusion of programming and communications within a general purpose environment, where the output is placed in the editor/retrieval environment.

14. The allowance of months of testing and re-working time in the schedule, so that purchasers of the system are not being used as test subjects.

This is only a barest sketch — the system specs run to some 50 pages — but we hope it gives you a feel for what led us to design SwyftCard the way we did.

SwyftCard Software Theory of Operation

The system is small and operates quickly partly because it is implemented in FORTH and assembly language, and partly because it has an inherently clean and simple design. There are few commands, and they operate uniformly. Text is not cluttered with special markers. All of this minimizes programming effort.

The structure of the text, although not unique to SwyftCard, is key. The beginning of the text (from the first character up to the last character of the highlight) is stored in the lower portion of the text area, and the end of the text (from the character at the cursor or one after the cursor to the last character of the text) is stored in the upper end of the text area. Between the upper and lower texts lies the gap. This means that typing just puts characters into the gap, and thus can proceed very rapidly. Updating the screen is the only task that need be done while typing is going on, which is one reason why it is possible for SwyftCard to do word wrap and unwrap on the fly. The other reason is the uniformity of text.

Searching is fast because text lies entirely in RAM, contains no codes or other obstructions, and is in only two contiguous areas. The disk operates quickly because we do not use Apple's encoding scheme, and because we only write or read as much text as necessary. As a consequence of this decision, a ProDos conversion routine is provided to establish a link to other Apple software. Formatting is done on the fly, since we write a whole track at a time, including sync bits. Not only is this fast, but it eliminates the need for a separate formatting step on the part of the user.

The system pointers are stored on disk so that SwyftCard texts come up in the exact state they were last saved. A serial number is written on each disk so that we can detect whether it is the same disk that was booted or if the user has changed disks. When backup disks are made, the same serial number is written on the master and all backups.

Updating the display after a leap is sped up by having a table of pointers to the places where pages begin. Thus, in order to figure out how the text should be formatted, the display algorithm has only to go back to the nearest page break prior to the text that is to be displayed at the beginning of the screen.

When inserting text, large areas of memory may have to be moved bodily. This "brute force" approach is surprisingly fast, but with very large texts does lead to a perceptible slowing. Still, SwyftCard is much faster than any other program that does a similar task on the Apple.

A deletion puts the deleted text at the beginning of the text area and moves the lower part of text up out of the way. This means that there is no limit to the amount of text that can be deleted, since we do not have to set aside room for a separate delete buffer.

Decisions of this sort abound, ultimately leaving an unusually large amount of space for the user — in this case over 80% of the memory that Apple does not dedicate to specific uses. SwyftCard does not use the extra memory afforded by the extended 80-column card, as the bank switching required would make the program operate too slowly for the high-quality interaction we think important.

The CALC, PRINT, and SEND command are all fundamentally the same. They take the highlighted chunk of text and transmit it to the BASIC interpreter, the printer port, and the serial port respectively.

Of these, dealing with the BASIC interpreter is the most difficult, since SwyftCard operates in the same address space as BASIC, and because of Applesoft's documentation.

The keyboard tables are in RAM so that software developers can redirect commands to execute code that they provide. BASIC programs that amplify SwyftCard's abilities can also be written. By changing the value of one word (BT%), the bottom of text can be moved up so that developers can have room for their own code.

SwyftCard's software amounts to less than 16 Kbytes. Approximately half is in a tokenized FORTH, and half in assembly code.

SwyftCard Hardware Theory of Operation

The SwyftCard is a plug-in card for the Apple //e that operates in Slot 3. The card contains three integrated circuits which provide a power-on reset circuit, storage for the SwyftCard program, and control signals for the card. The card operates by asserting the Apple //e bus signal INH' which disables the built-in ROM and enables the SwyftCard ROM. This permits the SwyftCard program to take over the system at power-on and run the SwyftCard program. (Please refer to the schematic.)

The LM311 voltage comparator is connected to provide the power-on reset function. When the Apple //e is first turned on, the power-on reset circuit resets the PAL, turning on the SwyftCard and disabling the Apple //e internal ROM. The power-on reset circuit must be provided because the existing Apple //e reset function is used by many Apple //e programs for a "warm start": if Apple //e reset always started the SwyftCard, other programs could not use the "warm start."

The 27128 PROM is used to store the SwyftCard program. The PROM contains 16384 bytes which are mapped into the address space $D000 - $FFFF. Since the address space is only 12 Kbytes, there are two 4 Kbyte sections of the PROM mapped into the address space $D000-$DFFF.

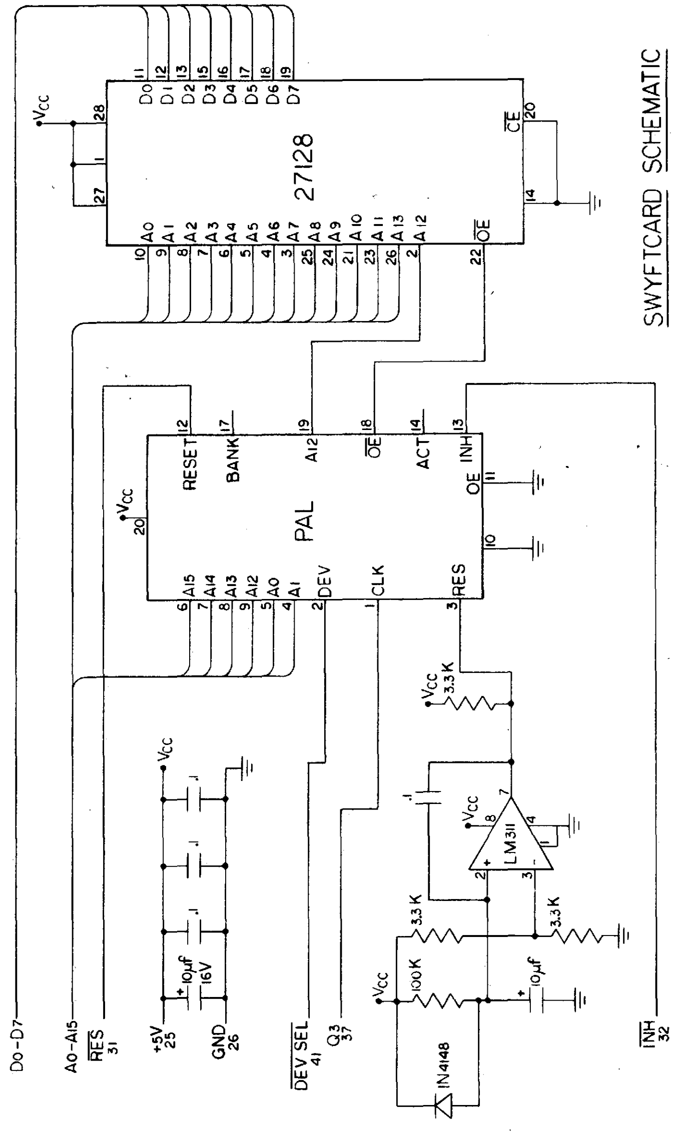
The card is controlled by the PAL. When the SwyftCard is active, the PAL asserts the INH' signal, enables the PROM, and bank switches the $D000-$DFFF address space. The card is controlled by two soft switches. The soft switches are controlled by accessing the following memory locations with either a read or a write operation.

$C0B0 - SwyftCard active, Bank 1
$C0B1 - SwyftCard inactive, Bank 1
$C0B2 - SwyftCard active, Bank 2

When the power-on reset circuit asserts the RES signal on Pin 3 of the PAL, the SwyftCard is made active in Bank 1. Accessing location $C0B1 deactivates the SwyftCard for normal Apple //e operation.

The INH' line is driven by a tri-state driver, so if another card in the Apple //e asserts the INH' signal there will not be a bus contention. However, there will be a bus contention on the data bus if another card attempts to control the bus while the SwyftCard is active.

SwyftCard Schematic



Appendix C: List of Features

1. ***Cursor***	Blinking object on screen
1.1 *Narrow cursor:*	cursor and highlight overlap
1.2 *Wide cursor:*	cursor and highlight adjacent
2. ***Highlight***	Solid rectangle on screen
3. ***LEAP BACKWARD***	Open-Apple (held while typing pattern)
4. ***LEAP FORWARD***	Solid-Apple (held while typing pattern)
5. ***LEAP AGAIN***	Leap and TAB
6. ***Creep backward***	Open-Apple (tapping)
7. ***Creep forward***	Solid-Apple (tapping)
4. ***Extend highlight***	Open-Apple and Solid-Apple together
8. ***DELETE backward***	Wide cursor and DELETE key
9. ***DELETE forward***	Narrow cursor and DELETE key
10. ***INSERT***	USE FRONT A
11. ***DISK***	USE FRONT L
12. ***COPY UP***	USE FRONT L and highlighted text
13. ***PRINT***	USE FRONT N
14. ***SEND***	USE FRONT D
14.1 *Stop reception:*	USE FRONT Z S
14.2 *Resume reception:*	USE FRONT Z Q
14.3 *Send bleep to receiver:*	USE FRONT Z G
14.4 *Test to see if receiver is a SwyftCard:*	USE FRONT Z E

15. ***CALC*** USE FRONT G

16. ***CALC commands*** *(require numerical input)*

No.	***Name***	***Default Value***	***Function***
16.1	MA%	0	Left margin of printer
16.2	LP%	3	Distance from bottom of page to page number
16.3	PL%	66	Lines per page
16.4	AB%	6	Lines in top margin
16.5	BE%	6	Lines in bottom margin
16.6	WI%	80	Characters per line
16.7	PF%	2	Lowest page number to be printed
16.8	PA%	0	Number of first page character in text
16.9	RO%		How many characters to go before full text
16.10	SP%	1	Line spacing
16.11	LE%	-1	Characters transmitted at the end of ever line during a SEND
16.12	SE%	5641	A value that controls most modem parameters
16.13	PR$	0,0,5,27,77, 27,108,8	Printer initialization (up to 29 characters
16.14	US$	27,45,49	Printer underline on (up to 5 characters)
16.15	UE$	1,27,45,48	Printer underline off (up to 5 characters)

Keyboard

17. ***Cast of Characters***

17.1 *Lowercase letters:*
a b c d e f g h i j k l m n o p q r s t u v w x y z

17.2 *Uppercase letters:*
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

17.3 *Numerals:*
0 1 2 3 4 5 6 7 8 9

17.4 *Punctuation:*
, . ; : ' " - ? !

17.5 *Special symbols:*
@ # $ % ^ & * ( ) = + [ ] { } \ | < > / ‘ ~

Underlining:

The first underline character prints a space and turns on underlining. Underline continues until you type another underline character, type two returns, or come to the end of the page (page break or page character). The second underline character also prints as a space.

17.7 *Format characters*

	Character	*Key*	*Function*
17.7.1	Space	SPACE BAR	Blank space between characters
17.7.2	Tab	TAB	Inserts several spaces at a time
17.7.3	Return	RETURN	Ends line, starts new line
17.7.4	Page	ESC	Ends page, starts new page
17.7.5	Page break		Shows end of page (not a character)

18. ***Other keys***

18.1 *USE FRONT:* Activates the commands designated on the fronts of keys.

18.2 *CONTROL:* Same as USE FRONT. Used in conjunction with Apple keys and RESET

18.3 *DELETE:* Erases whatever is in the highlight

18.4 *SHIFT:* Activates uppercase letters and symbols

18.5 *CAPS LOCK:* Switches all letters to uppercase. No effect on symbol keys.

18.6 *Arrow keys:* Nonfunctioning in SwyftCard

19. ***Emergency stop: CONTROL-RESET***

20. ***Emergency restart (emergency boot) CONTROL-Open-Apple-RESET***

Appendix D: Reassigning the Key Commands (for Experts Only)

SwyftCard allows you to change which keys perform which commands. There is generally no need to use this feature, but software developers who wish to add features to SwyftCard may find it useful. They might design a new command and assign it to USE FRONT R, for example.

But even if you are not a software developer, you can move the PRINT command – USE FRONT N – to USE FRONT U or USE FRONT P, if it suits you. In fact, you can move it to the DELETE key or to any letter of the alphabet. You can also move a command to USE FRONT 6, -, [,], or \ ..

You can even assign commands to the four "cursor control" or "arrow" keys. This is because pressing a cursor control key is exactly equivalent to holding down USE FRONT and pressing certain letters.

Why do we have this somewhat strange list of keys to which you can assign a command? It is a function of the way the Apple //e is designed, not a whim on our part. For example, in bygone days letter keys routinely doubled as cursor control keys, and the //e, among other systems, retains this wiring. That is why you can assign commands to the four arrow keys (and also why the N key rather than the "home row" key "J" is used for the PRINT command).

Talking to the Apple About Keys

To explain to the computer what you want to do with a key, you have to refer to the key by number. The following chart shows the keys that are available for commands and their numbers. Sometimes two different keys have the same number (this is indicated in the chart). These redundancies are usually quirks in the Apple.

Keys Available for Commands and Their Numbers

DELETE	5536	
a	5537	
b	5538	
c	5539	
d	5540	
e	5541	
f	5542	
g	5543	
h	5544	(this is the same as the left-arrow cursor key)
i	5545	(this is the same as the TAB key)
j	5546	(this is the same as the right-arrow cursor key)
k	5547	(this is the same as the down-arrow cursor key)
l	5548	
m	5549	
n	5550	
o	5551	
p	5552	
q	5553	
r	5554	
s	5555	
t	5556	
u	5557	(this is the same as the up-arrow cursor key)
v	5558	
w	5559	
x	5560	
y	5561	
z	5562	
[	5563	(this is the same as the ESC key)
\	5564	
]	5565	
6	5566	

Talking to the Apple About Commands

The commands are also numbered. As their names imply, some of these commands are for technical specialists. You should only play with commands you are comfortable with, and don't feel put off if there are some that don't mean anything to you.

Commands and Their Numbers

DELETE	1
INSERT	7
SEND	5
CALC	8
PRINT	2
DISK	4

Software developers may find some of the following routines helpful.

GET	6	This routine inserts the highlighted text of whatever disk is in drive 1 into the Text.
NOOP	0	Does nothing, but returns control to SwyftCard afterward.
MAGIC	29	A location that restores the normal command- key assignments when set to zero.
MONITOR	9	Gives control to the Apple monitor program.
CONTROL RESET		Gets you back if you haven't messed around too much while in the monitor.
TYPE	28	
CONTROL	12	The next character to be typed after this command is executed will be SENT as a control character.

How to Move the Command

To move a command from one key to another you write a brief statement telling the computer what you want to do, highlight it, and use the CALC command. The statement says in effect, "I want to remove the command from this key, and I want to put it on that key."

Suppose you want to move PRINT from USE FRONT N to USE FRONT U, for example. The first thing you need to do is write a brief statement saying you want to remove it from the N key. Look up the number for N, which is 5550, and combine it with the "do-nothing" command, NOOP. The number for the NOOP command is zero, so you write

```
POKE 5550,0:
```

Now you have to assign the PRINT command to the U key, so look up the number for U (5557), and the number for the PRINT command (2), and then finish the statement:

```
POKE 5550,0:POKE 5557,2
```

Highlight what you have typed, and then use the CALC command. The key assignment changes instantly.

In general, if a command with the number XX is on a key with the number AAAA, and you want to assign it to the key with the number BBBB, then use the CALC command on POKE AAAA,0:POKE BBBB,XX.

WARNING: *If you assign a command to G, the key that does the CALC command, you will lose the ability to do CALC at all. So remember to reassign CALC somewhere else first!*

Glossary

Glossary

AB%: Number of blank lines at the top of each printed page; default is 6; stands for "ABove."

accessory card: A circuit board that adds capability to your computer when it is plugged in.

acronym: A word formed from the first letters of the words of a descriptive phrase, e.g. "MADD" for "Mothers Against Drunk Driving."

address: See "machine address."

Apple //e: A later version of the venerable Apple II, a computer originally designed by Steve Wozniak.

Applesoft BASIC: Apple's dialect of the BASIC programming language.

ASCII: Acronym for American Standard Code for Information Interchange. This standard assigns a numerical code to each character. Most computers use this standard.

autorepeat: That ability of a key that causes it to act as if you were tapping it rapidly and repeatedly. This rapid repeat action is invoked by holding the key down for more than half a second.

automatic highlighting: When the PRINT, SEND, and CALC commands are invoked, the highlight will automatically extend as if both LEAP keys had been pressed at the same time and then the command had been given. The highlighted text will be the text covered in the last leap, or the text typed since the last leap. Creeping after leaping or typing will alter the boundary of the highlight.

backup, backup disk: An extra copy made for security's sake.

backward: Left of or above the cursor's present location; an earlier position in the text.

backward delete: Erasing text to the left of the cursor a character at a time. Only a "wide" cursor will delete backward. Also see "forward delete."

backspace: Moving the cursor backward by pressing the DELETE key; same as backward delete.

BASIC: Acronym for "Beginners All-purpose Symbolic Instruction Code. A popular computer language for personal computers. BASIC was created by Kemeney and Kurtz at Dartmouth College. Built into your Apple and accessible through your SwyftCard.

BASIC string: A phrase or sequence of characters given a name in the BASIC language. Such names always end with a dollar sign ($).

baud: A measure of the speed of communications, roughly equal to ten times the number of characters per second.

baud rate: Baud.

BE%: Number of blank lines at the bottom of every printed page. Default is 6. Stands for "BElow."

blank: Nothing there.

blank disk: A disk that contains no information.

blinking: Alternately appearing and disappearing. The only thing that blinks in the SwyftCard system is the cursor.

board: See circuit board.

boot, boot up: Computer jargon for "start;" to load a disk and automatically execute the program contained on it.

bottom margin: See BE%.

break: See "page break."

break, in communications: A very few external systems require that you be able to put the modem into a special condition called "break." The ability to send a break is not provided. Programmers can use BASIC to create a break condition.

bug: A software malfunction.

bulletin board: See "computer bulletin board."

byte: The amount of computer storage required to store one character.

CALC: One of the commands (USE FRONT G). Used to do arithmetic, adjust parameters for Print and Send, and execute small BASIC programs. Only 900 bytes are allocated to BASIC. See "automatic highlighting."

calculation: That art judged by a majority of the population to be second in popularity only to having one's teeth worked on; arithmetic.

CALL 3600: To erase a disk, type and use the CALC command on *CALL 3600.* After this, the disk is blank as far as SwyftCard is concerned. There is no way in the system to get back the information on the disk once this is done.

capital: An uppercase character; see "character."

CAPS LOCK: A nuisance which, when pressed, stays down and changes all lowercase letters to uppercase. If down when pressed, the CAPS LOCK key stays up. If this key is in the down position it can cause leaping to appear to be out of order.

card: A circuit board.

changed: See "DISK."

character: A building block of text. The SwyftCard system gives you 97 different characters from which to construct your text. These characters are usually classified into six groups: lowercase alphabet, uppercase alphabet, numerals, punctuation, special symbols, and formatters (for a table, see Appendix C, List of Features). Each character is created by pressing the appropriate key on the keyboard, sometimes in conjunction with a SHIFT key.

character-by-character (delete): Using the DELETE key to remove text one character at a time. This differs from using DELETE to remove a chunk of text, since text deleted a character at a time is *not* saved, and cannot be retrieved by the INSERT command. Deleted chunks of text can be retrieved (until the next chunk is deleted).

CHR$: A BASIC function that, given the ASCII code for a character, returns that character.

chunk: Two or more contiguous characters.

circuit: Any combination of wires and electronic parts.

circuit board: A sheet of material (typically fiberglass) on which wires areprinted and electronic parts attached. The SwyftCard is an example of a circuit board.

clobber: An informal term for "destroy".

Closed-Apple key: The key on the right side of the SPACE BAR. The Apple logo is printed on the key in a solid color (no outline). See "LEAP FORWARD"

code: A numerical designation.

column: Vertically aligned characters, words, or numbers; a vertical line on which characters may be aligned; see "TAB"

command: An action initiated by the use of a "command key."

command key: A key that will give a command to the computer if it is tapped while the USE FRONT key is held down; see "hold and tap."

commercial data services: Large computers set up so that you can, for a fee, phone in and retrieve a wide variety of information, including stock quotes, news, legal references, business directories, airplane schedules, and much more. You use a data service when you send and receive Telex messages with SwyftCard.

communications: Transferring information between your Apple and another computer.

computer: A device which runs programs computer bulletin board: a computer system set up so that you can postmessages to other people and see the messages they have left; an electronic kiosk.

CONTROL: A key on the middle left side of the keyboard, labeled "USE FRONT" in the SwyftCard system. When certain letter keys are tapped while this key is held down, the specified commands are invoked (for a list, see Part 5, "Other Features of the Keyboard"). Also used in conjunction with the RESET key to unfreeze the cursor (Part 20,"Frozen Cursor"), stop a PRINT, SEND, BASIC program, or repeating "?SYNTAX ERROR" message (see Part 19, "To Stop a BASIC Program"). Used in conjunction with Open-Apple and the RESET key to load a disk, overriding the DISK command (Part 20, "Overriding the DISK Command")

control character: Special characters not used by SwyftCard, but required by some data services that can be accessed through the SEND command. Generated in the following manner: Press and hold the CONTROL key, and, while holding it down, tap the Z key and then the desired control character. The control character will be sent to whatever interface is in Slot 2. Control characters can be sent to a printer via the printer initialization technique (Part 17, "Control of Printing Style").

controller, also disk controller: The circuit board that is the interface between your Apple and its disk drive.

copy: Duplicate (noun or verb). You can duplicate a chunk of text by highlighting it, deleting it, and then using the INSERT command two or more times. You can duplicate a disk by using the DISK command twice in a row, the second time on a blank or backup disk.

copy up: To use the DISK command to insert a highlighted chunk of text from one Text into a second Text.

copyright: In this case, our right to not allow you to copy our software, hardware or manuals; thank you.

COS: A mathematical function.

CR: Abbreviation for "return character" (from Carriage Return); used specifically where the ASCII code 13 is meant.

creep: Moving the cursor a character at a time by tapping on a single LEAP key.

crunch: A colloquial term for clobber.

cursor: The blinking object that shows where the next character you type will appear; see "leap" for cursor motion information; also see "narrow" and "wide."

data: Information.

data services: See "commercial data services."

debug: To remove the bugs from: to make a program or system correct.

default, default setting: SwyftCard's standard setting for various parameters.

delete: To cause one or more characters to be removed fom the text or from a pattern; erase.

DELETE: The key marked "DELETE". Erases the character or characters inside the highlight when pressed. When pressed while a LEAP key is being held down, removes the last character from the pattern. (See "backward delete" and "forward delete.")

different: The term for a disk in the drive that is not the same disk as the one that provided the text on the screen; see "DISK."

disk: A square plastic envelope that contains a round sheet of plastic coated with magnetizable particles (the same sort of stuff as recording tape) on which computer data is stored.

DISK: One of the commands (USE FRONT L). The DISK command loads, saves, or boots a disk, depending on which action is appropriate. If it can do none of these, it bleeps. Use DISK before you remove a disk, and after you place one in the drive. When you use the DISK command, SwyftCard looks at the text, and does the appropriate action based on whether

1. the text has been saved on disk in which case it is called "unchanged" and it is OK to load a new text,
2. there is no text at all, in which case it is called "empty" and it is also OK to load a new text,
3. some editing has been done in which case it is called "changed" and must be saved.

Before a load or save, SwyftCard also checks the disk to see if is appropriate for the operation. For example, it won't save on the wrong disk. There are four kinds of disks
1. the "same" disk you are currently working from,
2. a "different" disk made by SwyftCard,
3. a "blank" disk which is always saved to, or
4. a "non-SwyftCard" disk which is booted if the text is unchanged.

Another set of conditions checked for is to see if there is no disk at all in the drive, if the door is open, or if the disk is write-protected.

disk controller: See "controller."

disk drive: The device that stores and retrieves information from a disk.

diskette: Disk.

display: The physical device on which text is made visible.
document: A part of the text that you consider as being a single entity.SwyftCard itself has no concept of "document" as a special category of text.
double-space: To type or print on every other line.
drive: Disk drive.
duplex: Communication over a phone line may permit both people (or systems) to speak at the same time: this is full-duplex. If only one may speak at a time (as with CB radios) the system is said to be half-duplex. SwyftCard allowsfull-duplex communications.

editing: The process of changing your text.
editor: A word processor.
empty: When SwyftCard has no text whatever the text is termed "empty;" see "DISK."
end of line: See "return."
enter: To give information to the computer.
equal signs (a row of): Indicates a page character, which signifies the end of a page.
erase: To delete.
erase a disk: see "CALL 3600"
ESC: The key in the upper left-hand corner of the keyboard. It is used to put a page character into the text. Tó get an ASCII escape character, as used in printer setup, use CHR$(27). To Send an ESC, type CONTROL Z ESC.
exponentiation: Raising a number to a power.
expression: A meaningful combination of symbols.
extend: To enlarge the highlight so that it contains more than one character.
extended highlight: A highlight that contains more than one character.
extended 80-column card: An 80-column card (defined below) with extra memory; not needed to operate the SwyftCard system.

fanfold: Paper made from connected sheets folded in alternatedirections. Most printer paper takes this form.
fingers: The (usually) gold-plated strips on a circuit card which allow it to connect electrically with the computer.
flashing: Blinking on and off. The other definition is not suitable for a family computer.
font: A particular graphic design of an alphabet and other characters and symbols.

form feed: A signal from a computer to a printer that advances the paper so that printing can proceed on the next page.

format: The design features of something printed.

format characters: Characters that control the appearance of text on-screen and in the printout. SwyftCard has three such characters: space, return, and page.

formatting disks: The act of making disks usable on a system. This is never necessary with SwyftCard.

forward: Right of or below the cursor's present location; a later position in the text.

forward delete: Deleting text to the right of the cursor. The cursor must be narrow for a forward delete. While deleting, the cursor stands still. The characters to the right march toward the cursor, appearing to be "vacuumed" up.

full duplex: See "duplex"

gadgetry: See "paraphernalia."

global: Everywhere. A "global" search and replace means to find a term wherever it occurs in the text, and to replace every occurrence.

graphics, graphics output: Pictures drawn with a computer. Using BASIC you can draw pictures with SwyftCard.

half duplex: See "duplex."

hard copy: A printout in paper and ink of the things you write and store electronically in the computer.

hardware: The physical parts of a computer system.

highlight: (v) To mark a chunk of text so that something can be done to it all at once. Some of the things you can do to a highlighted chunk of text include deleting, printing, calculating, moving, and sending. (n) The unblinkingrectangular object that accompanies the cursor. Indicates which character or characters are highlighted. Anything in the highlight will be erased when the DELETE key is pressed.

hold and tap: A basic physical technique used extensively in SwyftCard. While one key is pressed and held down continuously, other keys are tapped. Four keys are used this way in SwyftCard: SHIFT, which enables you to type capitals and certain symbols; USE DFRONT, which enables you to use the commands; and the two LEAP keys, which enable you to move the cursor.

incantation: A written spell or charm that will produce certain effects inside SwyftCard.

indent: To set in from the margin; see "TAB ."

indentation: How far the first line of a paragraph is set in from the margin.

information retrieval: Giving a system a clue as to what you are looking for, and having the system try to find it. Information retrieval also includes the process of entering information into a system so that it can be retrieved.

information, storage of: Your text may be stored in the computer, on a disk, or both.

initialize, initialization: A step you have to take with some devices before you can operate them. This applies especially to printers and modems.

initialization sequence: The list of things you have to do to get something started. Printers and modems must sometimes be initialized.

INSERT: One of the commands (USE FRONT A). If a chunk of text has been deleted by a single tap of the DELETE key, INSERT will recall it to the screen. INSERT is used to undo accidental deletions, copy text, and move text from place to place. The deleted material is available for recall until another chunk is deleted.

instance: An example of something.

interface: The software and circuitry required so that two devices (or one device and a human) can communicate; somewhat trendy if used to mean talking to another person.

interface card: see "interface" and "card."

invoke a command: To use a command.

key: One of the pushbuttons on the keyboard.

keyboard: A device with many push buttons, designed for tiring your fingers. You use it to control and communicate with your computer.

keyword: A word or phrase that sticks out in your mind as being associated with some document or place in a document to which you'd like to move the cursor.

land on: To occupy the same position as.

leap: The term that describes SwyftCard's cursor motion. To leap to the target character, you hold down a LEAP key, then type a pattern consisting of the target character followed by the characters that come after the target character in the text. The cursor will leap to the nearest instance of that pattern. If the pattern cannot be found, the cursor will remain where it is. The leap mechanism is patent pending.

LEAP AGAIN: Alternate function of the TAB key, available when a LEAP key is held down; causes the cursor to leap to the next instance of whatever pattern you have typed without you having to retype the pattern.

LEAP AGAIN key: The key marked "TAB" on the left edge of the keyboard, when used in conjunction with a LEAP key; autorepeats when held down.

LEAP keys: The cursor control keys, Open-Apple and Solid Apple, which are located immediately to the left and right of the SPACE BAR. Pressing and holding a LEAP key readies the cursor to leap to a pattern of characters anywhere in the text. Tapping a LEAP key causes the cursor to "creep," that is, move one character at a time. Pressing both LEAP keys at once extends the highlight. Most typists prefer to operate the LEAP keys with their thumbs.

LEAP BACKWARD: The Open-Apple key on the left side of the SPACE BAR. Pressing and holding this key readies the cursor for a backward leap through the text. See "backward."

LEAP FORWARD: The Solid-Apple key on the right side of the SPACE BAR. Pressing and holding this key readies the cursor for a forward leap through the text. See "forward."

leap pattern: See "pattern."

LE%: Sets the ASCII code of the character to be sent out at the end of every line when using the SEND command. If a return is desired, set LE% to 29. Default is -1, which means nothing is sent out. Stands for "Line End."

left margin: See "margin," "PR$", and "MA%."

LF: Abbreviation for "line feed;" used specifically where the ASCII code 10 is meant.

line: A horizontal row of characters, or the space reserved on the screen or paper for a row of characters.

line feed: Moves the printer up one line.

line width: See "WI%."

LIST: A BASIC command that displays a program you have written.

load: To bring the information on a disk into the computer. No information is removed from the disk during a load; rather, the disk and feeds it into the computer. See "save"

LOG: A mathematical function.

lost: Leaping to a different part of the text than you intended. If you have not let go of the LEAP key, you can get unlost just by tapping a few characters at random.

lowercase: Letters that are not capitals. Lowercase letters in a leap pattern match both uppercase and lowercase letters in the text. See "character."

LP%: The number of lines the page number will appear above the bottom of the page. Default is 3. Stands for "Latitude of Page number."

machine address: A number (much like a house number) that tells where a particular piece of information is stored in a computer.

manual: The second most important part of any computer system (you're the most important part.)

margin: The space between the top or sides of a piece of paper (or the screen) and the borders of the printed material; see "PR$", "AB%," "BE%," "MA%," "WI%," and "PL%."

MA%: The number of spaces between the left edge of your text and the leftmost printing position of your printer. Applies to text on the printer only; does not affect the appearance of text on the display. Changing MA% changes the right margin; some adjustment of WI% (see also) may be necessary as a result. Default is 0. Stands for "MArgin."

memory: The part of the computer that is capable of storing information.

mistake: Something you do which proves you are a human being.

modem: A costly little box or circuit card that allows your computer to talk to other computers. The name is an acronym for MOdulator/DEModulator, so-called because it takes the computer signals and turns them into sounds that can be sent over telephone lines (modulation), and turns the sounds from other computers into signals your computer can understand (demodulation.)

modem card: An interface card for a modem, or a complete modem; installed in Slot 2 of the Apple //e.

monitor: The device that has the screen that you stare at for hours on end. It is usually sort of a TV.

move text: To delete text from one place and INSERT it at another.

munch: A state of the cursor: the cursor and the highlight are both on the same character. The cursor becomes narrow after a leap or creep. A narrow cursor forward deletes. See "wide"

NEW: A BASIC command.

non-SwyftCard disk: A disk that is not programmed in the SwyftCard system (see DISK.)

no disk: An empty disk drive (see DISK.)

on-line: Available via a telephone call or from the computer immediately.

on-screen: Visible on the monitor screen.

Open-Apple key: The key to the left of the SPACE BAR, used in conjunction with character keys to leap the cursor backward in the text; see "LEAP BACKWARD."

Outline-Apple key: The Open-Apple key

PA%: The page number of the first page character in a Text. All numbers are determined in relation to this page number. Default is 0. Stands for "PAge."

PAGE: The key that produces page characters. This key is marked "ESC" and should be labeled "PAGE" by the user of SwyftCard. See "page character."

page break: A dashed line that symbolizes the end of a page. The number in the middle of the line is the page number of the text above it. SwyftCard puts in a page break whenever you create enough text to fill a printed page. A page break is not a character, hence you cannot delete it, leap to it, or produce it with a key. See "page character."

page character: A character that forces the end of a page. The page character symbol is a double-dashed line with the page number in the middle. Since it is a bona fide character – just like a letter, numeral, or punctuation mark – you can type it, delete it, or leap to it. See "PAGE," and "page break."

page number: The number that appears in the middle of page characters, and which will print at the bottom center of the paper page; see "LP%," "PA%," "PF%."

pagination: The arrangement of page boundaries in the text.

paragraph: In SwyftCard, a chunk of text bordered by blank lines or enclosed by page characters.

parameters: A number or value that is needed by some part of the system, such as line width, number of lines per page, margins, baud rate, or page numbering.

paraphernalia: See "gadgetry."

pattern: One or more characters you type while holding down a LEAP key. The cursor leaps to the nearest instance of the pattern, landing on the first character (the "target character"). A pattern can be up to 31 characters.

perforations: The place fanfold paper is *supposed* to tear.

periferals: Accessories for a computer.

PF%: The lowest page number that you want printed. Default is 2, meaning that. Page numbers lower than 2 will not print. Stands for "Page Floor."

PL%: The number of lines per page. Default is 66.

power: A corrupting influence.

power supply: That portion of an electronic device that converts whatever form of electricity is available in your locale into the form of electricity required by the circuitry of your computer.

power up: Computer jargon for "turn on."

press: To push down a key on the keyboard; see "tap."

print: To translate text on screen into ink on paper.

PRINT: One of the commands (USE FRONT N); prints the text that was covered in the last leap, or the text that was typed since the last leap, or whatever text is highlighted; see "automatic highlighting."

printer: A device that reproduces your text on paper (provided it feels like working that day.)

printer card: An interface card that allows the Apple to talk to a printer; placed in Slot 1.

printer initialization: A task that must be done to make some printers operate properly. SwyftCard uses variables called PR$, US$, and UE$ for initialization. Once you put the required information into them, SwyftCard automatically sends them to the printer each time you use the PRINT command. The printer initialization is saved along with your disk when you use the DISK command. By using the CHR$ function in BASIC, any ASCII code (including ESC and control characters) can be put into PR$ and thus sent to the printer. Default is FX-80 or compatibles. See "variable", "CHR$."

print out, printout: The act of printing; the stuff you get from a printer.

program: (v) to write instructions for the computer; (n) a set of instructions that tells a computer how to accomplish a task.

programmer: A person who either no longer needs other humans for solace or companionship, or who doesn't realize this until they have all gone to sleep.

protocol: The art of being obsequious.

PR$: See "printer initialization"; stands for "PRinter."

quandary: A puzzling situation.

random: The technique used to write this glossary.

retrieve: To find something tucked away. The leap technique is an excellent retrieval mechanism. See "information retrieval."

RETURN: The RETURN key; types return characters.

return: One of the characters you can type. On SwyftCard, the return character puts the cursor on the next line and moves it to the left edge. After a return, the next character you type will display and print on the next line at the left border of the text.

right margin: See WI%.

ROOM%: The number of characters you can type before your Text is full.

RUN: A BASIC command.

same: A state of a disk recognized by the DISK command; means the disk in the drive is the same SwyftCard disk you started with; see "DISK".

search: Looking for an instance of a pattern in the text (something SwyftCard does every time you leap.)

SEND: One of the commands (USE FRONT D). Transmits text through an Apple Super Serial Card in Slot 2, and out the serial port. The text sent will be the text covered in the last leap, typed since the last leap or received text, or highlighted at the time the command is used. See "automatic highlighting."

serial port: Either the act of drinking one glass of wine after another, or a destination for ships laden with grain. Also, an electronic device which allows your computer to communicate with another device that also has a serial port. Serial ports are usually found on modems and sometimes on printers.

set: To give a parameter a value. This is done by using the CALC command on a formula that includes a variable, an equal sign, and the numerical setting. For example, use the CALC command on WI%=60 to set the line width to 60 characters (WI% is the variable.)

setup: Initialization .

SHIFT: One of the two keys marked with the word "SHIFT". Typing while holding either one of them gives you uppercase letters and the upper symbol on those keys which have two symbols .

SIN: A mathematical function.

single-space: Lines of text (or print) that follow each other with no blank lines between.

slot: One of the seven places you can plug accessory cards into an Apple //e.

software: Programs, i.e. instructions that tell a computer what to do. The phrase "software programs" is redundant.

Solid-Apple key: See "Closed-Apple key."

space: The only character in the English language more common than the letter "e". Put into the text by pressing the SPACE BAR. A space is a character just like any other character in the SwyftCard system.

SPACE BAR: One of the best scenes in the movie "Star Wars." Also the biggest key on the keyboard.

SP%: Line spacing in the printout. Set SP%=2 for double-spacing; set SP%=1 for single-spacing. Default is 1. Stands for "SPacing."

start bits: A communications parameter.

stop bits: Another communications parameter.

store: (v) Putting information into memory, or onto a disk.

string: A sequence of characters.

string variable: A BASIC variable that contains characters.

stuck cursor: Either a cursor that is flashing but will not move, or a cursor that stops flashing.

Super Serial Card: An Apple Computer Inc. interface card required for use with communications.

SwyftCard: An accessory card for the Apple //e that allows you to do word processing, calculations, communications, information retrieval, and short BASIC programs.

symbol: A raucous musical instrument of the percussion family.

SYNTAX ERROR: A message inserted into the text by BASIC when it detects something it can't understand.

system: Any collection of items and ideas that can be usefully thought of as a single entity. For example, the SwyftCard system consists of a computer, its peripherals, and a circuit card that embodies certain ideas that make a computer easier to use.

TAB: A key for typing several spaces at once. Each tap on the TAB key inserts enough spaces to move the cursor to the next column. TAB columns are five spaces apart. See "column" and "AGAIN."

TAN: A mathematical function.

tap: To quickly depress a key as far as it can go and immediately release it.

target: Target character.

target character: The character you want to move the cursor to; the first character in a pattern. It may be any letter, punctuation mark, numeral, symbol, space, return, or page character, anywhere in the Text.

text: All the characters that you can see on your screen or that can be reached by a leap is this different than Text? The text taken as a whole; all the information of yours that's in the computer at one time.

thumb: The best finger to use on the LEAP keys; may also be held down on a LEAP key to hold your place in the text, much as you might use a finger to mark your place in a book while you scan through the pages.

top margin: See AB%.

trademark: A word ruined for ordinary discourse in the pursuit of commercial interests.

tutorial: An instructional text.

typing: (1) The process of striking keys on a keyboard in order to create text. (2) An undeservedly low-paying job.

typewriter: An ancient instrument of torture that would penalize you by ruining a whole page when you merely mistyped a few characters.

UE$: The codes sent to the printer to turn off underlining. Stands for "Underline End". See "printer initialization."

unchanged: A state of the text recognized by the DISK command. Unchanged text has not been altered since it was last saved or loaded.

undo: To nullify something you have done. Deletion can be undone (with the INSERT command), highlighting text can be undone (by pressing either LEAP key), and an incorrect cursor leap can be undone (by typing a few letters at random before you've let go of the LEAP key.)

uppercase: Capital letters. Uppercase letters in a pattern match only uppercase letters in the text. See "character."

US$: The codes sent to the printer to turn on underlining. Stands for "Underline Start". See "printer initialization."

vacuum: The way delete seems to work when the cursor is narrow. See "forward delete."

variable: A short sequence of characters, such as WI%, I, or PR$, that can represent a number or a string in BASIC. See the Apple BASIC manual.

video: The signal going to the monitor.

whirr: A polite word for the dreadful sound the disk drive makes when it starts up.

WI%: Characters per line. Set WI% to the maximum number of characters you want in each line, both on the screen and on the printer. The minimum is 16, the maximum (and default value) is 80. A change in MA% will affect the right margin and may require an adjustment in WI%. "WI" stands for "WIDTH".

wide: A state of the cursor: the highlight and cursor are split apart (not on the same character), and the highlight contains only one character. See "narrow," "backward delete," and "extended highlight."

width: The maximum number of characters permitted in a line. See "WI%."

window: The portion of the text that you can see in the display

word: (1) In SwyftCard, a consecutive sequence of non-blank characters and up to two of the spaces that follow them. This is similar to the normal use of "word" except that English words don't include the spaces. (2) In computer jargon, all the information at one machine address (3) In communications, a package of information that conveys a single character.

word processing: Creating and editing text electronically; usually includes the means for storing and printing the text so created. All word processors make rearranging text far easier than on a typewriter.

word wrap: The process of automatically breaking text into lines without your having to type a return. Word wrap keeps words whole, it never breaks them in two.

write-protect: To install a safeguard on a disk that prevents the disk from being altered by the computer.

write-protect notch: A usually rectangular notch in the right side of the disk (when the label is in normal reading position) that can be covered with a write-protect tab in order to write-protect the disk.

write-protect tab: A piece of adhesive-backed paper or foil which is wrapped over the edge of the disk, covering the write-protect notch. Once this is done, the text on the disk cannot be altered by the computer.

8-bit word: See Word. Entries 2 and 3 apply.

80-column card: A circuit board that allows your Apple //e to show 80 characters of text per line, instead of the measly 40 you get without it.

80-column display: A system capable of showing 80 characters in each line.

? Placed in front of arithmetic expressions you want to calculate. For example, to calculate 2 + 2, type ? 2 + 2, and use the CALC command on this expression. Also placed in front of variables in order to ascertain their value. For example, to see how many characters you have per line, highlight and use the CALC command on ?WI%.

; Placed after arithmentic expressions you want to calculate if you don't want a return automatically inserted afterward.

Index

Index

The SwyftCard™ Reference Guide

The Commands

The following keys perform the indicated SwyftCard command when tapped while you hold down the USE FRONT key.

Key	*Function*
A	INSERT Places in text the last chunk of text that was DELETED
D	SEND Transmits the highlighted text via the super serial card
G	CALCulate Evaluates or executes the highlighted text
N	PRINT Delivers the highlighted text to the printer interface
L	DISK Directs the disk to do the appropriate disk operation
Z	Sends next keypress as a control character via super serial card

SEND, CALC, and PRINT autohighlight.

Special Keys

Certain Apple keys have special meanings to SwyftCard:

Apple	*Label*	*SwyftCard Use*
ESC	PAGE	generates a Page Character. Sends an ESC after a USE-FRONT-Z
TAB	LEAP AGAIN	when used in conjunction with a LEAP key. Tabs if pressed by itself. Tab stops in columns 5, 10, 15, ... , 75
CONTROL	USE FRONT	is used with certain keys to give SwyftCard commands.
DELETE	DELETE	deletes whatever is highlighted. DELETE operates to the left after typing, to the right after leaping or creeping.
OPEN-APPLE	LEAP BACKWARD	is held down while you type a pattern to which you wish to have the cursor LEAP. Will "creep" backward if tapped.
SOLID-APPLE	LEAP FORWARD	is like "LEAP BACKWARD" but the search goes forward through the text. Will "creep" forward if tapped.

Upper/Lower Case During LEAP

Lowercase letters in the pattern match both upper- and lowercase letters in the text. Uppercase letters in the pattern match only uppercase letters in the text.

Cursor and Highlight

The blinking "cursor" shows where the next letter will appear when you type, and where material will be placed by INSERT or from a modem. The unblinking "highlight" is usually alongside the cursor and showns what will be deleted if you press the DELETE key.
To Highlight A Chunk of Text: Move the cursor from one end of the chunk to the other and press both LEAP keys down at the same time. You can move the cursor to the left or right end of a highlight by pressing and releasing the left or right LEAP key respectively. You can then adjust the exact placement of that end of the highlight by creeping with either LEAP key and then re-highlighting by using both LEAP keys together again.

Disk Operations

Whenever you place a disk in the drive, use the DISK command. Before you take it out, use the DISK command. To force a disk to load, DELETE the entire Text and use the DISK command. When you change from one disk to another, anything highlighted on the first disk will be automatically INSERTed into the second disk: this is called COPY UP. To clear a disk irreversibly, type and highlight CALL 3600 and use the CALC command: this is dangerous, and should be used with due care.

Calculations

The CALC command sends the highlighted text to Applesoft BASIC. See Apple's manuals for details. Programs over 20 or so lines may not work.

Underlining

Precede and follow the text to be underlined with an underline character: __. You will have to have set US$ and UE$ for some printers for this to work. Two Returns or a Page Character also turn off the underline.

Values You Can Set or Values You Inquire About:

RO% How much space you have left, in characters.
If negative, use ?65536+RO%

PR$ The codes the printer needs.
If you are in a hurry, try one of these:
PR$=CHR$(0)+CHR$(0)+CHR$(0) or
PR$=CHR$(1)+CHR$(0)+CHR$(0)
If they don't work, read the manual.
Default:
PR$ = CHR$(0)+CHR$(0)+CHR$(5)+ CHR$(27)+CHR$(77)+
CHR$(27)+ CHR$(108)+CHR$(8)

WI% Line width.
Min 15 Max 80 Default 80

MA% Left margin when printed, not on the screen.
Min 0 Max 80 Default 0

AB% Top margin when printed.
Default 6

BE% Bottom margin when printed.
Default 6

PL% Page length, total of printed and margins.
Default 66

PF% Lowest page number to be printed.
Default 2

PA% The number to appear on the first page character in text.
Default 0

LP% Distance from page number to bottom of page, in lines.
Default 3

SP% SP%=2 for double spacing, for single spacing the default: 1

US$ The codes the printer needs to turn underlining on.
Default: US$=CHR$(27)+CHR$(45)+CHR$(49)

UE$ The codes the printer needs to turn underlining off.
Default: UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)

LE% Set LE%=29 for most communications. Between SwyftCards use default: -1

SE% For 2400 baud use 6665, for 1200 baud use 6153, for 300 baud use default: SE%=5641. Others in manual.

Stopping Incoming Communications

You can stop most incoming electronic communications with USE FRONT ZS and restart them with USE FRONT ZQ. This will always work between SwyftCards.

Bleeps

When using DISK command: No disk in drive, door open, disk is write protected, or you've switched disks without saving. When typing, INSERTing, or receiving from a modem: Text is full, you have more than 200 page breaks, or you were sent a "CONTROL-G". When using the CALC command: BASIC has encountered an error.

Hardware Requirements

Apple //e (enhanced or regular), 80-column card (with or without extra memory), 5 1/4" disk drive and controller, monitor.
For printing: printer and interface card.
For communications: modem and Apple Super Serial Card.

Slot Assignments

Turn off power while installing cards.

SwyftCard	Slot 3.
Printer Interface	Slot 1.
Super Serial Card	Slot 2.
Disk Controller	Slot 6.

If All Else Fails

Before you give up, press and hold USE FRONT and firmly tap the RESET key. This will usually restore normal operation to SwyftCard if something weird has happened.