

SUPERKEY[®] THE PRODUCTIVITY BOOSTER

*Simple macros turn
1000 keystrokes
into 1!*

*Powerful encryption
keeps confidential
files confidential!*



IBM[®] VERSION

PC, XT,[®] AT,[®] & True Compatibles

SUPERKEY

Owner's Handbook

Copyright© 1985, 1986
All Rights Reserved
BORLAND INTERNATIONAL, INC.
4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
USA

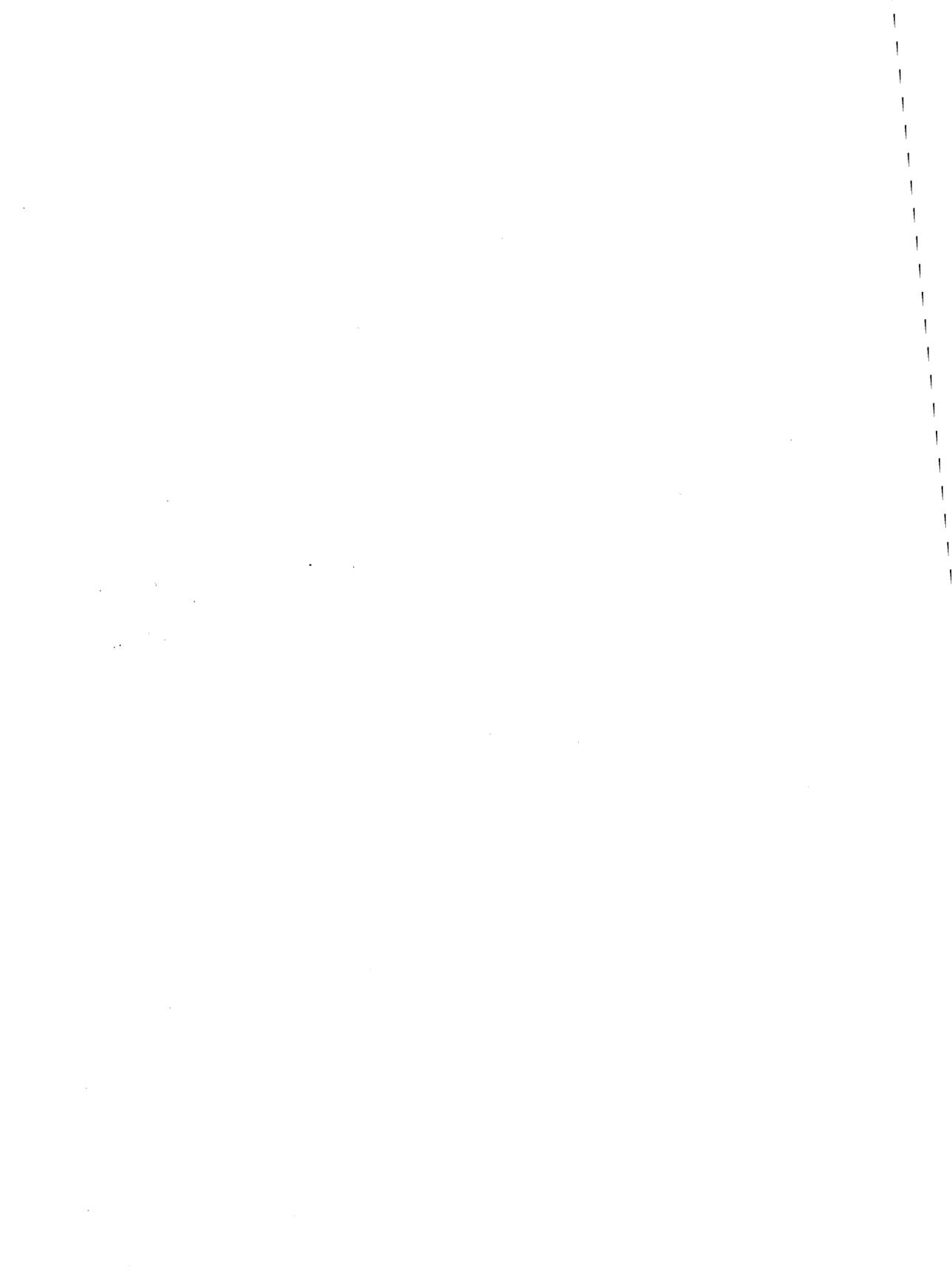


Table of Contents

Introduction	1
How to Use SuperKey	2
Structure of this Manual	2
The Distribution Disks	3
Minimum System Requirements	3
Acknowledgments	4
Chapter 1. Getting Started	5
Getting SuperKey Into Your System	5
Files on Your Diskette	6
Files Needed When Running SuperKey	7
Installation	8
Using SuperKey With Other Resident Programs	8
If You Use Tree-Structured Directories	9
How to Start SuperKey	9
How To Kill SuperKey	10
Chapter 2. Getting to Know SuperKey	11
Putting SuperKey to Work	11
How To Get Help	12
Record Your First Macro	13
DOS Commands Made Easy	15
Which Keys Can You Use?	16
Skip A Macro	17
If You Don't Like Menus	17
How to Stop a Macro	17

Save Macros On Disk	18
Clear Macro Memory	19
Load Macros From Disk	19
Delete A Single Macro	20
Macro Titles	20
Show Us Your Macros	20
Ready-Made Macros	21
The Functions Menu	22
Time Stamping Your Data: Time Function	22
May We Have Some Privacy, Please: The Screen Off Function	23
Freeze Those Keys: The Keyboard Lock Function	24
Encryption	25
DOS Command Stack	26
Type-ahead Buffer	27
Setup Menu	27
Encryption Directory	28
Macro Directory	28
Save Setup	29
The Options Menu	29
One Finger Mode	30
Suspend	31
Disk Wait	31
Bottom Line	32
Playback Delay	32
Protect Delay	33
Save Options	33
Chapter 3. How To Use SuperKey Without Menus	35
Chapter 4. Using SuperKey With Your Favorite Program	37
Cut and Paste	38
Using SuperKey with a Spreadsheet	39
Make a 1-2-3 Macro	41
Playback Delay	41
Graphics	41
Chapter 5. Advanced Macro Use	43
Auto-Start Macros	43
Display Macros	44
Pages	46
Freeze Your Keyboard with a Display Macro	47
Transparent Display Macros	48
Enhanced Text	49
Titles and Auto-Start	49

Fields	50
More On Variable Fields	51
Fixed Fields	52
Field Formats	52
Entry Control	53
Nested Macro Recording	55
The Macro Editor	55
Write A Macro	56
Edit The Macro	57
The Main Menu	57
End Edit	58
Using a Text Editor to Edit Macros	58
Chapter 6. The Layout Editor	59
Chapter 7. Reference to SuperKey	63
Arrow Key Defaults	64
Arrow Keys Option	66
Auto-start Macro	68
Beep Function	69
Begin Block Default	70
Begin Macro	71
Bottom Line Option	73
Clear Macros	74
Clear Keyboard Layout	75
Clear Screen Function	76
Command Line Parameters	77
Command Stack	80
Command Stack Option	82
Cut and Paste	84
Cut and Paste Function	86
Date function	87
Decimal Point Default	88
Defaults	90
Delay Function	92
Delimiter Default	94
Disk Wait Option	96
Display Macro	98
Edit Macro	101
Encryption	103
Encryption Directory	111
End Macro	113
Entry Mask	114
Fields	117
Field Formats	119

File Names	123
Fill Character Default	124
Fixed Field	125
Format Fields Option	127
Functions Menu	128
Interactive Macro Definition	130
Keyboard Buffer	131
Keyboard Click On/Off Option	132
Keyboard Lock/Unlock Function	133
Key Symbols	134
Kill SuperKey	138
Load Layout	139
Load Macro	140
Logged Drive Function	142
Macro	143
Macro Directory	144
Macro Memory	145
Main Menu	146
Merge Macro	148
One-finger Mode Option	150
Options	151
Path Function	153
Playback	154
Playback Delay Option	155
Protect Delay	157
Pull-Down Menu	159
Real Time Delay Function	161
Save Defaults	162
Save Macros	163
Save Options	164
Save Setup	165
Screen Off Function Screen On Function	166
Setup Menu	167
Skip Key	169
Show Titles	170
Stop	171
Suspend Option	172
Time Function	173
Titles	174
Variable Field	175

Appendix A. Using SuperKey With Other Borland Products	177
SideKick	177
GRAPH.MAC and GRAPH2.MAC	178
TAPE.MAC	178
Calling SideKick with SuperKey	178
Examples	179
Turbo Lightning	179
A Note for Reflex Users	180
Appendix B. Using SuperKey With Batch Files	181
The AUTOEXEC.BAT File	181
Loading Macros with .BAT Files	182
Appendix C. Installation	183
Screen Type	184
Default display mode	185
Monochrome display	185
Color display 80×25	185
B/W display 80×25	185
Blinking display?	185
Blank background?	186
Colors	186
Macro Memory Size	187
Appendix D. .MAC Files on the Distribution Disk	189
Appendix E. A DOS Primer	191
What Is DOS?	191
How to Load a Program	192
Directories	193
Subdirectories	194
Where Am I? The \$p \$g Prompt	194
The AUTOEXEC.BAT file	195
Changing Directories	196
Appendix F. HELP!!!	199
Appendix G. Glossary	203

Introduction

SuperKey is a *keyboard enhancer* that makes your keyboard a more efficient tool and saves you a lot of time. But SuperKey is much more than that. At the touch of a key it gives you instant access to a wealth of useful features found in no other single program:

- You can record sequences of characters in any key (a *macro*). A single key can thus hold lengthy commands for your spreadsheet, word processor, communications package, or any other program. One key can store frequently used numbers or phrases, or even hold entire paragraphs of text. This will save you thousands of keystrokes, because once recorded in a key, all this can be “played back” as often as you like at the touch of that single key. And you can record in as many keys as you want.
- SuperKey lets you “cut and paste” data from one program to another, bringing integration to “stand-alone” programs such as WordStar, MultiPlan, and many more.
- You can encrypt (scramble) any file for complete data and program security—even when you transmit files through the telephone line.
- You can design your own help systems for any of your programs with SuperKey’s *display macros*.
- You can redefine the layout of your keyboard. The standard keyboard can thus be made into a Dvorak keyboard, a Selectric keyboard, or any other keyboard of your choice.

- SuperKey gives you loads of “extras,” such as a screen-protect option which automatically turns off your screen after a preset time (to prevent “burn in” screen damage); a 128-character “type-ahead” keyboard buffer; a “command stack,” which lets you repeat your last 10 to 30 DOS commands without having to type them all over again; a “one-finger” mode which changes commands normally requiring you to press two or more keys simultaneously into a sequence of single keystrokes; and much, much more.

How to Use SuperKey

Although SuperKey is simple to use, and gives you detailed help information whenever you press **F1**, we suggest that you at least read Chapters 1, 2, 4, and 5, which take you on a quick tour of SuperKey and contain a number of useful hints.

These chapters provide a quick-and-easy introduction to SuperKey, but do not attempt to discuss it in every detail. Precise descriptions of all SuperKey commands, features, and concepts, along with examples, are collected alphabetically in Chapter 7. Use this chapter as a quick reference guide when you start exploring SuperKey on your own.

Structure of this Manual

- Chapter 1 tells you how to get SuperKey up and running. Also, all the files on your distribution disk are described here.
- Chapter 2 steps you through a typical session with SuperKey: making, saving, loading and deleting a macro; encrypting and decrypting a file; and using the SuperKey menus.
- Chapter 3 shows you how to sidestep the SuperKey menus and use “hot keys” to activate SuperKey features.
- Chapter 4 gives you some hints and techniques for using SuperKey with your favorite word processor or spreadsheet program.
- Chapter 5 provides details about advanced macro use. It discusses the SuperKey macro editor, which you use both to design complicated macros and to change a macro you’ve defined earlier. It also covers display and auto-start macros.
- Chapter 6 teaches you how to redefine your keyboard layout.

- Chapter 7 provides—in alphabetical order—detailed information about *all* SuperKey features. Examples are included.
- Appendix A describes how to use SuperKey with other Borland products.
- Appendix B explains how to automatically load SuperKey using batch (.BAT) files.
- Appendix C tells you how to custom-install SuperKey—screen type, colors, and macro memory size.
- Appendix D briefly describes each of the macro (.MAC) files supplied on your distribution disk.
- Appendix E explains some basic DOS concepts, such as directories and the AUTOEXEC.BAT file.
- Appendix F answers commonly asked questions about SuperKey.
- Appendix G is a glossary of terms used in this manual.

The Distribution Disks

Your distribution disk contains the main SuperKey program and several other files. Information about each of these files can be found in Chapter 1, pages 6 and 7.

SuperKey is not copy-protected. Please note that Borland's no-nonsense license statement licenses you to use your copy of SuperKey as if it were a book. It is not licensed to a single person, nor is it tied to one particular computer. The only restriction on using SuperKey is that *it must not be used by two different people at the same time*, just as a book cannot be read by two people at the same time. And, of course, giving away copies of SuperKey to others would be a violation of Borland's copyright.

Please sign and return the license statement so you can be kept up to date about SuperKey and other Borland products.

Minimum System Requirements

To use SuperKey, you should have the following:

- IBM PC or compatible
- PC-DOS or MS-DOS operating system, version 2.0 or later
- 128K RAM internal memory

Acknowledgments

In this book, references are made to a number of trademarks:

- Turbo Lightning is a trademark and SideKick, Turbo Pascal and SuperKey are registered trademarks of Borland International, Inc.
- IBM is a registered trademark of International Business Machines, Inc.
- Volkswriter is a registered trademark of Lifetree Software, Inc.
- Lotus 1-2-3 is a registered trademark of Lotus Development Corp.
- WordStar is a registered trademark of MicroPro Corp.
- Ready! is a trademark of Living Video Text, Inc.

C H A P T E R *1*

Getting Started

This chapter contains the information you need to make work disks and load SuperKey.

Getting SuperKey Into Your System

The first thing you should do with SuperKey is make a copy of the distribution disk. Always use this copy for any work you do with SuperKey. This way, if the work disk is damaged or erased, you can easily rebuild it from your original.

Use the DOS (operating system) COPY program to copy all the files on your SuperKey distribution disk onto your work disk. Assuming your work disk is in drive B: and the distribution disk is in drive A:, type

```
COPY A: *.* B: 
```

to transfer SuperKey to your work disk.

You now have a working copy of SuperKey, ready to load and run when you need it.

Before you do anything else, please type READ-ME on the DOS command line for important information.

If you have a hard disk, you could set up a subdirectory for SuperKey, for example one called \KEY. To do this, log onto the hard disk (usually drive C:), then type

```
MKDIR \KEY
```

Now put the distribution disk in the floppy drive, change to the \KEY directory, and copy all SuperKey files to the hard disk. Type

```
CHDIR \KEY  
COPY A:*. * C:
```

For more information about directories and other DOS features, see Appendix E.

Files on Your Diskette

Following is a brief description of the files on the distribution disk.

KEY.COM

The SuperKey program. All SuperKey functions are collected in this one file, and it is the *only* file you *must* have on your work disk when you start SuperKey. Once SuperKey is started, even this file is not absolutely necessary anymore.

KEY.COM

includes our highly efficient proprietary algorithm for file encryption. It protects your files against all but the most sophisticated intruders. For maximum security, use the DES standard file encryption in KEYDES.COM.

KEYDES.COM

KEYDES.COM is the same as KEY.COM, except that it uses the DES standard file encryption algorithm for maximum security. The cost of this higher security is that SuperKey won't perform quite as quickly.

KEY.HLP

Contains the SuperKey help information that appears when you press **F1**. This file may be left off your disk when running SuperKey if you don't want to use the built-in help system. If you want to use the help system, the file must be on the disk and directory from which you start SuperKey.

LAYOUT.COM

The keyboard layout editor. This is used to create new layout definition files or edit existing ones, and is only needed on your work disk if you want to edit your keyboard layout.

KEYINST.COM

Installation program which lets you change the macro memory size and colors used by SuperKey. Installation is purely optional, and you can leave this file off your work disk if you wish.

PCJRINST.BAT

This file contains the necessary commands to get SuperKey running properly on the PCjr. It needs to be run only the very first time you use SuperKey. Once you have installed SuperKey for the PCjr, you simply have to type KEY as you would with any other PC-compatible machine.

.MAC

Files with the last name MAC are SuperKey macro definitions. Their first names imply their use: WS.MAC, for instance, contains macros to be used with WordStar. See Appendix D for a list of .MAC files.

.LAY

Files with the last name LAY contain alternate keyboard layouts. Again, the first name indicates the type of layout.

LETTER.TXT

A sample text file that you can use to try out SuperKey's file encryption.

READ-ME.COM

This file contains the latest updates on SuperKey and any last-minute changes to this manual. Type READ-ME to see this information.

Files Needed When Running SuperKey

Once you have loaded SuperKey, the file KEY.COM (or KEYDES.COM) is only required on your disk when you want to save setups, options, and defaults. So if you are short on disk space, you *can* start SuperKey and then remove the disk with KEY.COM on it. Just remember to re-insert it in the same drive before attempting to save anything.

But be careful! DOS sometimes loses track of what's on your disks when you "swap" disks, and that can result in partially destroyed files, or even an entirely scrambled disk.

As a precaution, you should therefore press **Ctrl C** after you re-insert a disk and *before* you save anything on it. This assumes that you are in DOS when you swap disks. You should never swap disks if you are running any program (unless that program tells you to do so, of course; then you are safe).

If you are using MS-DOS 2.11, you must install the ANSI.SYS driver into your CONFIG.SYS file. If you already have a CONFIG.SYS file on your root directory, then edit it with your word processor or EDLIN and add the statement:

```
DEVICE=ANSI.SYS
```

If you do not have a CONFIG.SYS file, you can create one by typing from the DOS prompt (on your boot disk):

```
COPY CON CONFIG.SYS   
DEVICE=ANSI.SYS   
F6 
```

Be sure to put a copy of ANSI.SYS into your root directory.

Installation

SuperKey comes installed for an IBM PC or compatible computer. It is only necessary to go through the installation procedure if you are not satisfied with the pre-set values for color scheme, macro memory size, and screen type. Appendix C contains all the information you need to install SuperKey.

Note: If you want to install SuperKey on a PCjr, type PCjr on the DOS command line prior to using SuperKey the first time.

Using SuperKey With Other Resident Programs

SuperKey is a *resident* program, which means that once it is loaded into memory, it stays there until you switch off the computer. SuperKey is therefore available to you even while you run other programs, such as a word processor or a spreadsheet.

You may be using other resident programs with your computer, such as Borland's SideKick and Turbo Lightning, or a print spooler or RAM disk. If you are using other resident programs, it is essential that you load them in the following order:

- Load any non-Borland resident programs first (print spoolers, RAM disks, etc.).
- Load SuperKey next.
- If you have it, load Borland's Turbo Lightning.
- Finally, if you are a SideKick user, load SideKick last.

There is an exception to this rule: If you have another resident program that you want to use *with* SuperKey (and/or Turbo Lightning), load it *after* SuperKey (and Lightning). For instance, if you use Ready!, a resident outline processor, and want to write SuperKey macros and check spelling when you are within the Ready! program, you would load your programs in this order:

- SuperKey
- Turbo Lightning
- Ready!
- SideKick

If You Use Tree-Structured Directories

If you use a tree-structured directory, it is important that you change to the directory where you keep SuperKey's .COM and .HLP files before you start SuperKey.

This is because SuperKey must know where to find these files when you save a SuperKey setup or when you use the on-line help system.

If you use an AUTOEXEC.BAT file to set up your computer automatically each time you turn it on, please see Appendix B for instructions.

How to Start SuperKey

Once you have SuperKey on your work disk, type

K E Y ↵

to start it. The only file required on your disk at this point is KEY.COM. Please be sure to follow the previous instructions about using SuperKey with other resident software.

The message shown in Figure 1-1 will tell you that SuperKey is now loaded into memory and working. The memory information tells you how much RAM memory is left for running other programs. SuperKey's use of RAM is described in greater detail on page 145.

Although SuperKey is now waiting in the wings, you can't see it. To get in touch with it, you must press special keys, as described in the next chapter. You can also pass commands directly to SuperKey on the DOS command line. See page 77 for details.

```
Superkey          Version 1.15A
                  IBM-PC/XT/AT/PCjr
Copyright (C) 1984,85 BORLAND Inc.

655368 bytes total memory
597968 bytes were free
533424 bytes free
9588 bytes macro memory

C:\BORLAND\KEY >_
```

Figure 1-1 Log-On Message

How To Kill SuperKey

At some point you may need to “kill” SuperKey—remove it from memory if you need to free up space for other programs. This may be desirable if you need to create more room for other programs without resetting the computer.

But it should be done with caution, and only after observing the following rules:

- It must only be done if there are **no other programs** in memory **after** SuperKey. This means that you must not kill SuperKey if you have an application program running. That is, be sure you are on the DOS command line before you kill SuperKey. But first, be certain there are no other resident programs loaded after SuperKey. This means that if you have Turbo Lightning and/or SideKick in memory, you must kill them first.
- Kill the programs in the *reverse* order they were loaded in—i.e., SideKick, Ready!, Turbo Lightning, then SuperKey. *If you kill SuperKey while other programs occupy memory “above” it, you will create an unoccupied “hole” in memory—a situation that DOS cannot handle. You will have to reset the computer, and perhaps even turn it off and back on again.*

To kill SuperKey, call the main menu to the screen by pressing **Alt** **F**. Then, hold down the **Ctrl** key and press the **Home** and **End** keys.

You are now ready to use SuperKey. The following chapter takes you on a guided tour through the system. If you spend a little time with Chapter 2, you’ll be sure to get the most out of your SuperKey.

C H A P T E R 2

Getting to Know SuperKey

This chapter will get you off to a quick start with SuperKey. We'll explain how to use the menu system, how to create and use macros, how to encrypt and decrypt files, and more. This chapter introduces you to macros. For more detailed information, see Chapter 5.

Putting SuperKey to Work

To get in touch with SuperKey's menu system, press

that is, hold down  and press . You can press these keys at any time while your computer is running to call SuperKey to attention. The window shown in Figure 2-1 will appear at the top of your screen.

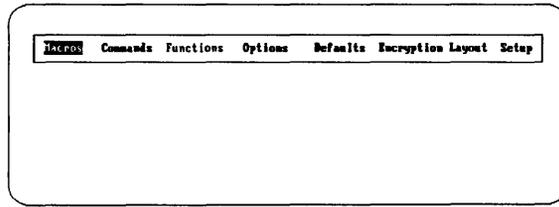


Figure 2-1 Main Menu

This is SuperKey’s main menu, which will allow you to access all its functions. There are two ways to select a menu item:

- Move to the right or left with the **→** and **←** keys. When the item you’re interested in is highlighted, press the **↵** key to open a pull-down menu containing details about that item.
- Press the key corresponding to the letter that is capitalized in the item. For instance, to select **Macros**, simply press **M** (upper or lowercase), and a pull-down menu for **Macros** will appear.

When SuperKey’s menus are on the screen, you are, in effect, inside the SuperKey world; your application program is temporarily suspended until you have finished telling SuperKey what you want it to do. To make the main menu disappear and return to your application program, press **Esc**.

SuperKey is still in your system, ready to appear again when you press **Alt /**. Whenever any of SuperKey’s menus or windows are on your screen, you can always make them disappear by pressing **Esc**.

Instead of using menus, SuperKey also allows you to use special “hot keys” to get in touch with its functions. See Chapter 3 for details.

Note that the **Functions** header is dim. This means that you cannot use **Functions** at this time; functions can only be called from within a macro (See page 12.)

How To Get Help

Chapter 7 contains detailed explanations about everything you need to know about SuperKey. Appendix F, *Help!!!* answers many of the most commonly asked questions about SuperKey. But there is an easier way that will answer many of your minor questions.

That’s the SuperKey help system. Whenever you would like a little assistance, just press

F1

and you'll get a helpful explanation about whatever you are doing. The KEY.HLP file must be present in the same drive and directory as it was when SuperKey was loaded.

Record Your First Macro

You are now ready to start recording a *macro*. A macro is simply one or more characters—text, numbers, or commands—which you record in SuperKey's memory and assign to a single key. Each time you press that key, all the recorded characters are played back, just as if you had entered them on the keyboard.

You can imagine SuperKey as a kind of tape recorder that you can start and stop at any time. When started, it will record everything you do on the keyboard; you can play back your recordings later just by pressing a key.

Right now you want to record a macro, so move the highlighted bar to **Macros** and press **←** or just press **M** to open the Macros pull-down menu. Figure 2-2 shows what you'll see. As on the main menu, you can either move the highlighted bar and press **←** or you can press the key corresponding to the upper-case letter of the desired item to select a menu item.

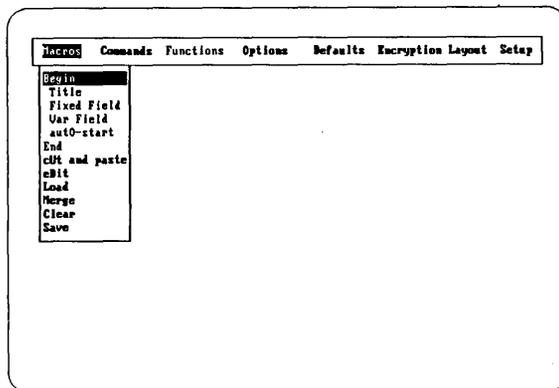


Figure 2-2 Macro Menu

Since you want to begin recording a macro, move the bar to **Begin** and press **←**, or just press **B**.

The menus disappear, and a small window now requests you to press the key you want to *define* (to hold the recording). For this example, press

F4

A line now appears at the top of the screen, informing you that you are in the process of recording a macro (unless you are currently using a graphics screen—e.g. Reflex; in that case, the status line does not appear). It shows you which key you are recording in, and tells you how to stop recording with the **Alt** **—** hot key.

Anything you enter now will be recorded in SuperKey's *macro memory*, and assigned to the key **F4**. Let's enter a command to the operating system (we'll just call it DOS from now on—it's short for Disk Operating System). For example:

DIR **↵**

Note that this command goes to the computer as usual, as well as to SuperKey's macro memory, so you will get a directory listing on your screen. SuperKey is completely *transparent* to DOS and other programs; that is, programs do not "know" that SuperKey is there in the background, recording everything you type. Now, to end the recording of this macro, press

Alt **/** **M**

to get the **Macros** menu back, and

E

to select **End**. The macro "tape recorder" stops, and the **F4** key contains the keystrokes DIR **↵**. To *play back* the macro, just press **F4**. Each time you press **F4**, the command you recorded (DIR) is played back and gives you a directory listing, just as if you had typed it on the keyboard.

As a matter of fact, the program (whether it is DOS, a word processor, or any other program) actually "thinks" it is still you typing away out there. From the program's point of view, there is no difference between normal typing and macro "playback."

This is the *interactive* way of recording a macro: you record the macro while doing your work as usual and you immediately see what the macro will do. There is another way to record macros or to make changes to the ones you record interactively. This involves the built-in *macro editor* which is described in Chapter 5.

Now, that was simple—and in principle that’s all there is to it. Of course, macros can be more sophisticated or they wouldn’t be of much use. But even with complicated macros, it all boils down to these simple steps:

1. Turn on recording: **Alt / M B**.
2. Press the key to hold the recording.
3. Enter the macro using the keyboard as usual.
4. Turn off recording: **Alt / M E**.
5. Press the key again and the recorded macro plays back.

DOS Commands Made Easy

When you are working with DOS or other programs, such as WordStar or Lotus 1-2-3, there are a number of commands you will probably use repeatedly. Some examples:

DIR to view the contents of your disk
 CHKDSK to check the disk status
 COPY to copy files
 MKDIR to make a new directory
 WS to start WordStar
 LOTUS to start Lotus 1-2-3

It would be nice to avoid typing these commands over and over; pressing a single key would be much more convenient. Easy! You have already recorded the first command, DIR, in **F4**; let’s record the other five. Press

Alt / M B F2	CHKDSK	Alt / M E
Alt / M B F3	COPY	Alt / M E
Alt / M B F5	WS	Alt / M E
Alt / M B F6	LOTUS	Alt / M E
Alt / M B F7	MKDIR	Alt / M E

Now, each time you press **F2**, you will run the CHKDSK program, **F3** will start copying, etc. If you get this message from DOS when you try these macros:

Bad command or file name,

it just means that DOS can’t find the command (for example, if WordStar isn’t on the current directory). All a macro does is record keystrokes—it can’t actually execute commands. Some of these macros execute immediately when you press the function key because you entered a **Return** (RETURN) into them, while, for ex-

ample, **F3** (COPY) waits and lets you enter a space, then the name and destination of the file you want to copy, then expects you to press **↵** before executing. (This special kind of macro uses *fields*, which are described fully in Chapter 5.)

These six macros, of course, are just small examples of what you can make macros do for you. Remember that a macro will do anything you can normally do on the keyboard.

Which Keys Can You Use?

Once you start using macros extensively, you will need more than the ten function keys **F1** through **F10**. SuperKey lets you use almost any key on the keyboard for macros with the following *shift-state* keys:

Alt **Ctrl** **⇧** (either shift key)

A key combination is achieved by holding down one of these three shift-state keys while pressing another key. You know this technique from a normal typewriter: to produce uppercase letters, you hold down the shift key, then you type the letter. Same thing with the **Alt** and **Ctrl** keys: to produce an **Alt F6** combination, for example, you hold down the **Alt** key and then press **F6**.

In this handbook, key combinations are symbolized by two keys close together:

Ctrl 9

This means: hold down the **Ctrl** key and press the **9** key. Sequences of separate characters are separated by white space, as shown here:

E ↵

This means: first press and release **E**, then press and release **↵**.

Although you *can* use normal alphabetic or numeric keys such as **D** or **5** for macros if you wish, you will get a warning if you try it. This is to protect you from inadvertent use of these keys. We suggest that you primarily use:

- the function keys **F1** through **F10**, alone or with **⇧**, **Ctrl**, or **Alt**
- the dedicated function keys **Home**, **End**, **PgUp**, **PgDn**, **Ins**, and **Del**, alone or with **Ctrl** or **Alt**
- **A** through **Z** and **1** through **0** in combination with **Ctrl** or **Alt**

This gives you a total of 130 keys for macros, and that should be plenty. The reason we discourage the use of normal keys for macros is that it could easily cause confusion if, for example, your **C** key no longer produced a C, but started a COPY. Better use **Alt C** or **Ctrl C** for such a macro.

There are a few keys and key combinations that *cannot* be used for macros; please check page 143.

Skip A Macro

While we are on the subject, you might find yourself in a situation where you have recorded a macro in a certain key, but now you want to use the original meaning of that key. No problem—SuperKey has a command that lets you skip a macro and access the original key.

If you have recorded the macros we have suggested, your **F3** now contains a COPY command. If you want to use **F3** as usual (to replay a command you give DOS), simply activate the skip command before you press **F3**, as follows:

Alt / C K F3

This will skip the macro in **F3**, and the key will have its usual effect.

If You Don't Like Menus

You might think it's a bit too much to have to press four keys just to skip one macro key. Well, there is an easier way to activate the skip command. The **⌘** key (the “back quote” located just above the right-hand **⌥**) will do the same as the **Alt / C K** sequence.

In fact, each of the SuperKey commands you use often can be activated by a single keystroke. The drawback to single-keystroke commands is that they have to be learned and memorized—and they are not terribly obvious. So we'll stick to the menu system in this chapter. We describe how to use SuperKey without menus in Chapter 3 and throughout the alphabetical lookup in Chapter 7.

How to Stop a Macro

Suppose you want to stop and get out of a macro recording—maybe because you started it by mistake, you lost your way, or you made some errors and want to start over.

Or suppose you want to stop macro playback. Say, for example, you made a macro “call itself.” Now it runs in an infinite loop, like a dog chasing its own tail.

The *stoP* command will stop a macro in its tracks. Just press

Ctrl Esc or **Alt / O P**

and playback, as well as recording, stops immediately. Anything you have recorded is discarded—it's as if it never happened.

Save Macros On Disk

The macros you have made are all stored in SuperKey's *macro memory*. The *macro memory* is a part of your computer's main memory that SuperKey has reserved for macros. However, if you don't save your macros on disk, they will be lost when you reboot the computer (by pressing **Ctrl Alt Del**) or when you turn it off.

If you want to use the macros later, they must be stored in a file on disk. To do that, press **Alt /** to see SuperKey's main menu. Move the cursor to **Macros** and press **←** or just press **M**.

Now select **Save**. Move the cursor to **Save** and press **←** or just press **S**. The window shown in Figure 2-3 appears, requesting a name for the file in which to save the macros. Enter a file name, followed by **←**. This name can be up to eight letters long. It's a good idea to let the name indicate the purpose of the macros in it. The macros we created for DOS could be called **COMMANDS**, since they contain DOS-command macros.

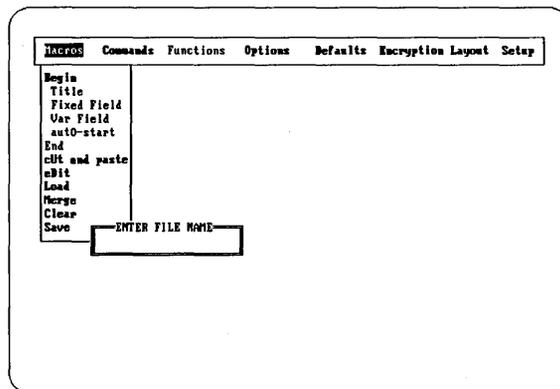


Figure 2-3 Save Macro

All the macros currently in macro memory are saved under the same file name when you select **Save**. If you want to save macros in separate files, you have to

reset the computer for each file, enter macros *individually*, then save them under different file names. File names are discussed in detail on page 123. Right now you just need to know that a file name consists of a first name, which is the one you enter, and a last name of up to three letters, separated from the first name by a period. The last name of a macro definition file is automatically MAC, so if you enter the name COMMANDS, the file will be called COMMANDS.MAC on disk. If you don't like .MAC you can enter a full name, complete with last name, for example COMMANDS.DEF (remember the separating period).

The macros in SuperKey's macro memory have now been written to disk, and you can reset or turn off the computer without losing them. By default, all .MAC files are written to the same disk and/or directory where the SuperKey program is stored. If you want to change this, use the Setup menu. See page 27.

Clear Macro Memory

Suppose you want to use your keyboard as usual, without any macros. The simplest way to do this is to clear all macros from memory. Now that you have saved your macro definitions in a disk file, you can safely *clear* them, since you can always reload them later (see the next section). Press

Alt / M C

to select **Macros Clear** from the menu. All macros in memory are now erased. .

Load Macros From Disk

Now let's read back the macros you just saved on disk. Press

Alt / M L

to select **Macros and Load**. Enter the file name you used when you saved your macros. The macros in the file will be loaded into memory and become active again.

If there are any macros in memory, they are replaced by the macros you load. There is a way to *merge* two or more macro files together in memory; see page 148.

Delete A Single Macro

What if you want to delete just one macro and leave the rest intact? You simply record an “empty” macro in that key, by pressing the *key itself* when recording. Let’s delete the macro in **F4**. Press

Alt / M B F4 Alt / M E

The macro in **F4** is deleted, and the normal use of the key is restored.

Macro Titles

Giving your macros *titles* (a longer description than the file name) helps you remember what each macro is used for and also makes it easier for others to use macros that you design. Once you have given a macro a title, it will be displayed along with the name of the key in the shoW titles window (see the next section).

Let’s redefine **F4** (the DIR macro) and give it a title. Press

Alt / M B F4 DIR

to define the macro as the DOS DIR command. SuperKey will ask you if you want to redefine the **F4** key. Answer “Y”. Then press

Alt / M T

to make the title. A window appears, where you enter the title you want to give the **F4** key. A good title might be the following:

List of files on this disk

You can enter up to 30 characters. When you press **Enter**, the window closes, and the title is recorded in the macro. Finally, press

Alt / M E

as usual to end the macro.

Show Us Your Macros

Although macros are indeed very useful, this is true only as long as you know they are there—and even more important, what each of them does. SuperKey therefore provides a window that gives you this information. Select shoW titles from the Commands menu:

Alt / C W

A window now appears, listing the keys currently defined as macros and their titles. If there is more than a page full, you can page through them with the **PgDn** and **PgUp** keys. You must press **Esc** to close the window before you continue working. As long as the shoW titles window is on the screen, you are not in touch with the program underneath, and you cannot play back macros.

Ready-Made Macros

One of SuperKey's strong points is its ability to create customized macros for each application program you use. Some samples of ready-made macros for various uses are provided on the distribution disk in the files with the last name MAC (for MACro file). To find out about the macro, load the file into SuperKey's macro memory. Information about each of these files can be found in Appendix D. Help information will appear in a window.

If you followed our example earlier, you have already created your own set of macros for use with DOS and saved it in the file COMMANDS.MAC, but the file DOS.MAC on the distribution disk contains a more sophisticated set of macros for use with DOS.

To load the DOS macros, press **Alt /** to access the main menu, then select first Macros, then Load. When asked for the name of the file to load, enter:

DOS **↵**

SuperKey supplies the last name MAC automatically. The DOS macros are now read into memory, ready to use. But something unexpected happens when you load this macro: a window with a helpful description of the macro pops up in the middle of the screen. This is a *display macro*, a macro that is not sent to the computer via the keyboard as usual, but instead displayed on the screen. But how did it get there? You didn't press anything. Well, it is also an *autO-start* macro; one that always starts automatically when the macro file is loaded. In Chapter 5 you will learn how to make your own display and autO-start macros, but let's play with this one for a minute.

What you see is only the first page of help on your DOS macros. You can press **PgDn** to see more, and **PgUp** to page back again. Once you have read the text, press **Esc** to close the window. You *must* close it before you can do anything else. As long as the display macro is on the screen, you are not in touch with the program underneath. Like the shoW titles window, display macros are not transparent.

Before you go on, maybe you should try some of these DOS macros to get really familiar with using SuperKey macros. Remember that **Alt / C W** will display a list of currently loaded macros and their titles. You can press **F1** to get help.

By now, you should have a basic idea about macros and how to use them. More advanced information can be found in Chapter 5.

The Functions Menu

Functions make a macro perform various tasks when it is played back, such as displaying the time or date, or switching the screen off and on. You record a function in a macro by selecting it on the **Functions** menu *while you record the macro*. This is the *only* way you can get to the **Functions** menu—while you are recording a macro. Otherwise, the menu will appear to be dimmer than the other menus and won't be accessible.

Let's take a look at some functions you'll probably use often. Detailed information about all SuperKey functions can be found in the alphabetical lookup in Chapter 7.

Time Stamping Your Data: Time Function

Suppose you keep notes on your telephone conversations. If you have a real time clock installed in your computer, it would be useful to have a macro that prints the time, followed by a **↵** to let you start writing the note on the next line. Let's record such a macro in **Alt T** (for time). Press

Alt / M B Alt T Alt / F

This starts recording the macro, brings up SuperKey's main menu, and opens the **Functions** pull-down menu, which is shown in Figure 2-4. Now press **T** to select the **Time** function (or press **↓** to move the bar down to **Time**, then press **↵**, if you prefer). You have now recorded the time function in the macro.

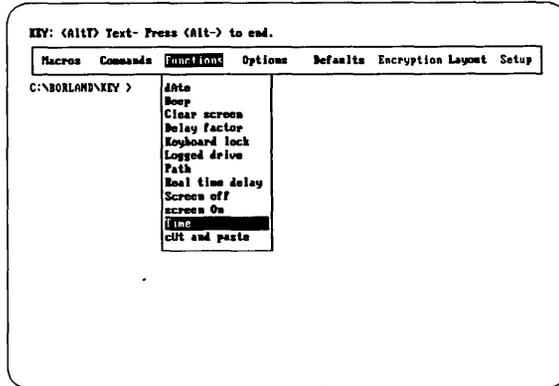


Figure 2-4 Functions Menu

Finally, press

← Alt / M E

to stop recording.

Each time you now press **Alt T**, the time will be read from DOS and printed out, followed by a return.

May We Have Some Privacy, Please: The Screen Off Function

Have you ever been in the middle of a confidential document when someone walked in and looked over your shoulder? Embarrassing, right? What does one do? Ask the person to leave, hurry to save the work, or what?

No problem. Just make yourself a couple of macros that turn the screen off at the click of a button, then back on again, without disrupting your work. For example:

Alt / M B Ctrl O Alt / F S Alt / M E

Alt / M B Ctrl I Alt / F O Alt / M E

Now all you need to do to hide your work from prying eyes is to press **Ctrl O**. Press **Ctrl I**, and your screen display is back.

Freeze Those Keys: The Keyboard Lock Function

For even more security, you could add a keyboard lock function to the screen off macro. Then you could safely leave your computer, and no one can gain access to it. Press

Alt / M B Ctrl O

Alt / F K Alt / F S

Alt / M E

When you play back this macro (by pressing **Ctrl O**), you are prompted for the keyword that will unlock the keyboard again. Enter it once, then re-enter it for verification. Press **↵** each time.

Your keyboard is now locked. The next thing you recorded in the macro was the Screen off function, so the screen is turned off.

The only way to gain access to the computer now is to enter your secret keyword again. This time, you need not press **↵**. As soon as the keyword is recognized, the keyboard is unlocked. You can then press **Ctrl I** to turn the screen back on.

For even greater security, you can activate the keyboard lock function on the DOS command line. This means you can load SuperKey from a batch file (such as AUTOEXEC.BAT) and lock the keyboard at the same time. Even if someone comes along and turns off your computer, the keyboard will lock again when the computer is turned back on.

Here's what to do: Insert the following line in your AUTOEXEC.BAT file:

```
KEY/FKnnnnnnnn
```

where nnnnnnnn is your password.

Now, when you reboot your computer, your keyboard will lock automatically. All you have to do is type your password to gain access.

Another alternative would be to include the keyboard lock function *and* your password in a macro file. That way, you don't have to enter your password each time you want to lock your keyboard.

For example, you could program the **Alt L** key sequence to lock your keyboard with the password *freeze* as follows:

Using a text editor (or the SideKick Notepad) create the following file:

```
<BEGDEF><AltL><CMD> FKfreeze <CMD><ENDDF>
```

Load this file into SuperKey, then press **Alt L**. Try your keyboard—it should be locked. Now type *freeze* to unlock it again.

Encryption

Now let's look at some other SuperKey features you are likely to use a lot. The first one is *file encryption*. Encrypting a file means scrambling it so no one can make use of it—that is, unless they have the keyword that will *decrypt* (or unscramble) it again.

SuperKey offers two distinct file encryption schemes:

- A fast method which should cover the need of most users. Unless you are dealing with government secrets or other high-security data, you'll be safe with this. If you want to use this type of encryption, you should start SuperKey with the command

```
KEY ↵
```

because this algorithm is in the file KEY.COM.

- The DES file encryption scheme. This encryption complies with the Data Encryption Standard issued by the National Bureau of Standards. It provides the maximum security needed for highly sensitive data, prone to threat by intruders with access to sophisticated code-breaking facilities. The security is not without cost, however—this encryption is significantly slower than the other.

If you want to use DES encryption, you should start SuperKey with the command

```
KEYDES ↵
```

because this algorithm is in the file KEYDES.COM. Now, let's try it. If you haven't yet done so, you should copy the file LETTER.TXT from the distribution disk onto your work disk. Then enter this command to display the letter on the screen:

```
TYPE LETTER.TXT ↵
```

You will see that the letter is perfectly readable. Now invoke SuperKey's encryption by pressing

```
Alt / E
```

The Encryption menu first lets you choose between **E**ncrypt file and **D**ecrypt file. Encryption takes a normal file and turns it into unreadable ciphertext. Decryption reverses the process—takes the ciphertext and turns it back to normal.

Select **E**ncrypt file. Enter the name of the file you want to encrypt:

```
LETTER.TXT ↵
```

Now enter your keyword. It can be anything you like (up to 30 characters long), but be sure to choose something you have a good chance of remembering, because you can *only* decrypt the file with the keyword you used for encryption. After you've entered the keyword, you are asked to retype it to make sure you made no mistakes. If your second entry does not match the first, you must enter the keyword again. Finally, you are asked whether you want *Text mode* encryption. Don't worry about that now; just press  for No.

SuperKey now encrypts the file LETTER.TXT. It physically writes the encrypted file on top of the original one, so no trace of the original file is left on disk. The file is completely safe from prying eyes.

After you have encrypted a file, it's a good idea to make a backup copy of the encrypted file in case the original encrypted file is corrupted by tampering or by decryption with the wrong password.

Now tell DOS to TYPE the file again. All you get on the screen is garbage, because that's all the file contains, at least to people without the keyword.

To retrieve the original file, choose **Decrypt file** from the **Encryption** menu. Be sure to *use the same keyword* you used for encryption. The encryption process is reversed, and you can again TYPE the original file on the screen.

Encryption works with any file or program. It even provides a way to turn a binary file (a program, for instance) into an encrypted text file. This text file can then be transmitted over the telephone line with any communications package. You can even use the bulletin boards on the public networks—only the authorized recipient with the right keyword can turn it back into a program.

Please read more about encryption on page 103 before you start using this feature extensively. There are a few things you should be careful with. In particular, *never try to decrypt a file that has not been encrypted.*

DOS Command Stack

Another useful SuperKey feature is the *DOS command stack*. This little "tape recorder" runs all the time, continually recording all the commands you give to DOS. You can see these commands by selecting **Command stack** from the **Commands** menu. Press

to open the command stack window. You will see the commands you have given DOS since you loaded SuperKey. Instead of having to retype DOS commands you use often, you can use the  and  keys to move the bar to the command you want to repeat. Then press  to send that command to DOS again, or edit the command using the , , , and  keys before you press .

Commands appear in reverse order from how you entered them, with the most recent command at the top of the stack. You can page through the stack with **PgDn** and **PgUp**. The stack contains the last 255 characters you entered on the DOS command line. Excess commands are deleted from the bottom of the stack as it fills up.

Press **Esc** to close the window without executing any commands.

Type-ahead Buffer

This is one SuperKey feature that you never actually see, although you use it all the time. Whenever SuperKey is loaded, the keyboard's buffer is expanded from the normal 16 characters to 128 characters. This means you can go on typing even while the computer is busy doing something else, like reading or writing on the disk. You should never hear the "keyboard buffer full" beep again!

Setup Menu

The Setup menu lets you specify where SuperKey looks for its macro and layout files and where it looks for files to be encrypted.

The standard (default) setup does not specify any disks or directories, so SuperKey will look for its files on the disk and/or directory where the SuperKey program is located.

If this is OK with you, you need not bother with Setup. But if you are using a hard disk and want your macros placed in a separate directory, or if you are using floppy disks and you can't fit the macro and layout files on your work disk, you can use Setup to tell SuperKey to look for its files elsewhere.

To open the Setup menu, press

Alt / S

As with all SuperKey menus, you select an entry on this menu by moving the bar up and down, and pressing **←** to select the desired entry, or you can just press the key for the uppercase letter in the entry you want.

Encryption Directory

When you select this entry, a window opens, asking for the name of the directory where SuperKey will look for files to be encrypted. The directory name you enter can be a full DOS path name (see Appendix E for an explanation of directories and paths), or it can just be a drive name, for instance A: or B:. If you just enter a drive name, the currently active directory on that drive will be used (DOS has an active directory for each individual drive).

When you press , the window closes and the directory or drive you enter becomes active. If you want SuperKey to find its files on the drive/directory where the SuperKey program is located, don't enter anything, just press .

If you want SuperKey to use the new setup next time you load it, you must record it permanently by selecting Save setup.

Suppose you only want to encrypt files on the root directory of the A: drive. You would then enter

    A:\ 

If you have a hard disk and want to encrypt files on a specific directory, say, \TEXT\LETTERS\PRIVATE, you would enter

    \TEXT\LETTERS\PRIVATE 

Or, to be more specific, you could include the drive name:

    C:\TEXT\LETTERS\PRIVATE 

If the window is not wide enough for the path name, it will scroll horizontally when you reach the right edge of the window. (This horizontal scrolling automatically takes place in all SuperKey windows.)

Macro Directory

This entry lets you specify the directory where your macro (.MAC) and layout (.LAY) files are stored. As with Encryption directory, the directory name can be a full DOS path name or it can just be a drive name.

If you are using a floppy disk computer and cannot fit your macro files onto your work disk, you can put them on the other disk and specify that drive name as the Macro directory. For example, if you want to place your macro and layout files on the B: drive, you would enter

    B: 

or, you could place them in a separate subdirectory on the B: drive, for example

B:\MAC

Alt / **S** **M** B:\MAC **↵**

If you are a hard disk user, you might want to collect all macro and layout files in a separate directory, for example \MAC. You would specify your Macro directory as

Alt / **S** **M** C:\MAC **↵**

Save Setup

Any changes you make to the setup become active immediately, but they are not permanently recorded in SuperKey. So if you want your new setup to take effect the next time you load SuperKey, you must press **S** to save it.

When you save a setup, the KEY.COM file must be on the disk and directory that was active when you first started SuperKey, because the new values are saved directly in this file. Please see page 9 for details.

The Options Menu

The main menu has an entry called **Options**, which is used to control several SuperKey features. A few of these features are described in detail here; see the alphabetical lookup in Chapter 7 for complete information about all the items on the **Options** menu. Press

Alt / **O**

to open the **Options** pull-down menu, which is shown in Figure 2-5. Most options are *toggles*: if you select an option that is currently ON, it will turn OFF, and vice-versa. You can also press **+** or **-** to turn an option ON or OFF, regardless of its current state. As usual, you select an item on the **Options** menu either by moving to it with **↓** or **↑**, then pressing **↵**, or by pressing the key corresponding to the capital letter in the item.

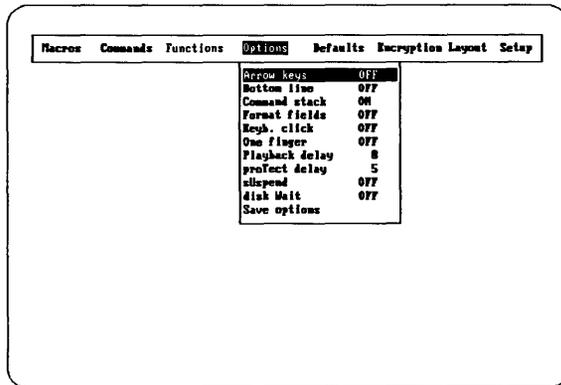


Figure 2-5 Options Menu

The two *delay* options need a little more input. When you turn them ON, you are required to enter the length of the delay.

One Finger Mode

The *one finger* option lets you enter commands which normally require you to press two or more keys at the same time as a *sequence* of keystrokes. This will come in handy if you only have the use of one hand or must use a mouth stick.

To reboot your computer, for example, you normally press

Ctrl **Alt** **Del**

all at the same time. That not only requires three fingers, but also two hands. If for some reason, you can't do that, switch on One finger mode. Then you can press the three keys individually, and the effect will be as if you had pressed them at the same time.

The normal way to turn on One finger mode is:

Alt **/** **O** **O**

There is another way to do it without pressing **Alt** **/**: You can pass the command on the DOS command line. Instead of starting SuperKey with the DOS command

KEY **←**

you would use the command

KEY /OO **←**

`/OO` is a *command line parameter* that is passed on to SuperKey. You can pass any command to SuperKey in this way at any time. Command line parameters are described in detail on page 77.

One last thing. If you would like SuperKey's commands to be single key-strokes instead of **Alt** and **Ctrl** key combinations, you can redefine all commands. See how on page 90.

Suspend

This is a way to temporarily deactivate SuperKey without removing it from memory, and without destroying macros or any of SuperKey's setup.

You simply press

Alt / **O** **U**

to sUspend SuperKey. It is still there, but it no longer monitors your keyboard. Therefore (since SuperKey is no longer able to receive its usual commands from you), the only way to turn OFF sUspension and get SuperKey back in action is to enter the **O** **U** command on the DOS command line:

KEY /OU ↵

Disk Wait

When programs read or write on a disk, they cannot also receive input from the keyboard, or from a macro. This usually doesn't cause any problems, because input is stored in a *keyboard buffer* until the program is ready to receive it.

Some programs, however, have a bad habit of throwing away characters sent to them while the disk is spinning (being read or written to). When you use such programs, you must stop and wait for the disk to stop spinning before you can type anything. You can tell SuperKey to do the same by switching on disk Wait. Press

Alt / **O** **W**

This only works on floppy disks. A hard disk spins all the time, so it makes no sense to wait for it to stop.

Bottom Line

When you record macros, SuperKey displays a line at the top of the screen so you can see what you're doing, and what you can do next. If your program also uses the top line of the screen for its status or entry line, you can move SuperKey's *interactive recording message* to the bottom line of the screen by switching Bottom line ON.

To do that, press

Alt **/** **O** **B**

The next time you record a macro, the interactive recording message will appear at the bottom of the screen. To make the change permanent, you must save your options by pressing

Alt **/** **O** **S**

Playback Delay

Macros are normally played back at full speed, which is close to 1000 characters per second. Some programs have problems receiving input at that speed. They may lose characters or start beeping furiously at you. If that happens, you can slow down playback speed by setting a Playback delay.

The Playback delay is 0 by default, which means no delay. Values from 1 to 999 introduce a delay between each character, thereby slowing down playback. The effect of a particular delay depends on the program that receives the macro, so you may have to do some experimenting to find the right values.

To set the Playback delay to 5, for instance, press

Alt **/** **O** **P** **5** **←**

If you want to make this delay permanent, you must save your options by pressing

Alt **/** **O** **S**

To a degree, the Playback delay is a matter of taste. Take WordStar, for instance. If you play back a macro with a lot of text at full speed, WordStar will start displaying exclamation marks on the screen. Everything is really OK; it's just that WordStar has trouble updating the screen while receiving input at full speed. As soon as the macro finishes, the screen will be corrected, and you'll see that everything went well. If you don't want WordStar to do that, however, you could set the Playback delay to 2.

Protect Delay

This option lets you protect your valuable screen from “burn in.” That’s what happens when the same image is displayed on the screen for an extended period of time. The screen’s phosphor becomes “worn” in places where characters are displayed, so after some time you’ll see faint after-images on the screen, even when it is turned off.

The way many of us use computers, they just sit there for long periods, waiting for us to enter something on the keyboard. During that time, there’s really no reason to have anything on the screen.

SuperKey will protect your screen by turning it off if there has been no activity on the keyboard or the screen for a certain time. Whenever you touch a key, or your program writes on the screen, the screen display comes back.

To activate the screen proTect option, press

Alt / O T

then enter the number of minutes you want the computer to wait before it turns off the screen. The default time is five minutes, and you can press

←

to accept that value.

As with the other options, you can make the screen proTect time interval you just defined permanent if you save your options by pressing

Alt / O S

Save Options

When you set your options as described above, they only stay in effect as long as SuperKey remains in memory. The next time you start SuperKey, the options are back to their defaults.

If you want your new Options setup to be permanent, you must deliberately save it with the command

Alt / O S

Remember: Whenever you save anything in SuperKey, the KEY.COM file must be present on the disk and in the directory from where you started SuperKey.

How To Use SuperKey Without Menus

Once you become familiar with SuperKey, you may want to take a few shortcuts and circumvent the menu system.

All the frequently used SuperKey commands, such as **B**egin macro, **E**nd macro, **s**ho**W** titles, **C**ommand stack, etc., can be activated with **Alt** or **Ctrl** and one other key. The drawback of these one-key commands is that they are not very obvious; they have to be learned. But once learned, they are faster to use than the menus.

There is a good reason why these one-key commands are so cryptic. We could have chosen mnemonic commands like **Alt B** for **B**egin macro, **Alt E** for **E**nd macro, etc., but that would render a lot of good key combinations unusable for the macros you create. So we let SuperKey use commands like **Alt =**, **Alt -**, etc., and leave the alphabetic keys free for you to use. If you still prefer to use mnemonic commands, you can change them as described on page 90.

Table 3-1 lists the commands as we have installed them, along with the commands you would use in the menu system. Note that **Alt -** uses the **-** key on the upper row of keys, next to the **=** key—not the **-** key on the numeric keypad.

Table 3-1 One-key SuperKey Commands

Command	One-key Command	Menu System Keystrokes
Menu	Alt /	
Begin macro	Alt =	Alt / M B
Title	Alt -	Alt / M T
Fixed field	Ctrl	Alt / M F
Var field	Ctrl -	Alt / M V
auto-start	Alt -	Alt / M O
End macro	Alt -	Alt / M E
cut and paste	Alt	Alt / M U
Command stack	Alt \	Alt / C C
skip macro	~	Alt / C K
stop	Ctrl Esc	Alt / C P
show titles	Alt PrtSc	Alt / C W

C H A P T E R 4

Using SuperKey With Your Favorite Program

This chapter provides information about using SuperKey with your text editor or spreadsheet. For information about using SuperKey with other Borland products, see Appendix A.

The distribution disk contains a number of ready-made macro files for use with various popular word processors and spreadsheets (see Appendix D for more information). Don't be upset if yours is not among them—you'll not miss much. They are merely meant as an inspiration for your own personal macros, not as something we think you will actually use (not for long, anyway).

After all, the whole idea with macros is to let you create your own personal set of commands for your programs. It would be ridiculous if we tried to do that for you; SuperKey lets you decide for yourself. Any time you find yourself repeatedly typing the same words, commands, or numbers, create a macro to make things easier.

One idea is to create and use the same macros to do similar things in different programs. For example, if you have different text editors, they are sure to use different commands for things like *search-and-replace*, *save-and-resume*, etc. With SuperKey macros, you can use the same command no matter which editor you use.

Take WordStar, for instance. It uses the command sequence **Ctrl** **K** **Ctrl** **R** to read a file into your document. If you want to do the same in Volkswriter, you must press **Alt** **F2**. If you use both these word processors, wouldn't it make life easier to have a *read file* macro like this for WordStar?

Alt **/** **M** **B** **Alt** **R**
Ctrl **K** **Ctrl** **R**
Alt **/** **M** **E**

And, in another macro file, this one for Volkswriter:

Alt **/** **M** **B** **Alt** **R**
Alt **F2**
Alt **/** **M** **E**

When you run WordStar, load the file with the WordStar macros, and when you run Volkswriter, load the VW macro file. Then you can use the same command **Alt** **R** to do a read file in either word processor!

To make things even easier, you could make a .BAT file load the macro files for you automatically. Here's a file called WS.BAT:

```
KEY WS /ML  
WS  
KEY /MC
```

This .BAT file will first load the macro file WS.MAC, and then start WordStar. When you exit WordStar, the .BAT file will clear the macros from memory and return your keyboard to normal. A similar VW.BAT file will do the same for Volkswriter:

```
KEY VW /ML  
VW  
KEY /MC
```

Note that we are using SuperKey command line parameters here—passing commands directly to SuperKey on the DOS command line. See page 77 for more information about command line parameters.

Cut and Paste

Have you ever wanted to grab text from somewhere else and move it into your word processor? SuperKey's cut-and-paste feature is easy to use and lets you cut text from your screen and paste it where you want it. Let's take a simple example. Exit to the DOS prompt (**A**> or similar) and type

```
DIR ←
```

This displays a list of all your files on the screen. Now let's move this list into your word processor. But first, press

Alt / M U

to activate SuperKey's cut and paste feature.

Then press the key to hold your "cutting." For example:

Alt C

Now use the arrow keys **↓**, **←**, **→**, and **↑** to move the cursor to where you want to start cutting. Then press

B

to begin cutting. Use the arrow keys again to move to the end of the cutting. You'll see it being marked on the screen as you move along.

When the cursor is at the end of the block of text you want to cut, simply press

←

to end cutting. Your cutting is now in the **Alt C** key.

Get into a file with your word processor, move to where you want the text pasted, and press

Alt C

Voila! The cutting is neatly pasted into your document.

Some word processors might wince a little when you paste, especially if it's a large block of text. This is because they don't like to receive characters at the high speed SuperKey uses (nearly 1000 characters per second). If that happens, you can slow down SuperKey as described on page 155.

Using SuperKey with a Spreadsheet

When working with a spreadsheet, most people use the same commands over and over again. That amounts to a lot of keys pressed during the day. SuperKey macros can reduce these keystrokes to a minimum.

The complexity of a spreadsheet program often baffles the casual user. Even though you really only need a handful of commands to do a few simple things, these commands must be found in different places in an often deeply nested hierarchy of commands.

SuperKey lets you design personalized macros for everyone who uses the spreadsheet program, turning complex commands into single, easy-to-remember keystrokes.

You can even use display-only macros to write personal help screens to help each user carry out specific jobs; display macros can provide help screens tailored for each job, or even for each person using the spreadsheet. If one person does data entry, and another person produces graphs, they need different help. See page 98 for more information.

Take the 1-2-3 macros on the distribution disk, for example.

Press

Alt / **M** **L** 123 **←**

to load this macro file. A display macro with information about these macros is shown automatically when you load the file. Press **Esc** to close the display macro. Then start 1-2-3. You can press **Ctrl** **F1** whenever you want to see the help screen again. These macros are not sophisticated, but they *do* show you how to turn some of the commands you use all the time into single mnemonic keystrokes.

So, for example, whenever you want to switch between automatic and manual recalculation, you don't have to go through four levels of 1-2-3 menus; simply press

Alt **M**

for manual recalculation, and

Alt **A**

when you want to switch back to automatic.

Same thing with global worksheet protection, for instance. If you normally have this feature on, but occasionally need to access one of the protected cells, you have to go to the main menu, then select Worksheet, then Global, then Protection, and finally Enable or Disable. It's only five keystrokes, but anyone who is not really familiar with 1-2-3 has to laboriously dig those keystrokes out of the menus. That's confusing and takes time.

With SuperKey, you could just press

Alt **D**

to disable protection, and

Alt **E**

to enable it again.

Make a 1-2-3 Macro

The sample 1-2-3 macros we provide, although useful, are only meant as examples, and deal with some general commands. Once you get inside a worksheet, what you want to do is highly individual. It depends on how your worksheet looks.

It's easy to make your own macros. All you do is press

Alt =

to turn on SuperKey's macro recording. Then press the key you want to hold the recording, and go to work. Anything you do now is recorded in a macro.

When you finish whatever you are doing, press

Alt - to stop recording.

You can now repeat everything you did by pressing the one key in which you recorded the macro.

If you don't want SuperKey to print its message at the top of the screen while you record a macro, you can move it to the bottom of the screen. Press

Alt / O B

to turn on SuperKey's *bottom line* option. In 1-2-3 you will definitely want to do that, because 1-2-3 also uses the top line for its messages.

Playback Delay

Most spreadsheets can't handle the speed at which SuperKey sends characters, so you'll want to use a playback delay in your spreadsheet macros. Try a delay of 5; if that doesn't work, increase the number gradually. See page 155 for more information.

Graphics

Whenever SuperKey displays anything on the screen, it changes the display to *text mode*. So if you have a graph on the screen, it will turn into gibberish. Also, when SuperKey is used over a program that is in graphics mode, the command line is not displayed while you define a macro.

But don't worry. When SuperKey closes its window, it will turn the screen back to graphics. In any case, there isn't much for SuperKey to do while graphics are displayed. If you don't want this to happen, use the KEYINST installation program to blank the screen in graphics mode. See Appendix C.

Advanced Macro Use

This chapter takes a more detailed look at macros. We'll cover auto-start and display macros, and discuss how to use variable and fixed fields in macros. We'll also show you how to use the macro editor to create more complicated macros and edit ones you already have.

Auto-Start Macros

An auto-start macro is a macro like any other, except that it executes automatically each time the file containing it is loaded.

Let's add a new macro to the DOS macros on your distribution disk (described on page 21). If you haven't already, load the DOS macros into your computer. You will see help text displayed automatically—an *autostart macro*, which in this case is also a *display macro* (described in the next section). The new macro will display the active directory and the system time at the DOS prompt. We'll put it in an out-of-the-way key like **Ctrl F10**, because it is not a macro you are likely to use from the keyboard; it's just there to execute automatically when you load the macro file. Record the macro as usual:

Alt / M B Ctrl F10 PROMPT \$P \$T \$G **↵**

Now, to make it autO-start, select autO-start on the Macros menu:

Alt / M O

and finally end the macro as usual:

Alt / M E

Now, save your DOS macros with this change in a file called NEWDOS.MAC:

Alt / M S NEWDOS ↵

And finally load your new DOS macros:

Alt / M L NEWDOS ↵

The macro you just defined plays back automatically when you load the file. The DOS macros on your distribution disk include a macro that displays help text automatically (autO-start). But a set of macros can have only one autO-start macro, so whenever you define a new one, the old autO-start macro turns into a plain old macro.

If you still want the help text displayed automatically whenever you load this file, you can let the new autO-start macro activate the help text in **F1**. To make **Ctrl F10** activate **F1**, just add **F1** to the end of your macro:

Alt / M D Ctrl F10

With the arrow keys, move to the end of the macro file and type

< F 1 > Alt Esc Y

Now you must save this file so the next time you load it, you will get the display macro

Alt / M S NEWDOS Y ↵

These are just a few simple uses of autO-start macros, but they can be used for much more. You could, for example, design large self-running demos or tutorials with SuperKey, having one set of macros chain another, which then autO-starts, chaining yet another, and so on as far as your imagination takes you.

Display Macros

Display macros are a SuperKey specialty which really add value to your computer. With them, you can design your own individual help system for any of your favorite programs. For instance, you might want to create a little menu that shows all your macros, their titles and what they do, so you can share them with your friends or co-workers.

Since display macros are not merely strings of text or commands being sent to the computer, but text that is displayed on the screen, you will want to format them nicely. You must therefore use a text editor to write your display macros. This can be any text editor producing standard ASCII text files, such as EDLIN, WordStar in non-document mode, SideKick's Notepad, or the Turbo Pascal editor.

Many other editors can produce standard text files. If you are not sure what your editor does, ask DOS to TYPE one of your text files on the screen. If the text is displayed legibly, it's a standard text file. Even if your editor does not produce standard text files, it may have a way of doing so, perhaps by *printing* the text to a disk file.

Let's try an example. Enter your text editor and create a new document. You could call it DISP.MAC, since it's going to be a display macro. On the first line of this file, you enter the information SuperKey needs to identify the file as a display macro. Enter everything exactly as shown below, including angle brackets. Both upper and lowercase letters can be used for all material enclosed in angle brackets. The first thing you should enter is

```
<BEGDISP>
```

This is a *symbol* that tells SuperKey that a display macro starts here. Right after that symbol, enter the name of the key in which you want to record this macro. Enclose the key label in angle brackets, so it also becomes a symbol. For example:

```
<AltF10>
```

This symbol identifies the key combination **Alt F10**. The next information you'll enter is the position and size of the window that will display this macro:

```
25,9,30,8.
```

The first two figures specify the column and line of the upper left corner of the window, the third figure is the number of columns (the width), and the last figure is the number of lines (the height). So here we have specified a window with its upper left corner at column 25, line 9. It is 30 characters wide and 8 lines high. Put a period after the window size. The first line of your file should now look like this:

```
<BEGDISP><AltF10>25,9,30,8.
```

Beginning on line 2, you should now type the text you want the macro to display. Remember that you have specified a window width of 30 characters, so this should be the maximum length of your lines. Longer lines will be truncated when displayed.

Although you have specified the window height to be 8 lines, this does not mean that you are limited to writing 8 lines of text. If you type more than that, SuperKey automatically divides the text into pages, and you can page through it with **PgDn** and **PgUp**.

Now type 10 or 12 lines of text. For example:

```
Welcome to MoneyManager
```

```
With MoneyManager, you will  
never be short on cash, nor  
will you ever exceed your  
personal credit limit.
```

```
The following pages will show  
you how to make MoneyManager  
perform this incredible feat,  
so please stay with us...<ENDDDEF>
```

The last thing you should type is **<ENDDDEF>**. This symbol tells SuperKey the macro ends here.

Now save this file on disk and load the file into macro memory:

```
Alt / M L DISP.MAC ↵
```

Then press **Alt F10**. The text you just wrote will pop up in the middle of the screen, with its upper left corner in column 25, line 9.

Note that the upper left corner you specified will be the upper left corner of the frame surrounding the text, and the number of lines and columns you specified will be the number of frames within the window. So, counting the frame, the whole thing takes up two extra columns and two extra lines. (SuperKey gives you a frame to enhance the display of your text.)

Pages

If you copied the example faithfully, you will see that your 8-line window breaks the text in the middle of a paragraph. You can avoid that by forcing a page break where you want it. The way to do that is to enter a Control-L where you want the page break. Call up your editor again and place a symbol representing a Control-L between the two paragraphs:

```
<BEGDISP><AltF10>25,9,30,8.  
Welcome to MoneyManager
```

```
With MoneyManager, you will  
never be short on cash, nor  
will you ever exceed your  
personal credit limit.  
<CtrlL>
```

```
The following pages will show  
you how to make MoneyManager  
perform this incredible feat,  
so please stay with us. <ENDDF>
```

If you use SideKick, you could insert a Control-L character in the document with the Notepad by pressing

```
Ctrl P Ctrl L
```

Save the file and load it into SuperKey again. Press **Alt F10** to see the result. Only the first page is displayed. Press **PgDn** to see page two, and **PgUp** to return to page one.

A display macro can be as long as your macro memory allows (determined with the KEYINST program; see Appendix C), with approximately 65,000 characters as the limit. Each window can be as small as one character by one line, or as large as 78 characters by 23 lines (the entire screen, when you add the frame), macro memory permitting.

If you are running out of macro memory space, SuperKey will shrink the display window to conserve memory and still allow the macro to be displayed.

Freeze Your Keyboard with a Display Macro

You could use a display macro to freeze your keyboard and cover the entire screen to show that the computer is in use. Use your text editor to write this display macro:

```
<BEGDISP><CtrlL>1,1,78,23  
I'M OUT TO LUNCH  
-----  
Don't bother to call me unless it's VERY important!  
<ENDDF>
```

Now save it in a file, then merge it into macro memory. Press

```
Alt / M M
```

and enter the name of the file containing the display macro. The file is now read into memory, but the macros already in memory are still there also.

Unlike loading a new macro, merging macros does not clear other macros from memory, but lets you *combine* many macro files into memory “in parallel.” See page 148 for more information about merging macros.

Now let’s make the **Ctrl O** macro activate the display macro in **Ctrl I**:

Alt / M B Ctrl O

Alt / F K Ctrl I Esc

Alt / M E

When you press **Ctrl O** and enter your keyword, the display macro will cover the screen, which is just as good as turning it off. When you return from lunch, you’ll enter your keyword to unlock the keyboard, then press **Esc** to end the display macro and get your screen back.

By the way, you could save the macros you just merged. That way, you would only have to load *one* file in the future. Press

Alt / M S

and enter the name of the file in which you want to save your combined macros.

Transparent Display Macros

To make your display macros more efficient and easy to use, you can include a **<TRANSP>** symbol in a display macro. This convenient method allows the macro to be cleared when any key is pressed, and also “passes through” the keystroke to whatever application you’re currently using. In other words, that keystroke is “transparent” to SuperKey.

How might this come in handy? Suppose you have a display macro that displays a menu of numbered selections. If you include the **<TRANSP>** symbol in that macro, a user could press the number for the menu item s/he wants. The menu would disappear (since the macro is cleared) and the selected menu item would appear on the screen.

The **<TRANSP>** symbol can be placed anywhere in the display macro file after the dimensions of the display window. An example:

```
<BEGDISP> <CtrlF1> 1, 1, 20, 10. <TRANSP>
```

Your distribution disk contains two macros that include the **<TRANSP>** feature—**DOSTRNSP.MAC** and **123TRNSP.MAC**. These are the same as **DOS.MAC** and **123.MAC** except that they are cleared when you press a key. Try them.

Enhanced Text

The display macro window sports three different colors: one for the frame and two different colors for the text. On a monochrome or B/W monitor, this translates into different intensities. The actual colors or intensities vary; you can use them as they are, or you can use the installation program to choose other colors. See page 186.

In any case, you select different colors by entering control characters into the text: Control-B's switch between the two text colors, and Control-D's switch the frame color on and off.

You can either actually enter control characters (by pressing, for instance, **Ctrl B**, if your editor lets you, or type the symbols shown here. (WordStar won't let you enter the actual control characters, but SideKick's Notepad will.)

<CtrlB> for Control-B, and

<CtrlD> for Control-D

For example:

```
<BEGDISP><Alt1>1,1,50,5.  
This is normal text.  
<CtrlB>This is the secondary text color<CtrlB>  
<CtrlD>And this is the color of the frame<CtrlD>  
All <CtrlB>three<CtrlB> on <CtrlD>one<CtrlD> line  
You <CtrlD>can <CtrlB>even<CtrlB> nest<CtrlD> them<ENDDF>
```

Now copy this example, save it in a file, and load it into SuperKey to see the effect on your screen.

Titles and Auto-Start

Finally, let's see how we can give the display macro a title, and how to make it auto-start.

The trick is to use SuperKey *symbols*. You have already used a couple of symbols: <BEGDISP> to signal the start of the macro and <ENDDF> to signal the end of it. Meet two more:

<TITLE> is used to mark the beginning and end of a title

<AUTO> makes the macro auto-start

Let's use both of these symbols in the DISP.MAC file we made before. Re-enter your editor and add the following to the end of the first line of the file:

```
<TITLE>MoneyManager Instructions<TITLE><AUTO>
```

The text between the two <TITLE> labels is the title of the macro that shows up in the shoW titles window. Now the first line should look like this:

```
<BEGDISP><AltF10>25,9,30,8.<TITLE>MoneyManager Instructions<TITLE><AUTO>
```

Save the file with these changes, load it into SuperKey, and watch what happens.

Fields

Macros can do more than just play back commands and display pretty screens. Suppose you want to copy a number of files from the logged disk to the B: drive. You could repeat this command for each file:

```
COPY FILENAME B: 
```

Let's make a macro do most of that for you; the only thing you'll have to enter each time is the file name. Let's start recording the command in  :

```
      COPY
```

It would be silly to actually put the file name into the macro; then you would have to make a macro for each file you want to copy. SuperKey provides a way to let you enter the file name when you play back the macro. A *field* will temporarily stop macro playback and let you type in something. When you press , the field is terminated, and the rest of the macro plays back.

SuperKey has fields of either *variable* or *fixed* length. You'll want to use a *variable field* here, because a file name could be as short as one character, or as long as 14 (actually 64 including a file path).

So, continue your   macro by pressing

```
    
```

to start the variable field. Note the space, represented by a blank key, that separates the file name from the COPY command. Then press

```
   
```

to end the field. The reason you have to both *begin* and *end* the variable field is that you could put something in between start and end to make the field do special things. We'll look into that on page 51.

Continue with the rest of the COPY command. Again, note the space that separates the file name from the destination:

```
 A:
```

DOS will tell you it couldn't COPY a nonexistent file, but you can ignore that; this is all for the purpose of creating a macro, not for telling DOS anything—yet. Now, end the macro:

Alt / M E

then press **Alt C** to play back the macro. It will write COPY and then wait for you to enter a file name. Do that and press **↵**. The rest of the macro plays back, and your file is copied.

More On Variable Fields

Suppose you are writing letters to customers who have ordered various articles, informing them that the ordered item is now in stock:

Dear Mrs. Jones,

We are happy to inform you that the oven you ordered on May 20th has now arrived.

The only things that would vary in this letter are the name, the article, the date, and the verb (which could be *has* or *have*). If you use a variable field, there's no need to write the entire text each time. Define a macro as follows:

Alt = Alt L Dear Ctrl - Ctrl - , ↵

↵

We are happy to inform you that the **Ctrl - Ctrl -** you ordered on **Ctrl - Ctrl -** now arrived.

↵

Alt -

The first time you press **Ctrl -** (hold down the **Ctrl** key and press **-**), you mark the beginning of a *variable field*. The next **Ctrl -** terminates it.

Now press **Alt L** and watch what happens. Each time SuperKey meets a variable field, it pauses and lets you enter whatever you want. Terminate the entry with **↵**, and SuperKey resumes playback.

If you “lose” characters on playback, it is probably because SuperKey's playback speed is too fast for your program. You can set a *playback delay* to reduce the speed (see page 155 for details).

Fixed Fields

Instead of variable length fields, you can use fixed length fields. Fixed length fields can save you time when the data you enter is the *same length*. A fixed field does not require you to press **↵**; playback resumes as soon as you enter the last character in the field. This is useful for entering such data as item numbers, dates, etc.

The following example records a month followed by a fixed two-character field for the day and, finally, the year:

Alt = **Alt D**
August- **Ctrl J** **Ctrl J** -1985
Alt =

As you can see, **Ctrl J** marks the beginning and end of the fixed field. The two spaces between define the length of the field: two spaces for two characters. Each time you press **Alt D**, SuperKey plays back "August-", waits for you to enter the day, and then prints the year "-1985".

You don't have to press **↵** to end the field. It ends automatically when filled. However, if you don't want to fill the entire field you can press **↵** to end it.

Field Formats

Field formats let you control the appearance of what you enter into a field. *Variable* and *fixed* fields alike let you define how numbers are formatted.

In *fixed* fields only, you can also format your data as Left adjusted, Right adjusted, or Centered.

You format a field simply by writing special characters between the beginning and end of a field. In order to use formatted fields, you must first switch on the Format fields option on the Options menu. Press

Alt / **O** **F**

When the Format fields option is off during macro recording, none of what you enter between the start and the end of a field is recorded in macro memory. This is usually how you want it, but if you want to format fields, you must record the special format characters into the field.

Field formats are described in full detail on page 119, but let's try a few simple formats now. First, we'll create a variable field which lets you enter a number

and then displays it with two decimal places and a comma for every third digit:

Alt = Alt N
Ctrl - , . ## Ctrl -
Alt -

If you want to use a macro to align columns of numbers, you will want them right-adjusted so the digits line up vertically. You do that by making a *fixed* field:

Alt = Alt N
Ctrl | ##,###,###.## Ctrl |
Alt -

Numeric formats are always right-adjusted in fixed fields. If you want to right-adjust text, enter an *R* as the first character when you define the field:

Alt = Alt N
Ctrl | R _ _ _ _ Ctrl |
Alt -

This fixed field will be four characters wide (because of the four spaces). The *R* does not count in the width of the field.

To *center* text in a fixed field, enter a *C* as the first character:

Alt = Alt C
Ctrl | C _ _ _ _ _ _ Ctrl |
Alt -

This field is six characters wide, and if you enter anything shorter than that, it will be centered over the six spaces.

Note that you cannot mix *formatted* numbers with text within one field, but you *can* have two or more fields right next to each other—some fields for text and some for numbers.

Entry Control

SuperKey also lets you control the type of data you can enter into variable and fixed fields, again simply by including special characters—an *entry mask*—between the field delimiters (beginning and end of the field). You can use an entry mask to specify any of the following:

- Numeric only
- Uppercase alphabetic (convert lowercase to upper)
- Lowercase alphabetic (convert uppercase to lower)
- Alphabetic
- Any character

Again, the Field formats option (on the Options menu) must be ON to use entry masks. In the example under Field Formats, the #'s (apart from describing the format), limit the field to numbers only; any other characters you try to enter are ignored. Press **Alt N** and enter

123x456.7y **↵**

The letters are ignored; what you should see on the screen is

123,456.70

If you want to make sure that numbers are entered in certain positions in a field, you can use an *N* in the entry mask. The fixed field below accepts only numbers in the first three positions, and any other characters in the remaining five:

Ctrl J NNN **Ctrl J**

Note that you must enter a space before and after the *N*s. Let's define a ten-character fixed field which will let you enter a name. We want to force the first character to uppercase (by entering a *U*), and all following characters to lowercase (by entering nine *L*s).

Alt = **Alt N**
My first name is **Ctrl J** ULLLLLLLLL **Ctrl J**
Alt -

Notice that this macro definition uses the same key, **Alt N**, as you used in the example with numbers above. This means that the old macro will be replaced by the new one. (SuperKey will warn you that you are about to erase a macro.)

Now press **Alt N**. SuperKey plays back the text *My first name is*, and then waits for you to type. Enter your name in all lowercase letters, for example:

john **↵**

The following will appear on the screen:

John

no matter which combination of upper and lowercase letters you type.

Try different combinations of upper and lowercase letters, and try to include numbers or other non-alphabetic characters. Each time SuperKey will show the name in the specified format.

In a variable field, the *last* entry control character determines what you can enter into the *rest* of the field. The variable field below will translate a lowercase letter to uppercase for the *first* character and vice-versa for all following characters.

Ctrl - **U** **L** **Ctrl -**

Entry control is described in more detail on page 114.

Nested Macro Recording

The COPY macro you made on page 50, using a variable field, saves you some typing when you copy files. But suppose you want to delete each original file once it is copied, like this:

```
COPY FILENAME A:
ERASE FILENAME
```

This macro would do the trick, of course:

```
Alt / M B Alt C
COPY [ ] Alt / M V Alt / M V [ ] A: [ ]
ERASE [ ] Alt / M V Alt / M V [ ]
Alt / M E
```

but then you would have to enter the file name twice. So instead of using a variable field, record the file name in another macro, and then use this macro in the ERASE command:

```
Alt / M B Alt C
COPY [ ] Alt / M B Ctrl F10 Alt / M E [ ] A: [ ]
ERASE [ ] Ctrl F10 [ ]
Alt / M E
```

When you play back this macro, it will type COPY, then stop for you to enter the file name and press []. When you do that, the file name is recorded in the macro [Ctrl F10], which is then in turn used by the ERASE command.

We call this *nested macro recording* because you record one macro while you play back another. [Ctrl F10] is chosen arbitrarily; it could be any key or key combination. We chose an out-of-the-way key combination because it is not a macro you will use directly; it's just there to hold the file name for the ERASE command. (If you know about programming, you'll see that [Ctrl F10] is actually used as a *variable*.)

The Macro Editor

Until now you have defined all your macros *interactively*, that is; while you were actually working with your application program. This is very convenient for many macros, especially because it lets you see what the macro actually does, right then and there.

However, if you want to design a very long and complicated macro, or if you want to make changes to an already defined macro, an interactive definition

could be very tedious. SuperKey, therefore, has another way to define and edit macros: the *macro eDitor*. Press

Alt / M D

to activate the eDitor. Now press the key that holds the macro you want to edit. If that key already holds a macro, you will see it displayed on the screen below the macro editor's *status line*. Otherwise the screen will be blank, ready for your entry.

Figure 5-1 shows the status line. The status line is discussed in detail on page 101. The three things you need to know right now are:

- the symbol on the far left shows you the key you are editing
- the arrow keys **←** and **→** are used to move around
- you press **Alt Esc** to exit the editor

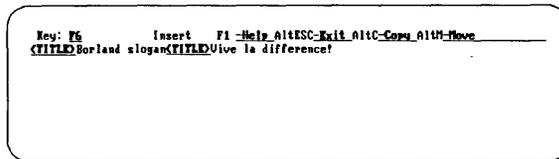


Figure 5-1 Macro Editor

Write A Macro

Now enter this text (spelling errors and all) and press **↵**.

This i a samble macro **↵**

Something unexpected happened, right? The cursor did not move down a line when you pressed **↵**. Instead, you got the symbol

<ENTER>

Many of the keys on the keyboard don't normally produce printable characters—they have some other effect. When you press **Esc**, for example, you don't get anything on the screen, but your program reacts to it in its own way.

In the macro editor, however, we want to *see* the **Esc** and all other keys that normally don't show. SuperKey displays these keys as symbols—descriptive text enclosed in angle brackets:

- Esc** is <ESC>
- ↵** is <TAB>
- ⏶** is <PGUP>
- ↵** is <ENTER>

and so on. There is a full listing of key symbols on page 135. If you want to insert arrow key symbols in a macro, you must first press the sKip key, then the arrow key. This is because the arrow keys are used to move around in the editor. For instance, to insert a right arrow key symbol (RGT), press  .

Edit The Macro

Remember the spelling errors in the text you typed in? Now we're going to use them to show you how to *edit* a macro.

Press  to move the cursor back to the space right after the *i*. It should have said *is*, so enter an *s* here. Now move the cursor to the *b* in *samble*. That should have read *sample*, of course, so you want to put a *p* where the *b* is. You can either press  to delete the *b* and then insert the *p*. Or you can first press  to switch from *Insert* mode to *Overwrite* mode. Notice the change on the status line at the top of the screen. Now enter the *p* on top of the *b*. Press  again to switch back to *Insert* mode.

The Main Menu

Even though you are in the macro editor, you can still use SuperKey's main menu to enter commands. That is, the commands you enter via the menu show up as symbols when you look at the macro in the macro editor. This is faster than typing in all the characters and angle brackets yourself. It also shows you what the symbols look like, so you can create and edit the file later with your own text editor.

Let's give the sample macro a title, for instance, *sample*. Press

as usual to call up the main menu. Then press

and type *sample*. When you return to the editor, you'll see this:

```
<TITLE> sample <TITLE>
```

This symbol does not represent a key on the keyboard, but a SuperKey command. Other examples of command symbols are:

<CMD> for functions

<VFLD> for variable fields

<FFLD> for fixed fields

You'll find a complete list of command symbols on page 137.

End Edit

To end editing and record the macro, press

Alt Esc

You are then asked if you want to save your changes. Press **Y** to save. **N** would exit the editor without recording the macro.

Using a Text Editor to Edit Macros

There is a limit to what the macro editor can do. First of all, it is limited to macros that will fit on the screen. Secondly, it only lets you see one macro at a time.

If you have very long macros, or want to see all your macros at the same time, you must use your normal text editor to edit macros. You can use any editor that reads and writes standard ASCII text files, such as EDLIN; WordStar (in non-document mode), SideKick's Notepad, or the Turbo Pascal editor.

Any macros you want to edit must first be written to a file with the Save macros command (on the macros menu).

To edit the macros, simply load this file into your editor. It contains all the macros that were in macro memory when you saved, in the order in which they were recorded.

Each macro begins with the symbol `<BEGDEF>` (or, for display macros `<BEGDISP>`) and ends with the symbol `<ENDDDEF>`

Anything entered in between those symbols appears the same in the macro editor. When editing macros this way, of course, you should be very careful to get everything right, especially the symbols. Look at other macros and see how symbols are used. If you make any mistakes, SuperKey will tell you so when you try to load the file.

Important. Be careful not to disturb the `<BEGDEF>` and `<ENDDDEF>` symbols. Each macro must start with `<BEGDEF>` and end with `<ENDDDEF>`.

C H A P T E R 6

The Layout Editor

The layout editor is used to create or alter keyboard layout files. This SuperKey feature can really come in handy if you don't like where some of your keys are located. Enter the layout editor by typing

LAYOUT 

on the DOS command line. The program will ask you for the name of a layout file. If you enter the name of an existing file, this file will be read in; otherwise, a new file will be created. The file name can be any legal name, and if you don't enter a last name, the last name .LAY will automatically be chosen.

The editor then shows you the picture of two keyboards, shown in Figure 6-1. The keys in the top picture are organized as they are on your actual keyboard. This is a reference to your *physical* keyboard, and it never changes.

The bottom picture shows the new key positions. This is your keyboard as it will appear when you load the layout file into SuperKey. The keys in this picture will change positions as you edit them.

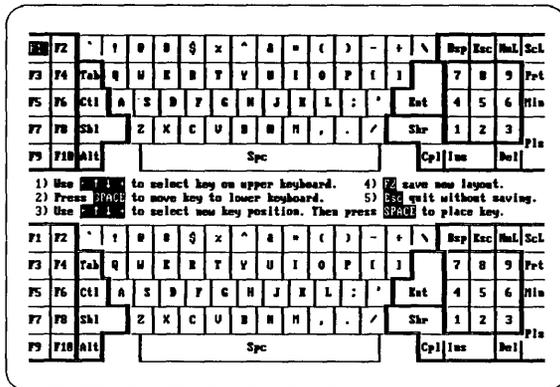


Figure 6-1 Layout Editor

Changing a key position is simple:

- Use the arrow keys to move the cursor to a key on the upper keyboard that you want to “relocate.”
- Press (the space bar) to select the key. The cursor now jumps to the lower keyboard.
- Use the arrow keys to move the cursor to the position where you want the key to be.
- Press again to position the key. You’ll see that the key name changes.

The cursor then jumps back to the upper keyboard, and you can repeat the process with another key.

Note that you move one key at a time, so if you want to swap two keys, you must first move one, then the other.

Let’s look at an example. We’ll switch the left hand key and the . This eliminates one of the most frequent complaints about the standard IBM PC keyboard: its shift key is in the wrong place.

First move the cursor to the left hand key on the upper keyboard and press . Then move the cursor to the key on the lower keyboard and press . That puts on the key. But since nothing is yet moved onto the key, you have two of them for the moment.

Now select the key on the upper keyboard, press , move to the leftmost key on the lower keyboard, and press again.

Voila! Your keys are where you want them.

Finally, press **F2** to save your new layout; the .LAY filename extension is added automatically. If you change your mind and want to leave the layout editor without saving your work, just press **Esc**.

To use a layout you create with the layout editor, you must *load* it. Press

Alt / L L

and enter the name of the layout file you want to load. This file must be on the same disk and directory as your macro files. It is normally the logged drive/active directory, but if you have defined another location in the Setup, your layout files must be there.

It's likely you'll want to use your new keyboard layout all the time. In that case, you can use command line parameters to load the new layout automatically when you load SuperKey. For example, to load a layout file called NEW.LAY, you would start SuperKey by typing the following on the DOS command line:

KEY NEW/LL **↵**

Of course, you can also use an AUTOEXEC.BAT file to load both SuperKey and the new layout automatically when you turn on your computer. See Appendix B.

Reference to SuperKey

This chapter provides, in alphabetical order, complete information about all commands, features, and concepts associated with SuperKey. Use this chapter as a handy reference whenever you need it.

For each command or feature, the following information is given:

- How to invoke the command: what keys to press or what to type on the DOS command line (if applicable).
- Default value: SuperKey factory settings.
- Complete description.
- Example using both “hot keys” and DOS commands.
- Cross reference to other commands and features.

Some of the information in this chapter is collected in the SuperKey help system which you use by pressing

F1

whenever SuperKey is loaded, provided the help file KEY.HLP is on the disk and/or directory where the SuperKey program is located.

Arrow Key Defaults

Press

Alt / D A (or use /DA parameter on DOS command line)

Default values

F7 = **↑**
F8 = **↓**
F9 = **←**
F10 = **→**

Description

The default value for the four alternate arrow keys (see page 66 for definition) are **F7** through **F10**, but you can choose any other definable keys you like. Activate the main menu (press **Alt /**), then select Defaults, and finally select Arrow keys. A small window opens showing which four keys are currently selected as alternate arrow keys. You can enter four other keys by pressing the desired keys. The first key you define is the alternate **↑** key. When you've done that, you automatically move to the **↓**, and so on. Even if you want to change only the last alternate arrow key, **→**, you must define the three others. You can press **Esc** any time to close the window.

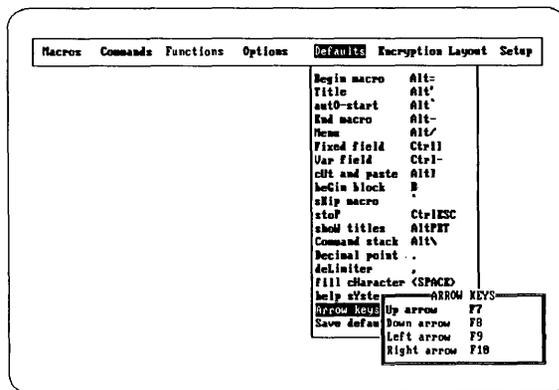


Figure 7-1 Alternate Arrow Key Definition Window

Example

To use the control characters CtrlE, CtrlX, CtrlS, and CtrlD (the “WordStar diamond”) as alternate arrow keys, press

Alt / D A Ctrl E Ctrl X Ctrl S Ctrl D

to automatically load the new defaults along with SuperKey. Start SuperKey with this command:

```
KEY /DA[CtrlE][CtrlX][CtrlS][CtrlD]
```

See Also

Arrow keys option

Defaults

Save defaults

Arrow Keys Option

Press

Alt / O A (or use /OA parameter on DOS command line)

Default value

OFF

Description

When you press **Num Lock** to activate the numbers on the numeric keypad, its cursor-movement arrows are no longer accessible (unless you press **Ctrl**). This can be inconvenient, since you often want the numbers and the arrows to be active at the same time, notably in a spreadsheet.

SuperKey therefore lets you use four other keys as **↑**, **↓**, **←**, and **→**. They are only active when **Num Lock** is active, rendering the normal arrow keys useless.

When you first start SuperKey, the alternate arrow keys, although defined, are not active. You activate them by opening the Options window on the main menu, then selecting Arrow keys. You can also press **+** or **-** to explicitly turn the alternate arrows ON or OFF.

The default values for the alternate arrow keys are

F7 = **↑** **F8** = **↓**
F9 = **←** **F10** = **→**

but you can choose to use other keys by changing the default (see page 64).

Examples

Use the following keystrokes to toggle the alternate arrow keys ON:

Alt / O + Esc

or start SuperKey with this command:

KEY /OA+

You can also make a macro that toggles the status of the alternate arrow keys.
Let's use **Alt** **A**; press:

Alt **=** **Alt** **A**
Alt **/** **O** **A**
Alt **-**

See Also

Arrow key defaults

Options

Save options

Auto-start Macro

Press

Alt / M O

Description

A macro can be made to start automatically each time the macro file is loaded or merged into memory. This is useful for displaying help text or menus, or for starting self-running demos or tutorials.

To make an auto-start macro, first activate the main menu: press **Alt /**. Then select auto-start on the Macros menu.

There can only be one auto-start macro in memory at one time. If there's already another in memory, it will lose its auto-start status.

Example

To make **Alt F1** start automatically each time the file containing it is loaded or merged into memory, enter

Alt = Alt F1 Alt / M O

: (The body of the macro is entered here.)

:

Alt -

See Also

Load macro

Merge macro

Beep Function

Press

Alt **/** **F** **B**

Description

Beep is a function that can be included in a macro. When the macro is played back, a sound is generated.

The beep function is entered while you are recording a macro. Press **Alt /**, select Functions, then select Beep. The small window shown in Figure 7-2 opens. You must now define the start and end frequencies and the duration of the tone. Finally, press **Enter** to record the beep.

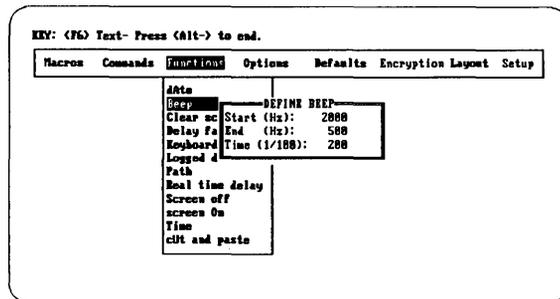


Figure 7-2 Define Beep Window

Example

If you enter:

400 **Enter**
500 **Enter**
25 **Enter**
Enter

you will get a tone that starts at 400 Hz, ends at 500 Hz, and lasts for one quarter of a second.

If you look at this macro in the editor, it appears as follows:

```
<CMD>FB00400 00500 00025<CMD>
```

with the **Alt /** command symbolized by <CMD>.

Begin Block Default

Press

Alt / D G (or use /DG parameter on DOS command line)

Default value

B

Description

The begin block command is used to mark the beginning of a block to cut and paste. You can change its default value from **B** to any other definable key(s). You might want to do this if your word processing program uses a different key to begin a block. It's easier to only have to remember one command.

Example

If you want to use **Alt Home** to begin a block, press

Alt / D G Alt Home

or start SuperKey with this command:

```
KEY /DG{AltHome}
```

See also

Cut and paste function
Defaults
Functions Menu

Begin Macro

Press

Alt **=** (can be redefined, see page 90) or
Alt **/** **M** **B**

Description

The Begin macro command is used to define a SuperKey macro. When you press the keys above, a small window opens and requests you to press the key you want to hold the macro.

After you press the key you want to define, a line appears at the top of the screen, informing you that you may now enter your macro. Anything you type now will be sent to SuperKey's macro memory as well as to your application program. SuperKey is thus totally "transparent" to your application program. This method of defining a macro is called *interactive*, since it takes place while you are working with your application program.

If your other program also uses the top line of the screen, you can move SuperKey's message to the bottom of the screen with the command

Alt **/** **O** **B**

While you record a macro, you can use the following commands:

Ctrl **-** Start and end a variable field (see page 175).

Ctrl **|** Start and end a fixed field (see page 125).

Alt **-** End the macro recording (see page 123).

Press **Alt** **/** to access the main menu. You can now include functions in the macro by selecting from the Functions pull-down menu, or you can let the macro change Options or Defaults by selecting items from these menus. You can even load or merge other macro files by selecting from the Macros menu. In fact, you can do almost anything you normally do with the main menu, and it will become part of the macro you are defining.

Examples

To make **Alt H** send the message *Hi, I'm a macro* each time you press it, type the following:

```
Alt = Alt H  
Hi, I'm a macro  
Alt -
```

Let's make a macro that first defines another macro and then uses this macro for further processing. For example, here is a macro that asks for a file name, then copies this file to drive B:, and finally erases the original file:

```
Alt = Alt C Enter file name:  
Alt = Ctrl F10 Alt - ←  
COPY Ctrl F10 B: ←  
ERASE Ctrl F10 ←  
Alt -
```

More examples of functions and options used in macros can be found in the sections discussing each separate feature.

See Also

- Editor
- End macro
- Fixed field
- Functions
- Macro memory
- Main menu
- Titles
- Variable field

Bottom Line Option

Press

Alt / O B (or use /OB parameter on DOS command line)

Default value

OFF

Description

The Bottom line option selects the position of the message that SuperKey displays while you record a macro interactively. It is normally at the top of the screen, but can be moved to the bottom if the normal position interferes with other programs.

Switching this option ON moves the line to the bottom line of the screen.

Example

Use the following keystrokes to move the line to the bottom of the screen:

Alt / O B

or start SuperKey with this command:

KEY /OB+

Or, you can use a macro to toggle the position of the line. Let's use **Alt L**.
Press

Alt = Alt L
Alt / O B
Alt -

See Also

Options

Save options

Clear Macros

Press

Alt / M C

Description

The clear macros command erases all macros from macro memory. The keyboard is now back to normal, but SuperKey is still active, and you can make new macros. It is not necessary to use the Clear macros command if you are loading a new macro file, because loading new macros automatically erases the ones that were in memory before.

Remember to save your macros before clearing memory if you want to use them again.

Clear macros does not affect keyboard layout; if you have an alternate layout loaded, it is still in effect.

Clear A Single Macro

If you just want to clear a single key of its macro, simply reassign the key to its original meaning. To clear, for example **Alt D**, press

Alt = Alt D Y Alt -

See Also

Save macros

Clear Keyboard Layout

Press



Description

The clear layout command returns the keyboard to its original configuration if you have loaded an alternate layout. SuperKey is still active, and any macros in memory are not affected.

Clear Screen Function

Press

Alt / F C

Description

Clear screen is a function that can be included in a macro. When the macro is played back, the screen is cleared and the cursor is placed in the upper left-hand corner.

To enter a clear screen function into a macro, first activate the main menu by pressing **Alt /**. Now select Functions, then Clear Screen.

A cleared screen cannot be restored. Instead, you can use the screen off/on function to blank the screen and restore it again—a very useful precaution against prying eyes. The protect delay option will switch off your screen after a pre-set time, and switch it back on again when you touch a predefined key.

Example

To make **Ctrl C** clear the screen, enter:

Alt = Ctrl C
Alt / F C
Alt -

If you look at this macro in the editor, it looks like this:

<CMD>FC<CMD>

where the **Alt /** command is symbolized by <CMD>.

See Also

Functions
Protect delay
Screen off function

Command Line Parameters

Description

Command line parameters can be used to pass commands to SuperKey when you load it, (and after you load it, if you wish). The same commands can be given via the main menu, once SuperKey is active, but parameters provide a convenient way to control SuperKey from .BAT files, which is especially useful if the computer operator does not know SuperKey.

Parameters are one or two letters initiated by a slash: /; some are preceded by a file name, and some are followed by additional information. The way to specify each parameter is listed later in this section.

Parameters can be passed not only on the first call (that is, when you load SuperKey), but also later when SuperKey is already loaded. Suppose you start SuperKey with the following command:

```
KEY WS/ML DVORAK/LL /OT+10 /DM[Ctrl\]
```

This command will:

- load SuperKey into memory
- load the macro definition file WS.MAC
- load an alternate keyboard layout from the file DVORAK.LAY
- set the save screen protect delay to 10 minutes
- redefine the main menu key from **Alt** **L** to **Ctrl** **N**

SuperKey is now resident in memory, and subsequent attempts to load it again will fail, but parameters can be passed on. After loading SuperKey as above, you may want to use a different set of macros and the standard keyboard layout. Of course, you could use the main menu, but you could also let a .BAT file do it automatically:

```
KEY DOS/ML /LC
```

This command will:

- attempt to load SuperKey into memory, but since it is already there, it is not loaded again
- load the macro definition file DOS.MAC
- clear the alternate keyboard layout and return the keyboard to normal

Parameter Syntax

Table 7-1 lists all command line parameters, their syntax, and their use. Note that you must always type **KEY** before any command line parameter. Some words have special meanings in the table:

FileName	a file name
Number	a number
{ + - }	either + or -
[symbol]	a key symbol (see page 134)

Note that square brackets [and] are used instead of angle brackets < and > to enclose key symbols on the command line. This is because DOS uses < and > for “pipe” commands.

Parameters must be separated by spaces or commas. The following command lines are equivalent:

```
KEY DBASE/ML /OW+ /DA [CtrlE] [CtrlX] [CtrlS] [CtrlD]
KEY DBASE/ML,/OW+ ,/DA [CtrlE],[CtrlX],[CtrlS],[CtrlD]
```

Example

Suppose you want to use SuperKey only in your word processor and in your spreadsheet. You have a separate macro file for each of these programs, and you want everything done automatically. Just make the following two .BAT files:

WP.BAT:

```
KEY WS/ML /OU-
WS
KEY /OU+
```

CALC.BAT:

```
KEY 123/ML /OU-
LOTUS
KEY /OU+
```

These .BAT files will load SuperKey and pass parameters to it (or, if it's already loaded, just pass the parameters). The parameters will load the appropriate macro definition file and then turn suspension OFF, that is, activate SuperKey. The program is then started, and when you return to DOS, SuperKey is called with a parameter that turns suspension ON and SuperKey becomes inactive.

Table 7-1 Command Line Parameter Syntax

	Syntax	Use
	/DA [symbols]	Change default alternate Arrow keys
	/DB [symbol]	Change default Begin macro key
	/DC [symbol]	Change default Command stack key
	/DD [symbol]	Change default Decimal point
	/DE [symbol]	Change default End macro key
	/DF [symbol]	Change default Fixed field key
	/DG [symbol]	Change default beGin block key
	/DK [symbol]	Change default sKip macro key
	/DL [symbol]	Change default deLimiter
	/DM [symbol]	Change default Menu key
	/DP [symbol]	Change default stoP key
	/DU [symbol]	Default cUt and paste
	/DV [symbol]	Change default Var field key
	/DW [symbol]	Change default shoW titles key
	/K	Kill SuperKey
	/LC	Clear keyboard layout
FileName	/LL	Load keyboard layout
	/MC	Clear macros
FileName	/ML	Load macros
FileName	/MM	Merge macros
FileName	/MS	Save macros
	/OA { + - }	alternate Arrow keys option
	/OK { + - }	Keyboard click option
	/OC { + - }	Command stack option
	/OO { + - }	One finger mode option
	/OP Number	Playback delay option
	/OR Number	Real time delay option
	/OT { + - }	proTect screen ON/OFF option
	/OT Number	proTect screen delay option
	/OU { + - }	sUspend option
	/OW { + - }	disk Wait option

Command Stack

Press

Alt **↘** (can be redefined, see page 90)
or **Alt** **/** **C** **C**

Description

The Command stack is a small part of your computer's memory that SuperKey sets aside for the last 256 characters you have entered on the DOS command line. You can execute a command from the stack just by selecting it and pressing **↵**; no need to re-enter long commands repeatedly. This feature is similar to the way the **F3** key works in DOS; but **F3** remembers only the last command, whereas the command stack remembers as many commands as will fit into 256 characters.

You access the command stack by pressing

Alt **↘**

The entire stack is displayed in a window, with each command on a separate line; commands are listed in reverse order, with the most recent command on top. Select the command you want to execute again by moving to it with the **↓** and **↑** keys, then press **↵**. The command is sent as if you had typed it all over again, and it is put on top of the stack. To leave the command stack, press **Esc**.

If you like, you can edit a command before pressing **↵** to re-execute it. The **→** and **←** keys move right and left on the line, and **Del** deletes the character under the cursor.

The command stack can be switched off and on as explained on page 82.

Example

Suppose you are developing a program in assembly language and want the final result to be a .COM file. You will be repeatedly editing, assembling, linking, and EXE2BIN'ing your program using commands like these:

```
WS INVADER1.ASM
ASM INVADER1;
LINK INVADER1+\OBJ\INVMOD1+\OBJ\
INVMOD2,INVADER,INVADER1;
EXE2BIN INVADER.EXE \COM\INVADER.COM
```

When this process is completed, you test your new program and find an error. Back to the editor, correct the error—and now you must type the whole mumbo-jumbo all over again, without error! Not so with SuperKey. You just press  , move the bar to the desired command, and press .

See Also

Command stack option
Defaults

Command Stack Option

Press

Alt / O C (or use /OC parameter on DOS command line)

Default value

ON

Description

The command stack is normally active, which means that anything you type on the DOS command line is continuously recorded on the stack. You can switch off the stack to stop recording; when you later switch the stack back on, it contains whatever was in it when you switched it off. When the command stack is OFF, you can still use the commands that were previously recorded in the stack.

To switch the stack OFF and ON, select Options on the main menu, then Command stack on the pull-down menu. Press **←** to toggle the status to the opposite of its current value (OFF if it's ON and ON if it's OFF), or you can press **+** or **-** to explicitly turn the stack ON or OFF.

Examples

Use the following keystrokes to turn OFF the command stack:

Alt / O ↓ ↓ - Esc

or start SuperKey with this command:

KEY /OC-

You can let a macro toggle the command stack status each time you press a key.
Let's use **Alt M**; press

Alt = Alt M Alt / O C ← Alt -

See Also

Command stack
Options
Save options

Cut and Paste

Press

Alt J (can be redefined, see page 90)
or **Alt / M U**

Description

The cut and paste command allows you to pick up anything displayed on the screen and put it into a macro. This command can be used to move data from one program to another.

After you press **Alt J**, press the key you want to hold the cutting. SuperKey's windows then disappear from the screen, and it looks as if you are back in your program—but you aren't. All you can do at this point is move the cursor around on the screen with the arrow keys and mark a block, which will then be “cut” from the screen and read into the macro. The maximum size of a block is one screen.

First, move the cursor to the first character of the block you want to put into the macro and press **B** for **B**egin block. Then move the cursor to the end of the block. You will see the block being marked as you move along. Now press **←** to end the block and read it into the macro.

To paste the block somewhere else, simply press the key that holds the cutting.

If you want to redefine the beginning of the block, press **B** again. The current cursor position will then become the start of the block.

The block you mark can be anything from a single character to the entire screen. The begin and end block markers define the upper left and lower right corners of a rectangular block.

If a block contains more than one line, a carriage return will be inserted at the end of each line when it is read into the macro.

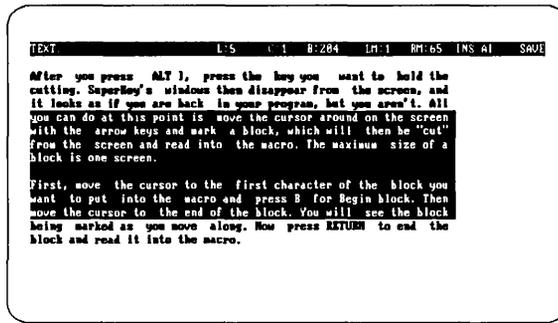


Figure 7-3 Block Marked For Reading

Example

You need to include the latest sales figures from your spreadsheet in a report you are preparing with your text editor. Activate the spreadsheet and get the figures you want on the screen. Then press

Alt J **Ctrl N**

to read data from the screen into **Ctrl N**. Use the arrow keys to move the cursor to the upper left corner of the block you want to copy and press **B**, then move the cursor to the lower right corner of the block.

Now press **←** to read the block into **Ctrl N**. Exit the spreadsheet and activate your word processor. Place the cursor where you want the upper left corner of the block and press **Ctrl N**.

If your word processor's margins are set too narrow for the block that is being pasted in, you may get some unwanted effects from automatic word wrap. Please be sure to check your margins.

See Also

Begin macro
Cut and paste function
Edit macro

Cut and Paste Function

Press

Alt / F U

Description

Cut and paste is a function that can be included in a macro. It lets you make a macro pick up anything displayed on the screen.

It differs from the cut and paste feature described on page 84 because it is included in a macro. When the macro is played back, you first mark a block of text or numbers on the screen.

You mark the upper left corner of the block with the begin block marker (**B** by default). Then you move to the lower right corner with the arrow keys, and end the block by pressing **↵**.

When you press the **↵** key, the marked block is moved to where the cursor was when you pressed the macro key(s).

See Also

Functions

Date function

Press

Alt **/** **F** **A**

Description

Date is a function that can be included in a macro. When the macro is played back, the date is read from DOS and transmitted to the application as a string of characters, usually in the following format:

MM/DD/YYYY

To enter a date function into a macro, press **Alt** **/**, then select Functions, then dAte.

Example

To make **Alt** **D** display the date, enter

Alt **=** **Alt** **D**
Alt **/** **F** **A**
Alt **-**

If you look at this macro in the macro editor, it looks like this:

<CMD>FA<CMD>

with the **Alt** **/** command symbolized by <CMD>.

See Also

Functions

Decimal Point Default

Press

Alt / D D (or use /DD parameter on DOS command line)

Default value

.

Description

The Decimal point we're talking about here is the character used to separate the integer part from the decimal part of a number that you enter into a formatted fixed or variable field. The format mask (see page 119) determines where in the field the decimal point is placed.

The decimal point is normally a period, but you can select any other displayable character. Press **Alt /** to activate the main menu, then select Defaults, then Decimal point. The current value of the decimal point is displayed to the right of the text, and you can now enter any other character you want to use.

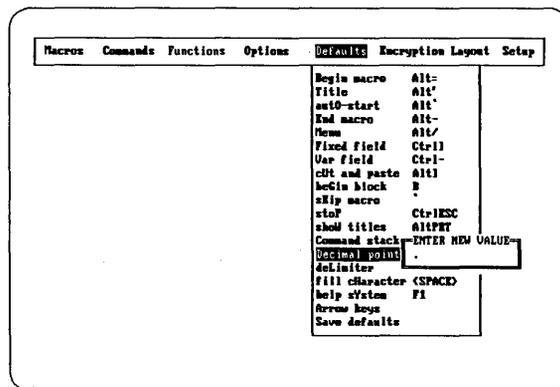


Figure 7-4 Decimal Point Definition

Example

If you want to use a comma as a decimal point, press

or start SuperKey with this command:

KEY /DD,

See Also

Defaults

Delimiter default

Entry control

Save defaults

Defaults

Description

Defaults are keystroke values assumed by SuperKey when nothing else is specified. A number of SuperKey's default values can be changed by activating the Defaults pull-down menu from the main menu: first press **Alt /**, then **D**. You can now move the bar up and down in the window and select an item by pressing **←**, or by pressing the capital letter in the name. Once you have selected an item, you can replace the current value by pressing the desired key or key combination; don't try to type in the *name* of the key(s).

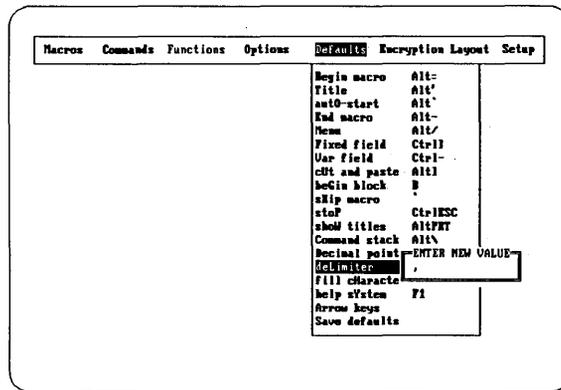


Figure 7-5 Defaults Menu

Default values can be changed when you first start SuperKey. Each parameter on the command line must be followed by the symbol of the desired key, using the symbols listed on page 136. See page 77 for how to use command line parameters.

If you want to make the new default values permanent, you can save your defaults by pressing

Alt / D S

Table 7-2 lists all default values that can be changed either on the Defaults menu or with a command line parameter.

Table 7-2 List of Installable Defaults

	Parameter	Default key(s)
Begin macro	/DB	Alt =
Title	/DT	Alt '
autO-start	/DO	Alt ^
End macro	/DE	Alt -
Menu	/DM	Alt /
Fixed field	/DF	Ctrl]
Var field	/DV	Ctrl -
cUt and paste	/DU	Alt]
beGin block	/DG	B
sKip macro	/DK	'
stoP	/DP	Ctrl Esc
shoW titles	/DT	Alt PrtSc
Command stack	/DC	Alt ^
Decimal point	/DD	\
deLimiter	/DL	.
fill cHaracter	/DH	,
help sYstem	/DY	F1
Arrow keys	/DA	F7 F8 F9 F10
Save defaults	/DS	Alt / D S

Example

To change the stop Playback command from **Ctrl Esc** to **Alt P**, press

Alt / D P Alt P

or start SuperKey with this command:

`KEY /DP[Alt]]`

Delay Function

Press

Alt / F D

Description

Delay is a function that can be included in a macro. When the macro is played back, playback pauses for the time given by the delay.

To enter a Delay into a macro, first press **Alt /**, select Functions, then Delay. A small window opens, requesting you to enter the length of the delay. You can enter a number from 1 through 999. The number you specify is not exactly a time. Rather, it defines how many times a program checks the keyboard status. The actual delay time therefore varies depending on which program is running. If you want to specify an actual time delay, use the Real Time delay function.

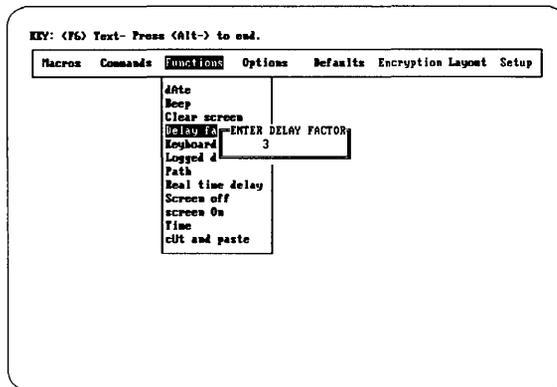
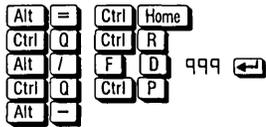


Figure 7-6 Set Delay

Example

To make **Ctrl Home** go to the top of the document (in WordStar), then pause, and finally go back where you were, enter the following:



If you look at this macro in the macro editor, it appears as follows:

```
<CtrlQ><CtrlR><CMD>FDqqq<CMD><CtrlQ><CtrlP>
```

See Also

Functions

Real time delay

Delimiter Default

Press

Alt **/** **D** **L** (or use /DL parameter on DOS command line)

Default value

.

Description

The delimiter we're talking about here is the character used to separate groups of digits in a formatted fixed or variable field.

The delimiter is a comma by default, but you can select any other displayable character you want to use. Press **Alt** **/** to activate the main menu, then select Defaults, and finally select delimiter. The current value of the delimiter is displayed to the right of the text, and you can now enter any other character you want to use.

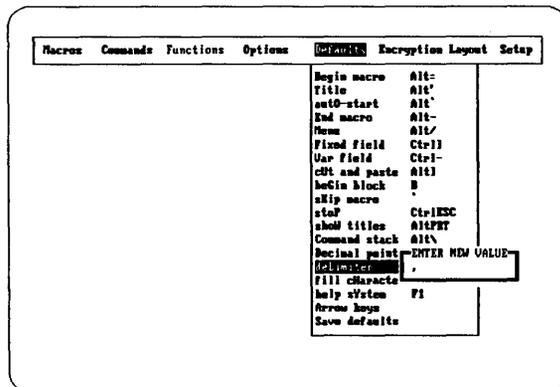


Figure 7-7 Delimiter Definition

Example

If you want to use a period as delimiter, press

Alt / D L .

or start SuperKey with this command:

KEY /DL.

See Also

Decimal point

Defaults

Field formats

Save defaults

Disk Wait Option

Press

    (or use /OW parameter on DOS command line)

Default value

OFF

Description

A computer program can do only one thing at a time, so while a program is reading or writing on the disk, it cannot receive characters you are typing. Your computer therefore has a *keyboard buffer*, a small part of its memory where it stores the characters it receives, whether they are typed in on the keyboard or sent by SuperKey. When the program is through reading or writing, it then reads the characters from the buffer—nothing is lost.

Some programs, however, do not follow this procedure and may lose characters sent to them while the disk is spinning. There are even programs that deliberately throw these characters away by “flushing” the buffer. If you run into a problem like this, you should turn disk wait ON. This will prevent SuperKey from sending any characters while the disk is spinning.

Examples

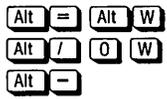
Use the following keystrokes to turn disk wait ON:

or start SuperKey with this command:

KEY /OW

You could also make a macro that turns disk Wait ON or OFF each time you press a key. Let's use **Alt W**; press:



See Also

Keyboard buffer
Save options

Display Macro

Description

A display macro is a macro that does not transmit characters to the application program, but instead displays text in a window on the screen. With display macros you can easily create your own customized help texts.

You must use a text editor to write your display macros. This can be any text editor producing standard ASCII text files, such as EDLIN, WordStar in non-document mode, SideKick's Notepad, or the Turbo Pascal editor.

Many other editors can produce standard text files. If you are not sure what your editor does, ask DOS to TYPE one of your text files on the screen. If the text is displayed legibly, it's a standard text file. Even if your editor does not produce standard text files, it may have a way of doing so, perhaps by printing the text to a disk file.

The file must contain the following information on the first line:

```
<BEGLISP><F10>1,1,30,10.
```

The first symbol identifies the macro as a display macro, while the second symbol is the key this macro (**F10** in this example) is assigned to. The numbers after the symbol specify the position and size of the window in which the macro is displayed. The first two figures are the column and line of the upper left corner of the window, the third figure is the number of columns (the width), and the last figure is the number of lines (the height). The numbers must end with a period.

If you want to add a title or make the display macro auto-start, you must also do that on the first line. For example:

```
<BEGLISP><F10>1,1,30,10.<TITLE>Read Me<TITLE><AUTO>
```

On following lines, write the text to be displayed. The maximum length of the lines is the specified window width (30 in the example). If you make your lines longer, they will be truncated. The window height you specify gives the number of lines in the window, but the macro can contain more lines. If it does, SuperKey automatically divides the text into pages, and you can page through it with **PgDn** and **PgUp**.

The display macro must end like any other macro, with the symbol:

```
<ENDDF>
```

Pages

An automatic page break will occur for every Nth line in a window N lines high. This may produce undesirable results, so you can force page breaks anywhere you want by entering a Control-L (a form-feed character) into the file. The Control-L can be entered either as a real control character, your text editor permitting, or as the symbol:

<CtrlL>

A display macro can be as long as your macro memory allows; the limit is approximately 65,000 characters. And each window can be as small as one character by one line, or as large as 78 characters by 23 lines (the entire screen, when you add the frame), if macro memory permits.

Video Attributes

You can use three different video attributes in a display macro text:

- Normal
- Boldface
- Frame

Normal video is used if you don't specify anything else.

Boldface text is indicated with Control-B's. The first Control-B in the text will toggle Boldface ON, the next Control-B will toggle it OFF again.

Frame is the attribute used for the window frame but it can also be used in the text. A Control-D toggles the Frame attribute ON, another Control-D toggles it OFF again.

These control characters can either be entered as real control characters or as symbols:

<CtrlB>
<CtrlD>

How the three attributes appear on your screen depends on your screen type: monochrome, black and white, or color. If you are not satisfied with the attributes as they are, use the installation program to change them (see page 186).

Pop-up Windows

Instead of entering specific values for the window position, you can enter a zero as either Line or Column value, or both. That will cause the window to open at the current cursor position. For example:

```
<BEGDISP><F1>0,1,30,10.
```

The top line of the window frame will be on line 1, and the left side of the window will be in the column where the cursor is presently located. If zero is specified for both Line and Column, the window will open with its top left corner at the current cursor position. Making help text windows “follow the cursor around” like this is often very effective, since it catches the user’s attention (which is usually focused on the cursor).

Transparent Macros

Transparent display macros are display macros that are cleared when a key is pressed. Just include the symbol <TRANSP> in the macro. See page 48 for details.

Example

Please see page 44 for an elaborate example.

See Also

Key symbols

Edit Macro

Press

Alt / M D

Description

The macro editor lets you create new macros and edit old ones. If you create a long and complicated macro containing many commands, you can fine-tune it later with the eDitor.

To activate the editor from the main menu, press **Alt /** to get the main menu, then press **M** for Macros and **D** for editor. You must then press the key whose macro you want to edit. If a macro is already assigned to that key, the editor screen will show you the macro; otherwise, the editor screen will be blank, except for the status line.

Status Line

Figure 7-8 shows the status line, which is the top line of the editor screen.

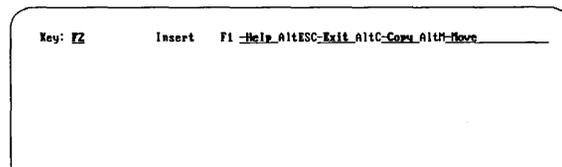


Figure 7-8 Editor Status Line

Key

Shows which key you are editing.

Insert

Indicates that you are in insert mode—that is, anything you type will be inserted at the cursor position. The **Ins** key toggles between Insert and Overwrite modes. In Overwrite mode, pressing a key will replace the character or symbol at the cursor position.

Help

As anywhere else in SuperKey, you can press **F1** to get help.

AltEsc--Exit

Press **Alt Esc** to exit the editor. When you do that, you can either save the macro or quit without saving.

AltC--Copy

Copies the contents of another macro key into the macro you are editing, at the cursor position. The macro you copy from is unchanged.

AltM--Move

Moves the contents of another macro key into the macro you are editing, at the cursor position. The macro you moved is cleared.

See Also

Begin macro

Interactive macro definition

Macro memory

Skip key

Encryption

Press



Description

SuperKey's encryption scheme lets you encrypt (scramble) *any* file for total data and program security. You can choose between two different encryption methods:

- The SuperKey in the file KEY.COM uses a proprietary algorithm for fast encryption. Files encrypted this way are safe against all but the most sophisticated intruders. If you want to use this method, type KEY  when you load SuperKey.
- The file KEYDES.COM contains a SuperKey that uses the DES encryption algorithm. This algorithm complies with the Data Encryption Standard issued by the National Bureau of Standards, U.S. Department of Commerce (FIPS publication 46, January 15, 1977). If you want to use this method, type KEYDES  when you load SuperKey.

Both algorithms use a unique keyword, supplied by you, as the basis of encryption and decryption. The same keyword must be used to decrypt an encrypted file, so don't call us if you forget your keyword; we can't help you.

Be Careful

Encryption can put you in a terrible mess if you're not careful. Please follow these guidelines:

- Never encrypt a file that's already encrypted.
- Only decrypt files that are encrypted.
- Never change the name of an encrypted file.
- Never mix the two encryption programs.

The encryption algorithm in KEY.COM uses the file name as part of the encryption/decryption process, so if you change the file name, you'll not be able to get your file back.

The DES algorithm (in KEYDES.COM) uses different methods for encryption and decryption. So if you encrypt a file twice, or decrypt a file that's not encrypted, the only way to get it back is to repeat the process in reverse order. This process does not use the file name.

The two encryption algorithms are totally different, so be sure to use only one of them, or at least keep files separate if you want to use them both.

An encrypted file contains absolutely no information about the encryption, and as we said before, **Borland cannot help you restore scrambled files. We are not responsible for any damage that may be caused by careless encryption procedures!**

If several people work together, and have different keywords, it's a good idea to keep files separate, so that one group of users does not encrypt or decrypt somebody else's files with a wrong keyword. Use separate diskettes, or different subdirectories if you have a fixed disk.

Encrypting and Decrypting

Use the Encryption menu to choose between Encrypt file and Decrypt file.

Use Encryption to scramble a normal file into unusable cipher, and Decryption to turn this cipher back to normal. Never encrypt an already encrypted file or decrypt a file that isn't encrypted!

Press **[E]** to Encrypt a file, or **[D]** to Decrypt a file. A new window then opens where you:

1. enter the name of the file(s) to be encrypted/decrypted
2. enter and verify the keyword
3. choose between non-text mode and text mode

File Names

The file name you specify for encryption can be any legal file name (see page 123).

If you want to encrypt a series of files whose names match a certain pattern (all your data files, for example, may have the last name DTA), you can include special characters, so-called *wildcards*, in the file name. There are two such wildcards—asterisk: * and question mark: ?. An asterisk stands for any *series* of characters, and a question mark stands for any *single* character. Let's look at some examples:

- *.* Denotes files with any first name and any last name, i.e. all files.
- *.TXT Denotes all files with any first name and the last name TXT.
- ??.* Denotes all files with one- or two-character first names and one-character last names.

And so on, ad infinitum. You can use any pattern, mixing asterisks, question marks, and text.

Verification

If you specify a file name containing wildcards, you will be asked if you want verification before encrypting or decrypting. If you answer Yes (or just press ) , the name of each file matching the file name pattern will be displayed before encryption, and you can accept or deny processing. If you answer No to "Verification?," all files matching the pattern will be processed without further warning.

We advise extreme caution in the use of wildcards. Verification should be skipped only when you are absolutely certain that the pattern indicated by your wildcards doesn't include file names you didn't mean to encrypt/decrypt. Otherwise you might end up having files encrypted or decrypted "behind your back," so to speak.

Protected Files

A few files are protected against encryption. These are:

COMMAND.COM
AUTOEXEC.BAT
KEY.COM
KEYDES.COM

If you encrypted these files, you could lock up your computer, or be unable to decrypt your files again.

Keyword

After you have specified the file name, you must enter a keyword. This can be any word or sequence of words, up to 30 characters long.

The keyword you enter does not show on the screen—all you will see is a string of asterisks. This is to prevent someone from looking over your shoulder and reading your secret keyword. But since you can't see what you type, you must enter the keyword twice to make sure that you didn't make any spelling mistakes.

Be sure to use the same keyword when you encrypt and decrypt. If you don't, you'll quickly end up in trouble. If that happens to you, the way to get out of it is *first* to reverse the process using the last keyword you used—that gets you back to where you were before you made the mistake—and *then* use the right keyword.

Lower and uppercase characters in keywords are not distinct, so you won't get in trouble if you enter the encryption keyword in lowercase letters and the decryption keyword in uppercase. If you enter any spaces before or after the keyword, they will be deleted, but embedded spaces count as part of the keyword.

Text Mode and Non-Text Mode

Now that you have entered the file name and keyword, you are finally asked whether you want Text mode or not. Press **N** or **←** if you don't want text mode; press **Y** if you do. Let's look at the two modes.

Non-text Mode This is the method you will want to use to secure the files on your floppy or hard disk. The encrypted file is written directly on top of the original file.

This means that no extra disk space is needed, and also that no part of the original file is left on the disk for anyone to look at. Each and every byte of the original file is physically encrypted. The encrypted file is *binary*, that is, it contains information in all eight bits of each byte.

Files encrypted in non-text mode must be decrypted in non-text mode!

Text Mode Text mode encryption is used to produce an encrypted file that can then be sent and received via modem communications programs that don't handle binary files.

Text mode encryption leaves the input file untouched and produces a separate output file containing only uppercase letters from A through Z. The output file is much larger than the input file, so this method requires free space on your disk, approximately equal to twice the size of the input file.

Since the original file is left intact, this method does not secure your data on the disk against unauthorized access. It only produces an encrypted file suitable for transmission over a modem.

Files encrypted in text mode must be decrypted in text mode! The recipient of a file that has been encrypted in text mode must select text mode when decrypting that file.

Also, the file to be decrypted must have the *same name* as when it was output by encryption. Text mode decryption produces a separate output file that is a duplicate of the file that was originally encrypted.

Example 1

Let's suppose you want to secure a database file called PERSONAL.DTA on your disk. Type:

```
Alt / E
E
Personal.dta ↵
MY OWN KEYWORD ↵
MY OWN KEYWORD ↵
↵
```

After you enter the keyword the first time, you are asked to enter it again. This is to protect you against mistakes. If the two entries do not match *exactly*, you must enter the keyword twice again.

The last accepts the default answer (No) to the question "Text mode?". You don't want to transmit the encrypted file via a modem, so you will encrypt the file directly on top of itself.

Your file is now encrypted. You can still copy it, back it up, and restore it, but your database program cannot use it.

It could have been any other type of file—a text file, a program—in fact, any file you want to secure.

Now, to decrypt that file and get back the original contents, repeat the process. Type:

```
Alt / E
D
Personal.dta ↵
my own keyword ↵
my own keyword ↵
↵
```

Note that upper and lowercase characters in the keyword are not distinct. You won't get into a mess if you enter your keyword in capitals when you encrypt, and use lowercase when you decrypt, or vice versa.

Remember that the keyword must be exactly the same as you used for encryption, including embedded spaces. Any spaces before or after the keyword are deleted.

Example 2

Let's repeat example 1, only this time we will encrypt *all* files with the last name .DTA:

```
Alt / E
E
*.dta ↵
Y
MY OWN KEYWORD ↵
MY OWN KEYWORD ↵
↵
```

The **Y** accepts verification, so you will be asked to accept or deny processing for each file matching the pattern. Now decrypt them all again:

```
Alt / E
D
*.dta ↵
Y
MY OWN KEYWORD ↵
MY OWN KEYWORD ↵
↵
```

Example 3

If your communications package won't let you transmit "object files" (files containing information in all eight bits of each byte, such as programs and files encrypted "on top" of themselves), you must use text mode to prepare a file for transmission over the phone.

Type:

Alt / E

E

Myprog.com

transmission keyword

transmission keyword

Y

Transmit.fil

This time, you answer Y to the question "Text mode?" because you want to produce an encrypted file that can be transmitted over a modem.

You are now asked for the name of the output file. You can't use the same name as the input file, because SuperKey now produces a separate output file and keeps the input file intact.

The output file contains only normal ASCII characters, and you can transmit it with any communications program. Note that the file contains a "header," which identifies it as an encrypted file, and a "terminator," to mark the end.

The receiving party must reverse the process, again using text mode:

Alt / E

D

Transmit.fil

transmission keyword

transmission keyword

Y

Newprog.com

The file now to be decrypted must have the *same* name as when it was output by encryption, but the output file name may be different from the original name.

If you download the file from a bulletin board, it may change names along the way. Be sure it has the right name before decrypting.

Because of the header on the encrypted file, SuperKey recognizes it as a text file. The output file will be an exact duplicate of the file you originally encrypted (and much smaller than the encrypted text file).

Some communications packages don't allow you to receive files, but can only "dump" your entire communication to disk, including log-on procedures and miscellaneous chit-chat. In that case you will have some irrelevant text in your file before the encrypted text starts, and probably some after it ends.

The receiving party must remove any irrelevant text at the start of the file before decrypting the file. The encryption header *must* be placed at the very start of the file. Use your text editor to delete the first part of the file, up to, but not including, the encryption header which appears as follows:

```
***SUPERKEY TEXT-ENCRYPTION START***
```

SuperKey automatically removes any irrelevant text at the end of the file because it stops encryption when it meets the encryption terminator, which looks like this:

```
***SUPERKEY TEXT-ENCRYPTION END***
```

See Also

Encryption directory

File names

Encryption Directory

Press

Description

Encryption directory lets you specify a location where SuperKey will look for files to encrypt. This location may be a full DOS path name (see page 193 for an explanation of directories and paths), or it may just be a drive name like A: or B:. If you just enter a drive name, the files will be found on the currently active directory of that drive.

The drive or path you specify becomes active immediately when you press . If you want SuperKey to find its files on the logged drive/current directory, then don't enter anything, just press .

The default directory you specify will be used only with file names without a path name. You can override the default by entering a full file path when you start Encrypt.

The directory you specify is stored only in your computer's internal memory. If you want SuperKey to use the new setup the next time you load it, you must record it permanently by selecting Save setup.

Example

Suppose all your program files are in a directory called \SYSTEM. You could encrypt all .COM files here, rendering your computer useless to people without the password. Specify the path to SuperKey as follows:

    \SYSTEM 

Now, each day before you shut down your computer, you can encrypt all your programs by entering:

```
Alt / E
E
*.COM
N
MY KEYWORD
MY KEYWORD
N
```

Because of the path specified in the Encryption directory, this will encrypt all .COM files in the \SYSTEM directory, but leave all other directories untouched.

COMMAND.COM (essential for starting the computer), KEY.COM, and KEYDES.COM are protected from encryption. Even if they are in the \SYSTEM directory, you'll still be able to start your computer and decrypt your files the next morning:

```
Alt / E
D
*.COM
N
MY KEYWORD
MY KEYWORD
N
```

See Also

Save setup

End Macro

Press

  (may be redefined, see page 90)
or    

Description

The End macro command terminates the interactive recording of a macro. When you press  , characters are no longer sent to macro memory, and the macro is ready to be used.

This command is also used to end the definition of a Title; see page 175.

See Also

Begin macro
Titles

Entry Mask

Description

SuperKey lets you specify the type of data you can enter into variable and fixed fields, by writing special characters—an entry mask—between the field delimiters **Ctrl** **[** or **Ctrl** **]**. The entry mask can specify the following types:

- Numeric only
- Uppercase alphabetic only (converts lowercase to upper)
- Lowercase alphabetic only (converts uppercase to lower)
- Upper or lowercase alphabetic
- Any character

Note that the Format fields option must be ON to record entry masks interactively.

Numeric Only

An **N** in a field means that only numeric data may be entered—digits, decimal points, plus and minus. Any other character will be ignored.

Uppercase Alphabetic Only

A **U** in a field means that only the uppercase characters **A** through **Z** may be entered. If the lowercase equivalents are entered, they will be converted to uppercase. Any other character will be ignored.

Lowercase Alphabetic Only

An **L** in a field means that only the lowercase characters **a** through **z** may be entered. If the uppercase equivalents are entered, they will be converted to lowercase. Any other character will be ignored.

Alphabetic Only

An **A** in a field means that only alphabetical characters **A** through **Z** and **a** through **z** may be entered. Any other character will be ignored.

Any Character

Any other character in a field (for example, a space) allows you to enter any character. Note, however, that the characters **C**, **R**, periods, and commas control the field format (see page 119).

The way entry control masks work differs between variable and fixed fields, as described next.

Variable Fields An entry control mask in a variable field determines the type of entry allowed character-by-character. However, since the length of the field is variable, the *last* entry control character entered determines the legal input types in the *rest* of the field.

Examples

Ctrl **-** **N** **Ctrl** **-**

—you can enter only numeric information into this field.

Ctrl **-** **A** **Ctrl** **-**

—you can enter only alphabetical characters into this field.

Ctrl **-** **UL** **Ctrl** **-**

—the first character entered into this field will be converted to uppercase, and all following characters will be converted to lowercase.

Ctrl **-** **UULLN** **Ctrl** **-**

—the first two characters in this field will be converted to uppercase, the next two will be converted to lowercase, and the rest of the field accepts only numbers.

Fixed Fields The entry control mask in a fixed field controls the contents of the field character-by-character. Different control characters within one field let you enter one type of data in some positions in the field, and other types of data in other positions.

Examples

Ctrl J N **Ctrl J**

—a one-character fixed field into which you can enter only numeric information.

Ctrl J A **Ctrl J**

—a one-character fixed field into which you can enter only an alphabetical character.

Ctrl J AN **Ctrl J**

—a two-character fixed field into which you can enter only an alphabetic character in the first position and numeric information in the second position.

Ctrl J UUU NNN **Ctrl J**

—a seven-character fixed field that accepts uppercase characters in the first three positions, any character in the fourth position, and numeric information in the last three positions.

See Also

Field formats

Fixed field

Format fields option

Variable field

Fields

Press

Ctrl **-** or **Alt** **/** **M** **V** to start and end a variable field

Ctrl **J** or **Alt** **/** **M** **F** to start and end a fixed field
(may be redefined, see page 90)

Description

A field included in a macro temporarily stops playback and lets you enter a value. When you're finished, macro playback resumes. Format and entry control masks can be used to control what can be typed into a field, and how it is displayed on the screen.

Fields come in two flavors: variable and fixed. A variable field can contain as few or as many characters as you enter into it, while a fixed field always contains a certain number of characters, as specified in the definition of the field.

Variable fields, *fixed fields*, *field format masks*, and entry control masks are described separately in this section. A detailed step-by-step example can be found in Chapter 5.

As long as a field is not yet terminated, you can press the backspace key, **⌫**, to delete your characters from right to left.

A variable field must always be terminated by pressing **⌫**. A fixed field is automatically terminated when you enter the last character of the field, but it can be terminated earlier by pressing **⌫**.

Fields are limited to a maximum of 255 characters.

Examples

Ctrl **-** **Ctrl** **-**

—declares a variable field

Ctrl **J** **Ctrl** **J**

—declares a fixed field 6 characters long

See Also

Entry control

Field formats

Fixed field

Variable field

Field Formats

Description

SuperKey lets you format the data you enter into variable and fixed fields by writing special characters—a *format mask*—between the field delimiters **Ctrl** **[-]** or **Ctrl** **[I]**. The following formats are available:

- Left adjusted (default)
- Right adjusted
- Centered
- Numeric formats

The first three apply to fixed fields only. Because variable fields are only as long as the entry, there is nothing to adjust.

Note that the Format fields option must be ON to record field formats interactively.

Left Adjusted

By default, the contents of a fixed field are written left-adjusted within the field—that is, the first character starts at position one. If the field is terminated before it is full, the rest of the field will be padded with fill characters. The fill character is a space by default, but any other character can be chosen with the fill cHaracter command on the Defaults menu.

Right Adjusted

Right-adjust the contents of a fixed field by placing an **R** right after the first field delimiter. The **R** does *not* count in the length of the field. When you type characters into a field, the first character will still start at position one, but as soon as you press **Ctrl** **[←]** to terminate the entry, it will be rewritten with its last character in the rightmost position of the field. The rest of the field to the left of your entry will be padded with fill characters (spaces by default).

Centered

Center the contents of a fixed field by placing a **C** right after the first field delimiter. The **C** does *not* count in the length of the field. When you type characters into a field, the first character will still start at position one, but as soon as you press  to terminate the entry, it will be rewritten with an equal number of fill characters on either side to fill out the format (in case of an odd number of fill characters, the left side will be the larger).

Numeric Formats

The format for numbers is specified by one or more **#** signs, and optionally a period to specify a decimal point and/or one or more commas to define the position of delimiters.

The effect of delimiters in a numeric format differs among variable and fixed fields.

Variable Fields In a variable field, a comma must be entered as the *first* character in the field definition. It will cause a comma to be printed for every third digit of the integer part of a number. In the following field

  , . ##  

the number 1234.5 would be printed as 1,234.50.

Fixed Fields In a fixed field, comma delimiters must appear in the format mask for every position where you want the delimiter. You can “force” a format in your numeric data with comma delimiters. In the following field

  ##,##.##  

the number 1234.5 would be printed as 12,34.50.

Examples

Ctrl **-** #.## **Ctrl** **-**

—a variable field that will display any number with two decimals:

123 will be displayed as 123.00

123.456 will be displayed as 123.45

.123 will be displayed as 0.12

Ctrl **-** ,.## **Ctrl** **-**

—a variable field that will display any number with two decimals and a comma for every third digit to the left of the decimal point:

123 will be displayed as 123.00

123456 will be displayed as 123,456.00

.123 will be displayed as 0.12

Ctrl **|** ##.# **Ctrl** **|**

—a four-character fixed field that will display a two-digit number with one decimal (\ symbolizes a fill character):

12 will be displayed as 12.0

.1 will be displayed as \0.1

Ctrl **|** ##,##,## **Ctrl** **|**

—an eight-character fixed field that will display a six-digit number with a comma for every second digit (\ symbolizes a fill character):

123 will be displayed as \\\1,23

123456 will be displayed as 12,34,56

.1 will be displayed as eight blanks, since there are no decimals in this format; the decimal point and anything following it are ignored.

Ctrl **|** C\\ **Ctrl** **|**

—a twenty-character fixed field that will display any entry centered within the field length (\ symbolizes a fill character):

CENTERED will be displayed as \\\\\\CENTERED\\\\\\

See Also

Defaults

Entry control

Fill character

Fixed field

Format fields option

Variable field

File Names

Description

A file name consists of a first name of 1 to 8 characters, optionally followed by a period and a last name of 1 to 3 characters.

A file name may also specify a *path*. A path is used to identify subdirectories in a DOS tree-structured directory. For example:

```
\KEY\MAC\WS.MAC
```

specifies the file WS.MAC in the directory MAC which is a subdirectory to KEY, which is a subdirectory to the root directory.

File names specified for *encryption* may also include *wildcards* to make the name match a series of files. There are two such wildcards: asterisk (*) and question mark (?). An asterisk stands for any series of characters, and a question mark stands for any single character. Let's look at some examples:

- *.* Denotes files with any first name and any last name, i.e., all files.
- *.MAC Denotes all files with any first name and the last name MAC.
- ???.?? Denotes all files with first names of one to three characters and one- or two-character last names.

You can mix asterisks, question marks, and letters to produce any file-name pattern.

Examples

```
LETTER  
LETTER.TXT  
\LETTERS\LETTER.TXT  
\DATA\DBASE\*.DTA
```

See Also

DOS primer (Appendix E)

Fill Character Default

Press

Alt / D H (or use /DH parameter on DOS command line)

Default value

Space

Description

SuperKey uses fill characters to pad fixed format fields if they are not completely filled in.

The fill character is a blank space by default, but you can select any other displayable character. Press **Alt /** to activate the Command menu, then select Defaults, and finally select fill cHaracter. The current value of the fill character is displayed to the right of the text, and you can now enter any other character you want to use.

Example

If you want to use a hyphen as fill character, press

Alt / D H -

or start SuperKey with this command:

KEY /DH-

See also

Defaults
Fixed fields
Save defaults

Fixed Field

Press

Ctrl **J** (may be redefined, see page 90)
or **Alt** **/** **M** **F**

Description

A fixed field included in a macro defines a space of a given size into which you can enter text or numbers during macro playback. When you're finished, macro playback resumes.

The definition of a fixed field specifies the length of the field, and special characters (format and entry control masks) can be used to control what can be typed into a field, and how it is displayed on the screen.

As long as your entry is not terminated, you can press **←** (the backspace key) to delete characters from right to left.

A fixed field is automatically terminated when you enter the last character of the field, but it can be terminated earlier by pressing **←**.

The maximum length of a fixed field is 255 characters.

Example

Alt **=** **Ctrl** **N**
Ctrl **J** **Ctrl** **J**
Alt **-**

Defines a fixed field six characters long.

See Also

Entry control
Field formats
Variable field

Format Fields Option

Press

Alt / O F (or use /OF parameter on DOS command line)

Default value

OFF

Description

By default, anything you enter between the start and end of fixed and variable fields is not recorded in the macro when you record interactively. This allows you to use the macro over and over again, entering different characters each time.

If you want to make entry control masks or formatted fields, however, the characters you enter *must* be recorded. This is accomplished by switching this option ON.

Example

To switch it ON, press

Alt / O F 

or start SuperKey with this command:

KEY /OF

You can let a macro toggle the status of the Format field option each time you press a key. Let's use **Alt F**; press

Alt = Alt F

Alt / O F

Alt -

See also

Entry control
Field formats
Options
Save options

Functions Menu

Press

Alt / F

Description

Functions are small “programs” you can include in macros to make them perform various tasks during playback, such as making sounds, switching the screen off and on, displaying current time or date, and so on.

The Functions menu can only be activated while you are defining a macro, because functions are only used during macro playback.

To include a function in a macro, select it from the Functions pull-down menu while you are defining the macro, either interactively or in the macro editor.

Press **Alt /** to get the main menu, then **F**. To enter a function into the macro you are defining, move the highlighted bar to the function and press **↵**, or just press the capital letter in the name. The Beep and Delay functions require a little additional input, so a small window will open, prompting you to enter the necessary values.

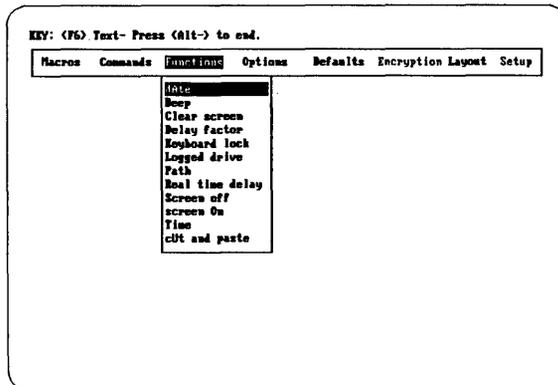


Figure 7-9 Functions Menu

You'll find each function described separately in this section. Table 7-3 lists all functions and the keystrokes used to access them.

Table 7-3 *List of Functions*

Function	Press
dAte	/
dAte	/
Beep	/
Clear screen	/
Delay	/
Keyboard lock	/
Logged drive	/
Path	/
Screen off	/
screen On	/
Time	/
cUt and paste	/

Examples

This macro will start a program called TONES, and sound a beep when the program finishes:

```
 =  TONES 
 /   500  500  300   -
```

This beep will be 500 Hertz and last for three seconds, enough to wake anyone. You could add a more musical note by making three beeps with different frequencies, with a short delay between them:

```
 =  TONES 
 /   400  400  100 
 /   500 
 /   500  500  150 
 /   500 
 /   600  600  200 
 -
```

Interactive Macro Definition

Description

You define a macro *interactively* when you define it while you actually work with your application program. Anything you put into the macro is also sent to the application, so you can immediately see the effect. This makes it easy to define complicated command sequences.

Interactive macro definition is started with the Begin macro command; see page 71.

If you don't want characters sent to your application program, the macro editor can be used to create and edit macros. See page 101.

See Also

Begin macro

Edit macro

Keyboard Buffer

Description

The IBM PC BIOS gives you a 16-character keyboard “type-ahead” buffer that allows you to enter up to 16 characters on the keyboard even when the computer is busy doing something else, like reading or writing on the disk; the characters will be saved and read by your computer when it is finished with its task. Often you’ll find that 16 characters is not enough; the computer starts beeping at you.

With SuperKey loaded (even if it is not active), you have a 128-character type-ahead buffer instead. You should never run into another beep.

Keyboard Click On/Off Option

Press

Alt / O K (or use /OK parameter on DOS command line)

Default value

OFF

Description

When Keyboard click is ON, you'll hear a click each time you press a key. If you use the standard IBM keyboard, you probably have all the clicking you want in hardware—so SuperKey's click is OFF by default. But if you have another keyboard, you may choose to emulate the IBM sound by setting this option ON.

Examples

Turn keyboard click ON with the following keystrokes:

Alt / O K +

or start SuperKey with this command:

KEY /OK+

You can design a macro that will toggle keyboard click on and off each time you press a key. Let's use **Alt K**; press

Alt = Alt M Alt / O K ← Alt -

See also

Options

Save options

Keyboard Lock/Unlock Function

Press

    (or use /FK parameter on DOS command line)

Description

Keyboard lock/unlock is a function that can be included in a macro. When the macro is played back, you will be asked to enter a keyword. The keyboard is then locked until you enter the keyword again.

To enter a keyboard lock/unlock into a macro, first define the key, then select Functions, then Keyboard lock.

Example

The following keystrokes will make   lock the keyboard:

If you look at this macro in the macro editor, it appears as follows:

```
<CMD>FK<CMD>
```

with the   command symbolized by <CMD>.

You can also lock your keyboard when you load SuperKey on the DOS command line by typing

Key /FKpassword

where *password* represents a password up to eight characters long.

See Also

Functions

Key Symbols

Description

You only need concern yourself with key symbols if you want to use the macro editor. When you look at a macro definition in the editor, you will notice that keys and key combinations are symbolized by a short descriptive text enclosed in angle brackets. For example:

⟨F1⟩ = the **F1** key
⟨AltD⟩ = the **Alt D** key combination

A key combination is achieved by holding down the first key while pressing the other. To get an ⟨AltD⟩, for example, you hold down the **Alt** key and then press the **D** key.

The special SuperKey commands, such as Begin macro, End macro, and Variable field, have special, unchanging descriptive symbols:

⟨BEGDEF⟩ = the Begin macro command (**Alt =** by default)
⟨ENDEF⟩ = the End macro command (**Alt -** by default)
⟨VFLD⟩ = the Variable field command (**Ctrl -** by default)

(Remaining command symbols can be found on page 137.)

The reason for these special symbols is that the SuperKey commands can be redefined (by changing the Defaults, see page 90). If, for example, the Variable field command **Ctrl -** were symbolized as ⟨Ctrl-⟩ in the macro definition, it would no longer be recognized as a command should you later change the Variable field command to something else—for example **Alt V**. With unchanging symbols, the command will always be recognized, no matter what keystrokes are used.

Key Symbols

Table 7-4 lists all keys and key combination symbols. Symbols are enclosed in angle brackets and the default command keys are marked with a •.

Table 7-4 List of Keys and Key Symbols

Key	Alone	With 	With 	With 
	a	A	<CtrlA>	<AltA>
	b	B	<CtrlB>	<AltB>
	c	C	<CtrlC>	<AltC>
	d	D	<CtrlD>	<AltD>
	e	E	<CtrlE>	<AltE>
	f	F	<CtrlF>	<AltF>
	g	G	<CtrlG>	<AltG>
	h	H	<CtrlH>	<AltH>
	i	I	<CtrlI>	<AltI>
	j	J	<CtrlJ>	<AltJ>
	k	K	<CtrlK>	<AltK>
	l	L	<CtrlL>	<AltL>
	m	M	<CtrlM>	<AltM>
	n	N	<CtrlN>	<AltN>
	o	O	<CtrlO>	<AltO>
	p	P	<CtrlP>	<AltP>
	q	Q	<CtrlQ>	<AltQ>
	r	R	<CtrlR>	<AltR>
	s	S	<CtrlS>	<AltS>
	t	T	<CtrlT>	<AltT>
	u	U	<CtrlU>	<AltU>
	v	V	<CtrlV>	<AltV>
	w	W	<CtrlW>	<AltW>
	x	X	<CtrlX>	<AltX>
	y	Y	<CtrlY>	<AltY>
	z	Z	<CtrlZ>	<AltZ>
	[{	<Esc>	<Alt{>
	\		<Ctrl\>	<Alt\>•
]	}	<Ctrl]>•	<Alt]>•
	Space	Space	Space	Space
	<F1>	<ShftF1>	<CtrlF1>	<AltF1>
	<F2>	<ShftF2>	<CtrlF2>	<AltF2>
	<F3>	<ShftF3>	<CtrlF3>	<AltF3>
	<F4>	<ShftF4>	<CtrlF4>	<AltF4>
	<F5>	<ShftF5>	<CtrlF5>	<AltF5>
	<F6>	<ShftF6>	<CtrlF6>	<AltF6>
	<F7>	<ShftF7>	<CtrlF7>	<AltF7>
	<F8>	<ShftF8>	<CtrlF8>	<AltF8>
	<F9>	<ShftF9>	<CtrlF9>	<AltF9>
	<F10>	<ShftF10>	<CtrlF10>	<AltF10>

Table 7-4 List of Keys and Key Symbols, continued

Key	Alone	With 	With 	With 
Upper row:				
	1	!	<Ctrl1>	<Alt1>
	2	@	<Ctrl2>	<Alt2>
	3	#	<Ctrl3>	<Alt3>
	4	\$	<Ctrl4>	<Alt4>
	5	%	<Ctrl5>	<Alt5>
	6	^	<Ctrl6>	<Alt6>
	7	&	<Ctrl7>	<Alt7>
	8	*	<Ctrl8>	<Alt8>
	9	(<Ctrl9>	<Alt9>
	0)	<Ctrl0>	<Alt0>
	-	_	<Ctrl- >	<Alt- >
	=	+	<Ctrl= >	<Alt= >
Numeric keypad (with  OFF):				
	<End>	<1>	<CtrlEnd>	<AltEnd>
	<Dn>	<2>	<CtrlDn>	<AltDn>
	<PgUp>	<3>	<CtrlPgDn>	<AltPgDn>
	<Lft>	<4>	<CtrlLft>	<AltLft>
	<Ctrl>	<5>	<CtrlCtr>	<AltCtr>
	<Rgt>	<6>	<CtrlRgt>	<AltRgt>
	<Home>	<7>	<CtrlHome>	<AltHome>
	<Up>	<8>	<CtrlUp>	<AltUp>
	<PgUp>	<9>	<CtrlPgUp>	<AltPgUp>
	<Ins>	<0>	<CtrlIns>	<AltIns>
		<. >	<CtrlDel>	<AltDel>
	<Min>	<Min>	<CtrlMin>	<AltMin>
	<Pls>	<Pls>	<CtrlPls>	<AltPls>
	<*>		<CtrlPrt>	<AltPrt>
	'	"	<Ctrl' >	<Alt' >
	.	~	<Ctrl' >	<Alt' >
	,	<	<Ctrl, >	<Alt, >
	.	>	<Ctrl. >	<Alt. >
	/	?	<Ctrl/ >* >	<Alt/ >* >
	;	:	<Ctrl; >	<Alt; >
	<Esc>	<Esc>	<CtrlEsc >* >	<AltEsc>
	<Tab>	<ShftTab>	<CtrlTab>	<AltTab>
	<Bks>	<Bks>	<CtrlBks>	<AltBks>
	<Enter>	<Enter>	<CtrlEnter>	<AltEnter>

* Only on the IBM PCjr after running the PCJRINST program.

Command Symbols

When a SuperKey command appears in a macro, it is represented by the special symbols shown in Table 7-5. These symbols allow redefinition of command key-strokes, while retaining compatibility with old macros. When key symbols are used as parameters on the DOS command line (see page 77), square brackets [and] must be used instead of the angle brackets. This is because DOS understands the angle brackets as “pipe” commands.

Table 7-5 List of Command Symbols

Command	Symbol	Default Key
Menu	<CMD>	Alt /
Begin/end title	<TITLE>	Alt *
Begin macro	<BEGDEF>	Alt =
Fixed field	<FFLD>	Ctrl J
Var field	<VFLD>	Ctrl -
Auto-start	<AUTO>	
End macro	<ENDDEF>	Alt -
cUt and paste	<CUT>	Alt J
Command stack	<STACK>	Alt \
sKip macro	<SKIP>	`
stoP	<STOP>	Ctrl Esc
shoW titles	<SHOW>	Alt PrtSc
Sidekick	<SIDEKICK>	

See Also

Defaults
Edit macro
Keys

Kill SuperKey

Press

     (or use /K parameter on DOS command line)

Description

At some point you may need to “kill” SuperKey—remove it from memory to free up space for other programs. This may be desirable if you need to create more room for other programs without resetting the computer.

But it should be done with caution, and only after observing the following rules:

It must only be done if there are **no other programs** in memory **after** SuperKey. This means that you must not kill SuperKey if you have an application program running. That is, be sure you are on the DOS command line before you kill SuperKey. But first be certain there are no other resident programs loaded after SuperKey. This means that if you have Turbo Lightning and/or SideKick in memory, you must kill them first. Kill the programs in the *reverse* order they were loaded in—i.e., SideKick, Turbo Lightning, then SuperKey. *If you kill SuperKey while other programs occupy memory “above” it, you will create an unoccupied “hole” in memory—a situation that DOS cannot handle. You will have to reset the computer, and perhaps even turn it off and back on again.*

To kill SuperKey, call the main menu to the screen by pressing  . Then, hold down the  key and press the  and  keys.

Load Layout

Press

Alt / L L (or use FILENAME /LL parameter on DOS command line)

Description

The Load layout command reads a keyboard layout from a disk file created with the layout editor (described in Chapter 6). The layout becomes effective immediately.

To load a layout, press **Alt /** to get the main menu, then press **L** twice. The window shown in Figure 7-10 opens, requesting you to enter the name of the file you want to load: The name you enter here can be any legal file name, optionally including a DOS path name (see page 123 for a discussion of file and path names). If you do not specify a last name, the last name .LAY is assumed.

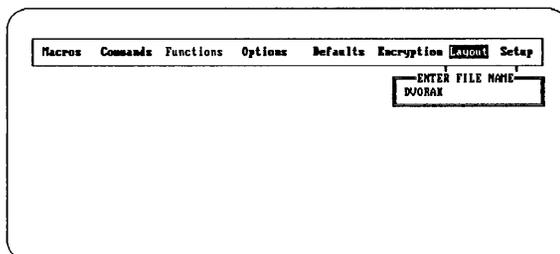


Figure 7-10 Load Layout Window

Example

To load the Dvorak layout from the file DVORAK.LAY, type

Alt / L L DVORAK or start SuperKey with this command:

KEY DVORAK /LL

Load Macro

Press

Alt / M L (or use FILENAME /ML parameter on DOS command line)

Description

The Load macro command loads a macro that was stored in a disk file into macro memory. Any macros you've been using so far will be deleted. (If you don't *want* to delete them, see Merge Macro on page 148.)

To load a macro file, press **Alt /** to get the main menu, then press **M** and **L**. A window opens, requesting you to enter the name of the file you want to load: The name you enter here can be any legal file name, optionally including a DOS path name (see page 123 for a discussion of file and path names). If you do not specify a last name, the last name .MAC is automatically assumed.

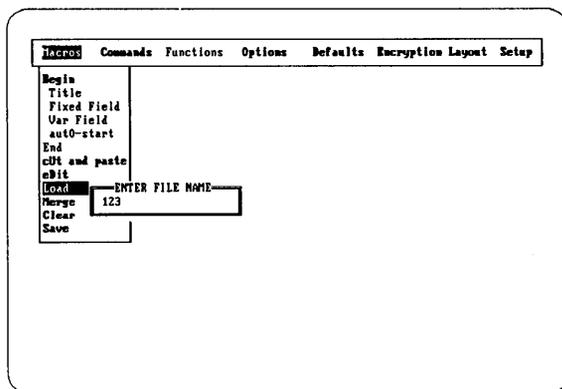


Figure 7-II Load Macro Window

On your distribution disk are a number of files containing macros; these files have the last name .MAC. You can load any of these files, or ones you make yourself.

See page 189 for more information about the .MAC files.

Example

To load the WS.MAC file on your distribution disk, press

   
WS 

When this macro is first loaded, you will see a brief message describing the macros contained in WS.MAC (special macros to be used with WordStar).

To automatically load a macro when you start SuperKey from a .BAT file, include a command like this in the file:

```
KEY MYMACRO /ML
```

That will cause the macro file MYMACRO.MAC to be loaded.

See Also

Auto-start macro

Merge macro

Save macro

Logged Drive Function

Press

Description

Logged drive is a function that can be included in a macro. When the macro is played back, the letter of the currently logged drive is displayed.

To enter a Logged drive function into a macro, first activate the main menu, then select Functions, then Logged drive:

See Also

Functions

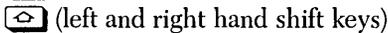
Macro

Description

A macro is any number of characters that you store in a key. This key no longer contains its original one character, but all the characters in the macro.

When the key is pressed, all the characters in the macro are played back, exactly as if all the keys stored in the macro were pressed again.

SuperKey lets you assign macros to almost any key on the keyboard, and to combinations of these keys with the following “shift-state” keys:



A key combination is achieved by holding down one of these shift-state keys while pressing another key. To get a , for example, you must hold down the key and then press the key.

The keys to which you *cannot* assign macros are the shift-state keys alone:



and any of SuperKey’s command keys. These are listed on page 137, and can also be found on the Defaults menu (press to see them).

Although you can assign macros to any other key, it could cause confusion if you inadvertently assign macros to letters or numbers. If you assign a macro to an alphabetic key (through) with or without or to a numeric key, you will therefore get a warning.

See Also

Key symbols

Skip key

Macro Directory

Press

Alt / S M

Description

Use Macro directory to specify where SuperKey will look for its macro (.MAC) and layout (.LAY) files. This location can be a full DOS path name (see page 123 for an explanation of directories and paths), or it may just be a drive name such as A: or B:. If you just enter a drive name, the files will be found on the currently active directory of that drive.

The drive or path you specify becomes active immediately when you press **Alt /**. If you want SuperKey to find its files on the logged drive/current directory, don't enter anything—just press **Alt /**.

The directory you specify is stored only in your computer's internal memory. If you want SuperKey to use the new setup next time you load it, you must record it permanently by selecting Save setup on the Setup menu.

The directory you specify is used only when you type a file name with no path name. You can override the default directory by entering a full file path when specifying macro and layout file names.

Examples

Suppose you want to collect all your macro and layout files in a directory called \MAC. Simply enter:

Alt / S M \MAC Alt /

SuperKey will now look for .MAC and .LAY files on the \MAC directory unless you override it as described above. If you are working with floppy disks, you are likely to want SuperKey to find its files on the A: drive, so you'll enter

Alt / S M A: Alt /

See Also

Save setup

Macro Memory

Description

Your computer contains electronic circuitry where programs and data can be stored and retrieved. This circuitry is your computer's *memory*, also called RAM (Random Access Memory), because information can be read and written anywhere in a large array of memory "cells."

The amount of memory varies among computers. The usual minimum is 64 Kbytes (one Kbyte = 1024 bytes) one byte being equal to one memory "cell." Many 16-bit computers have much more memory, however, and many can be expanded beyond their standard configuration by adding extra chips or extra boards to the computer.

SuperKey uses only part of your computer's memory. The resident part of SuperKey (the part of the program that remains in memory at all times) uses some memory, and more memory is set aside to hold macros. We call this part of memory the macro memory.

The amount of memory required for the resident part of SuperKey cannot be varied, but you *can* control the amount of memory reserved for macros. The default macro memory size is 8000 bytes, but you can use the KEYINST installation program to choose a smaller area if you have memory problems, or a larger area if you plan on defining many or very long macros (see page 183 for details on using KEYINST).

When you load SuperKey, some information about the memory in your computer is printed on the screen, as shown in Figure 7-12.

```
655360 bytes total memory
597968 bytes were free
530424 bytes free
9588 bytes macro memory

C:\BORLAND\KEY >_
```

Figure 7-12 SuperKey Memory Information

It first tells you how much memory you have in total, then the memory available after the operating system (DOS), and other resident programs (buffers, drivers, spoolers, RAM disks, or what have you). The third line shows the amount of memory left after SuperKey was loaded. This figure tells you how much memory is left for your application program. If it is too small, you can either add more memory to your computer or install SuperKey with a smaller macro memory. The fourth line shows the size of the currently installed macro memory.

Main Menu

Press

Alt / (may be redefined, see page 90)

Description

Figure 7-13 shows the main menu, which is the window that appears at the top of the screen when you press **Alt /**. The main menu can be accessed at any time, even while you are defining a macro (as in the following example).

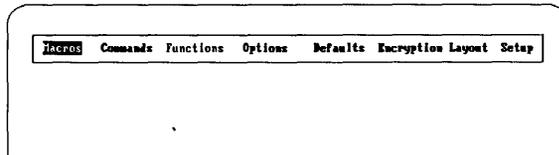


Figure 7-13 Main Menu

You pull down a submenu from the main menu in the following ways:

- either use the **←** and **→** keys to move the highlighted bar to the desired item, then press **↓** to pull down the menu
- or press the capital letter in the item you want. For example, to pull down the options menu, press

Alt / O

If you select by pointing with the bar and pressing **↓**, you must press **Esc** to exit a menu. If you select by pressing letters, your choice is immediately executed, and you automatically exit the menu. This saves quite a few keystrokes.

Once you have opened a pull-down menu, you can select an item in the same two ways you can use on the main menu:

- either use the **↓** and **↑** keys to move the bar to the item you want, then press **→** to select that item. Then press **Esc** to exit the pull-down menu.
- or press the capital letter in the desired item. If you are in the Functions pull-down menu, you would press **T** to select the Time function. The Time function is then recorded in the macro, and you immediately exit the menu.

When you use this selection method, you don't press **←** to select the item, and you don't press **Esc** to exit the pull-down menu. It is much faster than the first method, since it requires fewer keystrokes.

In some cases, that's all there is to it. The Time function, for example, is entered into the macro just by choosing it. Other things on the menus require you to enter a value, for example the duration of a delay, or the default value of a command key. All menu items are described in detail throughout this section.

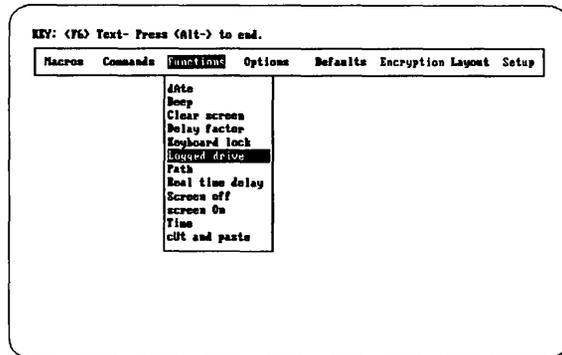
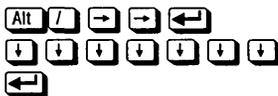


Figure 7-14 Pull-down Menu (Functions)

Example

To enter the Screen off function into a macro you are defining, press the following keys:



Or, you could select from the menus by pressing the capital letters in the desired items:



If you press these keys rapidly, you will just see the menus flash by on the screen.

You can also combine the two methods. For example, select Functions by pointing, and Time by letter:



In either case, the macro will look like this if you look at it in the macro editor:

<CMD>FT<CMD>

Merge Macro

Press

Alt / M M

Description

The Merge macro command loads a macro contained in a disk file into macro memory without deleting other macros already there. Merging comes in handy when you have defined different macros in several sessions with your computer, but want to use the macros all at once. Only if macro memory and the file *both* contain a macro assigned to the *same* key will the old macro be deleted.

To merge a macro file into memory, press **Alt /** to get the main menu, then press **M** twice. The window shown in Figure 7-15 opens, requesting you to enter the name of the file you want to merge. The name you enter here can be any legal file name, optionally including a DOS path name (see page 123 for a discussion of file and path names).

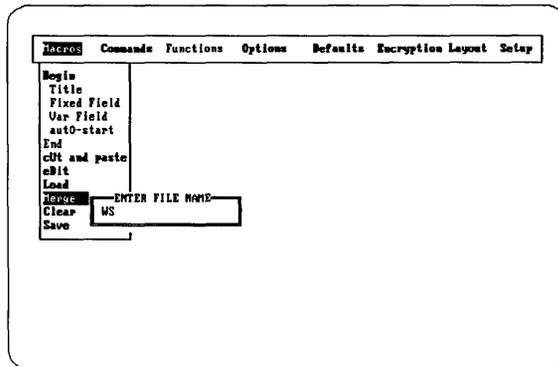


Figure 7-15 Merge Macro Window

Example

The distribution disk contains three files, MERGE1.MAC, MERGE2.MAC, and MERGE3.MAC, which you can merge in different orders to get acquainted with the way merging works. Consider Merge1 and Merge2 below, with characters assigned to the function keys **F1** - **F4**:

```
Key:  F1  F2  F3  F4
-----
Merge1  ABC      GHI
Merge2      DEF      JKL
```

If you first load Merge1 into memory, then merge Merge2, the result will be:

```
Key:  F1  F2  F3  F4
-----
Merge1  ABC      GHI
Merge2      DEF      JKL
-----
Memory:  ABC      DEF      GHI      JKL
-----
```

Now, if you merge a third file that contains a macro key already in memory, the existing definition will be replaced:

```
Key:  F1  F2  F3  F4  F5
-----
Memory:  ABC      DEF      GHI      JKL
Merge3:      123      JKL      456
-----
Memory:  ABC      DEF      123      JKL      456
-----
```

See Also

Load macro
Save macro

One-finger Mode Option

Press

Alt / O O (or use /OO parameter on DOS command line)

Default value

OFF

Description

One-finger mode lets you enter commands that must normally be pressed *simultaneously* as a *sequence* of keystrokes. This is useful if you can only use one hand, if you use a mouth stick, or if you have only one hand free for the computer at any one time.

To switch ON one-finger mode, select Options on the main menu, then One finger on the pull-down menu that appears. You can then press **←** to toggle the status to the opposite of its current value (OFF if it's ON and ON if it's OFF), or you can press **+** or **-** to explicitly turn one-finger mode ON or OFF.

Example

Use the following keystrokes to toggle one-finger mode on and off:

Alt / O O

or start SuperKey with this command:

KEY /OO

Now, if, for example, you want to reboot your computer using the **Ctrl Alt Del** sequence, you can press these three keys in sequence instead of holding down **Ctrl** and **Alt** and then pressing **Del**.

See also

Command line parameters

Options

Options

Description

Options are SuperKey features that you can turn on or off whenever you need to. They are controlled from the Options menu. To reach them, press **Alt / 0**. To switch an option on and off, move the highlighted bar to it or press the capital letter in the name. Then press **←** to toggle the status, or press **+** or **-** to explicitly switch the status to ON or OFF. Playback delay and proTect delay require a little extra information. A small window will open, prompting you to enter the necessary values.

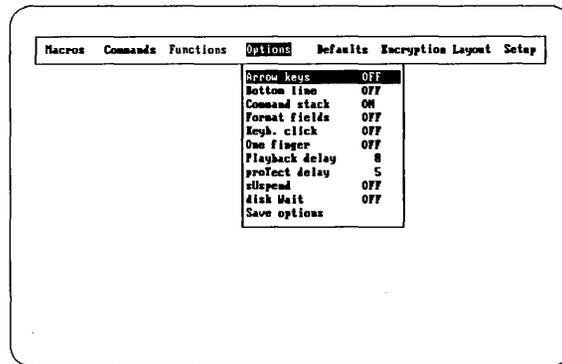


Figure 7-16 Options Menu

You can also control options with command line parameters when you start SuperKey. Each parameter on the command line must be followed by a **+** or a **-** to turn the option ON or OFF, and in the case of Playback speed and proTect delay, a number also. See page 77 for information about using command line parameters.

If you want to make the new status of the options permanent, you can save your options by selecting Save options on the Options menu.

Each option on the menu is described separately in this section. Table 7-6 lists all options, their default values, and the command line parameter you can use to set them (which, incidentally, are the same keys you press to access the options via the main menu).

Table 7-6 List Of Options

	Parameter	Default	Press
Arrow keys	/OA	OFF	
Bottom line	/OB	OFF	
Command stack	/OC	ON	
Format fields	/OF	OFF	
Keyboard click	/OK	OFF	
proTect delay	/OT	OFF	
One finger	/OO	OFF	
Playback delay	/OP	0	
Real time delay	/OR	0	
sUspend	/OU	OFF	
disk Wait	/OW	OFF	

Examples

If you want to toggle keyboard click on, press these keys:

or start SuperKey with this command:

KEY /OK+

You could also make a macro that would toggle the option ON and OFF:

See Also

Command line parameters

Save options

Path Function

Press

Alt / F P

Description

Path is a function that can be included in a macro. When the macro is played back, the path name of the currently logged directory is displayed. (See page 123 for a description of path names.)

To enter a Path function into a macro, press **Alt /** to activate the main menu, then select Functions, then Path.

Example

To make **Alt P** display the current path, enter

Alt = Alt P Alt / F P Alt -

If you look at this macro in the editor, it looks as follows:

```
<CMD>FP<CMD>
```

where the **Alt /** command is symbolized by `<CMD>`.

See Also

File names

Functions

Playback

Playback is the term used in this book to describe what happens when you press a key to which a macro has been assigned.

SuperKey is like a “tape recorder” placed between the keyboard and the computer. You can order it to “record” your keyboard activities, assign them to a single key, then order it to “play back” what you’ve recorded.

The “recording” takes place either interactively or with the aid of the macro editor. The “recording” is stored in SuperKey’s macro memory and can be just one character, or thousands of characters.

When the key associated with the macro “recording” is pressed, the characters in macro memory are played back to the computer, just as if they had been entered on the keyboard.

See Also

Macro

Playback Delay Option

Press

Alt / O P (or use /OP parameter on DOS command line)

Default value

0

Description

The Playback delay option controls the speed at which characters are transmitted by SuperKey. The default delay 0 gives maximum speed, about 1000 characters per second. If your program cannot receive input at that speed—and many programs can't, especially if the macro is long or contains many commands—then you must reduce the playback speed. You'll know you need to increase the playback delay if your program beeps furiously at you, creates "garbage" on the screen, mixes up macros, or otherwise bungles things.

To change the playback delay, first select Options on the main menu, then Playback delay on the pull-down menu. Now enter a number in the range 1 through 999. You'll probably want to experiment with this number until everything runs smoothly.

This number is not a time; it is the number of times SuperKey lets the receiving program check the keyboard status before sending the next character. The actual delay varies depending on the program you're using.

Example

Use the following keystrokes to set a playback delay of 10:

Or start SuperKey with this command:

KEY /OP10

See also

Options
Real time delay
Save options

Protect Delay Option

Press

Alt / 0 T (or use /OT parameter on DOS command line)

Default value

OFF

Description

Use the protect delay option to protect your screen against “burn in,” which occurs if the same image is displayed for a prolonged period of time. The image is “burned in” to the screen, leaving a “ghost” impression even when the image is not there—and thereby damaging your monitor. When this option is ON, the screen goes blank after there has been no activity on the keyboard for a time you specify—the protect delay. To turn the screen ON again, just press any key on the keyboard. Whatever was previously on the screen will reappear.

To activate screen protection and set the protect delay, first select Options on the main menu, then protect delay on the pull-down menu. Then press **←** to toggle the status to the opposite of its current value (OFF if it’s ON and ON if it’s OFF), or press **+** or **-** to explicitly turn the protect delay ON or OFF.

When you toggle protection ON, an additional window opens, and you can enter the desired delay time or accept the displayed value by pressing **←**.

You’ll probably want screen protection to be a permanent feature, so be sure to select Save options on the Options menu after specifying the delay. Otherwise, you’ll have to do it again the next time you turn on your computer.

Examples

Use the following keystrokes to toggle the status of the screen protection:

Alt / O T

or turn it ON with this command line parameter:

KEY /OT+

The following keystrokes turn ON screen protection and set the delay time to three minutes:

Alt / O T ← 3 ←

and so does this command line:

KEY /OT+3

See also

Options
Save options

Pull-Down Menu

Description

A pull-down menu is a window that opens below the main menu, displaying various functions or features that you can select. Once you have opened a pull-down menu, you can select an item in the following ways:

- either** use the **↓** and **↑** keys to move the highlighted bar to the item you want, then press **↵** to select that item, or to turn it ON or OFF. You can also press **+** to explicitly switch the function ON, or **-** to switch it OFF. You can then move to another item on the pull-down menu. When you're through, press **Esc** to exit the pull-down menu and return to the main menu.
- or** press the capital letter in the desired item. If you are in the Options menu, you would press **K** to toggle the Keyboard click option to the opposite of its current status (ON or OFF). You exit the menu immediately—you don't have to press **↵** or **Esc**. This method is faster than selecting by moving the bar, since it requires fewer keystrokes.

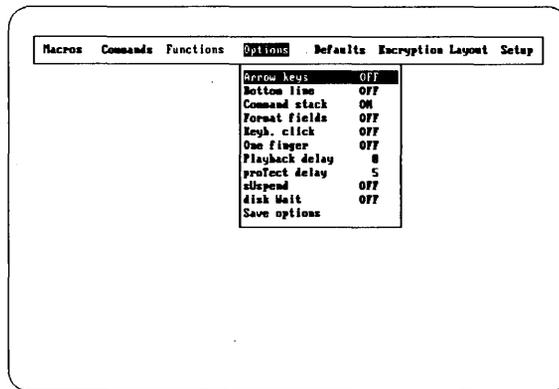


Figure 7-17 Pull-down Menu (Options)

In some cases, that's all there is to it. The Time function, for example, is entered into the macro just by being chosen. Other things on the menus require you to enter a value, for example the duration of a delay, or the default value of a command key. How each item works on the pull-down menus is described separately in this chapter.

Examples

If you want to make a macro to switch ON the Keyboard click option, press the following keys:

Or, you could select from the menus by pressing the capital letters in the desired items:

If you press these keys in rapid succession, you will just see the menus flash by on the screen.

You can also combine the two methods. Select, for example, Options by letter and click with the bar:

The last  is to exit the main menu.

See Also

Main menu

Real Time Delay Function

Press

Alt / F R

Description

Real time delay is a function that can be included in a macro. When the macro is played back, playback pauses for the time you have specified.

To enter a real time delay into a macro, press **Alt / F R**. A small window opens, requesting you to enter a time delay. Move the cursor with the **→** and **←** keys and enter a number from .10 (tenth of a second) to 99:59:59.99.

See Also

Delay function
Functions
Playback delay

Save Defaults

Press

Alt / D S (or use /DS parameter on DOS command line)

Description

To save the currently selected defaults, press **Alt /** to get the main menu, then press **D** and **S**. The current defaults setup is now saved in the KEY.COM file used to start SuperKey. To save defaults, this file must be present on the drive and directory that was active when you first started SuperKey.

See Also

Defaults

Save Macros

Press

Alt **/** **M** **S** (or use `/MS` parameter on DOS command line)

Description

The Save macros command saves all macros that are in macro memory in a file. If you want to use the macros you've been defining again—the next time you turn on your computer—you need to save them before you turn off your computer.

To save macros, press **Alt** **/** to get the main menu, then press **M** and **S**. A window opens, requesting you to enter the name of the file in which to save the macros. It's a good idea to use a name that suggests the purpose of the macros, such as `CALCSTAR`, `DBASE`, or the like. The name you enter here can be any legal file name, optionally including a DOS path name (see page 123 for a discussion of file and path names). If you do not specify a last name, the last name, `MAC` is automatically assumed.

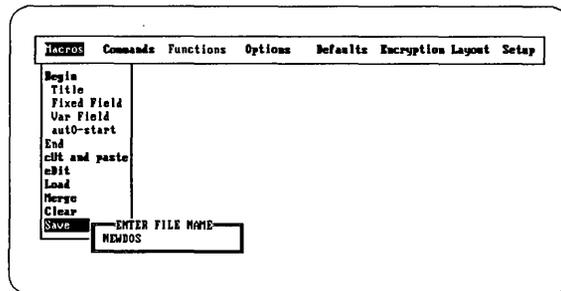


Figure 7-18 Save Macros Window

Example

To save some macros you are using with Turbo Lightning, press **Alt** **/** **M** **S**, then type `LIGHT` **↵**. Now, the next time you use Turbo Lightning, you can load the `LIGHT` file into SuperKey's macro memory.

You can also save the macros with the following command line parameter:

```
KEY LIGHT /MS
```

See Also

Load macros

Save Options

Press

Alt / **O** **S** (or use /OS parameter on DOS command line)

Description

To save the options you are using (on the Options menu), press **Alt /**, then press **O** and **S**. The current options selections are now saved in the KEY.COM file used to start SuperKey. For your options to be saved, this file must be present on the drive and directory that was active when you first started SuperKey.

See Also

Options

Save Setup

Press

Alt / S S (or use /SS parameter on DOS command line)

Description

Any changes you make to the Setup menu (except macro memory size) become active immediately, but they are not permanently recorded in SuperKey. So if you want your new setup to take effect the next time you start SuperKey, you must first save it.

To save the current setup, press **Alt /** to get the main menu, then press **S** twice. The current setup is now saved in the KEY.COM file which was used to start SuperKey. For the setup to be saved, this file must be present on the drive and directory that was active when you first started SuperKey.

See Also

Setup

Screen Off Function

Screen On Function

Press

Alt / F S (Screen off)

Alt / F O (Screen on)

Description

These are functions that can be included in a macro. When played back in a macro, they turn the screen off and on, respectively. Unlike Clear screen, Screen off does not clear the contents of the screen—it only blanks it out temporarily. To turn the screen back ON, use the screen On function.

To enter these functions into a macro, first activate the main menu (press **Alt /**), then select Functions, then Screen off or screen On.

Example

To make **Alt F9** turn the screen OFF, enter:

Alt = Alt F9 Alt / F S Alt -

To make **Alt F10** turn the screen ON, enter:

Alt = Alt F10 Alt / F O Alt -

Setup Menu

Press

Alt / S

Description

Use the Setup menu to specify where SuperKey should look for its macro and layout files and files to be encrypted.

The default setup does not specify any disks or directories, so SuperKey will look for its files on the currently logged disk and the currently active directory. But if you are using a hard disk and want your macros placed in a separate directory, or if you are using floppy disks and you can't fit the macro and layout files on your work disk, you can use Setup to tell SuperKey where to look for its files.

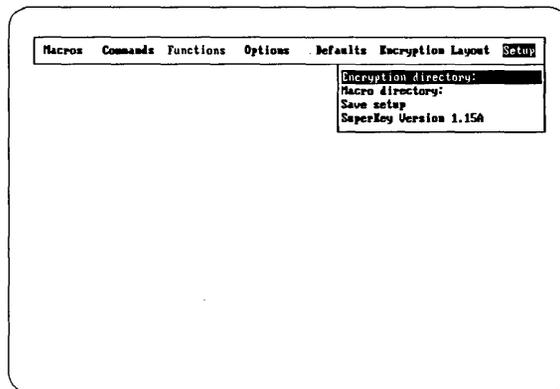


Figure 7-19 Setup Window

Each item on the Setup menu is described separately in this chapter.

Examples

Suppose your computer has two floppy disks, that you have all your programs, macro files, and layout files on your DOS system disk in the A: drive, and you are usually logged on the B: drive. Now you want SuperKey to find its files on the A: drive as well. Press

Alt / S M A: 

If you want to record this setup permanently, press

Alt / S S

If you have a hard disk, you may want to collect all your macro and layout files in a separate directory, for example \MAC. To make SuperKey find its files here, press

Alt / S M \MAC 

Finally, let's look at the encryption directory. The default is the current directory, so when you encrypt files, you will encrypt the files on the disk and directory that is currently active. You can use Setup to select another directory if, for example, you have a directory called \MAIL where you keep your electronic mail; you can encrypt files in that directory regardless of where you currently are by entering

Alt / S E \MAIL 

This also ensures that you will not accidentally encrypt files in a wrong directory. That can easily happen if encryption uses any currently active directory, because DOS does not ordinarily tell you where you are. There is a way to make DOS do that, however; check page 194.

See Also

Encryption directory
Macro directory
Macro memory size
Save setup

Show Titles

Press

Alt **PrtSc** (may be redefined, see page 90)

Description

The show titles command opens a window displaying the symbols and titles of all macros in memory. Figure 7-20 shows an example of a titles window. If there are more titles than will fit into a single window, you can page forwards and backwards with **PgDn** and **PgUp**.

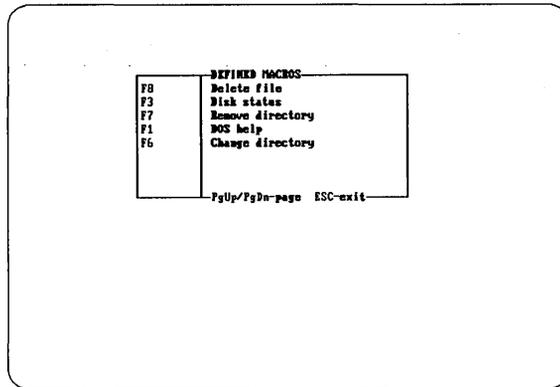


Figure 7-20 Titles Window

See also

Defaults

Titles

Skip Key

Press

 (may be redefined, see page 90)

Description

Use the skip key to access the original meaning of a key, temporarily bypassing the macro assigned to it.

Thus, to bypass a macro, first press , then the desired key. The macro is skipped only this one time. The next time you press the key, the macro is played back.

In the macro editor, the skip key has a slightly different effect. It is used to enter the editor's function keys into the macro (see page 101).

The skip key is essential when you want to swap the meaning of one key for another. For instance, let's say you want to switch the  and  keys so that their meanings are reversed. When you redefine the  key to  by typing the following:

 /      -

you no longer *have* an  key to define to the (former)  key! You would use the skip key to “get the  key back” temporarily, as follows:

 /      -

Example

You have assigned a macro to , and now you need to use —the original , not the macro—to access your application program's help system. Press

See Also

Defaults
Edit macro

Stop

Press

  (may be redefined, see page 90)

Description

The stop playback command immediately aborts a macro that is being played back. It is useful if you have inadvertently invoked a long macro, or if you have an “infinite loop”—a macro that calls itself. The only thing that will stop such a macro is the stop playback command.

Suspend Option

Press

Alt / O U (or use /OU parameter on DOS command line)

Default value

OFF

Description

Use the sUspend option to temporarily “switch off” SuperKey. You might want to use sUspend when you need to temporarily reuse the original meanings of macros you’ve defined, or the original layout of your keyboard (if you’ve loaded a new layout). When suspension is ON, all macros become ineffective, and the keyboard returns to normal, even if you have an alternate layout loaded.

Press **Alt / O U** to turn suspension on. SuperKey is now inactive, and can therefore not be activated again from the keyboard. To turn suspension OFF and reactivate SuperKey, you must therefore enter this command on the DOS command line:

```
KEY /OU
```

See also

Options
Save options

Time Function

Press

Alt / F T

Description

Time is a function that can be included in a macro. When the macro is played back, the time is read from the DOS clock and displayed as a string of characters in this format:

HH:MM:SS

To enter a Time function into a macro, press **Alt /**, then select Functions, then Time.

Example

To make **Alt T** display the time, enter

Alt = Alt T Alt / F T Alt -

Titles

Press

Alt **'** (may be redefined, see page 90)

or **Alt** **/** **M** **T**

Description

Use Titles to give your macros names. This will help you remember what your macros are used for, and make it easier for others to use the macros you design. Once you have given a macro a title, it will be displayed along with the name of the key in the show titles window.

After you have started a macro definition, press

Alt **/** **M** **T**

or

Alt **'**

to create a title. A window appears, and you enter a descriptive title up to 30 characters long. Press **↵** to end the title.

If you want to add a title to an existing macro, you can use the macro editor (see page 101).

Example

The following keystrokes will give a title to the macro stored in **F7**.

Alt **=** **F7**

Alt **'** Copy range of cells **↵**

:

:

Alt **-**

See also

Show titles

Variable Field

Press

Ctrl **-** (may be redefined, see page 90)

or **Alt** **/** **M** **V**

Description

A variable field included in a macro defines a space into which you can enter text or numbers during macro playback. When you stop typing and press **↵**, macro playback resumes.

A variable field is as long as what you type. Special characters (format and entry control masks) can be used to control what can be typed into a field, and how it is displayed on the screen.

As long as your entry is not terminated, you can press **⌫** (the backspace key) to delete characters from right to left.

A variable field is terminated when you press **↵**.

The maximum length of a variable field is 255 characters.

Example

The following keystrokes will establish a variable field in the macro in **F3**, following the words *Product Description*.

Alt **=** **F3**

Product description: **Ctrl** **-** **Ctrl** **-**

Alt **-**

See Also

Entry control
Field formats
Fixed field

A P P E N D I X **A**

*Using SuperKey With Other
Borland Products*

This appendix contains some suggestions for using SuperKey with Borland's other programs. If you're already a Borland veteran, you'll probably think of many other ways to use SuperKey in harmony with these programs. If you're not familiar with our other products, we hope this information will encourage you to try them out!

SuperKey can enhance the performance of *any* program, since it reduces your keystrokes. As examples, let's look at SideKick and Turbo Lightning for some SuperKey ideas.

SideKick

There are a couple of goodies for SideKick owners on the distribution disk, in the files GRAPH.MAC, GRAPH2.MAC, and TAPE.MAC.

GRAPH.MAC and GRAPH2.MAC

These two sets of macros turn the numeric keys on the numeric keypad into semi-graphic characters. With the SideKick Notepad in *Graphic* mode (press **Ctrl** **Q** **G**), you can now enter lines and corners for frames, flow diagrams, and the like.

Press **Alt** **/** **M** **L** **GRAPH** **↵**

to load the file GRAPH.MAC.

With **Ctrl** pressed, the numeric keypad now produces single lines, and if you press **Alt**, it produces double lines.

GRAPH2.MAC contains a secondary character set with symbols that combine single and double lines. Once the GRAPH file is loaded, you can switch between the two by pressing

Alt **Ins**

TAPE.MAC

This set of macros makes the SideKick Calculator use the Notepad as a “paper tape.” When you load this macro, the Notepad will record all calculations performed with the Calculator. (Whatever you’ve previously written in the Notepad will remain undisturbed.)

Calling SideKick with SuperKey

You normally activate and exit SideKick by pressing a combination of the **⇧**, **Ctrl**, and **Alt** keys—for example **Ctrl** **Alt** or **⇧** **⇧**. Since these shift-state keys cannot be recorded in macros by themselves, this method cannot be used to activate SideKick from a macro.

SuperKey therefore uses its own command to activate and leave SideKick. You’ll find this command on the Commands menu. Press

Alt **/** **C** **S**

Examples

This macro will alternately activate and exit SideKick:

Alt = F10
Alt / C S
Alt -

This macro will activate SideKick, enter the Notepad, and go to the bottom of the note file:

Alt = F10
Alt / C S
Alt N Ctrl Q C
Alt -

The SideKick Notepad is a great way to create and edit macros, too (see page 58).

Turbo Lightning

SuperKey macros can help you speed up Turbo Lightning's functions. This is especially useful for functions that you want to turn OFF and ON frequently. For instance, you may want to switch between screen and keyboard mode (selection M on the Edit Environment menu). We've included a macro to do this for you, under the file name KEYBOARD.MAC (Turbo Lightning Install/Supplemental Files Disk). This macro will let you toggle where your application receives its input—from the screen or the keyboard. After loading KEYBOARD.MAC, just press **Alt F4** when you want to change back and forth between keyboard and screen mode. This macro saves you five keystrokes.

You could set up another macro that would change to screen mode and check the word at the cursor; the macro would need to include this key sequence:

Alt F8 **Left Arrow F2** **M** **Esc** **Esc** **Alt F9**

After using this macro, you could then use the KEYBOARD.MAC macro (**Alt F4**) to switch back to keyboard mode.

You're certain to come up with other macros as you use Turbo Lightning and discover which of its functions you use most often.

A Note for Reflex Users

SuperKey will not work properly with Reflex versions earlier than 1.11. If you have an early version of Reflex and want to use it with SuperKey, contact Borland for an update.

A P P E N D I X **B**

Using SuperKey With Batch Files

The AUTOEXEC.BAT File

Starting SuperKey is as simple as typing KEY  on the DOS command line. But if you are using an AUTOEXEC.BAT file to start up your system, you might as well include SuperKey in it. (If you don't know what an AUTOEXEC.BAT file is, see Appendix E, "A DOS Primer.")

It is essential that SuperKey be loaded into your system according to the following sequence if you are using other resident programs:

- Load any non-Borland resident programs first (print spoolers, RAM disks, etc.).
- Load SuperKey next.
- If you have it, load Turbo Lightning.
- Finally, if you have it, load SideKick last.

Let's say you have an existing AUTOEXEC.BAT file that automatically loads a print spooler (spool) and your word processing program (wp) when you start up your computer. In addition, you are a SideKick owner. The file might look like this (assuming you use tree-structured directories; if you don't use directories, the CHDIR command is not necessary):

```
spool
sk
chdir / wp
ws
```

To include SuperKey in your AUTOEXEC.BAT file, you must insert it after the spool program, but before SideKick. Borland's SideKick must always be the last resident program loaded into your system. The following commands will do the trick:

```
copy con: autoexec.bat ⌨
spool ⌨
chdir / light ⌨
light ⌨
sk ⌨
chdir / wp ⌨
ws ⌨
Ctrl Z ⌨
```

Now, the next time you turn on your computer, SuperKey will be automatically loaded at the right time.

Loading Macros with .BAT Files

If you have a macro file for each program you use, it's a good idea to make a batch file that automatically loads the right macro file, then starts the program, and finally clears the macros from memory when you exit the program.

Here is an example .BAT file for Lotus 1-2-3:

```
KEY 123 /ML
LOTUS
KEY /MC
```

The reason to clear the macros when you exit is that some macros may have unwanted effects if used inadvertently in the wrong programs.

Of course, .BAT files can do much more than load macro files. In fact, you can pass almost any command to SuperKey on the DOS command line. Please see page 77 for a full description of all command line parameters.

A P P E N D I X **C**

Installation

This appendix tells you how to use the KEYINST.COM installation program to custom-configure SuperKey for your computer. Although SuperKey is automatically set up for an IBM PC or compatible, you may want to change some of the default settings in the program. If you want to install SuperKey on a PCjr, type PCjr on the DOS command line before using SuperKey the first time. KEYINST allows you to do the following:

- Change the monitor type.
- Change the colors of SuperKey's windows.
- Change macro memory size (the amount of space reserved for storing macros in your computer's memory).

Note: If you are using a one-color monitor with a graphics video board, type MODE BW80 on the DOS command line before running KEYINST.

During installation, all KEYINST files and the KEY.COM and KEYDES.COM files must be on the logged drive. To start installation, type

```
KEYINST 
```

The menu in Figure C-1 now appears on your screen. Unless you are using the DES encryption scheme (see page 104) choose KEY.COM. After you have chosen the SuperKey file to install, press  and the main installation menu will appear. Use the  and  keys to move to the item you want to change, then press .

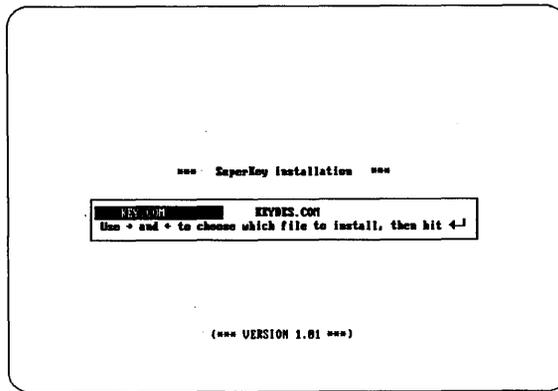


Figure C-1 First Installation Screen

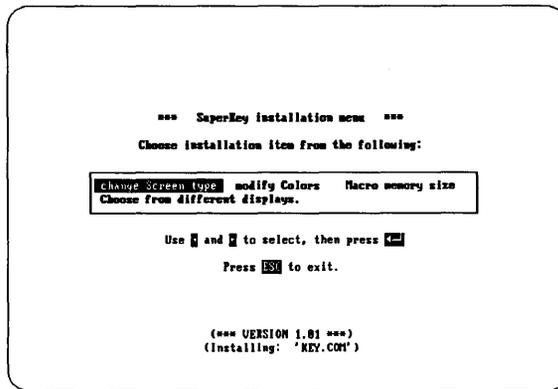


Figure C-2 Installation Main Menu

Screen Type

When you select Screen on the Installation menu, another pull-down menu appears which allows you to select the screen mode you want SuperKey to use.

Choose one of the following display modes by moving to it with the  and  keys, then press .

Default display mode

By default, SuperKey will always operate in the mode that is active when you load it. If you change the screen mode later, SuperKey will still use the mode which was active when you loaded it. SuperKey always uses 80×25 character mode even if 40×25 is active, and it will use color mode if graphics mode is active when you loaded it.

Monochrome display

SuperKey will use monochrome mode, no matter what mode is active, and switch back to the active mode on exit.

Color display 80×25

SuperKey will use color mode with 80×25 characters, no matter what mode is active, and switch back to the active mode on exit.

B/W display 80×25

SuperKey will use black and white mode with 80×25 characters, no matter what mode is active, and switch back to the active mode on exit.

At the bottom of the Screen Installation Menu are two questions you may need to answer:

Blinking display?

If you installed a monochrome display, you don't need to answer this question. A standard IBM color/graphics controller blinks when it scrolls, but some other controllers don't. Answer Y or N as appropriate. This causes SuperKey to use your video controller with as little screen flicker as possible.

Blank background?

When you select default or color screen mode, you will be asked whether you want the background cleared in graphics mode; default is Y (YES). When you use SuperKey in color graphics mode, the screen background will fill up with flashing “modern art” characters when you call up the menus. If you don’t mind this, you can save 4K of RAM by answering N (NO) to this question.

Colors

The Colors selection on the Installation Menu allows you to change the colors (and monochrome attributes, if you have a monochrome monitor) used for window frames and text in SuperKey. The standard colors and attributes have been carefully chosen to provide clear and informative screen images with standard IBM video controllers and monitors. Some other controllers and monitors, however, may display colors differently, so you can use this installation to design the windows to look their best with your particular hardware.

The Colors installation first lets you choose which of SuperKey’s windows you want to install. Select from the menu with the arrow keys, then press **↵**.

To the right, you’ll see a set of sample windows. Each window has three separate areas, and the selected area will be blinking. The **←** and **→** keys select an area. When the area you want to re-color blinks, press **↵** to chose a new color. The area will then stop blinking, and a bright frame will appear around the current color of that area. You can move this frame around to select a new color. The sample window to the right shows the colors as you move the frame.

If you install a monochrome screen, your choice is limited to five different video attributes.

When you’ve decided on a color, press **Esc** to return to the sample windows. Now you can select another area to color. When you’re satisfied with all the colors you’ve chosen, press **Esc** to exit to the window selection menu.

You can then choose another window to color, or press **F2** to save the colors you’ve chosen. If you change your mind about the colors and want to exit without saving them, just press **Esc**.

The changes you’ve made only affect the SuperKey files on disk; the SuperKey you may have already loaded into your computer’s internal memory is not installed. You have to re-load SuperKey and start again to see the new colors.

Macro Memory Size

SuperKey normally sets 8000 bytes of memory aside for macros. If you want to preserve memory, and you only plan to use a few or short macros, you can change this value to something smaller. On the other hand, if you have lots of memory, you can set as much as 65536 (64K) bytes aside for macros, and you will probably never run out of macro memory.

Press **M** to change macro memory size, and enter the new size as desired (in the range 128 through 65536). See page 145 for a full explanation of macro memory.

A P P E N D I X **D**

.MAC Files on the Distribution Disk

Following are the ready-to-use .MAC files on your distribution disk. To use a .MAC file, press

Alt / M L

to load the file into SuperKey's macro memory. Type in the name of the file you want to use; you don't have to type .MAC.

123.MAC	Macros to be used with LOTUS 1-2-3.
123TRNSP.MAC	Same macro file as 123.MAC but with the transparent (<TRANSP)) command included.
BASICA.MAC	Macros to be used with BASICA.
DOS.MAC	Macros that make common DOS commands a little easier to use.
DOSTRNSP.MAC	Same macro file as DOS.MAC but with the transparent (<TRANSP)) command included.
EDIX.MAC	Macros to be used with EDIX.
FOREIGN.MAC	Macros that allow you to use the letters most commonly found in foreign languages.
GRAPH.MAC	Macros that allow you to use extended ASCII characters to do graphics.
GRAPH2.MAC	More macros for graphics.
INVENT4.MAC	A macro demonstrating what beautiful music you can make with SuperKey's sound capability.

LUNCH.MAC	A macro that demonstrates keyboard lock.
MERGE1.MAC	Macros that demonstrate SuperKey's merge facility (explained on page 148).
MERGE2.MAC	Another merge macro.
MERGE3.MAC	Another merge macro.
SUPCALC3.MAC	Macros to be used with SuperCalc3.
TAPE.MAC	Macros to be used with SideKick's calculator that allow you to use the Notepad as a tape.
TAPEOFF.MAC	Macro that allows you to turn TAPE on and off.
TURBO.MAC	Macros to be used with Turbo Pascal.
VISICALC.MAC	Macros to be used with Visicalc.
VOLKSWTR.MAC	Macros to be used with Volkswriter.
WORDIX.MAC	Macros to be used with Wordix.
WORDPER.MAC	Macros to be used with Word Perfect 4.0.
WS.MAC	Macros to be used with WordStar.

A P P E N D I X *E*

A DOS Primer

If you are new to computers or the DOS (MS-DOS or PC-DOS) operating system, you may have trouble understanding certain terms used in this manual. This appendix will give you a brief overview of the following DOS concepts and functions:

- What DOS is and does
- The proper way to load a program
- Directories and subdirectories
- Using AUTOEXEC.BAT files

This information is by no means a complete explanation of the DOS operating system. If you need more details, please refer to the MS-DOS or PC-DOS user's manual that came with your computer system.

What Is DOS?

DOS is shorthand for Disk Operating System. MS-DOS is Microsoft's version of DOS, while PC-DOS is IBM's rendition. DOS is the traffic coordinator, manager, and operator for the transactions that occur between the parts of the computer system and the computer system and you. DOS operates in the background, taking care of many of the menial computer tasks you wouldn't want

to have to think about. But it's ready to leap to the foreground when you need it—that is, when you want to give it a command.

Some DOS transactions are invisible to you—for instance, the flow of characters between your keyboard and the computer, between the computer and your printer, and between your disk(s) and internal memory (RAM). Other transactions are ones that you initiate—by entering commands on the DOS command line; in other words, immediately after the DOS prompt. Your DOS prompt looks like one of the following:

```
A>
B>
C>
```

The capital letter refers to the logged disk (the one DOS and you are using right now). For instance, if the prompt is A>, it means you are working with the files on Drive A, and that commands you give DOS will refer to this drive. When you want to switch to another disk and make it the logged disk, all you do is type the letter of the disk, followed by a colon and a . For instance, to switch to Drive B, just type B:.

There are a few commands that you will use often with DOS, if you haven't already:

DEL or ERASE	To erase a file
DIR	To see a list of files on the logged disk
COPY	To copy files from one disk to another
KEY	To load SuperKey

DOS doesn't care whether you type in upper or lowercase letters, or a combination of both, so you can enter your commands however you like.

We'll assume you know how to use the first three commands listed above; if you don't, refer to your DOS manual. Next, we will explain the proper way to load a program like SuperKey, and that involves the last command—KEY.

How to Load a Program

On your System Disk, you'll find the main SuperKey program, under the file name KEY.COM. This program is necessary for all SuperKey functions, so you always need it when you first start the program. A file name with the "last name", or extension .COM or .EXE means a program file that you can load and run (use) by typing its "first name" on the DOS command line. So, to invoke SuperKey, you simply type KEY  and SuperKey will be loaded into your computer's memory.

Easy, right? It is, but there's one thing you need to remember about loading SuperKey and other similar programs: *you must be logged on the disk and directory where the program is located in order to load it*; otherwise, DOS won't know where to find the program.

For instance, if your System Disk with the KEY.COM program is in Drive A but the prompt you see on your screen is B>, DOS won't know what you're talking about if you type KEY  and will tell you

Bad command or file name

It's as if you were shuffling through the "School Records" file in your file cabinet looking for information about your home finances. You're in the wrong place. So if DOS gives you that message, all you have to do is switch to Drive A by typing A:. Then type KEY  to load SuperKey.

Directories

A *directory* is a convenient way to organize your floppy or hard disk files. DOS allows you to subdivide your disk into sections, much the way you might put groups of manila file folders into separate file boxes. For instance, you might want to put all your file folders having to do with finance—for instance, a bank statement file, an income tax file or the like—into a box labeled "Finances."

On your computer, it would be convenient to make a directory to hold all your SuperKey files, another for your SideKick files, another for your spreadsheet, and so on. That way, when you type DIR on the DOS command line, you don't have to wade through hundreds of file names looking for the file you want. You'll get a listing of only the files on the directory you're currently logged on.

Although you can make directories on either floppy or hard disks, they are used most often on hard disks. This is because a hard disk can hold a much greater volume of data, so there is a greater need for organization and compartmentalization.

When you're at the DOS level (rather than in SuperKey or another program), you can tell DOS to create directories, move files around between directories, and show which files are in a particular directory.

In the examples that follow, we assume you are using a hard disk system, and that you are logged onto the hard disk so that the prompt you see on your screen is C>. If you want to create directories on your floppy disks, just substitute A or B for C in the example.

To make a directory especially for all your SuperKey files, do the following:

- At the C> prompt, type MKDIR KEY . The MKDIR command tells DOS to make a directory called KEY.
- Type CHDIR KEY . The CHDIR command tells DOS to move you into the KEY directory.
- Now, put the SuperKey disk you want to copy from into one of your floppy drives—let's say A for this example—and type COPY A: * . * . (The asterisks are *wild cards* that stand for all files.) The COPY command tells DOS to copy all files on the A drive to the KEY directory on the C drive. As each file on the disk is copied, you will see it listed on the screen.

That's all there is to it. Treat a directory the same way you would a disk drive: to load SuperKey, you must be in the KEY directory before typing KEY , or DOS won't be able to find the program.

Subdirectories

If you are someone who really likes organization, you can further subdivide your directories into subdirectories. You can create as many directories and subdirectories as you like—just don't forget where you put your files!

A subdirectory is created the same way as a directory. For instance, to create a subdirectory from the KEY directory (for instance, for storing your macro files) do the following:

- Be sure you are in the KEY directory.
- Type MKDIR MACROS .
- Type CHDIR MACROS. You are now in the MACROS subdirectory

If you do put your macro files in a subdirectory, just remember to let SuperKey know where they are using the Setup menu.

Where Am I? The \$p \$g Prompt

You probably have noticed that when you change directories, you still see the C> prompt; there is no evidence of what directory or subdirectory you are in. This can be confusing, especially if you leave your computer for a while. It's easy to forget where you were when you left.

DOS gives you an easy way to find out. Just type prompt=\$p \$g and from now on (until you turn your computer off or reboot), the prompt will show you exactly

where you are. Try it. If you are still in the MACROS subdirectory, your DOS prompt should look like this:

```
C:\KEY\MACROS >
```

The AUTOEXEC.BAT file

In the example above, every time you turn on your computer, you must type the prompt command if you want to see where you are. If you'd rather not do this every time, you can set up an AUTOEXEC.BAT file to do it automatically each time you turn on your computer.

The AUTOEXEC.BAT file is a useful tool to get your computer to do things automatically. We won't go into great detail here—see your DOS manual for more information—but we will show you how to create an AUTOEXEC.BAT file that will automatically change your prompt so you know where you are in your directory structure, and then load SuperKey.

If you have an AUTOEXEC.BAT file in your root directory, your computer will do everything in that file when you first turn your computer on. (The root directory is where you see the C> or C:\ prompt, with no directory names following it.)

Here's how to create an AUTOEXEC.BAT file.

- Type CHDIR \  to get to the root directory.
- Type COPY CON AUTOEXEC.BAT . This tells DOS to copy whatever you type next into a file called AUTOEXEC.BAT.
- Type:

```
ECHO OFF  
PROMPT=$P $G   
CHDIR \KEY   
KEY   

```

The  sequence saves your commands in the AUTOEXEC.BAT file.

To test your new AUTOEXEC.BAT file, reboot your computer by holding down the  and  keys and then pressing . This is what you should see:

```
C:\KEY
```

Changing Directories

How do you get from one directory to another? It depends on where you want to go. The basic DOS command for changing directories is CHDIR. Use it as follows:

To move from one directory to another: For example, say you want to change from the KEY directory to one called WP. From the KEY directory, type

```
C:\KEY> CHDIR \WP 
```

Notice the backslash (\) before the directory name.

To move from a directory to its subdirectory: For example, you want to move from the KEY directory to the MACROS subdirectory. From the KEY directory, type

```
C:\KEY\ CHDIR MACROS 
```

In this case, you did not need the backslash, because the MACROS directory is a direct offshoot of the KEY directory. In fact, DOS would have misunderstood what you meant if you had used the backslash in this case. If you had included the backslash, DOS would have thought that MACROS was a directory off the main (root) directory.

To move from a subdirectory to its parent directory: For example, say you want to move from the MACROS subdirectory to the KEY directory. From the MACROS subdirectory, type

```
C:\KEY\MACROS> CHDIR .. 
```

DOS will move you back to the KEY directory. Anytime you want to move back to the parent directory, use a space, followed by two periods after the CHDIR command.

To move to the root directory: The root directory is the original directory. It is the parent (or grandparent) of all directories (and subdirectories). When you are in the root directory, you see this prompt:

```
C:\>
```

Anytime you want to go there, simply type

```
CHDIR \ 
```

The backslash without a directory name signals DOS that you want to return to the root directory.

This has been a quick look at moving around between directories. It is certainly enough to allow you to get around on your disk. For more information, see your DOS manual.

We hope this clears up any confusion you may have regarding DOS. Once you're familiar with the information given here, you may want to study your DOS manual and discover the many things you can do with your computer's operating system. There are many other DOS functions not mentioned here that can simplify and enhance your computer use.

A P P E N D I X *F*

HELP!!!

This appendix provides answers to the most commonly asked questions about SuperKey.

Q: What is the exact loading order if I want to load SuperKey, SideKick, and other memory-resident programs, such as RAM disks and print spoolers?

A: SideKick is always the *last program* to be loaded into memory, and SuperKey gets loaded *just before* SideKick (unless you use Turbo Lightning—see below). So, if you use other memory-resident programs, you load them before SuperKey and SideKick. A typical loading sequence would be:

- Clock
- RAM Disk
- Print Spooler
- SuperKey
- Turbo Lightning
- SideKick

Q: The Functions Menu is not available when I bring up SuperKey's main menu with **Alt J**.

A: Until you define macros, the Function menu is not available; the functions are for inclusion in the macros you define. They do nothing by themselves, but are available once you start defining a macro. Try again by starting a macro definition, then press **Alt J** to bring up SuperKey's main menu, and then press **F** for functions.

Q: When I play back a macro, something seems to be wrong as far as timing goes. The commands don't correspond to the prompts the computer is giving.

A: You should use delays when "playing back" macros. You do not always have to use delays, but when your application needs to pause, SuperKey needs to know this. A good example is when you are exiting WordStar. After pressing the X key to exit, anything you type is lost in the "bit bucket" until you get the DOS prompt back. There is no set rule for using delays, but to determine which delays your particular macro requires, simply play back the macros manually and notice where pauses are required and how long they should be. You can then determine the appropriate delays.

Q: With SuperKey loaded, I cannot use **[Alt]** with the numeric keypad to enter a string of ASCII characters.

A: For SuperKey to be able to define keys like **[Alt][PgDn]**, the standard use of **[Alt]** with the numeric keypad is modified. While SuperKey is resident, use the **[⇧][Alt]** keys in conjunction with the numeric keypad to enter a string of ASCII characters.

Q: While I am defining a macro, if I bring up the Defaults Menu, I cannot see the actual key sequences for default commands, but only their titles.

A: The actual key sequences for default commands are not shown while you are creating a macro, so you won't get confused as to what the current key sequence is. If they were shown, they might not reflect changes made to them by other macros. This function is here so you can change default command key sequences "on-the-fly," but not while defining a macro.

Q: What is the difference between text mode and nontext mode encryption?

A: In a nutshell, if you wish to encrypt a file and leave no traces of an original on your disk, use the default mode or answer N to the question "Text Mode?". If you wish to encrypt a file after making a copy of it, first use "Text Mode" encryption or answer Y to the question "Text Mode?". Text mode encryption is especially useful for changing binary files to encrypted text files for transmission across phone lines via modem to a bulletin board, Easy-Link, CompuServe or to another friend with a modem, etc.

Q: After I invoke a command from the main menu, sometimes the menus disappear and sometimes they stay there. Why?

A: If you use the cursor to select items from the menu, you are using novice mode, which makes the menus stay until you **[Esc]** out of them. If you use the letter of the commands instead of the cursor, the menus disappear immediately after issuing each command. This helps both first-time and expert users get maximum benefit from the pull-down menus.

Q: My display window shrinks when I add more text. Why?

A: As your total macro size grows closer to the “maximum macro memory” size (which defaults to 8000 characters, and is installable with KEYINST from 1000 to 64000 characters), the display window shrinks to conserve macro memory space, so that the macros function properly.

Q: I cannot use one-finger mode to invoke SideKick.

A: You must have SideKick 1.5 or greater to make use of one-finger mode.

Q: When I use one-finger mode to invoke SideKick, the screen flashes SideKick on and off.

A: You must press another key after pressing **Ctrl Alt**, because SuperKey remembers the status of the last-pressed key, and thinks you are still pressing **Ctrl Alt** until you press something else!

Q: When I run KEYINST with a color graphics card and a composite video one-color monitor (black and white), the display looks grainy.

A: Insert your DOS disk and type MODE BW80 on the DOS command line *before* running KEYINST. Then—using KEYINST—make sure you install your screen type for black and white.

Q: When I use either Protect Delay or Screen On/Screen Off, only the cursor turns off, not the whole screen.

A: If your video board is *not* an IBM video board, chances are it is a “non-blinking” video board. To determine if your video board is a “non-blinking” video board, perform the following test: hold down the **←** key (at the DOS prompt) and continue holding it down as the screen fills up with prompts. If prompts scroll off the screen smoothly and the whole screen DOES NOT blink with each scroll, then you have a “non-blinking” video board. If this is the case, you must run the installation program KEYINST and select “Change Screen Type.” When asked the question “Does your screen blink when it scrolls (y/n?)”, answer “N.”

Q: When I use Screen Protect, the manual says I can strike any key to bring back the display. But when I do, the key I strike is passed through to the program—causing unexpected results.

A: Any key you press to reactivate the screen will be passed through. The solution to this problem is to press keys that do not produce a character by themselves, e.g.:

<Alt>

<Shift>

<Ctrl>

The following keys should be used WITH CAUTION—because pressing them changes their status:

<CapsLock>
<Numlock>
<ScrollLock>

Q: When I call up SuperKey over a graphics screen (like Reflex), the screen turns into a pattern of nonsense characters.

A: This is because SuperKey has changed your display from graphics to text mode. When you **[Esc]** out of SuperKey, your display will return to normal. If you don't like this effect, use the KEYINST program to blank the screen in graphics mode; see Appendix C.

Q: I cannot page down through display macros that are larger than one page. When I press **[PgDn]** while the first display window is up on the screen, the display simply disappears.

A: The macro must be a transparent display macro. The command <TRANSP> is present somewhere in the file. With transparent display macros, only one display screen can be shown. **[PgUp]** and **[PgDn]** do not affect these display macros; instead, the keystrokes get sent through to the underlying application. For more information about transparent display macros, see page 48.

Glossary

auto-start macro: A macro that executes (plays back) automatically each time the file containing it is loaded.

beep function: Function that can be included in a macro to produce a sound tone of a certain frequency and duration.

burn-in: Damage caused to a computer screen when the same image is shown on the screen for an extended period of time. The screen's phosphor becomes "worn" in places where characters are displayed, and faint images are seen on the screen even when it is turned off.

command line parameter: One or two letters, preceded by a slash and possibly a file name, used to give commands to the operating system. It is possible to load SuperKey and give it instructions through command line parameters, instead of using the SuperKey main menu.

command stack: Area in your computer's memory that contains the last 256 characters you have entered while in DOS; the stack is accessible at any time from SuperKey, to allow you to re-execute any command without having to re-enter characters.

cut and paste: SuperKey feature that allows you to take (cut) anything displayed on the screen and include (paste) it in a macro.

decryption: Method used to unscramble an encrypted file, by way of a *keyword*.

default commands: Macro values assumed by SuperKey when no other values are specified.

DES encryption: Maximum-security *encryption* method that complies with the DES (Data Encryption Standard) issued by the U.S. National Bureau of Standards. Because it protects your data from even the most sophisticated code-breaker, it is slower than the other encryption method offered by SuperKey.

disk Wait: SuperKey option that prevents SuperKey from sending any characters to disk while the disk is spinning. This protects you from inadvertently losing data on a floppy-disk machine.

display macro: SuperKey macro that is not sent to the keyboard, but is simply displayed on the screen. You can create display macros with a text editor of your choice, to write your own customized, on-screen help text for yourself or other users.

encryption: Method used to scramble the contents of a file so that it is unusable by unauthorized users—i.e., those that don't know the *keyword* that will decrypt (unscramble) the file. SuperKey offers two methods of encryption: a fast algorithm contained in the KEY.COM file that should be used for files that are not of an extremely sensitive nature; and a slower, *DES file encryption* algorithm contained in the KEYDES.COM file, to be used for encrypting higher-security data.

entry mask: Special characters that you enter into a *field* to control the type of characters—numeric, alphabetic, uppercase or lowercase—that may be contained in a macro.

field: A special set of characters included in a macro that will stop playback of the macro, to allow you to enter a value that controls the contents and display of the macro. See *entry mask* and *format mask*.

fill character: Character used to “pad” a *fixed field* if it is not completely filled in. Fill characters are blank spaces by default, but can be specified to be any other displayable character.

fixed field: Set of characters included in a macro that defines a space of a specified size, into which you can enter characters during macro playback. See *variable field*, *format mask* and *entry mask*.

format mask: Special characters that you enter into a *field* to control the way the macro is displayed—left adjusted, right adjusted, centered, or with numeric formats.

interactive macro definition: SuperKey feature that allows you to define macros, using the Begin Macro command, while you are working with an application program other than SuperKey.

keyboard buffer: Area in your computer's memory that stores up to 16 keyboarded characters. The buffer allows you to type and store characters even while the computer is busy reading or writing to disk. SuperKey expands this buffer with its *type-ahead buffer*.

keyword: Sequence of characters, up to 30 characters in length, that is used to encrypt and decrypt files.

layout file: File that contains the format (layout) of your keyboard. This layout can be re-defined using the SuperKey layout editor.

macro: One or more characters that can be recorded in *macro memory* by SuperKey and assigned to a single key. Each time that key is pressed, the recorded characters are played back, as if they had been entered on the keyboard.

macro editor: SuperKey feature that allows you to create new macros and edit ones you've already created.

macro memory: Area of your computer's internal memory (RAM) that SuperKey reserves for storage of macros.

macro file: A file with a .MAC extension that contains a SuperKey macro definition. This file can be loaded into the computer via the **Macros** menu.

nested macro recording: Method of storing macros in which one macro is recorded while another is being played back.

non-text mode encryption: Mode of *encryption* in which the encrypted file is written directly on top of the original, unscrambled file, so that no part of the original file is available on disk. See *text mode encryption*.

one-finger mode: SuperKey feature that changes commands which normally require you to press two or more keys at the same time, into a sequence of single keystrokes.

path: Part of a file name, given on the DOS command line, that specifies sub-directories which lead to the file in a DOS *tree-structured file directory*.

playback: SuperKey function that enables you to press a key to display and/or execute the macro that was recorded (in *macro memory*) for that key.

playback delay: SuperKey option that controls the speed at which macros are transmitted to a program; playback delay is used to slow down transmission for programs that cannot receive input at SuperKey's default speed of 1000 characters per second.

resident program: Application program, such as SuperKey or SideKick, that is loaded into your computer's main memory, and stays there and remains available for your use until the computer is turned off.

root directory: Main directory, which may contain subdirectories, in a DOS *tree-structured file directory*.

shift-state key. The ALT, CTRL, or left or right hand SHIFT key. One of these keys is held down while pressing any other key on the keyboard to create some SuperKey macros.

status line: Top line of the macro eDitor screen that provides information about the macro you are currently editing.

text mode encryption: Mode of *encryption* in which the encrypted file is a separate file consisting of all uppercase characters that can be sent and received by communications programs that don't handle binary files; the original (unscrambled) file remains untouched and not protected from unauthorized disk access. See *non-text mode encryption*.

toggle: To turn a feature or function ON and OFF, often using the same key.

transparent macro: A display macro that clears when the user presses any key.

tree-structured directory: The way DOS organizes files. Some files are contained in *root directories*, which may branch to subdirectories that contain other files.

type-ahead buffer: SuperKey feature that expands your computer's normal 16-character *keyboard buffer* to 128 characters. This allows you to type up to 128 characters, even while the computer is reading or writing to disk.

variable field: Set of characters included in a macro that define a space into which text or numbers can be entered during macro playback. The variable field definition does not specify the length of the field; it will be as long as the entry. See *fixed field*, *entry mask*, and *format mask*.

Index

A

Arrow key defaults, 64
Arrow keys option, 66
AUTOEXEC.BAT file, 181, 195
Auto-start macros, 21, 43–44, 68

B

Batch files, 181
Beep function, 69
Begin macro, 13, 71
Bottom line option, 31, 73

C

Clear macros, 19, 74
Clear screen function, 76
Colors, 186
Command line parameters, 77
Command stack, *see* DOS
 command stack
Commands, one-key, 35–36
Cut and Paste feature, 38–39, 70, 84, 86

D

Date function, 87
Decimal point default, 88
Defaults menu, 90, 162
Delay function, 92
Delimiter default, 94
DES encryption, 104
Disk wait option, 31, 96
Display macros, 21, 44–46, 98
 auto-starting, 49

 designing help systems with, 44
 freezing keyboard with, 47
 page breaks in, 46–47
 text color and intensity, 48–49
 titles of, 49
 transparent, 48
Distribution disks, 3, 6–7
DOS, 191–197
 command line, used to pass
 commands, 77
 command stack, 26, 80, 82
 commands, using macros for, 15–16
 directories, and SuperKey, 9, 181, 193

E

Edit macros, *see* Macro editor
Encryption, 24–26, 103–110
 directory, 27–28, 111
End macro, 113
Entry mask, in fields, 54, 114–118

F

Fields, 49–55
 entry control, 54–55, 114–118
 fixed-length, 52–53, 125, 127
 formats, 53–54, 114–118, 119–122, 127
 variable-length, 50, 51–52, 175
File encryption, 24–26
File names, 123
Files on SuperKey disk, 6–7, 189
Fill character default, 124
Fixed field, *see* Fields
Format fields option, 127

Functions menu, 22–24, 128
 keyboard lock function, 23–24
 screen off function, 23
 time function, 22

G

Graphics, and SuperKey, 41–42

H

Help system, 12, 44

I

Installation 5, 8, 9, 183–187, 192
 and other resident programs, 8
 and tree-structured directories, 9

K

Keyboard buffer, 96, 131
Keyboard click option, 132
Keyboard layout, 59–61, 75
Keyboard lock function, 23–24, 47, 133
KEY.COM file, 6, 7
KEYDES.COM file, 6, 7
KEY.HLP file, 6
KEYINST.COM file, 6
Key position, changing, 59–61
Key symbols, 134
Keys for macros, 16
Killing SuperKey, 9, 138

L

.LAY files, 7
Layout editor, 59–61
LAYOUT.COM file, 6
LETTER.TXT file, 7
Load layout, 139
Load macro, 19, 140
Logged drive function, 142
Log-on message, 9
Lotus 1–2–3, 39

M

.MAC files, 7
Macro, 13–21, 143
 autO-start, 21, 43, 49
 begin, 13
 customized, 21
 deleting, 19
 directory, 28, 144
 display, 21, 44–49

 colors, 48, 99
 transparent, 48
editor, 55–58, 101
 editing a macro, 57
 ending the edit, 58
 main menu, 57
 status line, 55–56, 101
 using text editor, 58
 writing a macro, 56
end, 113
fields, 49–55
files on disk, 189
key symbols, 134
loading from disk, 19, 182
memory, 18–19, 145, 187
merging, 47, 148
 nested, 55
 ready-made, 21
recording, 13, 130, 154
 saving on disk, 18–19, 163
 skipping, 17, 170
 stopping, 17, 171
 suggested keys, 16
 suspending, 172
 titles, 20, 49, 169, 174
Main menu, 11–12, 146
Menu system, 11–12, 159

N

Nested macro recording, 55

O

One finger mode option, 30, 150
One key commands, 35–36
Options menu, 29–33, 151
 bottom line, 31
 disk wait, 31
 one finger mode, 30
 playback delay, 31–32
 protect delay, 32–33
 save options, 33, 164
 suspend, 30

P

Path function, 153
PCjr installation, 6–7, 8
Playback delay option, 31–32, 155
 with spreadsheets, 41
Protect delay option, 32–33, 157

R

RAM requirements, 18–19, 145, 187
READ-ME.COM file, 7
Real time delay function, 161
Reflex, 180
Resident programs, 8, 186–187

S

Save

defaults, 162
options, 33, 164
macros on disk, 18–19, 163
setup, 165
Screen off function, 23, 166
Screen protect option, 32–33, 157
Setup menu, 27–29, 167
encryption directory, 27–28
macro directory, 28
save setup, 28–29, 165
Show titles window, 20, 169
SideKick, 8, 58, 177
Skip key, 17, 170
Spreadsheets, using with SuperKey,
39–42
graphics, 41–42
playback delay, 41
1-2-3 macros, 41
Stop, 171
Suspend option, 30, 172
deactivating SuperKey, 30
System requirements, 3

T

Text editor, used to create macros, 58
Time function, 22, 173
Titles, macro, 20, 169, 174
Transparent display macros, 48
Turbo Lightning, 8, 179
Tutorial, 11–33
Type-ahead buffer, 26

V

Variable field, 50, 51–52, 175

W

Word processors, using with SuperKey,
37–39
WordStar, 38–39

Borland Software



4585 Scotts Valley Drive, Scotts Valley, CA 95066

Available at better dealers nationwide.
To order by credit card, call (800) 255-8008; CA (800) 742-1133;
CANADA (800) 237-1136.

QUATTRO™

THE PROFESSIONAL SPREADSHEET

Borland's super graphic new generation spreadsheet: Twice the power at half the price! Ten types of presentation-quality graphs. Compatible with 1-2-3®, dBASE®, Paradox® and other spreadsheets and databases.

Quattro, Borland's new generation professional spreadsheet, proves there are better and faster ways to get your work done—whether it's graphics, recalculations, macros, or search and sort.

Presentation-quality graphics

Quattro has excellent built-in graphics capabilities that help you create a wide variety of graphs. Bar graphs, line graphs, pie charts, XY graphs, area charts—you can create up to 10 types of graphs, and print them directly from the spreadsheet or store them for future use.

Smarter recalculation

When a formula needs to be recalculated, Quattro uses "intelligent recal" to recalculate only those formulas whose elements have changed. This makes Quattro smarter and faster than other spreadsheets.

Greater macro capability

You can create macros instantly by *recording* your actions and storing them in the spreadsheet. The number of macros is limited only by memory. A built-in macro debugging environment makes it easy to find and correct problem areas. Quattro also includes a set of over 40 macro commands which make up a programming language.

***Suggested retail price \$195.00
(not copy protected)***

Direct compatibility

Quattro can directly load and use data files created with other spreadsheet and database programs like 1-2-3, dBASE, and Paradox. Quattro can read and even write WKS, WK1, and WKE files. You can also import ASCII and other text files into the spreadsheet.

Easy installation

Quattro can detect most computers and screen types, so it's always ready to load and run!

Plus, like all other Borland products, Quattro is *not copy protected!*

Technical Features

- Understands your 1-2-3 macros
- 100 built-in financial and statistical functions
- Menu Builder add-in for customizing menus
- Supports 8087/80287 math coprocessors
- Supports EGA, CGA, and VGA graphics adapters
- Pop-up menus
- Shortcuts to menu commands
- Context-sensitive online help
- Three types of choice lists: @functions and syntax, macro commands, and existing block names
- Pointing lets you specify a block of cells using arrow keys
- Search (or Query) lets you find specific records or cells
- Lets you arrange/rearrange data in alphabetical, numerical, or chronological order
- Supports Expanded Memory Specification to create spreadsheets larger than 640K
- Supports PostScript™ printers and typesetters

Minimum system requirements: For the IBM PS/2™ and the IBM® and Compaq® families of personal computers and all 100% compatibles. PC-DOS (MS-DOS®) 2.0 or later. Two floppies or a hard disk. 384K.

Quattro and Paradox are trademarks of Borland International, Inc. Lotus and 1-2-3 are registered trademarks of Lotus Development Corp. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright ©1987 Borland International, Inc. BOR 0414

SIDEKICK® THE DESKTOP ORGANIZER

Whether you're running WordStar®, Lotus®, dBASE®, or any other program, SideKick puts all these desktop accessories at your fingertips—Instantly!

A full-screen WordStar-like Editor to jot down notes and edit files up to 25 pages long.

A Phone Directory for names, addresses, and telephone numbers. Finding a name or a number is a snap.

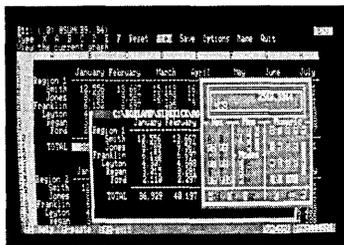
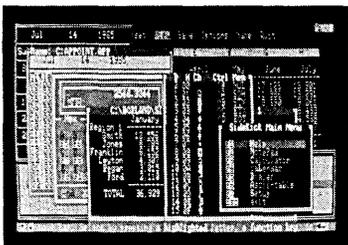
An Autodialer for all your phone calls. It will look up and dial telephone numbers for you. (A modem is required to use this function.)

A Monthly Calendar from 1901 through 2099.

Appointment Calendar to remind you of important meetings and appointments.

A full-featured Calculator ideal for business use. It also performs decimal to hexadecimal to binary conversions.

An ASCII Table for easy reference.



All the SideKick windows stacked up over Lotus 1-2-3.* From bottom to top: SideKick's "Menu Window," ASCII Table, Notepad, Calculator, Appointment Calendar, Monthly Calendar, and Phone Dialer.

Here's SideKick running over Lotus 1-2-3. In the SideKick Notepad you'll notice data that's been imported directly from the Lotus screen. In the upper right you can see the Calculator.

The Critics' Choice

"In a simple, beautiful implementation of WordStar's block copy commands, SideKick can transport all or any part of the display screen (even an area overlaid by the notepad display) to the notepad."

—Charles Petzold, *PC MAGAZINE*

"SideKick deserves a place in every PC."

—Gary Ray, *PC WEEK*

"SideKick is by far the best we've seen. It is also the least expensive."

—Ron Mansfield, *ENTREPRENEUR*

"If you use a PC, get SideKick. You'll soon become dependent on it."

—Jerry Pournelle, *BYTE*

Suggested Retail Price: \$84.95 (not copy protected)

Minimum system configuration: IBM PC, XT, AT, PCjr and true compatibles. PC-DOS (MS-DOS) 2.0 or greater. 128K RAM. One disk drive. A Hayes-compatible modem, IBM PCjr internal modem, or AT&T Modem 4000 is required for the autodialer function.



SideKick is a registered trademark of Borland International, Inc. dBASE is a registered trademark of Ashton-Tate. IBM, XT, AT, and PCjr are registered trademarks of International Business Machines Corp. AT&T is a registered trademark of American Telephone & Telegraph Company. Lotus and 1-2-3 are registered trademarks of Lotus Development Corp. WordStar is a registered trademark of MicroPro International Corp. Hayes is a trademark of Hayes Microcomputer Products, Inc. Copyright 1987 Borland International

BOR0060C

SUPERKEY[®] THE PRODUCTIVITY BOOSTER

RAM-resident
Increased productivity for IBM[®]PCs or compatibles

SuperKey's simple macros are electronic shortcuts to success. By letting you reduce a lengthy paragraph into a single keystroke of your choice, SuperKey eliminates repetition.

SuperKey turns 1,000 keystrokes into 1!

SuperKey can record lengthy keystroke sequences and play them back at the touch of a single key. Instantly. Like magic.

In fact, with SuperKey's simple macros, you can turn "Dear Customer: Thank you for your inquiry. We are pleased to let you know that shipment will be made within 24 hours. Sincerely," into the one keystroke of your choice!

SuperKey keeps your confidential files—confidential!

Without encryption, your files are open secrets. Anyone can walk up to your PC and read your confidential files (tax returns, business plans, customer lists, personal letters, etc.).

With SuperKey you can encrypt any file, *even* while running another program. As long as you keep the password secret, only *you* can decode your file correctly. SuperKey also implements the U.S. government Data Encryption Standard (DES).

- | | |
|---|---|
| <input checked="" type="checkbox"/> RAM resident—accepts new macro files even while running other programs | <input checked="" type="checkbox"/> Keyboard buffer increases 16 character keyboard "type-ahead" buffer to 128 characters |
| <input checked="" type="checkbox"/> Pull-down menus | <input checked="" type="checkbox"/> Real-time delay causes macro playback to pause for specified interval |
| <input checked="" type="checkbox"/> Superfast file encryption | <input checked="" type="checkbox"/> Transparent display macros allow creation of menus on top of application programs |
| <input checked="" type="checkbox"/> Choice of two encryption schemes | <input checked="" type="checkbox"/> Data entry and format control using "fixed" or "variable" fields |
| <input checked="" type="checkbox"/> On-line context-sensitive help | <input checked="" type="checkbox"/> Command stack recalls last 256 characters entered |
| <input checked="" type="checkbox"/> One-finger mode reduces key commands to single keystroke | |
| <input checked="" type="checkbox"/> Screen OFF/ON blanks out and restores screen to protect against "burn in" | |
| <input checked="" type="checkbox"/> Partial or complete reorganization of keyboard | |

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system configuration: IBM PC, XT, AT, PCjr, and true compatibles. PC-DOS (MS-DOS) 2.0 or greater. 128K RAM. One disk drive.



SuperKey is a registered trademark of Borland International, Inc. IBM, XT, AT, and PCjr are registered trademarks of International Business Machines Corp. MS-DOS is a registered trademark of Microsoft Corp. BOR 0062C

L I G H T N I N G
WORDWIZARD™

Lightning Word Wizard includes complete, commented Turbo Pascal® source code and all the technical information you'll need to understand and work with Turbo Lightning's "engine."

More than 20 fully documented Turbo Pascal procedures reveal powerful Turbo Lightning engine calls. Harness the full power of the complete and authoritative Random House® Concise Word List and Random House Thesaurus.

Turbo Lightning's "Reference Manual"

Developers can use the versatile on-line examples to harness Turbo Lightning's power to do rapid word searches. Lightning Word Wizard is the forerunner of the database access systems that will incorporate and engineer the Turbo Lightning Library® of electronic reference works.

The ultimate collection of word games and crossword solvers!

The excitement, challenge, competition, and education of four games and three solver utilities—puzzles, scrambles, spell-searches, synonym-seekings, hidden words, crossword solutions, and more. You and your friends (up to four people total) can set the difficulty level and contest the high-speed smarts of Lightning Word Wizard!

Turbo Lightning—Critics' Choice

"Lightning's good enough to make programmers and users cheer, executives of other software companies weep."

Jim Seymour, *PC Week*

"The real future of Lightning clearly lies not with the spelling checker and thesaurus currently included, but with other uses of its powerful look-up engine."

Ted Silveira, *Profiles*

"This newest product from Borland has it all."

Don Roy, *Computing Now!*

Minimum system configuration: IBM PC, XT, AT, PCjr, Portable, and true compatibles. 256K RAM minimum. PC-DOS (MS-DOS) 2.0 or greater. Turbo Lightning software required. Optional—Turbo Pascal 3.0 or greater to edit and compile Turbo Pascal source code.



***Suggested Retail Price: \$69.95
(not copy protected)***

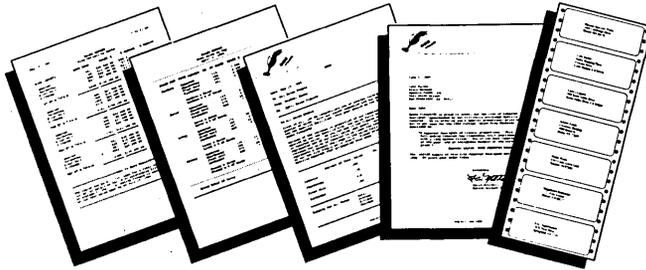
REFLEX[®] THE DATABASE MANAGER

***The high-performance database manager
that's so advanced it's easy to use!***

Lets you organize, analyze and report information faster than ever before! If you manage mailing lists, customer files, or even your company's budgets—Reflex is the database manager for you!

Reflex is the acclaimed, high-performance database manager you've been waiting for. Reflex extends database management with business graphics. Because a picture is often worth a 1000 words, Reflex lets you extract critical information buried in mountains of data. With Reflex, when you look, you see.

The **REPORT VIEW** allows you to generate everything from mailing labels to sophisticated reports. You can use database files created with Reflex or transferred from Lotus 1-2-3,[®] dBASE,[®] PFS: File,[®] and other applications.



Reflex: The Critics' Choice

"... if you use a PC, you should know about Reflex ... may be the best bargain in software today."

Jerry Pournelle, BYTE

"Everyone agrees that Reflex is the best-looking database they've ever seen."

Adam B. Green, InfoWorld

"The next generation of software has officially arrived."

Peter Norton, PC Week

Reflex: don't use your PC without it!

Join hundreds of thousands of enthusiastic Reflex users and experience the power and ease of use of Borland's award-winning Reflex.

Suggested Retail Price: \$149.95 (not copy protected)

Minimum system configuration: IBM PC, XT, AT, and true compatibles. 384K RAM minimum. IBM Color Graphics Adapter, Hercules Monochrome Graphics CArd, or equivalent. PC-DOS (MS-DOS) 2.0 or greater. Hard disk and mouse optional. Lotus 1-2-3, dBASE, or PFS: File optional.



Reflex is a trademark of Borland/Analytica Inc. Lotus 1-2-3 is a registered trademark of Lotus Development Corporation. dBASE is a registered trademark of Ashton-Tate. PFS: File is a registered trademark of Software Publishing Corporation. IBM, XT, AT, and IBM Color Graphics Adapter are registered trademarks of International Business Machines Corporation. Hercules Graphics Card is a trademark of Hercules Computer Technology. MS-DOS is a registered trademark of Microsoft Corp. Copyright 1987 Borland International BOR 0066C

REFLEX: THE WORKSHOP™

Includes 22 "instant templates" covering a broad range of business applications (listed below). Also shows you how to customize databases, graphs, crosstabs, and reports. It's an invaluable analytical tool and an important addition to another one of our best sellers, Reflex: The Database Manager.

Fast-start tutorial examples:

Learn Reflex® as you work with practical business applications. The Reflex Workshop Disk supplies databases and reports large enough to illustrate the power and variety of Reflex features. Instructions in each Reflex Workshop chapter take you through a step-by-step analysis of sample data. You then follow simple steps to adapt the files to your own needs.

22 practical business applications:

Workshop's 22 "instant templates" give you a wide range of analytical tools:

Administration

- Scheduling Appointments
- Planning Conference Facilities
- Managing a Project
- Creating a Mailing System
- Managing Employment Applications

Sales and Marketing

- Researching Store Check Inventory
- Tracking Sales Leads
- Summarizing Sales Trends
- Analyzing Trends

Production and Operations

- Summarizing Repair Turnaround

- Tracking Manufacturing Quality Assurance
- Analyzing Product Costs

Accounting and Financial Planning

- Tracking Petty Cash
- Entering Purchase Orders
- Organizing Outgoing Purchase Orders
- Analyzing Accounts Receivable
- Maintaining Letters of Credit
- Reporting Business Expenses
- Managing Debits and Credits
- Examining Leased Inventory Trends
- Tracking Fixed Assets
- Planning Commercial Real Estate Investment

Whether you're a newcomer learning Reflex basics or an experienced "power user" looking for tips, Reflex: The Workshop will help you quickly become an expert database analyst.

Minimum system configuration: IBM PC, AT, and XT, and true compatibles. PC-DOS (MS-DOS) 2.0 or greater. 384K RAM minimum. Requires Reflex: The Database Manager, and IBM Color Graphics Adapter, Hercules Monochrome Graphics Card or equivalent.



***Suggested Retail Price: \$69.95
(not copy protected)***

TURBO PASCAL[®]

VERSION 4.0

Turbo Pascal 4.0 has all the features

Turbo Pascal 4.0 has all the features of Turbo Pascal 3.0, *plus* an amazing compilation speed of 27,000 lines per minute,* support for programs larger than 64K, a library of powerful standard units, separate compilation, and much more.

The single-pass, native code compiler offers improved code generation, smart linking to remove unused code from your programs, built-in project management, separate compilation using units, output screen saved in a window, MAP files for use with standard debuggers, a command-line version of the compiler and MAKE utility, and built-in support for 8087/80287/80387 math coprocessors.

All these advanced features, plus the integrated programming environment, online help, and Borland's famous pull-down menus, make Turbo Pascal 4.0 the high-speed, high-performance development tool every programmer hopes for.

Built-in editor

An improved, full-screen editor for editing, compiling, and finding and correcting errors from inside the integrated development environment. Supports 25, 43, and 50 lines per screen, tabs, colors, and new command installation.

Interactive error detection

The compiler instantly locates errors, automatically activates the editor, and shows you the location of the error in the source code.

Pick list

Lets you pick a file from a list of the last eight files loaded into the editor and opens it at the exact spot where you last edited the file. It even remembers your last search string and search options.

Free MicroCalc spreadsheet

A new and improved version of the full-fledged spreadsheet included on your Turbo Pascal disk, absolutely free! You get the complete, revised source code, ready to compile and run.

Compatibility with Turbo Pascal 3.0

A conversion program and compatibility units help you convert all your 3.0 programs to 4.0.

Other Technical Features:

- Several powerful standard units (*System Dos*, *Crt*, and *Graph*)
- Device-independent graphics support for CGA, MCGA, EGA, VGA, Hercules, AT&T 6300, and IBM 3270 PC
- Extended data types, including *LongInt*
- Optional range- and stack-checking; short-circuit Boolean expression evaluation
- Support for inline statements, inline macros, and powerful assembly language interface
- Faster software-only floating point; toggle switch for 80x87 support including *Single*, *Double*, *Extended*, and *Comp* IEEE reals (with numeric coprocessor)
- Automatic execution of initialization and exit code for each unit
- Nested include files up to 8 levels deep, including main module and units
- Operating system calls and interrupts
- Interrupt procedure support for ISRs
- Variable and value typecasting
- Shell to DOS transfer

Minimum system requirements: For the IBM PS/2[™] and the IBM[®] and Compaq[®] families of personal computers and all 100% compatibles. Integrated environment: 384K; command line: 256K; one floppy drive.

*Run on an 8MHz IBM AT

***Suggested retail price \$99.95
(not copy protected)***

TURBO PASCAL[®]

TOOLBOXES AND DEVELOPER'S LIBRARY

An unsurpassed collection of TURBO PASCAL TOOLS that make you the expert, now upgraded to Version 4.0!

Turbo Pascal Tutor:

For both the novice programmer and the professional. Everything you need to write a simple program or handle advanced concepts like using assembly language routines with your Turbo Pascal programs. The programmer's guide covers the fine points of Turbo Pascal programming with lots of examples; and on accompanying disk gives you all the source code. A real education for just **\$69.95!**

Turbo Pascal Editor Toolbox:

Everything you need to build your own custom text editor or word processor including easy-to-install modules, source code and plenty of know-how. Includes all the popular features like word-wrap, auto indent, find/replace. Just **\$99.95!**

Turbo Pascal Database Toolbox:

A complete library of Pascal procedures that let you sort and search your data and build powerful applications. Includes Turbo Access files that use B+ trees to organize and search your data, and Turbo Sort to sort it. GINST even gets your programs up and running on other terminals! Includes a free database that you can use as is or modify to suit your needs. Just **\$99.95!**

Turbo Pascal Graphics Toolbox:

Gives you all the high-resolution graphics and graphic window management capabilities you need, with tools to draw and hatch pie charts, bar charts, circles, rectangles and a full range of geometric shapes. Save and restore graphic images to and from disk, plot precise curves, and create animation.* All for just **\$99.95!**

Turbo Pascal GameWorks:

Secrets and strategies of the masters with easy-to-understand examples that teach you how to quickly create your own computer games using Turbo Pascal. For instant excitement, play the three great computer games included on disk—Turbo Chess, Turbo Bridge and Turbo Go-Moku. They're all compiled and ready to run. Just **\$99.95!**

Turbo Pascal Numerical Methods Toolbox:

All the state-of-the-art applied mathematical tools you'll ever need. A collection of Turbo Pascal mathematical routines and programs and ten independent modules that you can easily adapt to different programs. Gives you the kind of mathematical routines IMSL[®] and NAG libraries provide for FORTRAN. Complete with sample programs and source code for each module. All for just **\$99.95!**

Buy them separately or get The Developer's Library, which includes all six, for just \$395 suggested retail price! Not copy protected!

System Requirements: For the IBM PS/2[™] and the IBM[®] and Compaq[®] families of personal computers and all 100% compatibles. Operating System: PC-DOS (MS-DOS) 2.0 or later.

*Turbo Pascal Graphics Toolbox also requires one of the following graphics adapters: CGA, EGA, Hercules, or IBM 3270.

All Borland products are trademarks or registered trademarks of Borland International, Inc. Borland Turbo Toolbox[®] products. Other brand and product name are trademarks or registered trademarks of their respective holders. Copyright ©1987 Borland International, Inc.

BOR 0486

TURBO PROLOG™

the natural language of Artificial Intelligence

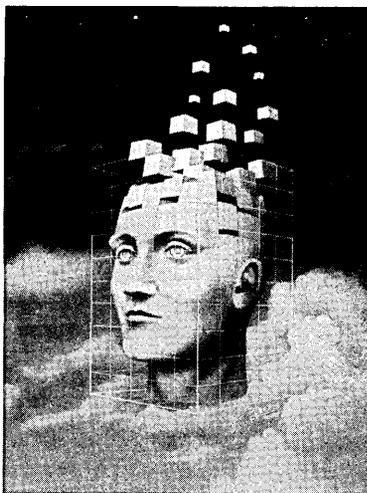
Turbo Prolog brings fifth-generation supercomputer power to your IBM®PC!

STEP-BY-STEP
TUTORIAL AND DEMO PROGRAMS
WITH SOURCE CODE INCLUDED!

Turbo Prolog takes programming into a new, natural, and logical environment

With Turbo Prolog, because of its natural, logical approach, both people new to programming and professional programmers can build powerful applications such as expert systems, customized knowledge bases, natural language interfaces, and smart information management systems.

Turbo Prolog is a *declarative* language which uses deductive reasoning to solve programming problems.



Turbo Prolog provides a fully integrated programming environment like Borland's Turbo Pascal,® the *de facto* worldwide standard.

You get the complete Turbo Prolog programming system

You get the 200-page manual you're holding, software that includes the lightning-fast Turbo Prolog six-pass

compiler and interactive editor, and the free GeoBase natural query language database, which includes commented source code on disk, ready to compile. (GeoBase is a complete database designed and developed around U.S. geography. You can modify it or use it "as is.")

Turbo Prolog's development system includes:

- A complete Prolog compiler that is a variation of the Clocksin and Mellish Edinburgh standard Prolog.
- A full-screen interactive editor.
- Support for both graphic and text windows.
- All the tools that let you build your own expert systems and AI applications with unprecedented ease.

Minimum system configuration: IBM PC, XT, AT, Portable, 3270, PCjr, and true compatibles. PC-DOS (MS-DOS) 2.0 or later. 384K RAM minimum.

**Suggested Retail Price \$99.95
(not copy protected)**



BORLAND
INTERNATIONAL

Turbo Prolog is a trademark and Turbo Pascal is a registered trademark of Borland International, Inc. IBM AT, XT, and PCjr are registered trademarks of International Business Machines Corp. MS-DOS is a registered trademark of Microsoft Corp.

Copyright 1987 Borland International

BOR 0016D

TURBO PROLOG™ TOOLBOX

***Enhances Turbo Prolog with more than 80 tools
and over 8,000 lines of source code***

***Turbo Prolog, the natural language of Artificial Intelligence, is the
most popular AI package in the world with more than 100,000 users.
Our new Turbo Prolog Toolbox extends its possibilities.***

The Turbo Prolog Toolbox enhances Turbo Prolog—our 5th-generation computer programming language that brings supercomputer power to your IBM PC and compatibles—with its more than 80 tools and over 8,000 lines of source code that can be incorporated into your programs, quite easily.

Turbo Prolog Toolbox features include:

- Business graphics generation: boxes, circles, ellipses, bar charts, pie charts, scaled graphics
- Complete communications package: supports XModem protocol
- File transfers from Reflex,® dBASE III,® Lotus 1-2-3,® Symphony®
- A unique parser generator: construct your own compiler or query language
- Sophisticated user-interface design tools
- 40 example programs
- Easy-to-use screen editor: design your screen layout and I/O
- Calculated fields definition
- Over 8,000 lines of source code you can incorporate into your own programs

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system configuration: IBM PC, XT, AT or true compatibles. PC-DOS (MS-DOS) 2.0 or later. Requires Turbo Prolog 1.10 or higher. Dual-floppy disk drive or hard disk. 512K.



Turbo Prolog Toolbox and Turbo Prolog are trademarks of Borland International, Inc. Reflex is a registered trademark of Borland Analytica, Inc. dBASE III is a registered trademark of Ashton-Tate. Lotus 1-2-3 and Symphony are registered trademarks of Lotus Development Corp. IBM, XT, and AT are registered trademarks of International Business Machines Corp. MS-DOS is a registered trademark of Microsoft Corp.

BOR 0240

TURBO BASIC[®]

The high-speed BASIC you've been waiting for!

You probably know us for our Turbo Pascal[®] and Turbo Prolog[®]. Well, we've done it again! We've created Turbo Basic, because BASIC doesn't have to be slow.

If BASIC taught you how to walk, Turbo Basic will teach you how to run!

With Turbo Basic, your only speed is "Full Speed Ahead"! Turbo Basic is a complete development environment with an *amazingly fast compiler*, an *interactive editor* and a *trace debugging system*. And because Turbo Basic is also compatible with BASICA, chances are that you already know how to use Turbo Basic.

Turbo Basic ends the basic confusion

There's now one standard: Turbo Basic. And because Turbo Basic is a Borland product, the price is right, the quality is there, and the power is at your fingertips. Turbo Basic is part of the fast-growing Borland family of programming languages we call the "Turbo Family." And hundreds of thousands of users are already using Borland's languages. So, welcome to a whole new generation of smart PC users!

Free spreadsheet included with source code!

Yes, we've included MicroCalc,[™] our sample spreadsheet, complete with source code. So you can get started right away with a "real program." You can compile and run it "as is," or modify it.

A technical look at Turbo Basic

- | | |
|---|---|
| <input checked="" type="checkbox"/> Full recursion supported | executable program, with separate windows for editing, messages, tracing, and execution |
| <input checked="" type="checkbox"/> Standard IEEE floating-point format | |
| <input checked="" type="checkbox"/> Floating-point support, with full 8087 coprocessor integration. Software emulation if no 8087 present | <input checked="" type="checkbox"/> Compile and run-time errors place you in source code where error occurred |
| <input checked="" type="checkbox"/> Program size limited only by available memory (no 64K limitation) | <input checked="" type="checkbox"/> Access to local, static and global variables |
| <input checked="" type="checkbox"/> EGA, CGA, MCGA and VGA support | <input checked="" type="checkbox"/> New long integer (32-bit) data type |
| <input checked="" type="checkbox"/> Full integration of the compiler, editor, and | <input checked="" type="checkbox"/> Full 80-bit precision |
| | <input checked="" type="checkbox"/> Pull-down menus |
| | <input checked="" type="checkbox"/> Full window management |

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system configuration: IBM PC, AT, XT, PS/2 or true compatibles. 320K. One floppy drive. PC-DOS (MS-DOS) 2.0 or later.



Turbo Basic, Turbo Prolog and Turbo Pascal are registered trademarks and MicroCalc is a trademark of Borland International, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.
Copyright 1987 Borland International

BOR 0265B

TURBO BASIC® DATABASE TOOLBOX™

With the Turbo Basic Database Toolbox you can build your own powerful, professional-quality database programs. And like all other Borland Toolboxes, it's advanced enough for professional programmers yet easy enough for beginners.

Three ready-to-use modules

The Toolbox enhances your programming with three problem-solving modules:

Turbo Access quickly locates, inserts, or deletes records in a database using B+ trees—the fastest method for finding and retrieving database information. (Source code is included.)

Turbo Sort uses the *Quicksort* method to sort data on single items or on multiple keys. Features virtual memory management for sorting large data files. (Commented source code is on disk.)

TRAINER is a demonstration program that graphically displays how B+ trees work. You can key in sample records and see a visual index of B+ trees being built.

Free sample database

Included is a free sample database with source code. Just compile it, and it's ready to go to work for you—you can use it as is or customize it. You can search the database by keywords or numbers, update records, or add and delete them, as needed.

Saves you time and money

If you're a professional programmer writing software for databases or other applications where search-and-sort capabilities are important, we can save you time and money. Instead of writing the same tedious but essential routines over and over again, you can simply include any of the Toolbox's modules in your own compiled programs.

Technical Features

- | | |
|--|---|
| <input checked="" type="checkbox"/> Maximum number of files open: 15 files, or 7 data sets | <input checked="" type="checkbox"/> Maximum number of records: +2 billion |
| <input checked="" type="checkbox"/> Maximum file size: 32 Mb | <input checked="" type="checkbox"/> Maximum field size: 32K |
| <input checked="" type="checkbox"/> Maximum record size: 32K | <input checked="" type="checkbox"/> Maximum key size: 128 bytes |
| | <input checked="" type="checkbox"/> Maximum number of keys: +2 billion |

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system requirements: For the IBM PS/2 and the IBM® and Compaq® families of personal computers and all 100% compatibles, running Turbo Basic 1.0. PC-DOS (MS-DOS®) 2.0 or later. Memory: 640K.



All Borland products are registered trademarks or trademarks of Borland International, Inc. or Borland/Analytica, Inc. A Borland *Turbo Toolbox* product. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright 1987 Borland International.

BOR 0384A

TURBO C[®]

Includes free
MicroCalc spreadsheet
with source code

A complete interactive development environment

With Turbo C, you can expect what only Borland delivers: Quality, Speed, Power and Price. And with its compilation speed of more than 7000 lines a minute, Turbo C makes everything else look like an exercise in slow motion.

Turbo C: The C compiler for both amateurs and professionals

If you're just beginning and you've "kinda wanted to learn C," now's your chance to do it the easy way. Turbo C's got everything to get you going. If you're already programming in C, switching to Turbo C will considerably increase your productivity and help make your programs both smaller and faster.

Turbo C: a complete interactive development environment

Like Turbo Pascal[®] and Turbo Prolog,[™] Turbo C comes with an interactive editor that will show you syntax errors right in your source code. Developing, debugging, and running a Turbo C program is a snap!

Technical Specifications

- Compiler:** One-pass compiler generating native in-line code, linkable object modules and assembler. The object module format is compatible with the PC-DOS linker. Supports small, medium, compact, large, and huge memory model libraries. Can mix models with near and far pointers. Includes floating point emulator (utilizes 8087/80287 if installed).
- Interactive Editor:** The system includes a powerful, interactive full-screen text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code.
- Development Environment:** A powerful "Make" is included so that managing Turbo C program development is easy. Borland's fast "Turbo Linker" is also included. Also includes pull-down menus and windows. Can run from the environment or generate an executable file.
- Links with relocatable object modules created using Borland's Turbo Prolog into a single program.
- ANSI C compatible.
- Start-up routine source code included.
- Both command line and integrated environment versions included.

"Steve" benchmark (25 iterations)

	Turbo C	Microsoft[®] C	Lattice C
Compile time	3.89	16.37	13.90
Compile and link time	9.94	29.06	27.79
Execution time	5.77	9.51	13.79
Object code size	274	297	301
Price	\$99.95	\$450.00	\$500.00

Benchmark run on a 6 Mhz IBM AT using Turbo C version 1.0 and the Turbo Linker version 1.0; Microsoft C version 4.0 and the MS overlay linker version 3.51; Lattice C version 3.1 and the MS object linker version 3.05.

Suggested Retail Price: \$99.95* (not copy protected) *Introductory offer good through July 1, 1987.

Minimum system configuration: IBM PC, XT, AT and true compatibles. PC-DOS (MS-DOS) 2.0 or later. One floppy drive. 320K.



Turbo C and Turbo Pascal are registered trademarks and Turbo Prolog is a trademark of Borland International, Inc. Microsoft C and MS-DOS are registered trademarks of Microsoft Corp. Lattice C is a registered trademark of Lattice, Inc. IBM, XT, and AT are registered trademarks of International Business Machines Corp.

BOR 0243

EUREKA: THE SOLVER™

The solution to your most complex equations—in seconds!

If you're a scientist, engineer, financial analyst, student, teacher, or any other professional working with equations, Eureka: The Solver can do your Algebra, Trigonometry and Calculus problems in a snap.

Eureka also handles maximization and minimization problems, plots functions, generates reports, and saves an incredible amount of time. Even if you're not a computer specialist, Eureka can help you solve your real-world mathematical problems fast, without having to learn numerical approximation techniques. Using Borland's famous pull-down menu design and context-sensitive help screens, Eureka is easy to learn and easy to use—as simple as a hand-held calculator.

X + exp(X) = 10 solved instantly instead of eventually!

Imagine you have to "solve for X," where $X + \exp(X) = 10$, and you don't have Eureka: The Solver. What you do have is a problem, because it's going to take a lot of time guessing at "X." With Eureka, there's no guessing, no dancing in the dark—you get the right answer, right now. (PS: $X = 2.0705799$, and Eureka solved that one in .4 of a second!)

How to use Eureka: The Solver

It's easy.

1. Enter your equation into the full-screen editor
2. Select the "Solve" command
3. Look at the answer
4. You're done

You can then tell Eureka to

- Evaluate your solution
- Plot a graph
- Generate a report, then send the output to your printer, disk file or screen
- Or all of the above

Some of Eureka's key features

You can key in:

- A formula or formulas
- A series of equations—and solve for all variables
- Constraints (like X has to be $<$ or $=$ 2)
- A function to plot
- Unit conversions
- Maximization and minimization problems
- Interest Rate/Present Value calculations
- Variables we call "What happens?," like "What happens if I change this variable to 21 and that variable to 27?"

Eureka: The Solver includes

- A full-screen editor
- Pull-down menus
- Context-sensitive Help
- On-screen calculator
- Automatic 8087 math co-processor chip support
- Powerful financial functions
- Built-in and user-defined math and financial functions
- Ability to generate reports complete with plots and lists
- Polynomial finder
- Inequality solutions

Minimum system configuration: IBM PC, AT, XT, PS/2, Portable, 3270 and true compatibles. PC-DOS (MS-DOS) 2.0 and later. 384K.

Suggested Retail Price: \$167.00
(not copy protected)



Eureka: The Solver is a trademark of Borland International, Inc. IBM, AT, and XT are registered trademarks of International Business Machines Corp. MS-DOS is a registered trademark of Microsoft Corp. Copyright 1987 Borland International BOR 0221B

SIDEKICK[®] THE DESKTOP ORGANIZER

Release 2.0
Macintosh™

The most complete and comprehensive collection of desk accessories available for your Macintosh!

Thousands of users already know that SideKick is the best collection of desk accessories available for the Macintosh. With our new Release 2.0, the best just got better.

We've just added two powerful high-performance tools to SideKick—Outlook™: The Outliner and MacPlan™: The Spreadsheet. They work in perfect harmony with each other and *while* you run other programs!

Outlook: The Outliner

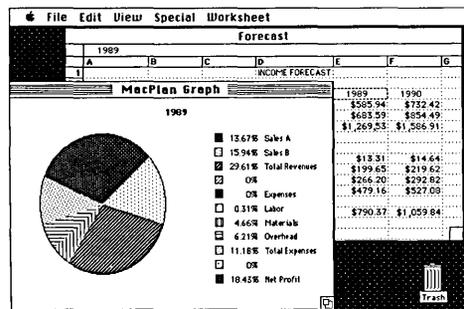
- It's the desk accessory with more power than a stand-alone outliner
- A great desktop publishing tool, Outlook lets you incorporate both text and graphics into your outlines
- Works hand-in-hand with MacPlan
- Allows you to work on several outlines at the same time

MacPlan: The Spreadsheet

- Integrates spreadsheets and graphs
- Does both formulas and straight numbers
- Graph types include bar charts, stacked bar charts, pie charts and line graphs
- Includes 12 example templates free!
- Pastes graphics and data right into Outlook creating professional memos and reports, complete with headers and footers.

SideKick: The Desktop Organizer, Release 2.0 now includes

- Outlook: The Outliner
- MacPlan: The Spreadsheet
- Mini word processor
- Calendar
- PhoneLog
- Analog clock
- Alarm system
- Calculator
- Report generator
- Telecommunications (new version now supports XModem file transfer protocol)



MacPlan does both spreadsheets and business graphs. Paste them into your Outlook files and generate professional reports.

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system configurations: Macintosh 512K or Macintosh Plus with one disk drive. One 800K or two 400K drives are recommended. With one 400K drive, a limited number of desk accessories will be installable per disk.



SideKick is a registered trademark and Outlook and MacPlan are trademarks of Borland International, Inc. Macintosh is a trademark of McIntosh Laboratory, Inc. licensed to Apple Computer, Inc. Copyright 1987 Borland International BOR 0069D

REFLEX[®] PLUS: THE DATABASE MANAGER Macintosh™

All the Power & Flexibility of a Relational Database Made Easy!
Reflex Plus: The Database Manager is the first relational database that's easy to learn, powerful, and aimed at your needs. Reflex Plus is not a mere file organizer, nor is it a monstrosity complicated behemoth aimed solely at consultants. Reflex Plus is the only relational database aimed at your needs and time constraints.

Reflex Plus accomplishes this by taking full advantage of the Macintosh's superior graphic ability while still giving users what they want: unlimited flexibility in creating databases, accessing data, and producing reports.

What puts the plus into Reflex Plus?

Borland listens to its customers and has added the most-asked-for features and improvements to Reflex Plus.

High-powered features of Reflex Plus:

- Multiple entry forms for the same database.
- Entry for more than one database in a single entry form.
- Your choice of having an entry form that shows one record at a time, or one that shows all the records at once.
- Calculated fields in entry forms.
- Display-only fields.
- Default (but editable) fields.
- New functions like GROUPBY, which lets you easily show records grouped by values in common.
- A selection of useful templates.
- Larger record size. (You can now choose record sizes of 1000, 2000, or 4000 characters.)

Check out these Reflex Plus features:

- Visual database design.
- A "what you see is what you get" design capability both for entry forms and reports.
- Compatible with all Macintoshes with at least 512K, including the SE[™], and Macintosh II[™].

The heart of Reflex Plus is in its special functions with which you create formulas. With over 50 function words to choose from, you are given all the power of programming without struggling with complex syntax. Reflex Plus functions are straightforward and can handle all types of data.

Database specifications: Maximum single field length: 4072 bytes. Maximum fields per record: 254. Maximum record length: 4080 bytes. Maximum records per file: limited only by disk capacity. Maximum number of linked database files: 200. Maximum number of open windows: 15. Maximum number of files that can be used by a report: no limit.

Suggested Retail Price: \$279.00 (not copy protected)

Minimum system requirements: Runs on any Macintosh with at least 512K memory. Minimum setup is one 800K (double-sided) disk drive or two 400K (single-sided) drives. Works with the Hierarchical File System, Switcher, and most hard disks. Supports printing on the ImageWriter and the LaserWriter.

Armed with these functions, you create formulas that sort, search, calculate, quantify, qualify—you name it. And if you don't feel up to writing the formula yourself, Reflex Plus will do it for you. Using the FormulaBuild dialog box, you can master even the most complicated formula.

Display grouped data. Reflex Plus gives you unlimited flexibility when you want to display your data grouped in meaningful ways.

Flexible entry forms. Most databases have a data entry form, and that's that. Reflex Plus lets you design your own (but if you don't want to bother, Reflex Plus will make one for you). Here are just some of the options available in your entry forms:

- View all records at once.
- View one record at a time.
- Enter data into many databases at once.
- Use calculated fields.
- Default values in fields, display-only values, and lots more.

Convenience and Ease

- Preset entry forms.** Let Reflex Plus create an entry form for you.
- Preset reports.** Let Reflex Plus create a table-style report for you.
- Paste Formula command.** Let Reflex Plus guide you through the steps of creating formulas for power searching and data manipulation.
- On-line help facility.** Reflex Plus has an extensive on-screen, context-sensitive help feature.
- Paste Choice command.** This command lets you paste in fields that duplicate all the attributes of another field. A great time saver. The command also lets you build formulas by pointing and clicking.
- Auto-save.** You'll never lose data again with Reflex Plus's auto-save feature.

TURBO PASCAL[®]

MACINTOSH™

Borland's Macintosh version of Turbo Pascal is so incredibly fast that it can compile 1,420 lines of source code in the 7.1 seconds it took you to read this!

And reading the rest of this takes about 5 minutes, which is plenty of time for Turbo Pascal to compile at least 60,000 more lines of source code!

Turbo Pascal does both Windows and Units

The separate compilation of routines offered by Turbo Pascal creates modules called Units, which can be linked to any Turbo Pascal program. This modular pathway gives you pieces that can be integrated into larger programs. You can use memory more efficiently and reduce the time it takes to develop large programs.

Turbo Pascal is so compatible with MPW that they should be living together

You can compile and run routines from Macintosh Programmer's Workshop Pascal and Inside Macintosh with only the subtlest changes. Turbo Pascal is also compatible with the Hierarchical File System of the Macintosh.

The 27-second Guide to Turbo Pascal

- Compilation speed of more than 12,000 lines per minute
- Unit structure lets you create programs in modular form
- Multiple editing windows—up to 8 at a time
- Compilation options include compiling to disk or memory, or compile and run
- No need to switch between programs to compile or run a program
- Streamlined development and debugging
- Compatibility with Macintosh Programmer's Workshop Pascal (with minimal changes)
- Compatibility with Hierarchical File System of your Macintosh
- Ability to define default volume and folder names used in compiler directives
- Search and change features in the editor speed up and simplify alteration of routines
- Ability to use all available Macintosh memory without limit
- Units included to call all the routines provided by Macintosh Toolbox

*Suggested Retail Price: \$99.95
(not copy protected)*

Minimum system configuration: Macintosh 512K or Macintosh Plus with one disk drive.

All Borland products are trademarks or registered trademarks of Borland International, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright ©1987 Borland International, Inc. BOR 0167B

TURBO PASCAL[®] TUTOR

From the folks who created Turbo Pascal. Borland's new Turbo Pascal Tutor is everything you need to start programming in Turbo Pascal on the Macintosh!™ It takes you from the bare basics to advanced programming in a simple, easy-to-understand fashion.

No gimmicks. It's all here.

The manual, the Tutor application, and 30 sample programs provide a step-by-step tutorial in three phases: programming in Pascal, programming on the Macintosh, and programming in Turbo Pascal on the Macintosh. Here's how the manual is set up:

Turbo Pascal for the Absolute Novice delivers the basics—a concise history of Pascal, key terminology, your first program.

A Programmer's Guide to Turbo Pascal covers Pascal specifics—program structure, procedures and functions, arrays, strings, and so on. We've also included Turbo Typist, a textbook sample program.

Advanced Programming takes you a step higher into stacks, queues, binary trees, linked structures, writing large programs, and more.

Using the Power of the Macintosh discusses the revolutionary hardware and software features of this machine. It introduces the 600-plus utility routines in the Apple Toolbox.

Programming the Macintosh in Turbo Pascal shows you how to create true Macintosh programs that use graphics, pull-down menus, dialog boxes, and so on. Finally, MacTypist, a complete stand-alone application featuring animated graphics, builds on Turbo Typist and demonstrates what you can do with all the knowledge you've just acquired.

The disk contains the source code for all the sample programs, including Turbo Typist, MacTypist, and Turbo Tutor. The Tutor's split screen lets you run a procedure and view its source code simultaneously. After running it, you can take a test on the procedure. If you're stuck for an answer, a Hint option steers you in the right direction.

Macintosh topics included are

- | | |
|--|--|
| <input checked="" type="checkbox"/> memory management | <input checked="" type="checkbox"/> menus |
| <input checked="" type="checkbox"/> resources and resource files | <input checked="" type="checkbox"/> desk accessory support |
| <input checked="" type="checkbox"/> QuickDraw | <input checked="" type="checkbox"/> dialogs |
| <input checked="" type="checkbox"/> events | <input checked="" type="checkbox"/> File Manager |
| <input checked="" type="checkbox"/> windows | <input checked="" type="checkbox"/> debugging |
| <input checked="" type="checkbox"/> controls | |

Suggested Retail Price: \$69.95

Minimum system requirements: Any Macintosh with at least 512K of RAM. Requires Turbo Pascal.



Turbo Pascal and Turbo Tutor are registered trademarks of Borland International, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright 1987 Borland International. BOR 0301

EUREKA: THE SOLVER™

If you're a scientist, engineer, financial analyst, student, teacher, or any other professional working with equations, Eureka: The Solver can do your Algebra, Trigonometry and Calculus problems in a snap.

Eureka also handles maximization and minimization problems, plots functions, generates reports, and saves an incredible amount of time. Even if you're not a computer specialist, Eureka can help you solve your real-world mathematical problems fast, without having to learn numerical approximation techniques. Eureka is easy to learn and easy to use—as simple as a hand-held calculator.

X + exp(X) = 10 solved instantly instead of eventually!

Imagine you have to solve for X, where $X + \exp(X) = 10$, and you don't have Eureka: The Solver. What you do have is a problem, because it's going to take a lot of time guessing at X. With Eureka, there's no guessing, no dancing in the dark—you get the right answer, right now. (PS: $X = 2.0705799$, and Eureka solved that one in less than 5 seconds!)

How to use Eureka: The Solver

It's easy.

1. Enter your equation into a problem text window
2. Select the "Solve" command
3. Look at the answer
4. You're done

You can then tell Eureka to:

- Verify the solutions
- Draw a graph
- Zoom in on interesting areas of the graph
- Generate a report and send the output to your printer or disk file
- Or all of the above

Some of Eureka's key features

You can key in:

- A formula or formulas
- A series of equations—and solve for all variables
- Constraints (like X must be $<$ or $=$ 2)
- Functions to plot
- Unit conversions
- Maximization and minimization problems
- Interest Rate/Present Value calculations
- Variables we call "What happens?," like "What happens if I change this variable to 21 and that variable to 27?"

Eureka: The Solver includes:

- Calculator+ desk accessory
- Powerful financial functions
- Built-in and user-defined functions
- Reports: generate and save them as MacWrite™ files—complete with graphs and lists—or as Text Only files
- Polynomial root finder
- Inequality constraints
- Logging: keep an up-to-the-minute record of your work
- Macintosh™ text editor
- On-screen Help system

Suggested Retail Price: \$195.00 (not copy protected)

Minimum system configuration: Macintosh 512K, Macintosh Plus, SE, or II with one 800K disk drive or two 400K disk drives.



Eureka: The Solver is a trademark of Borland International, Inc. Macintosh is a trademark of McIntosh Laboratory, Inc. licensed to Apple Computer, Inc. Copyright 1987 Borland International BOR 0415

TURBO PASCAL TOOLBOX™ **NUMERICAL METHODS**

Turbo Pascal Numerical Methods Toolbox for the Macintosh implements the latest high-level mathematical methods to solve common scientific and engineering problems. Fast.

So every time you need to calculate an integral, work with Fourier transforms, or incorporate any of the classical numerical analysis tools into your programs, you don't have to reinvent the wheel, because the Numerical Methods Toolbox is a complete collection of Turbo Pascal routines and programs that gives you applied state-of-the-art math tools. It also includes two graphics demo programs that use least-square and Fast Fourier Transform routines to give you the picture along with the numbers.

The Turbo Pascal Numerical Methods Toolbox is a must if you're involved with any type of scientific or engineering computing on the Macintosh. Because it comes with complete source code, you have total control of your application at all times.

What Numerical Methods Toolbox will do for you:

- Find solutions to equations
- Interpolations
- Calculus: numerical derivatives and integrals
- Matrix operations: inversions, determinants, and eigenvalues
- Differential equations
- Least-squares approximations
- Fourier transforms
- Graphics

Five free ways to look at Least-Squares Fit!

As well as a free demo of Fast Fourier Transforms, you also get the Least-Squares Fit in five different forms—which gives you five different methods of fitting curves to a collection of data points. You instantly get the picture! The five different forms are

1. Power
2. Exponential
3. Logarithm
4. 5-term Fourier
5. 5-term
Polynomial

They're all ready to compile and run as is.

Suggested Retail Price: \$99.95 (not copy protected)

Minimum system requirements: Macintosh 512K, Macintosh Plus, SE, or II, with one 800K disk drive (or two 400K).



All Borland products are trademarks or registered trademarks of Borland International, Inc. or Borland/Analytica, Inc. Macintosh is a trademark licensed to Apple Computer, Inc. Copyright 1987 Borland International. A Borland Turbo Toolbox product.

BOR 0419

Borland Software ORDER TODAY

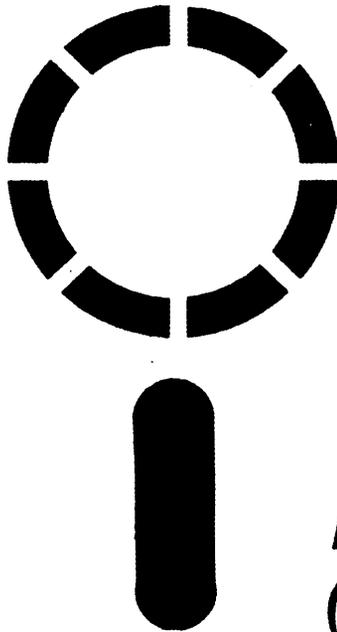


BORLAND

I N T E R N A T I O N A L

4585 Scotts Valley Drive Scotts Valley, California 95066

*To Order
By Credit
Card,
Call
(800)
255-8008*



*In
California
call
(800)
742-1133*

*In Canada call
(800) 237-1136*



SUPERKEY:[®] **THE** **PRODUCTIVITY** **BOOSTER**

RAM-resident

**Increased productivity
for IBM[®]PCs or
compatibles**

IBM[®] VERSION

PC, XT,[®] AT,[®] & True Compatibles

SuperKey's simple macros are electronic shortcuts to success. By letting you reduce a lengthy paragraph into a single keystroke of your choice, SuperKey eliminates repetition.

SuperKey turns 1,000 keystrokes into 1!

SuperKey can record lengthy keystroke sequences and play them back at the touch of a single key. Instantly. Like magic.

In fact, with SuperKey's simple macros, you can turn "Dear Customer: Thank you for your inquiry. We are pleased to let you know that shipment will be made within 24 hours. Sincerely," into the one keystroke of your choice!

SuperKey keeps your confidential files—confidential!

Without encryption, your files are open secrets. Anyone can walk up to your PC and read your confidential files (tax returns, business plans, customer lists, personal letters, etc.).

With SuperKey you can encrypt any file, *even* while running another program. As long as you keep the password secret, only *you* can decode your file correctly. SuperKey also implements the U.S. government Data Encryption Standard (DES).

- RAM resident—accepts new macro files even while running other programs
- Pull-down menus
- Superfast file encryption
- Choice of two encryption schemes
- On-line context-sensitive help
- One-finger mode reduces key commands to single keystroke
- Screen OFF/ON blanks out and restores screen to protect against "burn in"
- Partial or complete reorganization of keyboard
- Keyboard buffer increases 16 character keyboard "type-ahead" buffer to 128 characters
- Real-time delay causes macro playback to pause for specified interval
- Transparent display macros allow creation of menus on top of application programs
- Data entry and format control using "fixed" or "variable" fields
- Command stack recalls last 256 characters entered

Minimum system configuration: IBM PC, XT, AT, PCjr, and true compatibles. PC-DOS (MS-DOS) 2.0 or greater. 128K RAM. One disk drive.

SuperKey is a registered trademark of Borland International, Inc. IBM, XT, AT, and PCjr are registered trademarks of International Business Machines Corp. MS-DOS is a registered trademark of Microsoft Corp.

BOR 0294A

 **BORLAND**
INTERNATIONAL

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066

ISBN 0-87524-000-3