Super MacroWorks

by RANDY BRANDT

SUPER MACROWORKS

Enhancement for AppleWorks

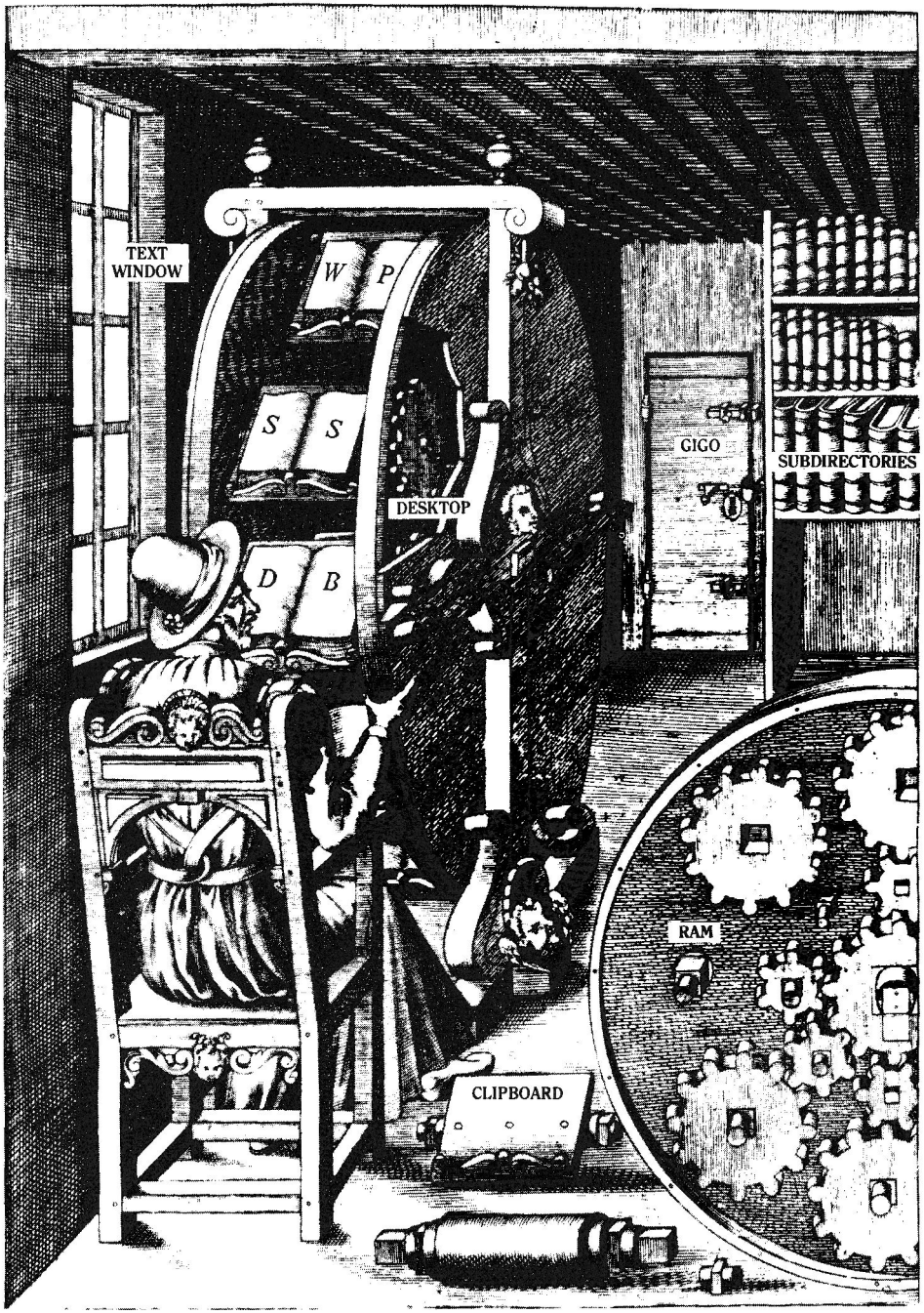
by Randy Brandt

ISBN 0-917085-00-0

Manual by Scott Myrom

Published by Beagle Bros Micro Software, Inc.
3990 Old Town Avenue
San Diego, California 92110

TABLE OF CONTENTS



APPLEWORKS VERSION 0.1

1. Welcome to Super MacroWorks

AppleWorks 2.0 or Newer, Please

Super MacroWorks requires AppleWorks version 2.0 or newer. Older versions of AppleWorks (1.3 and earlier) won't work with Super MacroWorks (but they will work with Beagle Bros' MacroWorks). If you have an earlier version of AppleWorks, call your Apple dealer and ask about Apple's AppleWorks trade-in policy. The new features of version 2.0 combined with the power of Super MacroWorks make it well worth the effort.

1a. Saving Time and Keystrokes With Macros

A *macro* is a single keystroke that does the work of many keystrokes. An AppleWorks macro is a SOLID-APPLE-key command; you simply hold down the SOLID-APPLE key while you press another key and a predefined sequence of keystrokes is performed. For example, you can set up a macro like SOLID-APPLE-N that types your name and address, or use a macro like SOLID-APPLE-I to indent a paragraph three spaces (*one* keystroke in place of the usual *seven*). Macros save you a lot of typing, and the fewer keystrokes you make, the fewer chances for error.

> *Note: The* SOLID-APPLE *key on the Apple IIe and IIc has been replaced by the* OPTION *key on the Apple IIGS. If you have a IIGS, use the* OPTION *key instead of the* SOLID-APPLE *key.*
>
> *In this manual, when we refer to the* SOLID-APPLE *key, we mean either the* SOLID-APPLE *key or the* OPTION *key.*

Remember: macros are SOLID-APPLE-key commands. The AppleWorks OPEN-APPLE-key commands perform the same functions as always.

Super MacroWorks is flexible. A wide variety of built-in macros are provided on the Super MacroWorks disk. They can be used "as is" or changed to suit your needs. You can also add new macros of your own design. You can easily create and try out custom macros directly from AppleWorks, store them on disk in a Word Processor file, and recall them at any time.

Super MacroWorks also provides you with many new OPEN-APPLE and CONTROL-key commands to use with AppleWorks and in macros.

This manual explains how to use Super MacroWorks. You will find that the power of Super MacroWorks extends to the limits of your own imagination and ingenuity.

Make a NOTES of it!

Before you do anything else, add the file NOTES from your Super MacroWorks disk to the Desktop and read its contents. NOTES contains important up-to-the-minute information about changes and additions to Super MacroWorks. Some of this information may not be included in this manual, so be sure to give NOTES a look before you use Super MacroWorks.

If you're too impatient to read the rest of this introductory section, turn to Section 2. You *must* perform the steps listed in Section 2a *before* you can use Super MacroWorks' macros and commands.

1b. Special Features of Super MacroWorks

In addition to macros, Super MacroWorks provides other features that make your life at the keyboard easier, more productive, and more fun. These include new AppleWorks commands, ways to customize AppleWorks, and mouse control.

New AppleWorks Commands

Super MacroWorks adds new commands to AppleWorks that save you time and effort. For example, you can organize your disks by reading file information into a Data Base file for sorting and reference, and you can set pathnames without having to remember subdirectory names. The new AppleWorks commands are described in Section 5.

Customized AppleWorks

You can use Super MacroWorks to customize the AppleWorks help screens or to tone down the error buzz. You can even download a custom font to your printer when you boot AppleWorks (this feature requires Beagle Bros' Power Print software). Read about these special features of Super MacroWorks in Section 7.

A Mouse in the House

Super MacroWorks allows you to use a mouse to scroll rapidly through AppleWorks files and make quick selections from menus. Here's how:

- Move the mouse to move the cursor (the same as holding down one of the four ARROW keys).
- Hold down the OPEN-APPLE key to make the cursor move farther and faster.
- Press the mouse button to (a) select an option from an AppleWorks menu (the same as pressing the RETURN key) or (b) scroll quickly through an AppleWorks file (the direction—up or down—is the same as the last vertical mouse movement).

You can use the Option Editor program of Super MacroWorks to adjust your mouse's sensitivity or to disable it completely. To use Option Editor, just boot the Super MacroWorks disk and select option O: Option Editor.

Option Editor will ask you to insert the disk that contains the file SUPER.SYSTEM. This is the file that Super MacroWorks uses to update AppleWorks with macros and new commands (the update procedure is described in Section 2a).

If you haven't yet updated AppleWorks, just leave the Super MacroWorks disk in the drive and press any key. If you have updated AppleWorks, insert the updated AppleWorks Startup disk (see Section 2a) in drive 1 and press any key.

If the copy of AppleWorks you are using is stored in a subdirectory, you must either:

- set the prefix from BASIC before you run the Option Editor program, or
- copy the file OPTION.EDITOR from the Super MacroWorks disk to the subdirectory that contains AppleWorks.

> *Note: Unless you specify otherwise, the mouse is turned on ("Mouse active? = Yes"). If you don't want to use a mouse, disable it with Option Editor ("Mouse active? = No"). If you have an Apple IIe with a mouse interface card but no mouse is plugged in, you must disable the mouse; otherwise, you will be treated to a spectacular (but useless) display of cursor jumping. (This also can happen on some IIc's that don't have a mouse plugged in.)*

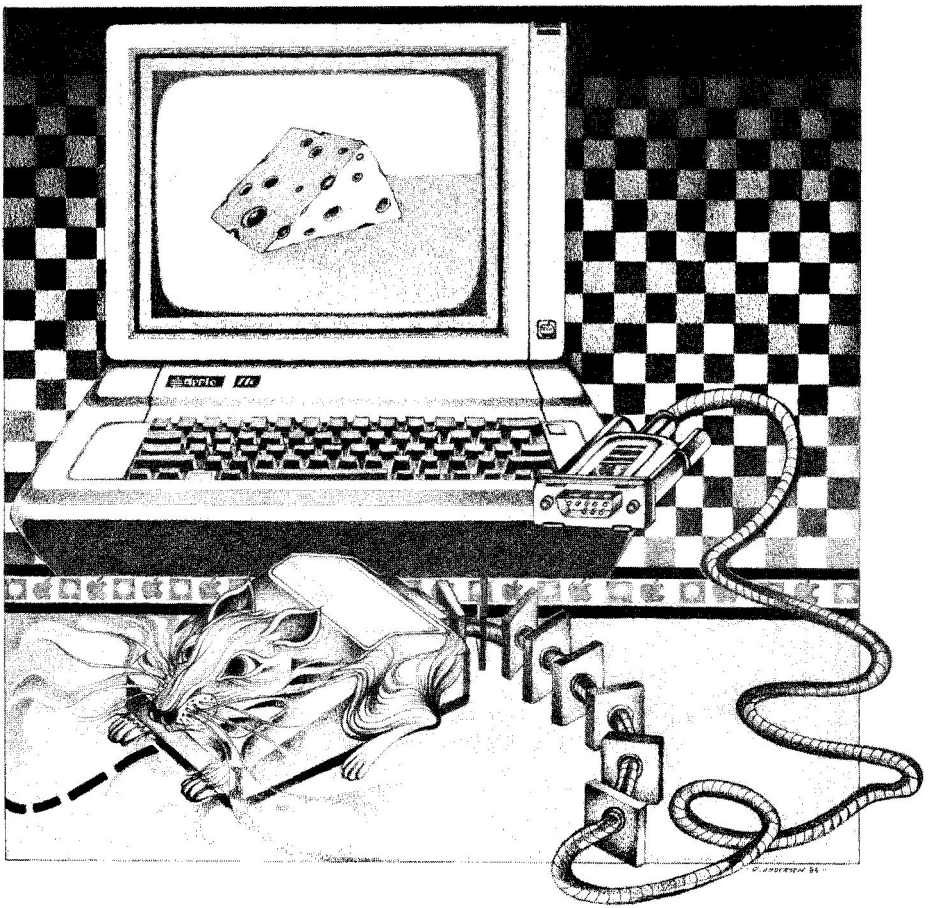
You can use a mouse with the Main Menu and Compiler programs of Super MacroWorks. Moving the mouse is the same as using the ARROW keys, and pressing the mouse button is the same as pressing the RETURN key.

A Mouse Tip

The mouse is especially handy for creating Data Base layouts. Just hold down the OPEN-APPLE key and use the mouse to drag fields into position.

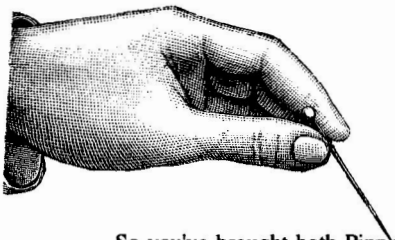
1c. Back It Up

Like all Beagle Bros disks, Super MacroWorks is copy*righted*, not copy protected. This means you can and should make a backup copy of the disk (use the copy program that came with your Apple). Use either the original Super MacroWorks disk or the copy; store the other disk in a safe place away from magnetic fields and teething infants. Please make backup copies for your own use only. Every illegal copy you give to a friend is a vote *for* copy protection and *against* friendly software. You support us and we'll support you.



Attention Pinpoint™ Users

Here's a suggestion from Pinpoint Publishing's July 1986 *Points of Interest* newsletter:



So you've brought both Pinpoint desktop accessories and MacroWorks, and they don't both work together in Memory. Both compliment AppleWorks nicely. What's a User to do?

RunRun (featured in the June issue of *Points of Interest*) is a handy *shell* to ProDOS that lets you launch applications. You can create two versions of AppleWorks on your disk, one containing Pinpoint desktop accessories, and one containing MacroWorks.

You can quickly switch between the two versions of AppleWorks by exiting one version and running the other. RunRun works well with the RAM Enhancement kit from Pinpoint, since you can load the two RunRun files — RUNRUN.SYSTEM and RUNRUN.APPLST up to the RAM drive in advance. This facilitates rapid switching between applications.

2. *The First Step*

This section tells you how to start using Super MacroWorks. It describes how to update your AppleWorks Startup disk and how to select the directory you want to use. It also gives you an overview of how to use Super MacroWorks' built-in macros and how to create your own custom macros. Sections 3, 4, and 5 provide detailed information about how to create and use custom macros.

2a. How to Update Your AppleWorks Disk

Before you can use macros, you must update a copy of your AppleWorks Startup disk. (*Note: The floppy disk version of AppleWorks uses two disks: a STARTUP disk (the one you boot) and a PROGRAM disk. If you are using AppleWorks on a hard disk or a UniDisk 3.5, there is only one disk.*)

If you are using a "desktop expander" from Applied Engineering or Checkmate (or any other firm that uses a similar method of modifying AppleWorks), perform steps A - E first. Otherwise, perform steps 1 - 8.

Desktop Expanders

You can use Super MacroWorks with desktop expanders, *but make sure your desktop expander program works with AppleWorks version 2.0.* Super MacroWorks will *not* work with programs like Pinpoint™ that use the same memory as Super MacroWorks.

Here's how to set up Super MacroWorks to work with a desktop expander:

A. Make a copy of your original *unmodified* AppleWorks Startup disk. (You may use any disk-copy program.)

B. Boot the Super MacroWorks disk and select option M: Main Menu.

C. Select option 4: Pre-Expander Patch.

D. Run your desktop expander program. Do *not* install printer buffers, if that is an option.

E. Go to Step 2 below.

Installing Super MacroWorks

1. Make a copy of your original *unmodified* AppleWorks Startup disk. (You may use any disk-copy program.)

2. Boot the Super MacroWorks disk and select option M: Main Menu.

3. Select option 1: Enhance AppleWorks.

4. Select option 1: Add Macros to AppleWorks.

5. Remove the Super MacroWorks disk from the main disk drive, and then insert the AppleWorks Startup disk you created in Step 1.

6. Press RETURN.

7. Select option 2: Slot 6 Drive 1 (see Section 2b if your disk drive is in a different slot).

   The cursor will return to option 1: Current Directory, and the disk name that follows "Current Directory:" will change (probably to /APPLEWORKS/).

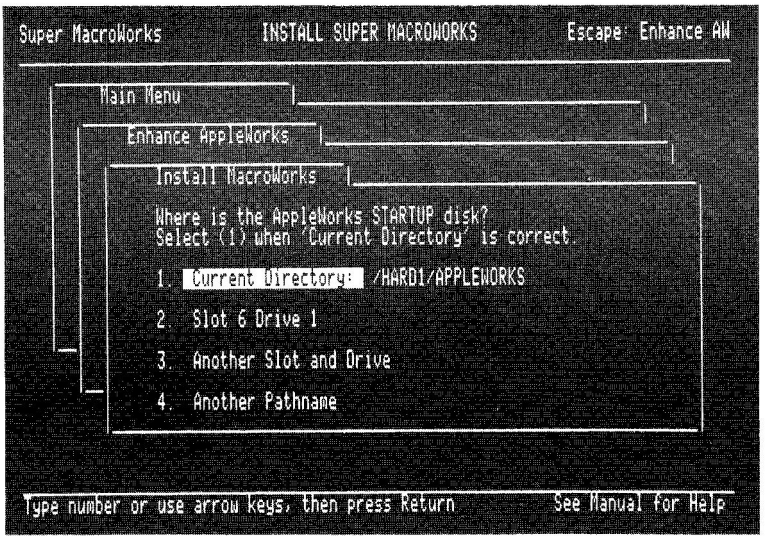
8. Press RETURN.

   Your AppleWorks Startup disk will be updated with Super MacroWorks' macros and commands. A message will appear at the bottom of the screen when the updating is complete. Press any key; you will be returned to Super MacroWorks' Main Menu.

The built-in macros supplied with the Super MacroWorks disk are now ready for use (see Section 2c). You can also create custom macros directly from AppleWorks (Section 3) and store custom macros on disk for permanent use (Section 4).

2b. How to Specify File Locations

Super MacroWorks needs to know the location of your AppleWorks files. If you are running either the Main Menu or the Compiler program and the program can't find your AppleWorks files, it will display this "Where's Your Disk?" menu:



This menu has four options:

Option 1: Displays the current location of the files. This is either the name of a disk (the volume name) or the name of a subdirectory.

If you are using floppy disks, option 1 probably will display a volume name. If you are using a hard disk or a UniDisk 3.5, option 1 probably will display the name of a subdirectory. In either case, the name is preceded by a slash.

If the location displayed by option 1 is correct, move the cursor to option 1 and press RETURN. If the location is incorrect, follow the steps listed below to specify the proper location.

Option 2: Displays the slot and drive numbers of the second disk drive, if you have one.

Option 3: Allows you to specify slot and drive numbers.

Use this option to locate AppleWorks files that are *not* stored in a subdirectory (see steps A1 - A4 below).

Option 4: Allows you to specify a pathname.

Use this option to locate AppleWorks files that are stored in a subdirectory (see steps B1 - B3 below).

The way you use these options to specify file locations depends on whether or not the files are stored in a subdirectory.

If the files are not stored in a subdirectory (floppy disks):

A1. Check option 2: Is it displaying the proper slot and drive numbers? If so, go to step A2. If not, go to step A3.

A2. Insert the disk you want to use in the drive indicated by option 2. Then move the cursor to option 2 and press RETURN. Go to step A4.

A3. Insert the disk you want to use in a drive. Move the cursor to option 3 ("Another Slot and Drive") and press RETURN. The slot number in option 2 will be highlighted. If the slot number is correct, just press RETURN. If the slot number is not correct, change it:

- Use the UP-ARROW or DOWN-ARROW key to change the number, then press RETURN; or
- Type the correct number.

Repeat this procedure for the drive number. Go to step A4.

Note: Ninety-nine percent of all Apples have drives connected to Slot 6 Drive 1 and, if there is a second drive, Slot 6 Drive 2. On the Apple IIc, Drive 1 is the internal drive and Drive 2 is the external drive. UniDisk 3.5's are often in Slot 5.

A4. Option 1 should now display the correct disk (volume) name. Press RETURN.

If the files are stored in a subdirectory (hard disks and UniDisk 3.5's):

B1. Move the cursor to option 4 and press RETURN.

B2. Type the ProDOS pathname that identifies the correct subdirectory, then press RETURN. (Type only the disk name and the subdirectory name; do *not* type any filenames.)

Note: Complete information about pathnames is contained in your Apple manuals, such as the ProDOS User's Manual. *A pathname always begins with a slash, followed by the disk (volume) name, then another slash, then the subdirectory name, like this: /DISKNAME/SUBDIR*

B3. Option 1 should now display the correct subdirectory. Press RETURN.

2c. How to Use Macros

To use Super MacroWorks, simply boot AppleWorks with the updated Startup disk you created in Section 2a above. A startup screen with the Super MacroWorks copyright message will be displayed before the usual AppleWorks copyright screen.

Super MacroWorks provides you with a variety of built-in macros that you can use "as is." You can also create your own custom macros and store them on disk for permanent use.

Built-In Macros

The "built-in" macros are the macros supplied on your Super MacroWorks disk. You may change most of these macros, but some of them are "reserved" and cannot be modified or deleted. The reserved macros are listed in Section 4f.

The definitions of Super MacroWorks' built-in macros are contained in this Word Processor file:

MACROS.SUPER

This Word Processor file contains written descriptions of the built-in macros:

MACRO.COMMENTS

To see what the built-in macros are and how they work, boot AppleWorks with your updated Startup disk and add these two files to the Desktop. Then use your printer to make printouts of the files.

When you look at the macro definitions in MACROS.SUPER, you'll see abbreviations enclosed in <brackets>. We call these *tokens*. Tokens are used to represent special keystrokes. For example, the token <rtn> represents the RETURN key and the token <ctrl> represents the CONTROL key. Tokens are a shorthand that helps you visualize the keystrokes that make up each macro. Section 4b below gives a complete description of macro tokens and how to use them.

Each time you boot AppleWorks with your updated Startup disk, the built-in macros are ready for use. To use one, simply hold down the SOLID-APPLE key while you press the appropriate key to activate the macro.

> *Note: Some of the built-in macros can be used in any of the three AppleWorks applications (Word Processor, Data Base, and Spreadsheet). We call these "global" macros. Some of the built-in macros can be used in only one of the applications; we call these "local" macros. A special token in each macro definition tells whether the macro is global or local. The token <all> means global; <awp> means Word Processor; <adb> means Data Base; <asp> means Spreadsheet. See Section 4b for more information.*

Custom Macros

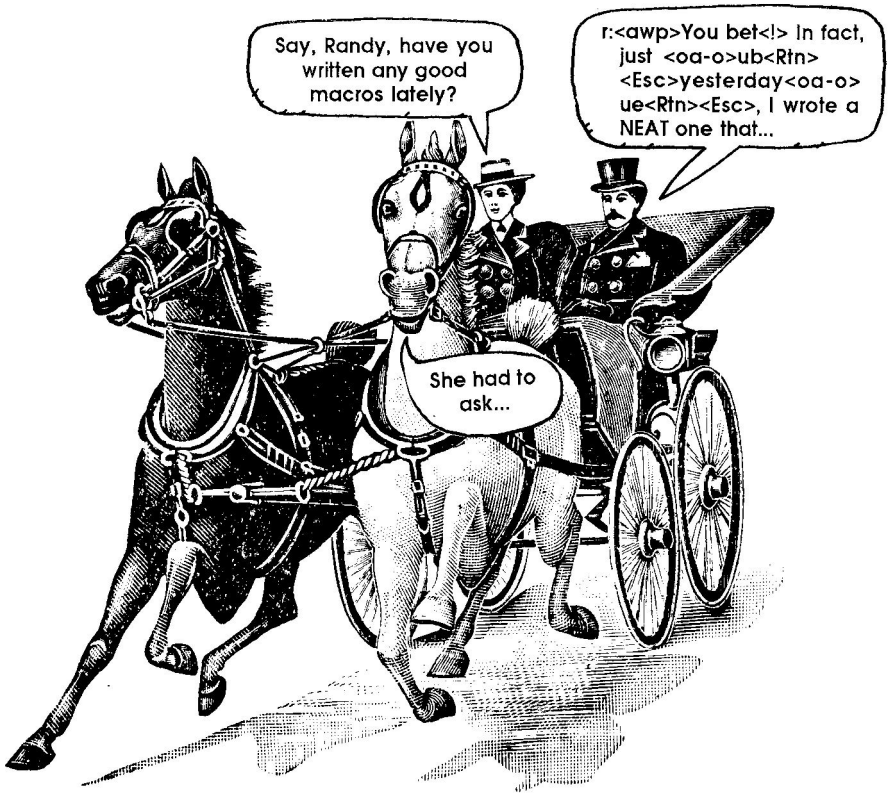
Super MacroWorks makes it easy for you to create custom macros that do whatever you want them to. There are two ways to create custom macros:

- *"Recorded" macros:* The OPEN-APPLE-X command places AppleWorks in the "Record a Macro" mode. Then you can create and test macros directly from AppleWorks.

  Recorded macros are ready for use immediately, but they disappear when you quit AppleWorks unless you save them. Section 3 tells you how to use the OPEN-APPLE-X command to record macros and how to save them for future use.

- *The MACROS. File:* Another way to create macros is to make changes to a renamed copy of the file MACROS.SUPER (the file that contains Super MacroWorks' built-in macros). Use AppleWorks to change the existing macros and add your own macros. Then use the Compiler program to compile the renamed file and save it on your AppleWorks Startup disk. The macros in the file will be ready for use each time you boot AppleWorks.

  Section 4 tells you how to create, compile, and save a MACROS. file.



3. *Recorded Macros*

Super MacroWorks makes it easy for you to create and experiment with macros directly from AppleWorks. The secret is the OPEN-APPLE-X command. This command signals that a macro is about to be defined and puts AppleWorks in the RECORD A MACRO mode. The keystrokes after OPEN-APPLE-X define the macro, and the command CONTROL-@ signals the end of the macro.

3a. How to Create a Recorded Macro

1. Boot AppleWorks with the updated Startup disk you created in Section 2a.

2. From within a Word Processor file, press OPEN-APPLE-X. Notice that the title at the top of the screen changes to "RECORD A MACRO."

3. The prompt "Select macro key:" appears at the bottom of the screen. Type a letter, number, or other keyboard character to identify the SOLID-APPLE command that will be used to activate the macro. For example, if you type 1, the command SOLID-APPLE-1 will activate the macro after it has been recorded.

   Some keyboard characters *cannot* be recorded as macros; refer to the Key Chart at the back of this manual. Keep in mind that the tilde (~) is the lower-case equivalent of the upper-case caret (^). If you try to record the ~, this error message will appear: Reserved^ (only the upper-case character appears in an error message). Likewise, DELETE is the lower-case equivalent of the upper-case _ (underscore).

   If you record Ø (zero), you can record up to 70 keystrokes *(but be careful with macro Ø; see the warning in Section 3c).* With all the other keys, you can record keystrokes until the macro table is full (which may allow anywhere from 2 to over 4,000 keystrokes, depending on what is already in the table).

4. The message "Recording *x* " appears in the lower right corner of the screen (*x* is the character you entered at step 3). Enter the keystrokes that cause the macro to do whatever it is supposed to do.

   If you make a mistake:

   - Press CONTROL-@.
   - Press OPEN-APPLE-X to re-start recording. Type the number, letter, or character that identifies this macro.
   - If the prompt "Replace global [or local] macro *x* ?" appears, select "Yes" (which means you want to re-record this macro).
   - Re-enter the keystrokes for this macro.

   Note: If the macro has many keystrokes and you don't want to re-type it, there is an easier way to correct mistakes. See "How to Change Recorded Macros" below.

5. Press CONTROL-@ to signal the end of the macro definition. The message "Done macro *x* " will appear in the lower right corner of the screen.

6. The macro is now ready for use. Just press SOLID-APPLE-*x*, where *x* is the character you entered at step 3 to identify the macro.

   Note: Recorded macros disappear when you quit AppleWorks, but there is a way to save them for future use. See "How to Save Recorded Macros" below.

How to Change Recorded Macros

When you try out recorded macros, you may discover that they don't work the way you thought they would. You may need to correct mistakes or to experiment with improvements and additions. You can re-record macros, but this may be difficult if there are a lot of keystrokes or you can't remember exactly which keystrokes you used. Here's an easy way to change recorded macros:

1. Make sure you're in the Word Processor. Press OPEN-APPLE-$. The definition of the macro you just recorded will be displayed. (OPEN-APPLE-$ does *not* work in the Data Base or the Spreadsheet.)

   Macro definitions use <bracketed> "tokens" to represent special keystrokes. For example, the token <Rtn> represents the RETURN key. See Section 4b for a complete explanation of macro tokens. An exclamation point (!) signals the end of each macro definition.

2. Use the Word Processor to change the macro definition(s).

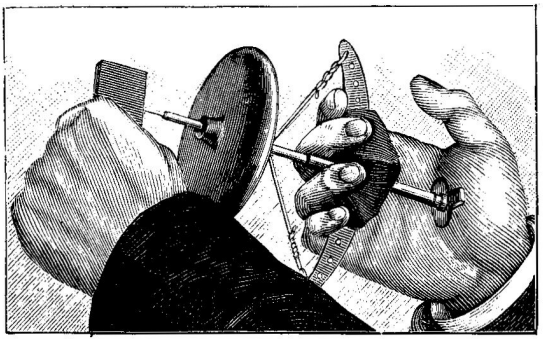
3. Insert the word START (followed immediately by a carriage return) on a line above the first macro in the file. You must do this in order to compile the macros (step 4), even if there is only one macro in the file. *(Note: If you don't want to compile all the macros in the file, insert a line with the word END just after the last macro that you want to compile. The macros after END will not be compiled.)*

4. Press OPEN-APPLE-=. This compiles the new macro definitions.

   If the compilation is successful, the message "No errors" appears in the lower right corner of the screen. Otherwise, an error message appears. For example, if you haven't inserted the START marker in the file (step 3), the message "No start" appears. The compiler error messages are listed in Section 4d.

5. Your new macros are ready to be used.

Each time you change a macro definition, you must re-compile the macros with the OPEN-APPLE-= command.



A Note About Compilers

Super MacroWorks actually provides two compilers:

- The OPEN-APPLE-= command. Use this compiler directly from AppleWorks.

  This compiler is contained in the file COMPILER.SYS, which is *not* available to Quark hard disk users.

- The Macro Compiler program. Use this compiler to compile macros contained in a MACROS. file (see Section 4). To use the Macro Compiler, boot the Super MacroWorks disk and select option C from the Main Menu (see Section 4d for details).

The two compilers work the same way, but the Macro Compiler program performs an additional function: it stores the macros in the MACROS. file on your AppleWorks Startup disk. That way, the macros are ready for use each time you boot AppleWorks.

Each compiler makes every effort to successfully compile every macro. If it doesn't recognize something in a macro, it treats it as text. For example, if you accidentally type a blank space inside a token (like <r tn> instead of <rtn>), the compiler won't recognize the token and will assume that it's text. The resulting macro may compile, but it won't work the way you thought it would.

Therefore, it's possible for a macro to compile with "No errors" but still not work properly. Look closely at the macro definition; if you see anything that doesn't make sense, re-define and re-compile the macro.

The compiler error messages are listed in Section 4d.

How to Save Recorded Macros

Recorded macros are lost when you quit AppleWorks. However, it is possible to save them so they can be used again in future sessions without your having to re-record them. Here's how:

1. At the end of your recording session, make sure you're in the Word Processor and press OPEN-APPLE-$ to display the macro definitions.

2. Save the Word Processor file that contains the macro definitions on disk.

3. Quit AppleWorks.

4. The next time you want to use the macros, add the file that contains the macro definitions to the Desktop.

5. Press OPEN-APPLE-=. This compiles the macros and activates them. The macros in the file are now ready to be used.

Another Alternative:

If you like, you can save recorded macros on your AppleWorks Startup disk so they will be available each time you boot AppleWorks. Here's how:

1. At the end of your macro-recording session, make sure you're in the Word Processor and press OPEN-APPLE-$ to display the macro definitions.

2. Save the Word Processor file that contains the macro definitions on disk. Rename the file; give it a name that begins with the seven characters MACROS. (including the period), like MACROS.NEW.

3. Follow the instructions in Section 4d to compile your MACROS. file (using the Macro Compiler program). The macros will be stored on your AppleWorks Startup disk.

3b. Try This Example

For practice, try creating this simple recorded macro. This macro causes SOLID-APPLE-5 to center and print the message "Macros are great".

1. From within AppleWorks, press OPEN-APPLE-X.

   The RECORD A MACRO title appears at the top of the screen. This means that every keystroke from this point on is being "recorded" as part of a macro definition.

2. The prompt "Select macro key:" appears at the bottom left of the screen. Press the 5 key. The message "Recording 5" appears at the bottom right of the screen.

3. Press OPEN-APPLE-O, then type CN, then press RETURN, then press ESC.

4. Type "Macros are great" (without the quotation marks), then press RETURN.

5. Press OPEN-APPLE-O, then type UJ, then press RETURN, then press ESC.

6. Press CONTROL-@ to signify the end of the macro definition. The message "Done macro 5" appears at the bottom right of the screen.

7. Try out the macro; press SOLID-APPLE-5.

8. Press OPEN-APPLE-$; the macro definition is displayed. It should look like this:

   ```
   5:<all><oa-o>cn<Rtn><Esc>Macros are
   great<Rtn><oa-o>uj<Rtn><Esc>!
   ```

9. Now we're going to change the message to read "Macros are super!" Use the Word Processor to change the word "great" in the macro definition to "super" and add an exclamation point.

   Caution: Be sure to enclose the exclamation point in brackets, like this: super<!>. If you don't, the Compiler will see the exclamation point as the end of the macro and treat the rest as a comment. To include an exclamation point as text in a macro, you must enclose it in brackets.

   The new macro definition should look like this:

   ```
   5:<all><oa-o>cn<Rtn><Esc>Macros are
   super<!><Rtn><oa-o>uj<Rtn><Esc>!
   ```

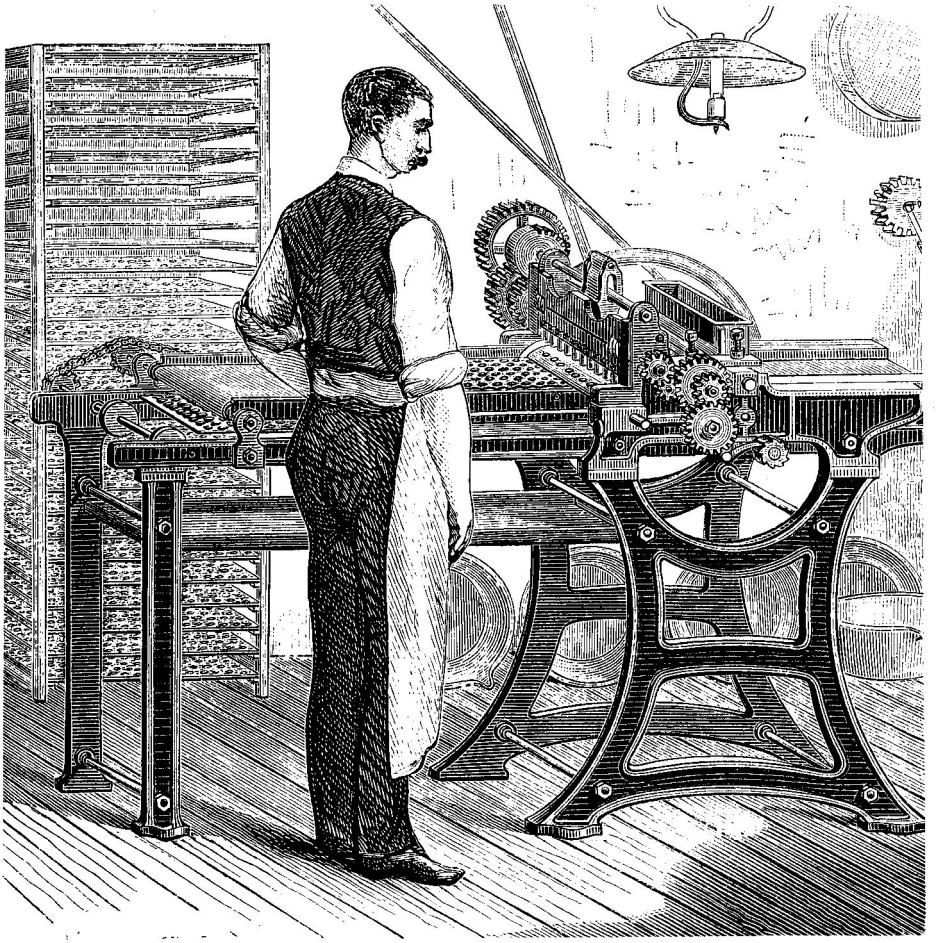
10. Type the word "start" on the line just above the macro definition and press RETURN.

11. Press OPEN-APPLE-=. This compiles the new macro definition. Now if you press SOLID-APPLE-5, the centered message "Macros are super!" will appear on the screen.

If you want to save this macro for future use, just save the Word Processor file that contains it. The next time you want to use the macro, add the file to the Desktop and press OPEN-APPLE-= to compile the macros in the file.

3c. *The Special Case of Macro Ø*

You can record SOLID-APPLE-Ø (zero) as a macro, but *be careful.* Macro Ø is used as a special storage area by several of the new AppleWorks commands provided by Super MacroWorks (these are described in Section 5b). Therefore, if you try to record the zero key, the definition you give it may be wiped out by another command. See Section 5b for more information about the special uses of macro Ø.



EARLY MACRO COMPILER

4. *The MACROS. File*

This section tells you how to create custom macros that will *automatically* be ready for use each time you boot AppleWorks with your updated Startup disk.

A file named MACROS.SUPER is supplied on your Super MacroWorks disk. This file contains Super MacroWorks' built-in macros. You can create a modified version of this file that contains your own custom macros. Super MacroWorks can compile your custom MACROS. file and store it on your AppleWorks Startup disk; then each time you boot AppleWorks, your custom macros are ready for use.

4a. The Two-Step Process

A custom MACROS. file is created in a two-step process:

1. *Create a MACROS. File.*

   A MACROS. file is a file that contains macro defintions and that has a name beginning with MACROS. (including the period). There are two ways to create such a file:

   - Add the file MACROS.SUPER from the Super MacroWorks disk to the Desktop. Rename the file; give it a name that begins with MACROS., like MACROS.NEW. Use AppleWorks to modify the file and create your own custom macros, then save the modified file on disk.
   - From within AppleWorks, use the OPEN-APPLE-X command to record macros (as described in Section 3). Save the file that contains the recorded macros; give it a name that begins with MACROS.

2. *Compile the MACROS. file.*

   Use Super MacroWorks' Compiler program to compile the MACROS. file created in step 1 and save it on your AppleWorks Startup disk.

Sections 4b and 4c give detailed information about the first step; Section 4d describes the second step.

4b. The Anatomy of a Macro

Before you start modifying MACROS.SUPER to create your own macros, you need to know how a macro is built. *If you haven't already, use your printer to obtain printouts of the files MACROS.SUPER and MACRO.COMMENTS.*

Tokens

Look at the macros in the MACROS.SUPER file (the macros come after the word START and before the word END). Each macro consists of a series of characters and *tokens.* A token is a symbol enclosed in <brackets> that represents a special keystroke. For example, the token **<rtn>** stands for the RETURN key, and the token **<left>** stands for the LEFT-ARROW key.

Here is a macro from the MACROS.SUPER file:

C:<awp><oa-O>CN<rtn><esc>! center text

Each macro begins with a character or token that represents the key used with SOLID-APPLE to activate the macro. In this case, the character C indicates that this macro is activated by the SOLID-APPLE-C command.

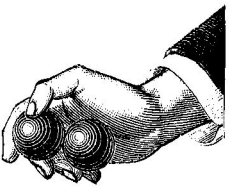
Next comes a colon, followed by a token that indicates whether the macro is local or global (this is explained below). This macro is local: it works only within the AppleWorks Word Processor (<awp>).

Next come the keystrokes (characters and tokens) that are executed by the macro. In this example there are five keystrokes: OPEN-APPLE-O, C, N, RETURN, and ESCAPE.

An exclamation point (!) signals the end of the macro. If you wish, you may add a comment after the exclamation point.

Here are some of the tokens you can use to create macros:

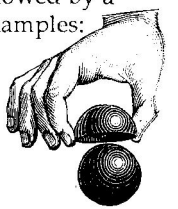
<del>	DELETE key
<esc>	ESCAPE key
<rtn>	RETURN key
<tab>	TAB key
<left>	LEFT-ARROW key
<right>	RIGHT-ARROW key
<up>	UP-ARROW key
<down>	DOWN-ARROW key
<spc>	space



(Instead of using the **<spc>** token, you can just type a space.)

The tokens for OPEN-APPLE, SOLID-APPLE, and CONTROL commands use the abbreviations **oa**, **sa**, and **ctrl** followed by a hyphen and the appropriate key. Here are some examples:

<oa-A>	OPEN-APPLE-A
<sa-A>	SOLID-APPLE-A
<sa-left>	SOLID-APPLE-LEFT-ARROW
<sa-ctrl-A>	SOLID-APPLE-CONTROL-A



Super MacroWorks provides you with some new CONTROL-key commands to use in macros. These commands and their tokens are described in Section 5.

Tokens may be typed in upper or lower case, but no spaces are allowed between the brackets. For example, **<rtn>**, **<RTN>**, and **<Rtn>** are all valid tokens for the RETURN key, but **<r tn>** is *not* valid. The Compiler would assume that <r tn> is text rather than a token; the macro would compile, but it wouldn't work the way you want it to.

Local and Global Macros

Each macro must be classified as either local or global. A *local* macro is a macro that works within only one of the three AppleWorks applications (Word Processor, Data Base, or Spreadsheet). A *global* macro is a macro that works within all three applications.

In a macro definition, the token just after the colon indicates whether the macro is local or global:

<all>	all applications (global)
<awp>	Word Processor only
<adb>	Data Base only
<asp>	Spreadsheet only

You cannot have more than one global macro with the same name (the second one will never be used), but you can give the same name to two or three local macros as long as they are in separate applications. For example, look at macro H (which stands for SOLID-APPLE-H) in MACROS.SUPER. Macro H has the same result in each application (it returns the cursor to the top left of the file), but it cannot be defined as global because the keystrokes differ between applications. Therefore, there are three different definitions of H, one for each of the three applications.

The order in which macros appear in MACROS.SUPER is important. Suppose you want to define two macros, one local and one global, and give them the same name. Super MacroWorks will not recognize the local macro unless it comes *before* the global macro in the file. For example, there are two macros named <left> in MACROS.SUPER. The local version of <left> appears first in the file; it is effective within the Word Processor. The global version, which uses different keystrokes, appears after the local version; the global version is effective within the Data Base and the Spreadsheet. Once you have defined a global macro, Super MacroWorks will not recognize a local macro of the same name.

Recorded macros (macros defined with the OPEN-APPLE-X command; see Section 3) are global by default. If you want to change a recorded macro to local: use OPEN-APPLE-$ to display the macro definition; then use AppleWorks to change the <all> token to the desired local token; then use OPEN-APPLE-= to compile the new definition.

When I'm Calling You

One macro can call another; that is, a macro can contain a SOLID-APPLE keystroke if that keystroke is defined elsewhere in the macro file. For example, here is macro Y in MACROS.SUPER:

Y:<all><sa-left><oa-M>T<down><left><rtn>! delete a line

This macro definition includes the SOLID-APPLE-LEFT-ARROW macro, which is defined elsewhere in the file (macro <left>).

> *Caution: When you delete a macro from a file, make sure the macro isn't called by another macro elsewhere in the file. (Use the OPEN-APPLE-F (Find) command to search for references to the macro.)*

4c. How to Create a Custom MACROS. File

Now that you know how macros are built, you are ready to create your own custom MACROS. file. Here's how:

1. Boot AppleWorks with the updated Startup disk you created in Section 2a.

2. Add the file MACROS.SUPER from the Super MacroWorks disk to the Desktop. (This file contains Super MacroWorks' built-in macros.)

3. Rename MACROS.SUPER (use the OPEN-APPLE-N command).

   Important: The new filename must begin with the seven characters MACROS. (including the period); for example, MACROS.NEW or MACROS.RANDY.

4. Use AppleWorks to make changes to the renamed file. Either modify existing macros or add new macros.

   The macros are located after the word START and before the word END. The comments and suggestions before START and after END are optional and may be changed or deleted.

   Suggestion: Use OPEN-APPLE-Z ("zoom") to reveal carriage returns and printer options.

   Another suggestion: Use the OPEN-APPLE-X command (see Section 3) to record macros from within AppleWorks. The OPEN-APPLE-$ command allows you to see the macros that have been defined. When the macros are "bug-free," use AppleWorks to copy or move the macro definitions to where you want them to appear in your MACROS. file.

5. Save your custom MACROS. file on disk.

6. Use Super MacroWorks' Compiler program to update your AppleWorks Startup disk with the macros in your MACROS. file.

   This procedure is explained in Section 4d.

   > *Note: Some macros cannot be changed or deleted. Section 4f lists these reserved macros and describes their uses.*

Auto-Startup

After you have used the Macro Compiler program to compile a MACROS. file (see Section 4d), the first macro in the file will be executed automatically each time you boot AppleWorks with your updated Startup disk. This can be a *very* helpful feature. For example, you can set up the first macro in the file to locate a file on disk, add it to the Desktop, and open it so you can work in it — all without your having to give a single command!

If you do *not* want the first macro to execute when you boot AppleWorks, you can disable the Auto-Startup feature. Just boot the Super MacroWorks disk, select option O (Option Editor), and then set "Auto-Startup Macro = No" on the option editor menu. Or make the first macro in the file a macro that doesn't do anything, like this: <ctrl-x >:<all>!

> *Note: A special file called MACROS.STARTUP is supplied on your Super MacroWorks disk. You can create an auto-startup macro that makes use of this file to help you get started when you boot AppleWorks. For more information, see Section 5c.*

4d. How to Compile a MACROS. File

After you have used AppleWorks to create a MACROS. file and store it on disk, the next step is to compile the file and store the macros in SUPER.SYSTEM on your AppleWorks Startup disk. To do this, you must use the Macro Compiler program on your Super MacroWorks disk.

Here's how to compile the macros in your MACROS. file and update your AppleWorks Startup disk:

1. Boot the Super MacroWorks disk. Select option C: Compiler.

2. Select option 1: Compile Macros.

3. You will be asked "Where is the MACROS. file?" Follow the procedure described in Section 2b to specify the volume (disk) name or subdirectory where the file is located.

4. One or more filenames will appear on the screen. Move the cursor to the filename of the file you want to compile, then press RETURN.

   Super MacroWorks will compile the macro file. If the compilation is successful, a "No errors found" message will appear. If there is an error, an error message will appear (see below).

5. You will be asked "Where is your Startup disk?" Insert your AppleWorks Startup disk in a drive, then follow the procedure described in Section 2b to specify the current directory.

   Super MacroWorks will update the Startup disk with the macros in your MACROS. file. The message "Default macros installed" will appear.

6. Press ESC to quit or RETURN to update another disk with the compiled macros.

The custom macros in your MACROS. file are now ready to be used, and they automatically will be ready for use each time you boot AppleWorks with the updated Startup disk.

Compiler Error Messages

The error messages listed below apply to both the OPEN-APPLE-= compile command and the Macro Compiler program. If an error applies to a specific macro, the error message will identify the macro. A caret (^) is used to represent the CONTROL key. For example, if you receive an error message like this:

```
Bad Appl ^D
```

it means there is an error in the definition of the macro <ctrl-D>.

Remember that the compiler assumes anything it doesn't recognize is text. Therefore, the fact that a macro compiles successfully doesn't guarantee that it has no errors; it may work, but perhaps not the way you want it to.

Here are the Compiler error messages:

No start

The Compiler couldn't find the START marker in the macro file. The word START must be on a line by itself, immediately preceded and followed by carriage returns (no spaces).

No Macros

The Compiler couldn't find a recognizable macro definition between the START and END markers in the macro file.

Bad Appl

You used an invalid global/local token. The valid tokens to indicate which AppleWorks application(s) a macro works in are: <all>, <awp>, <asp>, and <adb>.

Error in

You've made an error in a macro definition. This message usually means that the compiler didn't recognize a <sa- > or <oa- > token.

Reserved

You tried to define one of the special reserved macros, which cannot be changed or deleted (see Section 4f).

Table full

The macros in the file use up more than the maximum allowed number of bytes (see Section 4e). This message refers to the number of keystrokes defined in the macros themselves, *not* to the overall size of your MACROS. file.

No errors

At least none that the Compiler found.

Stopped at

You pressed ESC to stop the compiling process. The macros defined before the macro indicated by this message will be functional.

4e. Things to Watch Out For

Writing macros is easy, but there are a few limitations and "don'ts" you should keep in mind to ensure that your macro file compiles successfully and works the way you want it to.

Table Size Limit

The macro table can contain a maximum of about 4,000 bytes. This limit does *not* include the reserved macros (see Section 4f) and the space occupied by comments. If you exceed the table size limit, the error message "Table full" will appear when you try to compile the macros.

The table size limit is really not too limiting. To give you an idea, the macros in the MACROS.SUPER file supplied on your Super MacroWorks disk use up only about 2,000 bytes, or half the table capacity.

Here's how to count the number of bytes in a macro:

- Every macro includes three bytes of "overhead": the character or token that identifies the macro, the token that indicates whether the macro is global or local, and the end-of-macro marker.
- Each text character or token (including OPEN-APPLE keystrokes) counts as one byte.
- A call to another macro (i.e., a SOLID-APPLE keystroke) counts as two bytes.
- Comments use zero bytes.

For example, this macro:

N:<awp><oa-O>CN<rtn><esc>Hello<oa-O>UJ<rtn><esc>!

uses a total of 18 bytes:

3 bytes of "overhead" ("N", "<awp>", and "!")
6 bytes of <bracketed> keystroke tokens
9 bytes of text characters (C, N, H, e, l, l, o, U, J)
18

A Few "Don'ts"

- Don't forget to precede the macros in each macro file with the word START. Make sure that START (and END, if you use it) appears on a line by itself, immediately preceded and followed by carriage returns.

- Don't define a lower-case macro if you've already defined an upper-case macro for the same key, and vice versa. For example, you can't define a macro **n** if you've already defined a macro **N**. Keep in mind that the tilde (~) is the lower-case equivalent of the upper-case caret (^) and DELETE is the lower-case equivalent of the upper-case underscore (_). Refer to the Key Chart at the back of this manual.

- Don't let a macro call another macro that isn't defined somewhere in the file.

- Don't misspell a <bracketed> token name.

- Do have fun creating and experimenting with macros. Let us know about custom macros that you find especially useful or clever.

4f. Reserved Macros

The special macros listed below cannot be re-recorded, changed or deleted; you must use them "as is." You can use these macros at any time (unless otherwise noted): directly from the keyboard (press the appropriate SOLID-APPLE key), while recording a macro (press the appropriate SOLID-APPLE key), or in a macro definition (use the appropriate token).

<sa-del>
SOLID-APPLE-DELETE

Deletes the character at the current cursor position. (You may also use OPEN-APPLE-DELETE for this function.)

<date>
SOLID-APPLE-'

Displays the date in this format: July 4, 1986 (handy for printing reports).

<date2>
SOLID-APPLE-"

Displays the date in this format: 07/04/86.

<time>
SOLID-APPLE-=

Displays the time in this format: 21:46 (meaning 9:46 p.m.). 0:00 means you don't have a clock. In the Data Base, if a category has the word *time* included in its name, AppleWorks converts 24-hour times to 12-hour times (for example, 21:46 is converted to 9:46 PM).

<find> (in the Word Processor)
SOLID-APPLE-RETURN

Searches a Word Processor file for the next carriage return marker and moves the cursor to the marker. This command works only when the "Type entry . . . " message is visible at the bottom of the screen. See the <ctrl-D> macro in MACROS.SUPER for an example of how to use this macro.

<find> (in an Add, Delete, or List files menu)
SOLID-APPLE-RETURN

Searches the current file list for the filename stored in macro Ø and moves the highlight cursor to that filename. To find a file in a hurry, record macro Ø with the desired filename, go to a file menu, and press SOLID-APPLE-RETURN.

If the filename isn't found, the highlight cursor is moved to the last filename on the list.

This command can be used with the OPEN-APPLE-< (<store>) and OPEN-APPLE-> (<recall>) commands to "link" related files. (For example, you can connect specific macro files with specific AppleWorks applications.) See Section 5c for details.

<back>
SOLID-APPLE-,

Finds the first blank space to the left of the current cursor position. This macro works wherever AppleWorks allows you to edit characters, including Data Base fields, at Find prompts, and where AppleWorks prompts you to enter names.

<ahead>
SOLID-APPLE-.

Finds the first blank space to the right of the current cursor position.

<findpo>
SOLID-APPLE-^ (in the Word Processor)

Searches for the next caret (^) and moves the cursor to that position. (The caret may be either a printer option caret or a text caret.) This macro works only when the "Type entry or . . ." prompt is visible at the bottom of the screen.

This macro is easier to use to locate printer options in a file than the OPEN-APPLE-F command, which requires you to know which option you're searching for (and its two-letter code). SOLID-APPLE-^ simply searches for the next printer option of any kind in the file.

(no token; can be used only from the keyboard)
SOLID-APPLE-^ (in the Data Base)

Reads information about the files on the current disk or subdirectory into a Data Base file. This information includes the disk location, filename, file type, file size, date last altered, and notes. Here's how to use this macro to help you organize your disks:

1. Boot AppleWorks. Add the file DISK FILES to the Desktop from the Super MacroWorks disk.
2. Press SOLID-APPLE-^. The current disk or subdirectory file information will be read into DISK FILES.
3. If desired, change to a different disk or subdirectory, then use SOLID-APPLE-^ to add the file information to DISK FILES. Repeat for as many disks or subdirectories as desired.

4. Print reports. There are three pre-defined Report Formats:
   - All files: Prints the file information for all the files read into DISK FILES.
   - AppleWorks files: Prints the file information for only the AppleWorks files read into DISK FILES.
   - Notes: Prints the file information for only the AppleWorks files read into DISK FILES that have entries in their "Note" field.

If desired, you can create a renamed version of DISK FILES and add new categories, but *do not change the order of the original categories.*

Note: This macro won't work if you have only one disk drive (it won't pause for disk swaps).

5. *New AppleWorks Commands*

Super MacroWorks provides you with a variety of new AppleWorks commands. These commands will add tremendous power and flexibility to your macros. Some of the commands are CONTROL-key commands; these are described in Section 5a. Other commands are OPEN-APPLE commands; these are described in Section 5b. Most of the commands have tokens that you can use in macro definitions. Section 5c describes some special ways to use the new commands.

5a. New CONTROL-Key Commands

Note: These CONTROL-key commands cannot be used directly from the keyboard; they work only from within macros. If you are recording a macro, press the appropriate CONTROL-key combination. Some of these commands will work when you press the key; others won't work until you execute the macro. If you are creating or modifying a macro definition, use the appropriate token.

CONTROL-C
<if>

Tests whether the character at the current cursor position matches the character following the token (for example, "<if>." to test for a period). If so (true), continues with the next command in the macro. If not (false), stops the macro (if the macro was called from another macro, returns to the calling macro). For an example of how to use this command, see "Conditional Macros" in Section 5c below.

Note: This command works only with the blinking insert or overstrike cursor. It does not work when the cursor is a highlight bar.

CONTROL-X **<ifnot>**

Opposite of **<if>** command. Tests whether the character at the current cursor position does *not* match the character following the token. If true (i.e., if there is *not* a match), continues with the next macro command. If false (i.e., if there is a match), stops the macro (if the macro was called from another macro, returns to the calling macro). For an example of how to use this command, see "Conditional Macros" in Section 5c below.

Note: This command works only with the blinking insert or overstrike cursor. It does not work when the cursor is a highlight bar.

CONTROL-W **<ifkey>**

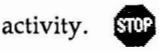
Checks whether a key is pressed and whether the key matches the character following the token. For example, "<ifkey><left>" to test for a LEFT-ARROW keypress or "<ifkey>C" to test for the letter C. Continues with the next macro command only if a key is pressed *and* it matches.

*Note: The <ifkey> command looks for an EXACT match. For example, **<ifkey>A** tests for an upper-case "A"; **<ifkey>a** tests for a lower-case "a". The <ifkey> command responds only to keystrokes, not mouse movements.*

CONTROL-R **<rpt>**

Repeats a macro indefinitely. The macro stops only when a conditional command (such as <if> or <ifnot>) stops it. If a macro calls itself *without* a <rpt>, the macro repeats only about 18 times. For an example of how to use this command, see "Conditional Macros" in Section 5c below.

CONTROL-Q **<stop>**

Stops all macro activity. 

CONTROL-E **<input>**

Allows you to enter text or OPEN-APPLE commands until you press RETURN (the RETURN is *not* passed to AppleWorks). To exit this macro prematurely, press CONTROL-@.

CONTROL-T **<key>**

Pauses for a keystroke; continues when a key is pressed and passes the keystroke to AppleWorks.

CONTROL-O **<list>**

Designates the start of a list of items to be passed to AppleWorks. The items are passed one at a time in the order they appear in the list. Each time the macro is called, the next item in the list is passed; when the end of the list is reached, it is reset to the first item.

List items must be separated by commas; the commas are *not* passed to AppleWorks. List items must be either text or OPEN-APPLE commands. See macro X in MACROS.SUPER for an example (the list items are the days of the week).

The <list> token can appear anywhere within the first 250 bytes of a macro. There can be only one list per macro. A list cannot be included in a macro that is called by another macro.

CONTROL-P **<msg>**

Prints a message on the second-to-last line of the screen. The message can be normal, inverse, or MouseText style. Enclose the message in quotation marks (") for normal style, apostrophes (') for inverse style, or asterisks (*) for MouseText style. For example:

```
<msg>"This is normal text."
<msg>'This is inverse text.'
<msg>*This is MouseText.*
```

In MouseText, upper-case characters appear as MouseText and all other characters appear as normal text (refer to the Key Chart at the back of this manual).

If you want to use " or ' or * in the message itself, enclose the message in a different keyboard character. The message will appear in normal style. For example:

```
<msg>#This is "normal" text.#
```

If you want to display more than one message on the same line, the <msg> commands must be back-to-back with no other text or commands between them.

If you want to mix more than one style on the same line, make each message an even number of characters long; otherwise, there may be some overlap and loss of characters. See macro V in MACROS.SUPER for an example.

Each time a message is displayed, the portion of the line to the right of the message text is re-drawn using whatever character (for example, a dash or a blank space) is currently in column 79 on the line. If you like, you can display a message with a special character in column 79, then display a second message. The portion of the line to the right of the second message will be drawn using the character set up by the first message.

5b. New OPEN-APPLE Commands

Note: These commands can be used directly from the keyboard as well as from within macros. If you are recording a macro, press the appropriate OPEN-APPLE-key combination. If you are creating or modifying a macro definition, use the appropriate token.

OPEN-APPLE-: **<uc>** — Changes the character at the current cursor position to upper-case.

OPEN-APPLE-; **<lc>** — Changes the character at the current cursor position to lower-case and turns on the overstrike cursor (the blinking rectangle).

OPEN-APPLE-! **<insert>** — Turns on the insert cursor (the blinking underscore). To turn on the overstrike cursor (the blinking rectangle), follow this command with an OPEN-APPLE-E command.

OPEN-APPLE-@ **<zoom>** Forces zoom *out* (hides printer options in the Word Processor; shows values rather than formulas in the Spreadsheet; shows multiple-record layout in the Data Base). Follow with OPEN-APPLE-Z to zoom in.

OPEN-APPLE-% **<end?>** Determines whether any files are left on the Desktop. If not, it stops all macro activity; otherwise, it continues the current macro. Use this command immediately after an OPEN-APPLE-Q command. See the <esc> macro in the MACROS.SUPER file for an example of how to use this command.

OPEN-APPLE-& **<disk>** Reads the current volume (disk) name or subdirectory pathname into macro Ø. See macro <ctrl-Q> in MACROS.SUPER for an example of how to use this command.

OPEN-APPLE-* **<path>** Reads the current volume name or subdirectory name and the currently highlighted filename into macro Ø. You can use this command only from a List, Add, or Delete files menu. See the <ctrl-S> macro in MACROS.SUPER for an example of how to use this command.

OPEN-APPLE-(**<menu>** Used with the OPEN-APPLE-) command (see below). Allows you to leave a Desktop file and then quickly return to it later. Saves the current Desktop file number and sets up macro Ø to send you to the Main Menu.

When you want to leave a file, press OPEN-APPLE-(followed by SOLID-APPLE-Ø.

OPEN-APPLE-) **<resume>** Used with the OPEN-APPLE-(command (see above). Sets up macro Ø to return you to the Desktop file you exited when you used the OPEN-APPLE-(command. Follow OPEN-APPLE-) with SOLID-APPLE-Ø to return to the file.

OPEN-APPLE-- **<swap>** Swaps macro Ø with its special storage area. In effect, it gives you two macro Ø's for use with commands such as <path>, <resume>, <menu>, and <find> (see above).

OPEN-APPLE-CONTROL-W **<inc>**

Increments the character at the current cursor position. For example, "a" becomes "b", "1" becomes "2", and so on. This command is useful as a "counter" when used with a conditional command such as <if> or <ifnot>. It can also be used to change spreadsheet values.

OPEN-APPLE-CONTROL-A **<dec>**

Opposite of <inc>; decrements the character at the current cursor position.

OPEN-APPLE-#

Turns on "single-step" mode: When a macro passes characters to AppleWorks, this command forces a pause before each character is sent; the character is sent only after you press a key. Press RETURN to return to normal speed. This command is useful for "debugging" complex macros.

OPEN-APPLE-$

Displays macro definitions. OPEN-APPLE and SOLID-APPLE commands are not "tokenized." For example, the OPEN-APPLE-^ command (see below) is displayed as <oa-^> rather than <read>, and SOLID-APPLE-RETURN (see Section 4f) is displayed as <sa-rtn> rather than <find>.

OPEN-APPLE-^ **<read>**

From the keyboard: Reads the character at the current cursor position into macro Ø. You can use the ARROW keys to move the cursor to a new position before you read the next character.

Recording a macro: OPEN-APPLE-^ reads the character at the current cursor position into the macro that is being recorded (the character becomes text in the macro).

<read> token in a macro definition: Reads the character at the current cursor position into macro Ø.

Here's an example of how to use this command:

1. Create a new Data Base file.
2. Press OPEN-APPLE-X 1 to define macro 1.
3. Press OPEN-APPLE-^ nine times to store the word "Category " (including a blank space at the end) in macro 1.
4. Press CONTROL-@ to end the macro definition.
5. Use the SOLID-APPLE-1 command to quickly enter new categories in the Data Base file.

See macros <ctrl-R> and <ctrl-P> in the MACROS.SUPER file for examples of how to use this command from a macro.

Note: The following commands are contained in the file COMPILER.SYS, which is not available to Quark hard disk users.

OPEN-APPLE-+ **<Lprint>** Prints the current screen line on the printer designated as the OPEN-APPLE-H printer.

OPEN-APPLE-= **<compile>** Compiles the macros in the current Word Processor file. If this command is called from within a macro (and if there are no errors in the compilation), the first macro in the file is executed automatically.

OPEN-APPLE-< **<store>** Stores the current contents of macro Ø in an unused area of a Word Processor or Spreadsheet file. The name being stored is displayed on the bottom line of the screen. This command is useful for "linking" application files and macro files; see Section 5c.

OPEN-APPLE-> **<recall>** Sets macro Ø equal to the filename stored by the OPEN-APPLE-< (<store>) command. This command is useful for "linking" application files and macro files; see Section 5c.

OPEN-APPLE-CONTROL-S **<saveØ>**

Copies the contents of macro Ø to a storage area. There are ten storage areas, numbered Ø - 9. If you press OPEN-APPLE-CONTROL-S from the keyboard, the prompt "Save to ?" will appear; enter the number of the storage area you want to use. If you are using the token in a macro definition, follow the token with the storage area number (for example, "<saveØ>3").

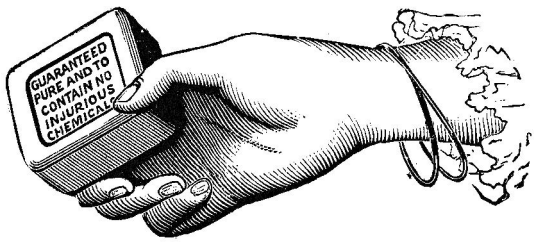
The OPEN-APPLE-CONTROL-L command (see below) loads the contents of one of the ten storage areas back into macro Ø.

Note: There are 12 macro Ø storage areas altogether: macro Ø itself, the <swap> temporary storage, and the 10 <saveØ> areas. These areas provide a total of 840 bytes. They are especially useful for doing <read>'s in one file and pasting the information in another file.

OPEN-APPLE CONTROL-L **<loadØ>**

Loads the contents of one of the ten <saveØ> storage areas into macro Ø. If you press OPEN-APPLE-CONTROL-L from the keyboard, the prompt "Load from ?" will appear; enter the number of the storage area you want to load from. If you are using the token in a macro definition, follow the token with the storage area number (for example, "<loadØ>3").

Important: In the Data Base, the SOLID-APPLE-^ command (see Section 4f) will wipe out the <saveØ> storage areas. Therefore, you cannot do a <loadØ> after a SOLID-APPLE-^ unless you first do a <saveØ>. In general, when you use <loadØ> be careful that you are loading data from the proper storage area.



OPEN-APPLE-CONTROL-D **<Ø=>**

Stores text in macro Ø. Starting with the first character after the command, it stores text until the next token, command, or end-of-macro marker, or until 70 characters have been stored.

This command can be used only from within a macro (not directly from the keyboard).

Here's a sample macro that uses <Ø=> to store the filename "My File" in macro Ø. The <find> macro is then used to locate that filename in a list of files:

L:<all><menu><sa-Ø><rtn><rtn><Ø=>My File<find><rtn>!

OPEN-APPLE-B **<ifØ>**

Compares macro Ø with one of the 10 <saveØ> storage areas. Continues with the next macro command if there is an *exact* match. For example, "<ifØ>3" compares the contents of macro Ø with the contents of storage area 3.

OPEN-APPLE-CONTROL-G **<bell>**

Sounds the AppleWorks error bell once. Drive co-workers crazy with this macro: G:<all><bell><rpt>!

The <var> Commands

Super MacroWorks provides you with a special system variable called <var>. You can use this variable as a "counter" to set up macros that "loop" a specified number of times. The commands listed below (which are included in COMPILER.SYS) can be used to set <var> to a starting value, increase or decrease <var>, and test whether <var> has reached a specified value.

OPEN-APPLE-Z **<var=>**

Sets <var> to a number from 0 - 9. For example, "<var=>3" sets <var> to 3.

OPEN-APPLE-N **<incvar>**

Increases <var>. The highest value <var> can have is z (74). Use the Key Chart at the back of this manual to determine the keyboard symbols for <var> values greater than 9. The sequence of <var> values is 0, 1, 2, . . . 9, : (10), ; (11), < (12), = (13), . . . A (17), B (18), . . . a (49), b (50), . . . z (74).

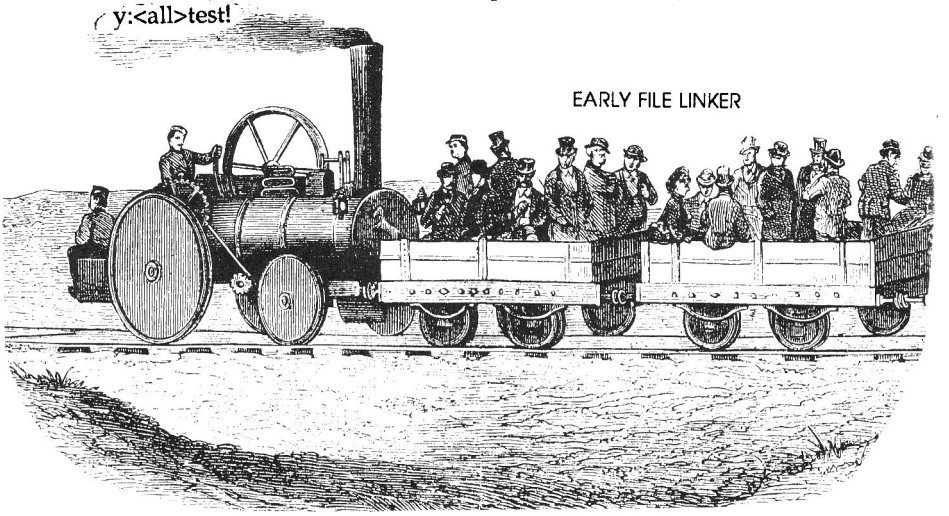
OPEN-APPLE-Y **<decvar>** Decreases <var>. The lowest value <var> can have is zero.

OPEN-APPLE-F **<varnot>** Allows you to repeat a macro a predetermined number of times. The macro continues as long as <var> does *not* equal the specified value and stops when <var> reaches the specified value. The example below demonstrates how to use <varnot>.

OPEN-APPLE-] **<var>** Use this command with <saveØ>, <loadØ>, and <ifØ>. For example, "<loadØ><var>" loads the contents of the storage area whose number corresponds to the current value of <var> into macro Ø. (To use <var> this way, its value must be in the range 0 - 9.)

An example: Here's a simple example of how to use the <var> commands to control the number of times a macro is executed. In this example, three macros are defined. Macro v sets <var> to a starting value of zero and calls macro x. Macro x calls macro y (which simply prints some text), increments <var>, and tests whether <var> has reached a value of z (74). If <var> is *not* equal to z, macro x continues; if <var> equals z, macro x stops. Thus, macro x is repeated 74 times. Here are the macro definitions:

```
v:<all><var=>Ø<sa-x>!     initialize <var>; call macro x
x:<all><sa-y><incvar><varnot>z<rpt>!   call macro y 74 times
y:<all>test!
```



5c. Some Special Tricks

As you experiment with the commands listed in this section, you will discover new and powerful ways to use macros. Here are a few ideas to get you started and spur your imagination.

Linking Files

You can use the <store> and <recall> commands to link application files (Word Processor, Data Base, and Spreadsheet) with specific macro files. For example, suppose you have created a Spreadsheet file named FINANCE and a Word Processor file named MACROS.FINANCE that contains macros you want to use with FINANCE. MACROS.FINANCE should be in the same directory as FINANCE.

To link these two files, use the macro shown below while you're in the Spreadsheet file (FINANCE). Then save FINANCE and remove both files from the Desktop.

```
Y:<all><Ø=>MACROS.FINANCE<store>!
```

Now, the following macro will go to the "Add files" menu and allow you to select FINANCE. Then it will <recall> the name of the associated macro file (MACROS.FINANCE), load and compile it, and execute the first macro in the file:

```
X:<all><oa-q><esc><rtn><rtn><input><rtn><recall><swap>
 <menu><sa-Ø><rtn><rtn><swap><find><rtn><compile>!
```

The first macro in MACROS.FINANCE must be defined as follows in order to return you to the Spreadsheet file FINANCE:

```
1:<all><resume><sa-Ø> . . . any other commands you want . . . !
```

Macro X (defined above) should also be in MACROS.FINANCE so you can repeat the process with a new application file when you're finished with the spreadsheet.

For an example of how to link files, try this:

1. Add the file MACROS.SS from your Super MacroWorks disk to the Desktop.

2. Press OPEN-APPLE-= to compile the macros in MACROS.SS.

3. Press SOLID-APPLE-X.

4. Select the file SAMPLE SS from the file menu.

The MACROS.STARTUP File

In the above example, macro X, which links the FINANCE and MACROS.FINANCE files, makes use of the <find> macro. The <find> macro searches an Add, Delete, or List files menu for the filename stored in macro Ø and moves the highlight cursor to that filename. But this requires that you first store a filename in macro Ø. What if you want to use <find> in your auto-startup macro (the macro that is automatically executed when you boot AppleWorks)?

No problem: *The default definition of macro Ø at boot-up time is MACROS.STARTUP.* If there is a <find> in your auto-startup macro, it will search for MACROS.STARTUP. If you follow the <find> with a <compile>, the macros in MACROS.STARTUP will be ready for use. You can customize MACROS.STARTUP so it does whatever you want it to. There is a sample MACROS.STARTUP file on your Super MacroWorks disk; examine it for more information.

Keep in mind that you can also use the <Ø=> command (see Section 5b) from within a macro to store a filename in macro Ø.

Conditional Macros

Some commands test for a condition (for example, whether the character at the current cursor position matches a specified character) and either continue the macro if the condition is true or stop if the condition is false.

These commands stop *all* macro activity if the condition is false:

<find> <end?>
<compile> <stop>

These commands stop only the current macro; they return to the calling macro, which continues:

<if> <ifkey>
<ifnot> <varnot>
<ifØ>

Here's an example of how to use conditional macros. Suppose you define three macros A, B, and C:

A: Tests for a period. If the cursor is on a period, it stops all macro activity; otherwise, the macro quits (if it was called from another macro, it returns to the calling macro).

B: If the cursor is at a blank space, substitutes a period for the blank space.

C: Calls both A and B. The overall purpose is to search through a text string, replace any blank spaces with periods, and stop when a period is encountered. C searches the text string character-by-character and repeats indefinitely until a period is encountered. This macro could be used to convert a text string to a ProDOS-legal filename (where periods are allowed but blank spaces aren't).

Here are the macros:

A:<all><if>.<stop>! test for end of sentence
B:<all><if><spc>.<left>! substitute period for blank space
C:<all><sa-A><sa-B><right><rpt>!

Notice that <left> and <right> are used to reset the cursor to continue the search. The <rpt> command causes macro C to search any length text string; if macro C called itself without the <rpt>, it could search through no more than about 18 characters.

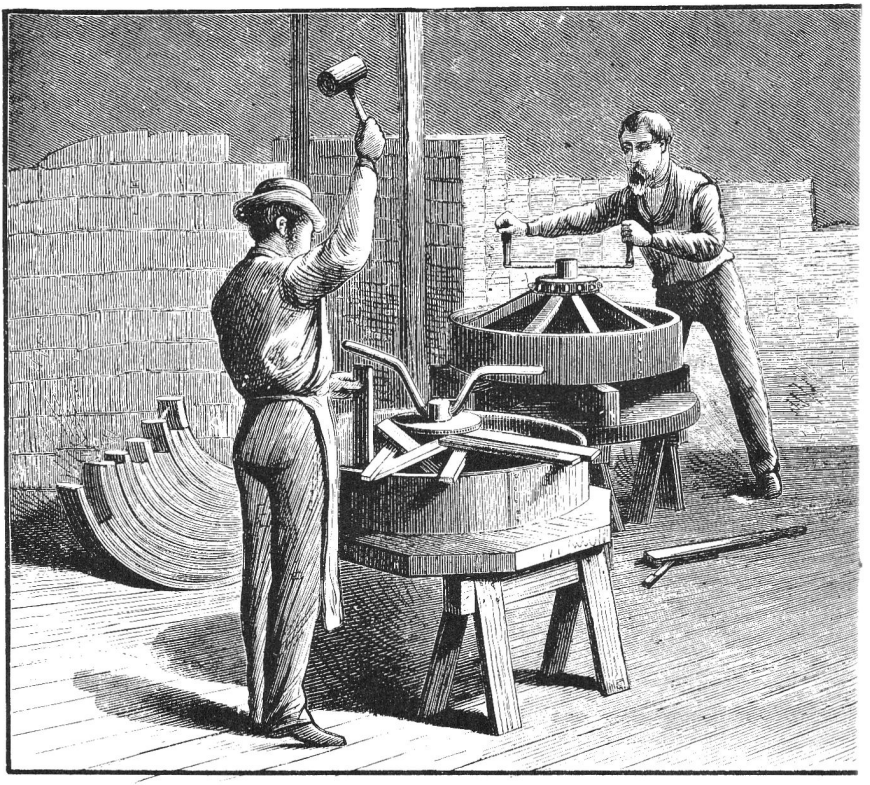
Putting It All Together

If you want to see an example of how to use both file-linking and conditional macros, just follow the steps listed below. This example demonstrates a search across multiple Data Base files.

1. Boot your updated AppleWorks Startup disk.

2. Add the file MACROS.DB from the Super MacroWorks disk to the Desktop.

3. Follow the instructions in MACROS.DB.



6. *Super MacroExtras*

Super MacroWorks gives you built-in macros, the ability to create your own custom macros, and a variety of new AppleWorks commands. But that's not all: Super MacroWorks also provides you with ways to customize AppleWorks itself to make your work more efficient and fun. The following pages describe these useful "extras."

6a. Custom Help Screens

Super MacroWorks allows you to change the AppleWorks OPEN-APPLE-? help screens. You can replace these screens with your own messages: new macro names, important birthdays, or whatever. Here's how:

1. Make a copy of your AppleWorks PROGRAM disk (*not* the Startup disk).

2. Boot the Super MacroWorks disk. Select option M: Main Menu.

3. Select option 1: Enhance AppleWorks.

4. Select option 3: Customize AppleWorks Help Screens.

5. You will be asked "Where is the AppleWorks PROGRAM disk?" Insert the Program disk you created in step 1 in a drive, then follow the procedure described in Section 2b to specify the current directory.

6. Select the application that contains the help screen you want to change: Word Processor, Data Base, or Spreadsheet.

7. The first line of the help screen will appear. Use UP-ARROW and DOWN-ARROW to move from one line to another.

   If you want to change a line, press RETURN. Then type the new text. You cannot delete or insert characters; if you make a mistake, press ESC to restore the original line. Your new message cannot be longer than the number of characters in the original line. If your message is shorter than the original line, include blank spaces at the end.

   Press RETURN when you're finished editing a line. Then either use the ARROW keys to select another line or press ESC to stop editing lines.

8. Select option 4: Save New Help Info. Super MacroWorks will update the AppleWorks Program disk with your new help messages. The message "New screen saved" will appear; press any key to continue.

9. Go back to step 6 to select another help screen to edit, or press ESC to stop editing help screens.

MouseText

You may use the 32 MouseText characters in your new help messages; just hold down the OPEN-APPLE key while you press another key. See the "Mouse Characters" column on the Key Chart at the back of this manual. Apple has been known to change the MouseText characters, so please don't send us dirty telegrams if the Key Chart is wrong.

6b. Beep, Don't Buzz

Here's how to use Super MacroWorks to change the AppleWorks error buzz to a short, friendly "beep" (if you have an Apple IIGS or an accelerator card, the buzz probably sounds more like a beep already):

1. Boot the Super MacroWorks disk and select option M: Main Menu.

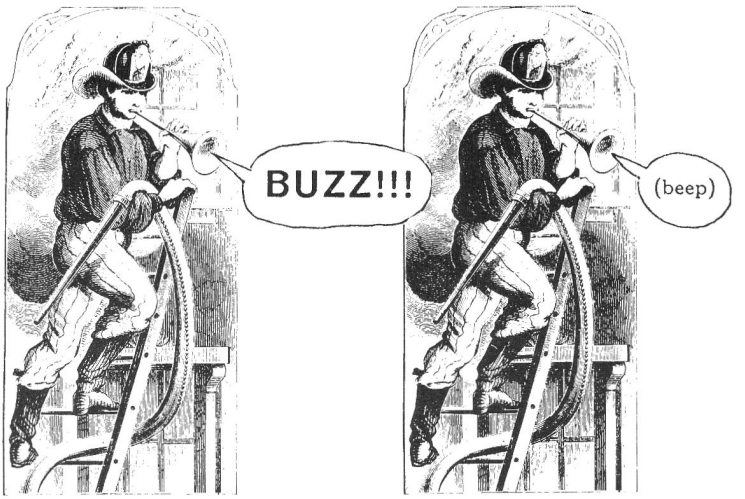
2. Select option 1: Enhance AppleWorks.

3. Select option 4: Change Error Buzz to MacroWorks Beep.

4. Press the space bar a few times to hear how the beep sounds. If you want to change from buzz to beep, press RETURN. If you don't want to make the change, press ESC.

5. You will be asked "Where is the AppleWorks STARTUP disk?" Insert your Startup disk in a drive and follow the procedure in Section 2b to specify the current directory.

   Super MacroWorks will update your Startup disk to change the error buzz to an error beep. You will be returned to the Enhance AppleWorks menu.



6c. Power Printing

Power Print™ is an exciting Beagle Bros product that allows you to create and download custom fonts to your dot matrix printer. You can even use Power Print to produce small graphics and company logos!

If you're lucky enough to have Power Print, you know that you can create a special file called a "Quickload File" (see the *Power Print User's Manual*). A Quickload File enables you to download a font to your printer without booting the Power Print disk.

You can combine the Quickload feature of Power Print with Super MacroWorks to cause a custom font to be downloaded to your printer when you boot AppleWorks. Here's how:

1. Use Power Print to create a Quickload File. Name the file QUICK.LOADER. (If you want to use a previously created Quickload File, rename it.)

2. Save the QUICK.LOADER file on the updated AppleWorks Startup disk created in Section 2a above. The file must be in the same subdirectory as the files SUPER.SYSTEM and APLWORKS.SYS.

3. Make sure your printer is turned on and on-line. Boot AppleWorks with your updated Startup disk. The font specified by the QUICK.LOADER file will be downloaded to your printer. This font will be used until you turn off the printer or until you use Power Print to download a different font.

Mix Graphics and Text

Power Print turns your printer into a creative tool. For example, you may design up to 96 custom printer-characters that combine to print a small illustration like a logo or your signature. You may then include this illustration as part of your regular word processor printouts.

COUNT: ABCDabcd6789*&?
BYTE: ABCDE12345
OUTLINE: XYZ OLD ENGLISH
BORDER FONTS TOO!

6d. Bird's Better "Bye"

(by Alan Bird of *The Software Touch*)

The Super MacroWorks disk includes a slightly-altered version of ProDOS. This version contains an improved "BYE" program selector that helps you quickly select System files from a menu. (System files are files like BASIC.SYSTEM, the file for Applesoft BASIC, and SUPER.SYSTEM, the file for Super MacroWorks.)

If you want to use this feature, copy the file PRODOS from the Super MacroWorks disk to your AppleWorks Startup disk. Then whenever you boot AppleWorks with the Startup disk, the special "BYE" code is brought into memory. When you quit AppleWorks (or when you type BYE from Applesoft), a menu appears that displays the names of all the System files and subdirectories on the current disk. The menu looks something like this:

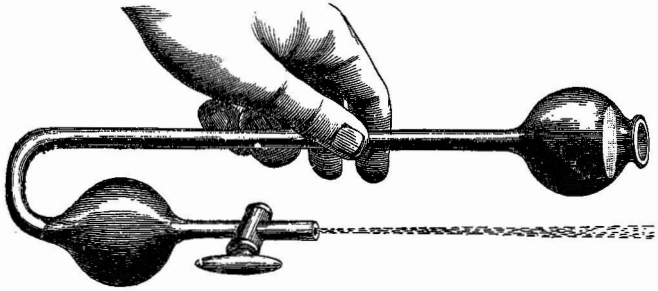
```
ESC: CHANGE VOLUME
RETURN: SELECT FILE

/APPLEWORKS
        PRODOS
        BASIC.SYSTEM
        APLWORKS.SYS
        SEG.00
        SUPER.SYSTEM
        /SUB1
```

In this example, /APPLEWORKS is the name of the current volume (disk). Below it are the names of the System files and subdirectories on that volume (subdirectory names begin with a slash).

To select a file or subdirectory from such a menu, just use the UP-ARROW and DOWN-ARROW keys to highlight the name, then press RETURN. If you select a file, that file is activated; if you select a subdirectory, another menu appears that shows the files contained in that subdirectory. If you want to switch to a different disk, press ESC.

In our example, you could use the "BYE" selector to switch between SUPER.SYSTEM (AppleWorks *with* macros) and APLWORKS.SYS (AppleWorks *without* macros). You could also select BASIC.SYSTEM for Applesoft BASIC.



7. *Super MacroTips*

Extra Custom Printers

We know of three ways to get around AppleWorks' limit of one custom printer:

1. Create macros to enter all the printer codes for each custom printer . This is the only way to avoid re-booting to change printers.

2. Make a different Program disk for each printer; you can use the same Startup disk.

3. Keep multiple versions of SEG.PR on a UniDisk 3.5 under different names. Use an Applesoft program to rename one to SEG.PR before you start to use AppleWorks.

A Use for Characters-Per-Inch

To trick AppleWorks into accepting special printer codes, try setting up a Custom Printer and using the different "characters-per-inch" options. For example, 10 characters-per-inch could send an ESC ABC to the printer, and 11 characters-per-inch could send ESC DEF.

Quick-Change Artist

If your printer has switches for changing print quality or typestyle, use Pause Here to let you change styles between paragraphs during printing.

How to Print Just One Page

To print only one Word Processor page, move the cursor to the beginning of your document and do an OPEN-APPLE-O PE (Pause Each page). Then put the cursor on the first line you want printed and do an OPEN-APPLE-P C (print from Cursor). When your printer finishes a page, press ESC to quit.

Printer Code Zappers

Use SOLID-APPLE-D to delete AppleWorks indents, page breaks, and so on. If you want, you can even move them to a new position with SOLID-APPLE-U.

Chip DIP

Mark your printer's DIP switch settings with a felt pen so you don't have to hunt for the user's manual the next time you need to change the settings.

I Thought I Saved That File. . . .

It's good practice to save a document *before* you print it, just in case something goes wrong. With AppleWorks, if you save before printing you are told that you have made changes to the file and should save it again. Actually you didn't change the file, *AppleWorks* did by inserting page breaks. You really don't need to save again. Of course, to play it safe, maybe you should. . . .

One solution is to do an OPEN-APPLE-K before saving (a new macro!). Then AppleWorks won't have to make changes before printing.

Missing a File?

In AppleWorks filenames, lower-case and upper-case letters are all the same. And AppleWorks doesn't care if you use a space or a period between letters. So saving a file three times as "File Name", "FILE.Name", and "FILE.NAME" will leave you with *one* file named FILE.NAME on the disk.

Re-Sort Your Files

AppleWorks displays filenames in an alphabetized list. You can set up subgroups within this list by starting each filename with an appropriate prefix. For example, all of your letters to people could start with the characters "LTR.".

Save/Delete/Rename

AppleWorks has a clever (careful!) way of saving files that lets you press ESC if you change your mind during a save. Here's what happens when you save a file called "Note" that's already on a disk:

1. Your file is saved as "APPLEWORKS.TEMP".

2. If ESC isn't pressed, "Note" is deleted.

3. APPLEWORKS.TEMP is renamed "Note".

NEVER Do This!

Never save any valuable Appleworks file under the name "APPLEWORKS.TEMP" (clue above).

Two-Sided Disk Tip

Copy your AppleWorks Startup disk onto one side of a floppy and your Program disk onto the back. Then you have to grab only one disk when you want to use AppleWorks. When you boot side one and are told to insert your Program disk, just flip the disk over and you're in business. If you choose a special color for this disk, like bright yellow, it will be that much easier to find on a messy desk. To copy onto the back of a floppy, you need to punch a second write-protect notch in its side. To make a disk bright yellow. . . . Never mind.

If You Have Only One disk Drive

Macros are fast and won't wait around for "disk swapping" unless specifically told to. If you only have one drive, some macros may require modification.

<<< Page Number >>> Tip

When writing a long document, we like to force page breaks and type in page numbers as we go. By using the format

<<< page 3 >>>

we can jump from page to page by simply searching for "< p".

Sorted Lists in the Word Processor

Use the Spreadsheet as a mini-Word Processor to enter a list. After using OPEN-APPLE-A to arrange it alphabetically, print to the Clipboard and then move it to the Word Processor.

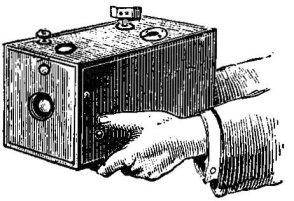
That Perforated Feeling

Flip the perforation on your printer paper with your index finger to start the tear. Then fold the pages before you rip off the perfed edges; it's easier and it saves time.

Ready For Anything

Use the Auto-Startup macro (see Section 4c) to add these files to your desktop first thing in the morning:

- NOTES (a Word Processor file)
- CALCULATOR (a Spreadsheet file)
- APPOINTMENTS (a Data Base file)



Max-Width Macro

Set up a Word Processor macro that sets the left and right margins to 0.0 inches:

W:<awp><oa-O>LM<rtn>0<rtn>RM<rtn>0<rt n><esc>!

Now you can fit more words on the screen and do less scrolling. Fix the margins before printing. Use maximum-width settings when writing custom macro files and you'll have more room for comments.

Find It Fast

Use the Word Processor's Find command to find sample tokens, macro definitions, etc. when you're defining your own macro commands.

Magic Markers

Markers are great for setting vertical "tab stops" throughout a Word Processor document. They let you scroll to another section and quickly find your way back. We have found that using only Marker #1 (instead of many different numbers) prevents confusion. Use SOLID-APPLE-M to set Marker #1's wherever you want: at page breaks, paragraph headings, etc. Use SOLID-APPLE-G to go to them.

Which reminds us: have you noticed that you can't search in the middle of a move? Sure you can:

1. Use SOLID-APPLE-M to set a marker at the point you're going to move to.

2. Go to the text you're want to move and use OPEN-APPLE-M or C to transfer it to the Clipboard (a-ha!).

3. Use SOLID-APPLE-G to go to your marker.

4. Use SOLID-APPLE-U to transfer the Clipboard onto the screen.

5. Use SOLID-APPLE-D to delete the marker.

Quick Statistics

Press OPEN-APPLE-Q to check on the available Desktop space. Or select "Remove Files" to check file sizes and status.

Macro Keystroke Saver

We just bet five bucks that you didn't know you could type a letter to select those AppleWorks options at the bottom of the screen. Did we win?

For example, when you press OPEN-APPLE-F, the bottom text line says:

Text Page Marker Case sensitive text Options...

Instead of moving the cursor with the RIGHT- ARROW key, you can select by pressing T, P, M, C or O. Use this trick when creating custom macros and save yourself a keystroke or three.

Key Chart

keys that can be defined as macros

	Upper-case	Lower-case	MouseText
➡ ctrl-@	sp	@ (16)	` (48)	@=
ctrl-A	!	A	a	A=
ctrl-B	➡ "	B	b (50)	B=
ctrl-C	#	C	c	C=
ctrl-D	$	D (20)	d	D=
ctrl-E	%	E	e	E=
ctrl-F	&	F	f	F=
ctrl-G Bell	➡ '	G	g (55)	G=
ctrl-H ←	(	H	h	H=
ctrl-I Tab	)	I (25)	i	I=
➡ ctrl-J ↓	*	J	j	J=
ctrl-K ↑	+	K	k	K=
ctrl-L	➡ ,	L	l (60)	L=
➡ ctrl-M Return	-	M	m	M=
ctrl-N	➡ .	N (30)	n	N=
ctrl-O	/	O	o	O=
ctrl-P	0 (0)	P	p	P=
ctrl-Q	1	Q	q (65)	Q=
ctrl-R	2	R	r	R=
ctrl-S	3	S (35)	s	S=
ctrl-T	4	T	t	T=
ctrl-U →	5 (5)	U	u	U=
ctrl-V	6	V	v (70)	V=
ctrl-W	7	W	w	W=
ctrl-X	8	X (40)	x	X=
ctrl-Y	9	Y	y	Y=
ctrl-Z	: (10)	Z	z (74)	Z=
ctrl-[Esc	;	[	{	[=
ctrl-\	<	\	|	\=
ctrl-]	➡ =	] (45)	}	]=
ctrl-^	>	➡ ^	➡ ~	^=
ctrl-_	? (15)	➡ _	➡ □ Delete	_=

➡ = Reserved macros (can't be changed or deleted)

Numbers in parentheses are <var> values (0-74).

Index

Disclaimer of All Warranties and Liabilities

Even though the software described in this manual has been tested and reviewed, neither Beagle Bros nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual, the software and/or the diskette; their quality, performance, merchantability, or fitness for any particular purpose. As a result, the diskette, software and manual are sold "as is," and you, the purchaser, are assuming the entire risk as to their quality and performance. In no event will Beagle Bros or its software suppliers be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the diskette, software, or manual, even if they have been advised of the possibility of such damages. In particular, they shall have no liability for any programs or data stored in or used with Beagle Bros products, including the costs of recovering or reproducing these programs or data. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

About ProDOS: This product includes software, ProDOS, licensed from Apple Computer, Inc. Apple makes no warranties, either express or implied, regarding the enclosed software package, its merchantability or fitness for any purpose. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so this limitation or exclusion may not apply to you.